Final Report

For Project 2:

The Real-Time Determination of Numbers of Passengers

BY

GROUP 9, VITAS

陶元杰 Tao Yuanjie 516370910068

张梓毅 Zhang Ziyi 516370910101

刘嘉晨 Liu Jiachen 516370910079

丁恺雯 Ding Kaiwen 516370910024

FOR

VG100 INTRODUCTION TO ENGINEERING (SUMMER 2017)
UNIVERSITY OF MICHIGAN-SHANGHAI JIAO TONG UNIVERSITY
JOINT INSTITUE
(UM-SJTU JI)
SHANGHAI, CHINA

INSTRUCTORS

Professor Yu Zheng, Ph.D.
Professor Cynthia Vagnetti, Ph.D.

Contents

List of Figures

List of Tables

1 Executive Summary

Since more and more people are gathering to metropolis like Shanghai, the burden on public transportation, especially subways and metros, is getting heavier and heavier. A common scene in metros is that some trains and carriages are overcrowded, while others are almost deserted. This inequality can cause not only passengers' discomfort, but also potential risks in operation. Our team found that both passengers and operation companies need one thing, real-time numbers of passengers in each carriage and train, so that they can make right decisions. Therefore, we decided to design a system to determine the real-time number of passengers in each carriage and each train, upload the real-time data onto the Internet, and provide a method (QR code) for everyone to achieve these data as conveniently as possible. To achieve the best accuracy and other functions, we make every effort on algorithm, mathematic principles and assemblage, etc. With two cameras to obtain the images, a computer (with Matlab) for image and data processing, the same computer for data transmission, a website to show real-time data and a QR code system to connect to the website, our whole system can meet our original demand. Also we meet with some problems and still have something to improve, but our system must have a good potential market and be welcomed by different stakeholders, passengers, operators, and people from all walks of life.

2 Introduction

2.1 Our Campus & Team

Our campus (**Figure 1**), the UM-SJTU Joint Institute (JI), was jointly established by Shanghai Jiao Tong University and the University of Michigan in 2006. This institute, located at 800 Dongchuan Road, Minhang District, Shanghai, China, is a successful example of international education cooperation. As all courses are taught in English by world-class faculty recruited from global leading universities, we, the students, not only acquire knowledge of academic fields but also gain creativity, teamwork, communication, and leadership.

We four (**Figure 2**), Kaiwen Ding （丁恺雯) from F1637002, Yuanjie Tao (陶元杰) from F1637003 and Jiachen Liu (刘嘉晨) & Ziyi Zhang (张梓毅) from

Figure 1. An overall view of our campus, the Joint Institute

F1637004, come together and form our team, VITAS, for the course VG100.

We name our team by VITAS according to our understanding of engineering, and each letter in this name stands for a principle our team believe in. First, “V” stands for **vision**. Engineers shoulder the responsibility to make changes and improve the world. The very precondition is using our own eyes to discover the world and find out what we can do. In another word, we need vision to find a theme for our creation. Second, “I” stands for **inspiration**. For thousands of years, human beings all knew that they needed more light at night, which was a common vision. However, nobody but Edison got the inspiration that we could electrify a material and make it emit heat as well as powerful light. This inspiration did change the world, and it proves the importance of inspiration for engineers. Third, “T” stands for **technology**. It can be quite obvious as technology is the essence for us to apply our knowledge, tools and materials we have to practice, so that we can produce inventions and creations. Fourth, “A” stands for **artistry**. What an engineer does is not only about technology, it is also about art. Only if an engineer regards his products as works of art can people accept his products. Last, “S” stands for **sociality**. In the modern world, good products is important, but advertising and marketing experience are also essential. We should try our best to communicate with all kinds of people and sell our own products.

Figure 2. Our Team, VITAS

Figure 3 is the logo for our team. As everyone can see, it is a slide of lemon. Lemon stands for a clear mind, with which we can have a better vision of the entire world and more chances to get perfect inspiration. We can also have a better ability of applying technology, comprehending artistry, and communicate with our potential customers. More than that, the lemon slide is divided into four parts, which means we four are individuals who combine together as a whole in Team VITAS.

Figure 3. Logo for Team VITAS

2.2 Problem

Nowadays, people are gathering to big cities. In Shanghai, for example, there has been 24 million people. In the same time, more and more people rely on public transportation, especially subways and metros, to travel. According to a survey (**Table 1**), Shanghai has about 10 million daily passengers in average, which is a great burden to transportation, which can bring both great convenience and trouble.

Ranking	City	Length	Line	Stations	Daily Passengers
1	Shanghai	617 km	14	366	10.83 million
2	Beijing	554 km	18	334	11.95 million
3	Guangzhou	266 km	9	164	8.79 million
4	Nanjing	225 km	6	121	2.98 million
5	Hong Kong	218 km	10	155	/
6	Chongqing	202 km	4	117	2.40 million
7	Shenzhen	178 km	5	131	3.94 million
8	Dalian	146 km	4	74	/
9	Tianjin	140 km	4	83	0.92 million
10	Taipei	131 km	5	117	3.03 million

Table 1. Metro conditions in major cities in China (2016)

A common scene in metros is that some trains and carriages are overcrowded (**Figure 4**), while others are almost deserted (**Figure 5**). This inequality can cause not only passengers' discomfort, but also potential risks in operation on trains and rails.

Figure 4. Overcrowded trains

Figure 5. Deserted trains

So, what people want to know is that where is relatively crowded, and where is not. For those passengers, they want to know the real-time number in each carriage and train so that they can choose a comfortable place. For metro operation companies, they also want to know the data so that they can make right decisions to balance the crowd and ensure public safety. Of course, except for subways, many other places have crowd and people also want to know the real-time data.

However, there is no that kind of system now. No existing system can share the real-time number of people. Therefore, our goal is to design and construct a system for passengers and metro operation companies to determine the real-time number of passengers in each metro carriage by using cameras to identify human heads, analyzing and counting by computer, store and transfer data. With this new system, both passengers and operators can obtain real-time data and make wiser decisions.

In short, the problems are

- **No real-time data, no way for passengers to choose a right train and carriage.**
- **No real-time data, no way for operation companies to lower pressure and ensure safety.**

2.3 Needs

What we need is a new system to determine the real-time number of passengers in each metro carriage, and deliver the data to anyone with a smartphone anywhere and anytime. With the help of this very system, all people, both passengers and metro operators, can embrace a efficient and intelligent life.

To be specific, all of our needs can be divided into two parts, criteria and constraints.

Criteria:

- **Attain the accurate number of people**
- **Transmit the real-time data**
- **Lower the budget**

Constraints:

- **Accuracy is hard to achieve**
- **Technology (algorithm & principles) is hard to improve**

3 Objectives

3.1 Methods to address the needs

3.1.1 Determine the accurate passenger flow

The best way to inform people the congestion degree is to offer them exact real-time numbers. After considering for many different potential ways of determination, identifying heads by camera can be a direct, accurate and feasible method. We should first obtain the whole picture of the carriage, then conduct image processing, and finally identify all human heads in the picture. **Task 2** and **Task 3** below are related to this objective.

3.1.2. Deliver and share the data

To make our potential customers use our product, we should make it achievable to as many people as possible. We deliver the real-time data we obtain and upload it onto our website. After comparing with history data, we can offer people analyzed data. And a QR code system should also be introduced as people love scanning QR code instead of using URL nowadays. With this we can share information to everybody on mobile terminals.

3.2 Design to solve the problems

First, the two cameras under the ceiling of the carriage, front and back, take real-time pictures and send them to a computer. After a series of image processing, like eliminating the coincide part and the original background, and comparison with the data base, our system will automatically find out the exact number of passengers (by detecting human heads) in the picture.

Then we integrate all the data in each carriage and each train, and add them into the data base and prepare for further machine learning. We also upload these real-time data onto our website. A QR code is prepared for everyone which can connect to by simply scanning. For passengers, all you need to do is using a cell phone to scan the QR code we offer, getting access to our website, and choosing the train or carriage you want to know about.

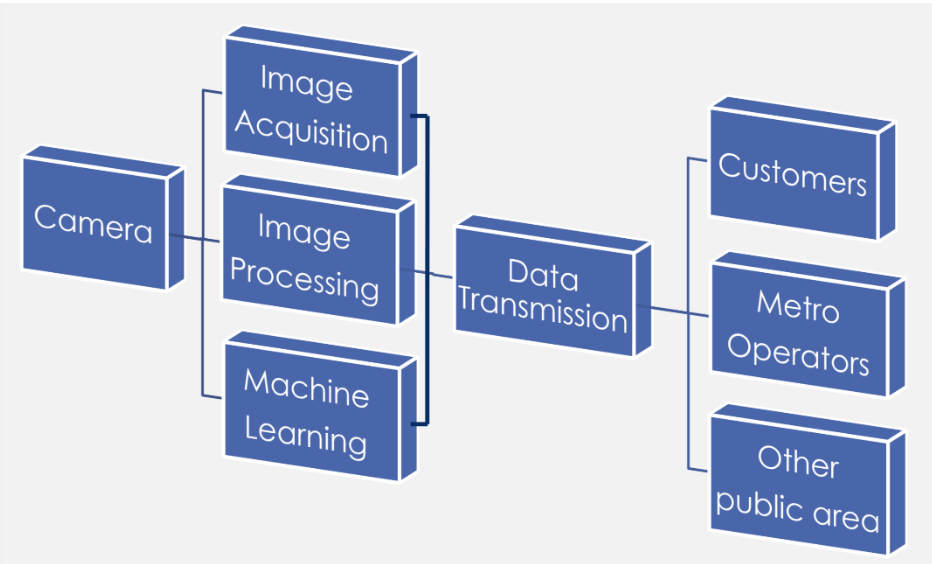


Figure 6. Block diagram for solutions

Our objectives aim to do everything for our stakeholders. All people need is a cell phone and they can know the exact number of passengers on any certain carriage of any metro line. Also, these data can be used by metro operators or anyone else. The system can even be applied to any place that can be crowded.

4 Solution

To solve all the problems and realize our system and goal, our solution entails:

1. **Two cameras to obtain the images**
2. **A computer (with Matlab) for image and data processing**
3. **A computer for data transmission**
4. **A website to show real-time data**
5. **A QR code system to connect to the website**

4.1 Cameras

We have two cameras on both sides to capture the whole picture of the carriage. We use USB cameras in experiments. Matlab toolbox 'Image Acquisition Toolbox' enables us to import pictures to Matlab workspace. The key problem is to avoid double counting the same head in two pictures. We managed to adjust the position of the cameras to make the only overlap area is a passenger who stands in the middle of the carriage.

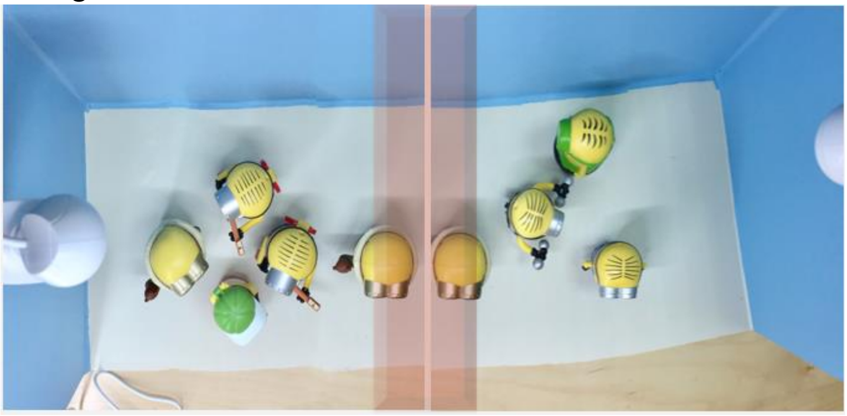


Figure 7. Image integration

This procedure ensures the only possible repeated count will only appear in a certain edge of the picture. After checking all detected heads' positions, our program automatically rules out those heads which exist in different pictures but the calculated positions coincide. Using this strategy, the final answer will cover the whole carriage with no blind angles and no repeated counts.

4.2 Image processing

4.2.1 Background

In preparation phase, we need to take a picture of the background. The next step is to cut the background of the new frame with the difference of two RGB color as a standard. The difference between two RGB colors is given by the Euclidean distance in RGB and another important indicator HSV distance without considering the Value coordinate.

Figure 8. An example of removing background

HSL and HSV are the two most common cylindrical-coordinate representations of points in an RGB color model. The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the Cartesian (cube) representation. Because HSL and HSV are simple transformations of device-dependent RGB models, the physical colors they define depend on the colors of the red, green, and blue primaries of the device or of the particular RGB space, and on the gamma correction used to represent the amounts of those primaries. As a result, each unique RGB device has unique HSL and HSV absolute color spaces to accompany it (just as it has unique RGB absolute color space to accompany it), and the same numerical HSL or HSV values (just as numerical RGB values) may be displayed differently by different devices [3].

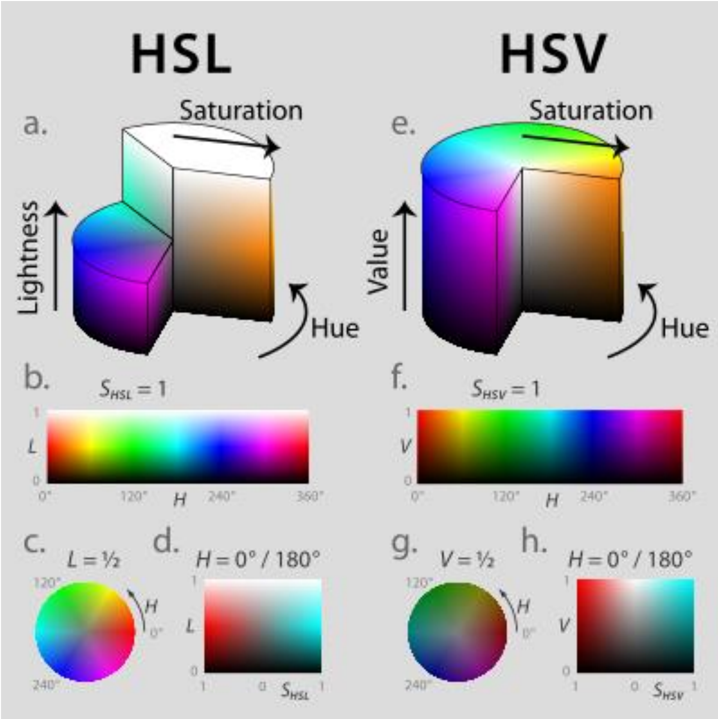


Figure 9. Illustration of HSL and HSV color space

For example, imagine we have an RGB display whose color is controlled by three sliders ranging from 0–255, one controlling the intensity of each of the red, green, and blue primaries. If we begin with a relatively colorful orange, with RGB values R = 217, G = 118, B = 33, and want to reduce its colorfulness by half to a less saturated orange , we would need to drag the sliders to decrease R by 31, increase G by 24, and increase B by 59, as pictured below [4].

Figure 10. Effect of subtle change of hue in RGB color space

If two colors are similar enough, we can consider the background does not change between those two frames, which will reduce the amount of computation later.

4.2.2 Finding head candidates

The Standard Hough Transform (SHT) uses the parametric representation of a line:

$$rho = x \cdot cos(\theta) + y \cdot sin(\theta)$$

The variable ***rho*** is the distance from the origin to the line along a vector perpendicular to the line. $\boldsymbol{\theta}$ is the angle of the perpendicular projection from the origin to the line measured in degrees clockwise from the positive x-axis. The range of $\boldsymbol{\theta}$ is $-90° \le \theta < 90°$. The angle of the line itself is $\theta+90°$, also measured clockwise with respect to the positive x-axis.

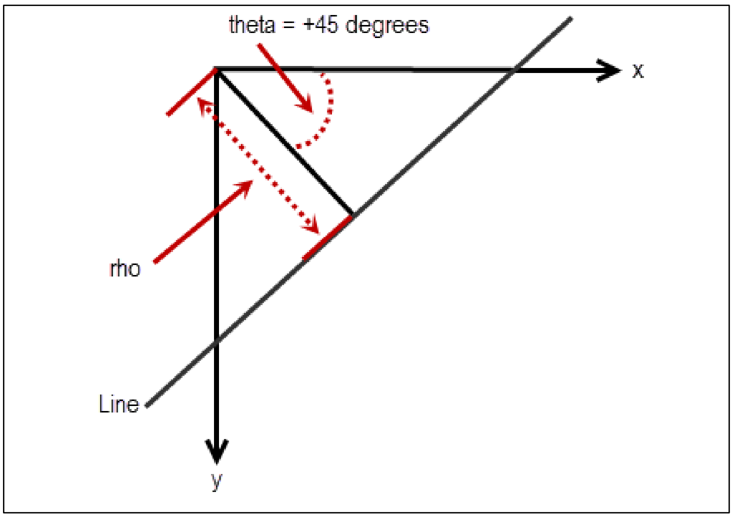
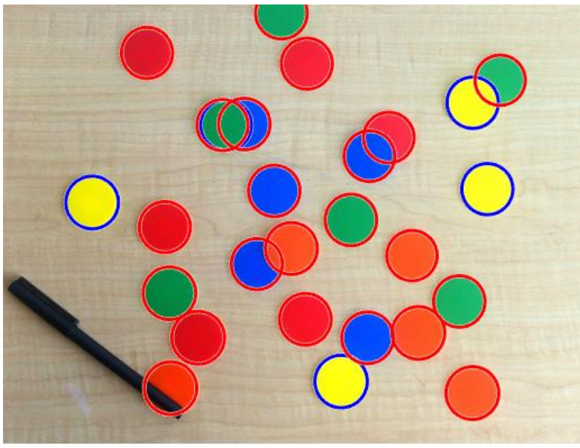


Figure 11. Standard Hough Transform (SHT) parametric representation

The SHT is a parameter space matrix whose rows and columns correspond to rho and theta values respectively. The elements in the SHT represent accumulator cells. Initially, the value in each cell is zero. Then, for every non-background point in the image, ***rho*** is calculated for every theta. ***rho*** is rounded off to the nearest allowed row in SHT. That accumulator cell is incremented. At the end of this procedure, a value of Q in SHT(r,c) means that Q points in the xy-plane lie on the line specified by $\boldsymbol{\theta}$(c) and ***rho***(r). Peak values in the SHT represent potential lines in the input image.

In this project we uses a Circular Hough Transform (CHT) based algorithm for



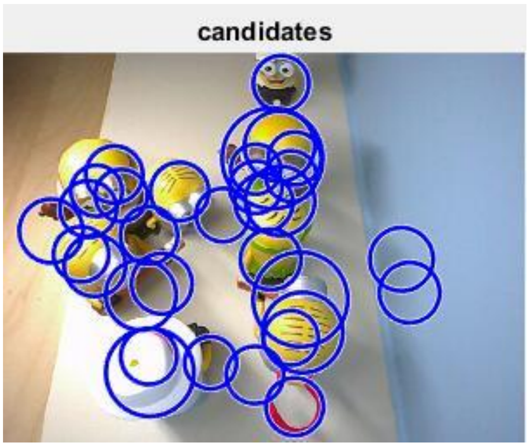


Figure 12. Circular Hough Transformation (CHT) under realistic and real situation

finding circles in images [5]. This approach is used because of its robustness in the presence of noise, occlusion and varying illumination.

While since the real situation is not as realistic as the example (left), there are too many candidates to be determined. Detailed solution will be given in other subsections.

The CHT is not a rigorously specified algorithm, rather there are a number of different approaches that can be taken in its implementation. However, by and large, there are three essential steps which are common to all.

1. Accumulator Array Computation.

Foreground pixels of high gradient are designated as being candidate pixels and are allowed to cast 'votes' in the accumulator array. In a classical CHT implementation, the candidate pixels vote in pattern around them that forms a full circle of a fixed radius. Figure 1a shows an example of a candidate pixel lying on an actual circle (solid circle) and the classical CHT voting pattern (dashed circles) for the candidate pixel.

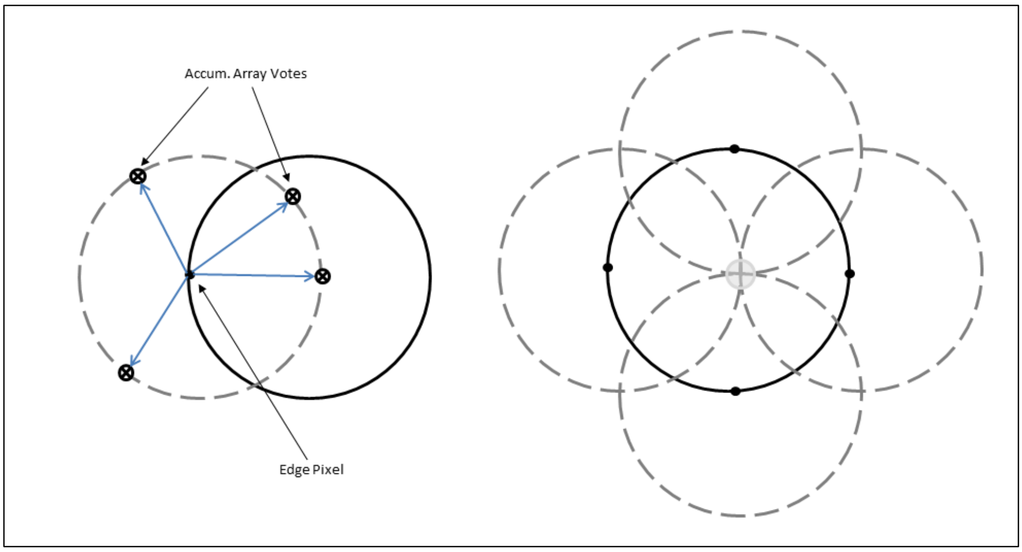


Figure 13. Circular Hough Transformation (CHT) accumulator array computation

2. Center Estimation

The votes of candidate pixels belonging to an image circle tend to accumulate at the accumulator array bin corresponding to the circle's center. Therefore, the circle centers are estimated by detecting the peaks in the accumulator array. Figure 1b shows an example of the candidate pixels (solid dots) lying on an actual circle (solid circle), and their voting patterns (dashed circles) which coincide at the center of the actual circle.

3. Radius Estimation

If the same accumulator array is used for more than one radius value, as is commonly done in CHT algorithms, radii of the detected circles have to be estimated as a separate step [6].

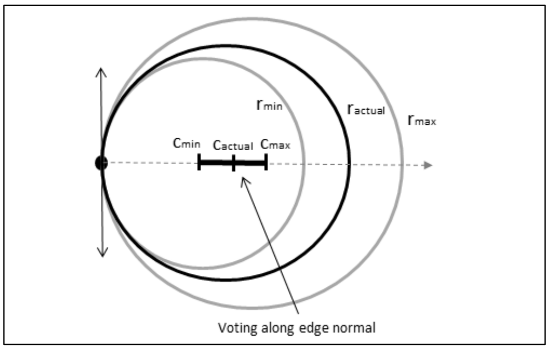


Figure 14. Circular Hough Transformation (CHT) voting pattern

4.2.3 Image de-noising

De-noising methods based on wavelet decomposition is one of the most significant applications of wavelets.

Wavelet de-noising attempts to remove the noise present in the signal while preserving the signal characteristics, regardless of its frequency content. It involves three steps: a linear forward wavelet trans-form, nonlinear thresholding step and a linear in-verse wavelet transform. Wavelet de-noising must not be confused with smoothing; smoothing only removes the high frequencies and retains the lower ones. Wavelet shrinkage is a non-linear process and is what distinguishes it from entire linear de-noising technique such as least squares. As will be explained later, wavelet shrinkage depends heavily on the choice of a thresholding parameter and the choice of this threshold determines, to a great extent the efficacy of de-noising.

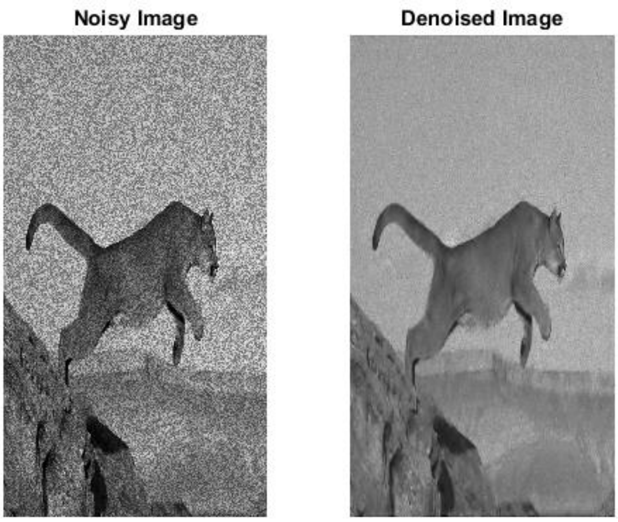


Figure 15. An example of wavelet image de-noising

4.2.4 Image segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics [7].

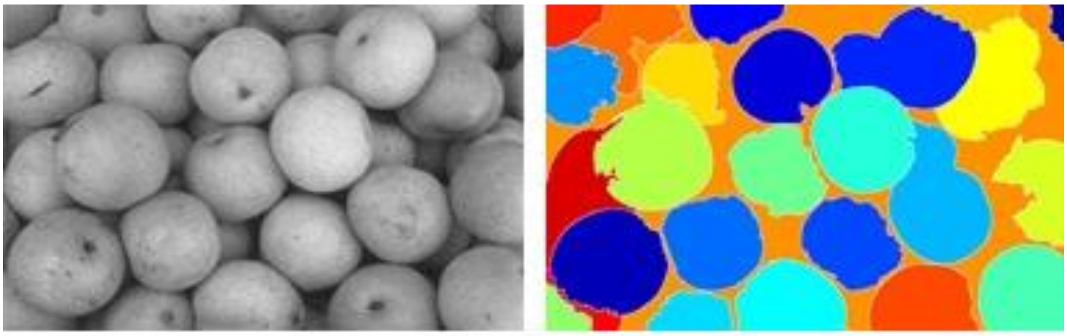


Figure 16. An example of image segmentation

In this project, we use **region-growing method**.

Region-growing methods rely mainly on the assumption that the neighboring pixels within one region have similar values. The common procedure is to compare one pixel with its neighbors. If a similarity criterion is satisfied, the pixel can be set to belong to the cluster as one or more of its neighbors. The selection of the similarity criterion is significant and the results are influenced by noise in all instances.

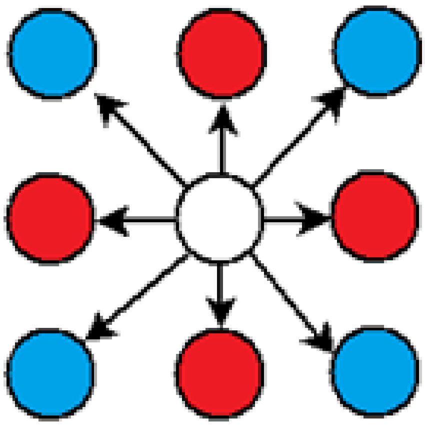


Figure 17. 8 connected neighborhood

The first step in region growing is to select a set of seed points. Seed point selection is based on some user criterion (for example, pixels in a certain grayscale range, pixels evenly spaced on a grid, etc.). The initial region begins as the exact location of these seeds.

The regions are then grown from these seed points to adjacent points depending on a region membership criterion. The criterion could be, for example, pixel intensity, grayscale texture, or color.

Since the regions are grown on the basis of the criterion, the image information itself is important. For example, if the criterion were a pixel intensity threshold value, knowledge of the histogram of the image would be of use, as one could use it to determine a suitable threshold value for the region membership criterion.

There is a very simple example followed below. Here we use 4-connected neighborhood to grow from the seed points. We can also choose 8-connected neighborhood for our pixels adjacent relationship. And the criteria we make here is the same pixel value. That is, we keep examining the adjacent pixels of seed points. If they have the same intensity value with the seed points, we classify them into the seed points. It is an iterated process until there are no change in two successive iterative stages. Of course, we can make other criteria, but the main goal is to classify the similarity of the image into regions.

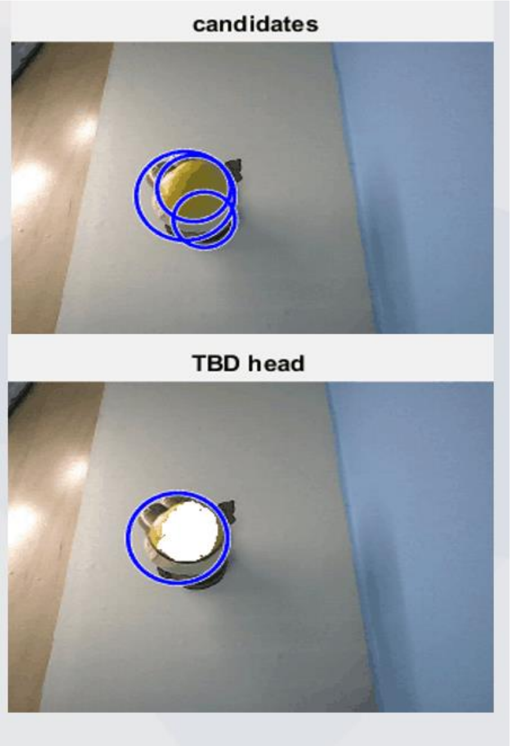


Figure 18. Image segmentation using region-growing method in real situation

4.3 Data Processing

In machine learning, **support vector machines** (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall [8].

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

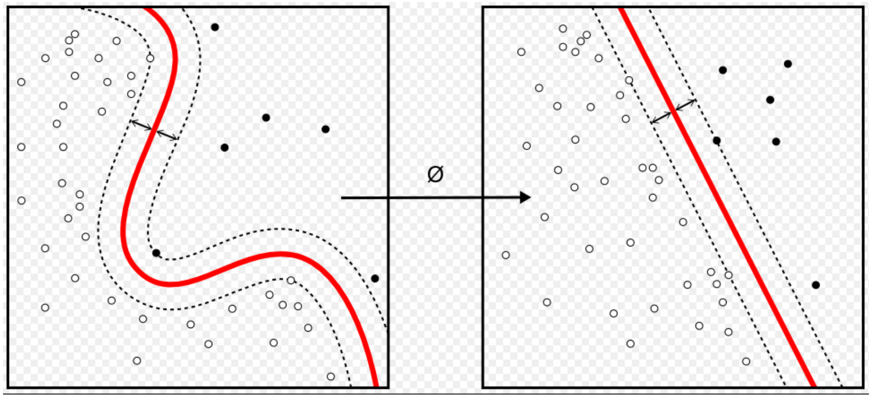


Figure 19. Non-linear classification using kernel

For two-class learning, if you specify the cost matrix C (see Cost), then the software updates the class prior probabilities p (see Prior) to pc by incorporating the penalties described in C.

Specifically, the SVM binary classification algorithm searches for an optimal hyperplane that separates the data into two classes. For separable classes, the optimal hyperplane maximizes a margin (space that does not contain any observations) surrounding itself, which creates boundaries for the positive and negative classes. For inseparable classes, the objective is the same, but the algorithm imposes a penalty on the length of the margin for every observation that is on the wrong side of its class boundary.

The linear SVM score function is

$$f(x) = x'\beta + b,$$

where:

x is an observation (corresponding to a row of X).

The vector β contains the coefficients that define an orthogonal vector to the hyperplane. For separable data, the optimal margin length is $2/\beta$.

b is the bias term.

More details are too complicated to be listed here.

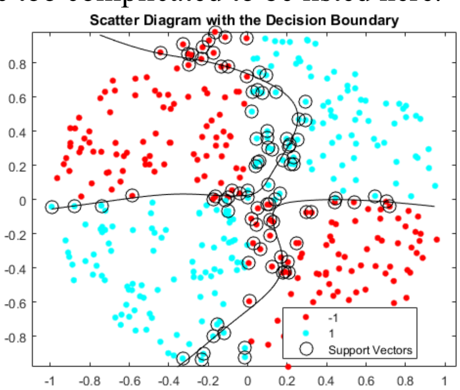


Figure 20. An example of non-linear binary two dimensional classification

Both dual soft-margin problems are quadratic programming problems. Internally, fitcsvm has several different algorithms for solving the problems.

For one-class or binary classification, if you do not set a fraction of expected outliers in the data, then the default solver is Sequential Minimal Optimization (SMO). SMO minimizes the one-norm problem by a series of two-point minimizations. During optimization, SMO respects the linear constraint

$$\sum \alpha_i y_i = 0$$

and explicitly includes the bias term in the model. SMO is relatively fast. For more details on SMO.

For binary classification, if you set a fraction of expected outliers in the data, then the default solver is the Iterative Single Data Algorithm. Like SMO, ISDA solves the one-norm problem. Unlike SMO, ISDA minimizes by a series on one-point minimizations, does not respect the linear constraint, and does not explicitly include the bias term in the model. For more details on ISDA.

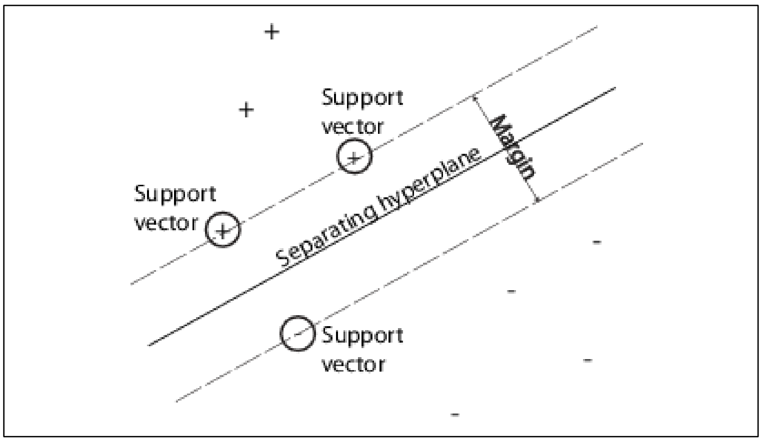


Figure 21. Illustration of support vectors and hypersurface

Some binary classification problems do not have a simple hyperplane as a useful separating criterion. For those problems, there is a variant of the

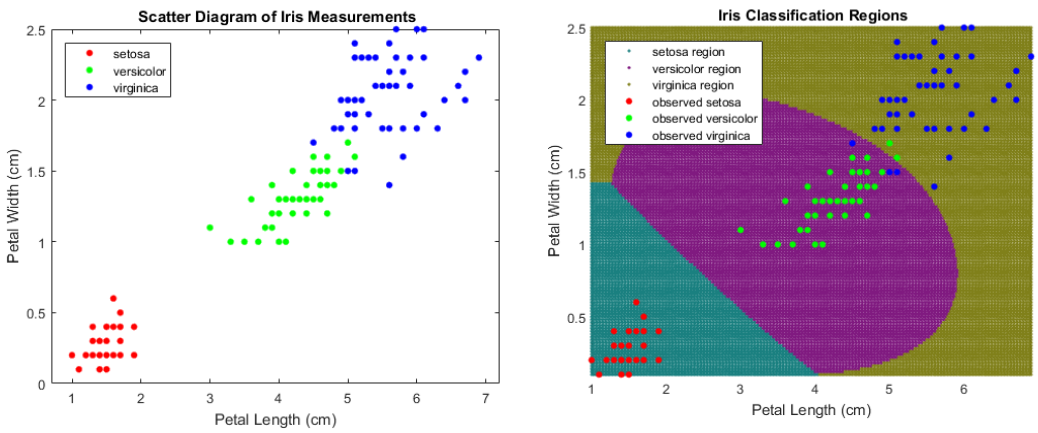


Figure 22. An example of non-linear ternary two dimensional classification

mathematical approach that retains nearly all the simplicity of an SVM separating hyperplane.

The mathematical approach using kernels relies on the computational method of hyperplanes. All the calculations for hyperplane classification use nothing more than dot products. Therefore, nonlinear kernels can use identical calculations and solution algorithms, and obtain classifiers that are nonlinear. The resulting classifiers are hypersurfaces in some space S, but the space S does not have to be identified or examined.

This approach uses these results from the theory of reproducing kernels [9]:

There is a class of functions $G(x_1,x_2)$ with the following property. There is a linear space S and a function φ mapping x to S such that

$$G(x_1,x_2) = <\varphi(x_1),\varphi(x_2)>.$$

The dot product takes place in the space S.

This class of functions includes:

Polynomials: For some positive integer p,

$$G(x_1,x_2) = (1 + x_1{'}x_2)p.$$

Radial basis function (Gaussian):

$$G(x_1,x_2) = \exp(-\| x_1-x_2 \|_2).$$

Multilayer perceptron or sigmoid (neural network): For a positive number p1 and a negative number p2,

$$G(x_1,x_2) = \tanh(p_1x_1{'}x_2 + p_2).$$

Here is an example of the data we measured using heads and an example of 3D boundary:

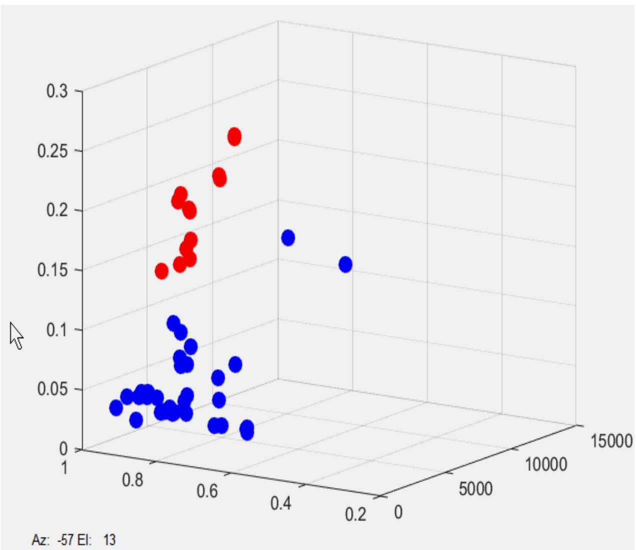


Figure 23. Scatter figure of real database (only choose three features)

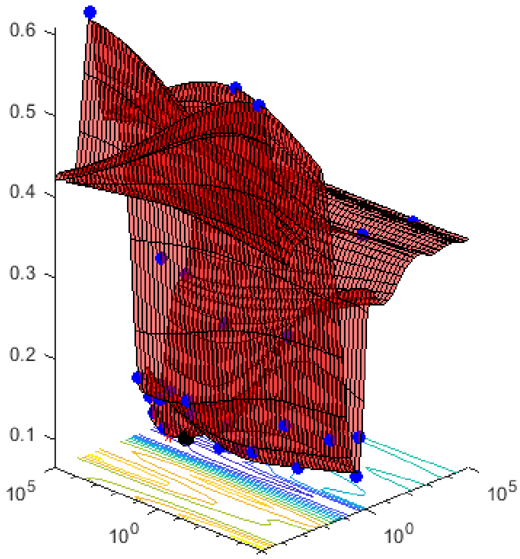


Figure 24. An example of three-dimensional classification

4.4 Website and the QR code system

To implement this function, we rent a virtual server at *aliyun.com* and build the backend system using *Laravel 5.0*. After the above system finishes measuring the number of candidates inside the box, it will automatically send the data to a remote server. The server validates the data and stores it in a remote database with MySQL.

To send data to the server, a security ID is needed along with the data. This prohibits unauthorized users sending data to the server and changing our data base. The data is automatically saved in the server when a post is sent to the server.

To view the data in a browser, a user sends a request to this server, which would then take out the data from the database and send it back to the browser. Data refreshing is a function included in HTML. The QR code can be obtained using the free QR code generators on the Internet.

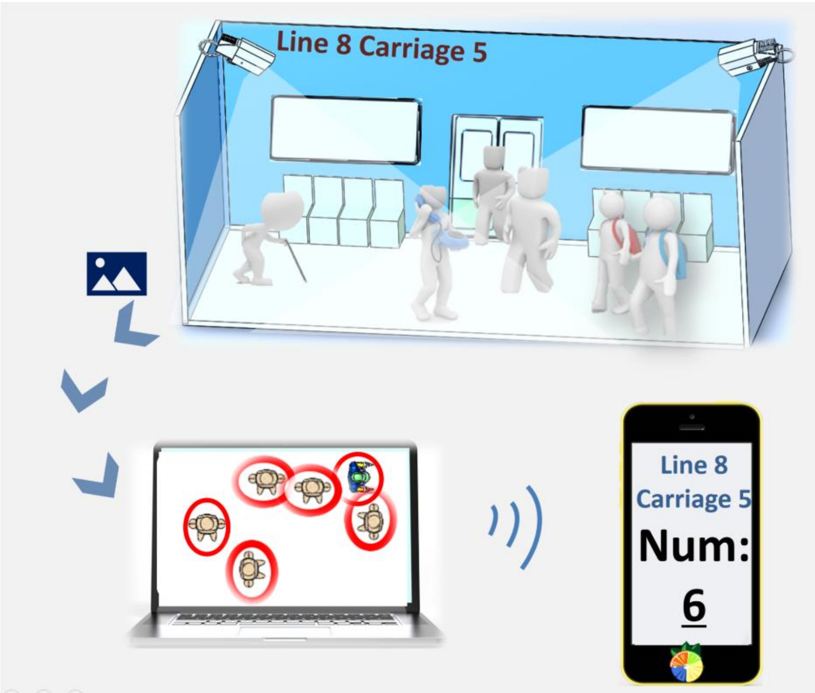


Figure 25. Concept diagram

5 Tasks

We have five tasks in all: market research, model assemblage, image processing, data transmission and data base improvement. **Task 2** and **Task 3** are basically applied to solve **Objective 1**, and **Task 4** is for **Objective 2**.

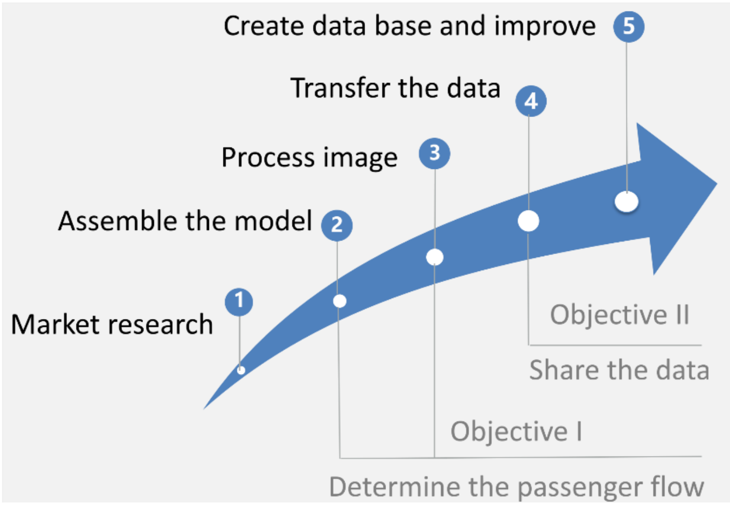


Figure 26. Flow chart for tasks

5.1 Market Research

We all know that the number of passengers in subways and metros are too large, but we do not know how large the number is exactly. To understand the real extent of the problem, we made a research online, and we obtained some very useful data. For example, from the data in Table 1 above and Figure 27 below, we can easily find that the daily metro passengers in Shanghai is more than 1/3 population of Shanghai. This exact number urged us to make a system to solve this current issue.

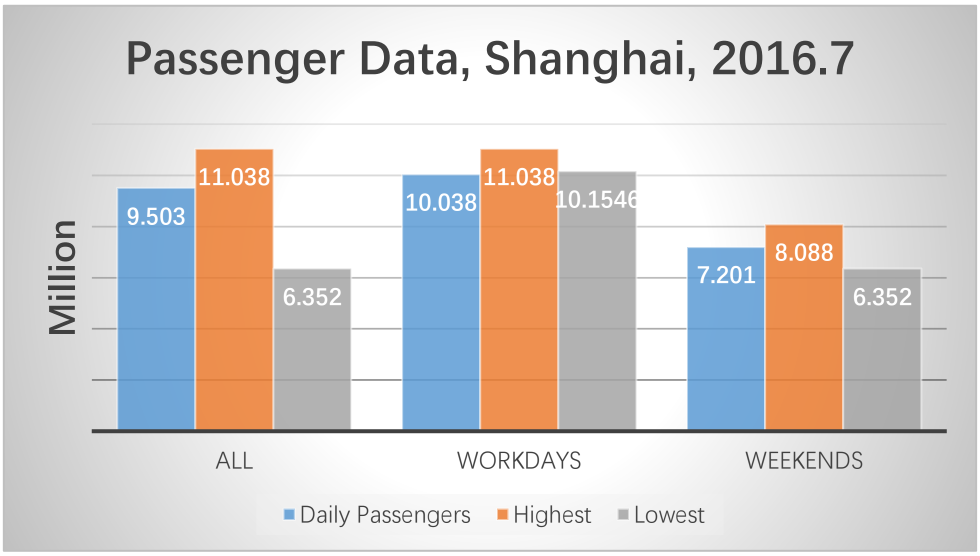


Figure 27. Passenger data we obtained

In the meanwhile, we also make a research on the current passenger data system. From Shanghai Metro, we do can find some kinds of passenger data, but not the exact data in each train and each carriage. We think the reason is that the current passenger data system in Shanghai is based on the ticket, not the real-time determination in each carriage. What we decide to achieve is a much more advanced system with better accuracy.

5.2 Model Assemblage: Camera and light source

To achieve our **Objective 1** (**Determine the accurate passenger flow**), we should first have the equipment to obtain the real-time picture of a carriage. Thus, we have two cameras prepared. In the other hand, we should also apply these

Figure 28. Camera we used

Figure 29. Dummy we used

camera to the real world, but the restriction of reality did not allow us to really do that. As a result, we must assemble a model carriage and prepare some dummies.

After assembling our model carriage, adding our cameras under the ceiling at front and back, and preparing all dummies, we found a new problem: shadows. As far as we have already designed to use "circle searching" (human heads can be approximately detected as circles), sometimes some round shadows can be detected as "heads". Our solution is to learn from astral lights and add light source from all directions, which can avoid the light influence from outer environment. In the same time, we made an on-the-spot investigation in subways, we found that the light source in metros is good enough so that there is hardly any shadows in subways.

5.3 Image processing: Process photos and identify "human" heads

To achieve our **Objective 1** (**Determine the accurate passenger flow**), we also have to process our images obtained from cameras by computer (Matlab). To

Figure 30. Overview of the model

integrate a picture of a whole carriage, we combine these two photos together and eliminate the coincide part. Then, we capture the objects (possible human heads) from it, and remove the background.

By judging the shape, size and color of those potential objects, we can make primary selection and identify those potential objects (all the principles have been shown above in **Solution**), which can possibly be human heads. Also, we need to compare our data with those in our data base. From this, our computer can detect all human heads in the carriage, and conclude the data. The establishment of the data base will be shown below in **Task 5**.

5.4 Data transmission: Analyze data and transfer

To achieve our **Objective 2** (**Share the data**), we need to transfer our data to

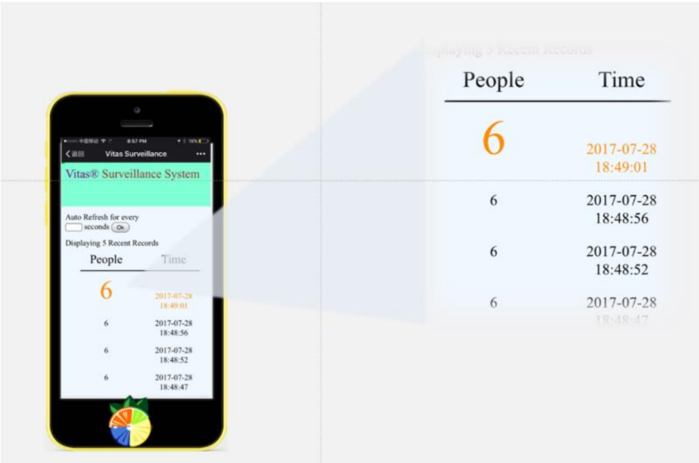


Figure 31. User interface

Figure 32. QR code to our website

the Internet. We have to obtain a website, and build a data base on the server. As we have mentioned above in Solution, we prepared a website and designed its interface.

Then we need to transfer our data to the server and save the data in the data base. We also need to present our data clearly in the browser. To give users easier access to the data, we need a QR code of the website. The Internet has been intelligent enough that the website can automatically generate a QR code for us.

5.5 Machine Learning: Create data base and Improvement

To achieve the best performance of **Task 3**, apart from identifying only based on objects' shape color and size, we introduce the cutting-edge technology —— machine learning. We need to create a data base for the models we use to test our algorithm, so that the identification will be more specific and accurate. By manually telling which primary identifications are correct, our computer will record these choices based on several specific features. After that, every time our system tries to make a final decision, it will be compared to the history data and delete the unmatched objects automatically. As long as our data base is large enough, the accuracy can be closer to 100%.

6 Schedule

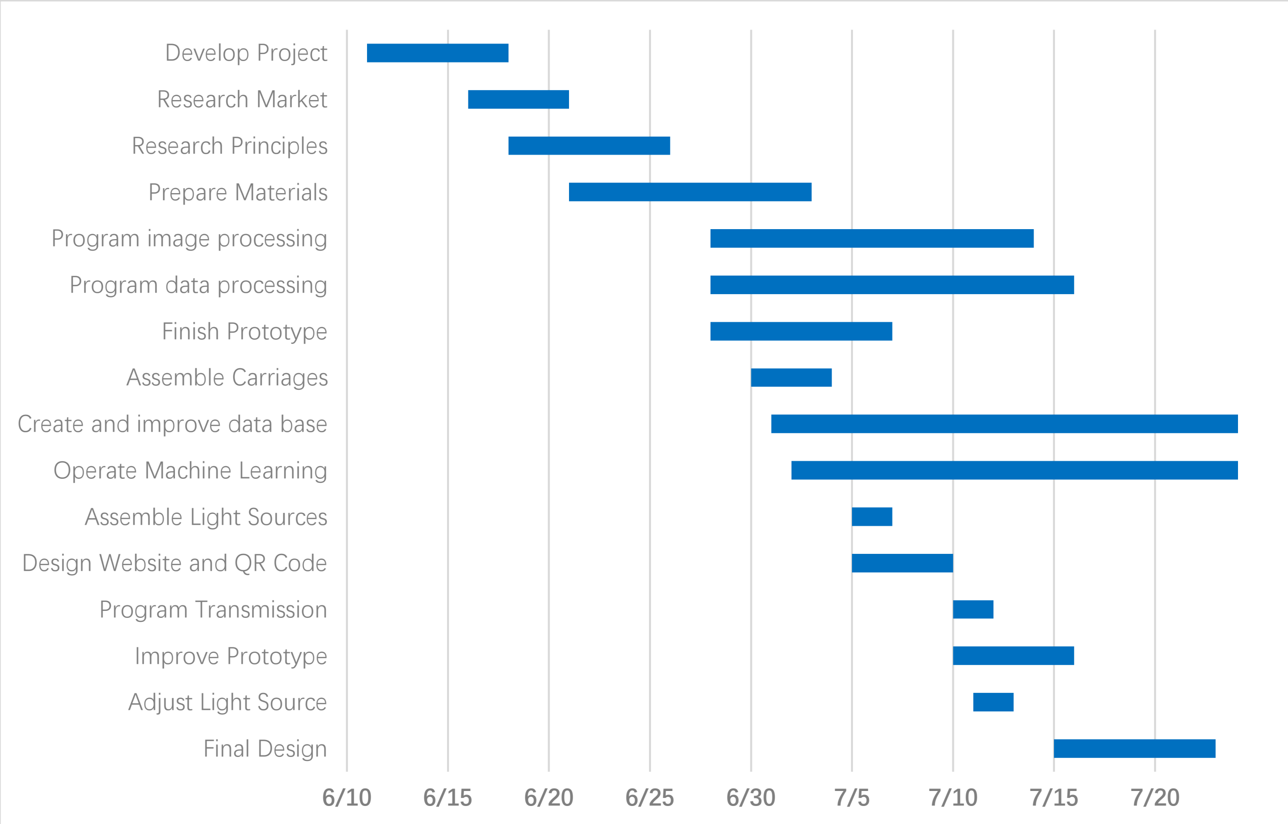


Figure 33. Gantt chart for our tasks

The Gantt chart above shows the all work we have done to make our project come true. Below, we will show five detailed Gantt charts to demonstrate five tasks in clear timetables.

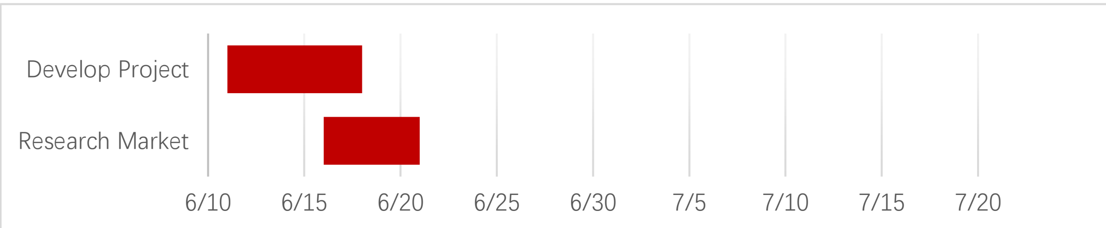


Figure 34. Gantt chart for Task 1

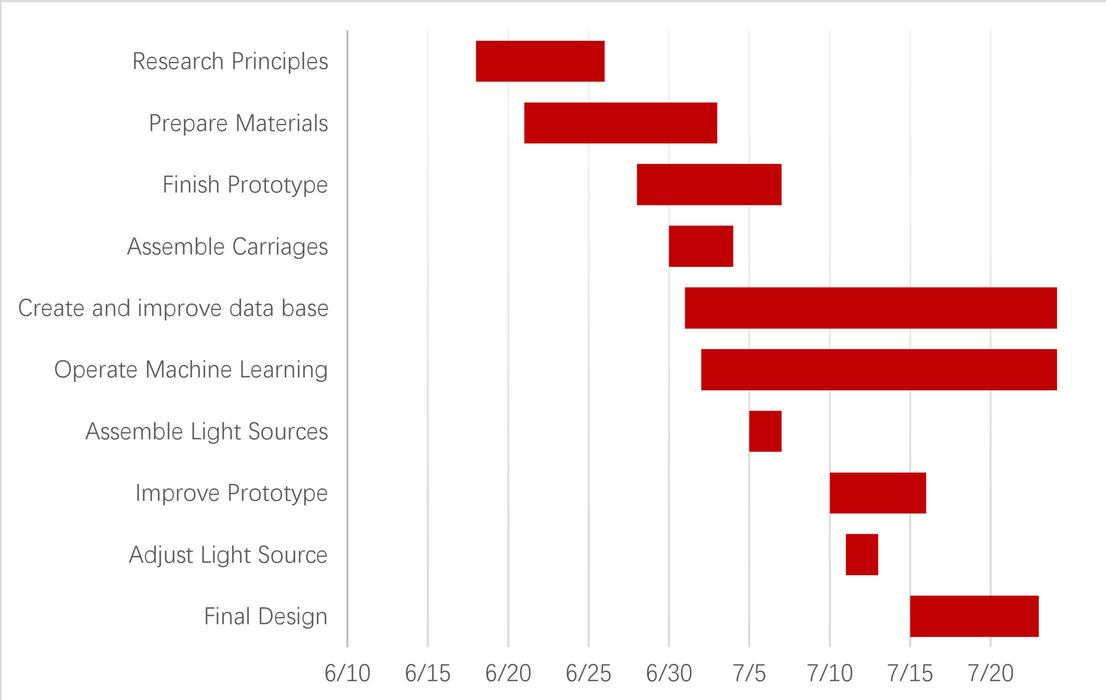


Figure 35. Gantt chart for Task 2

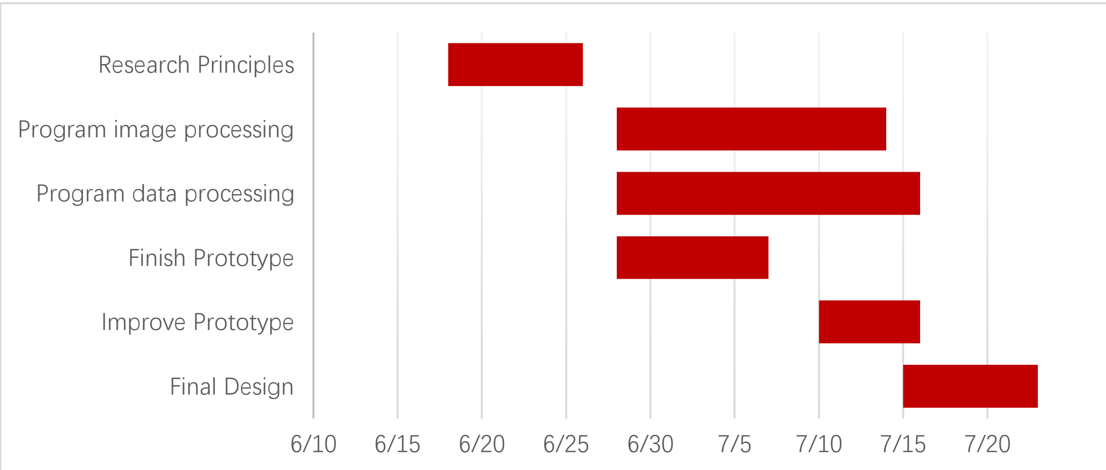


Figure 36. Gantt chart for Task 3

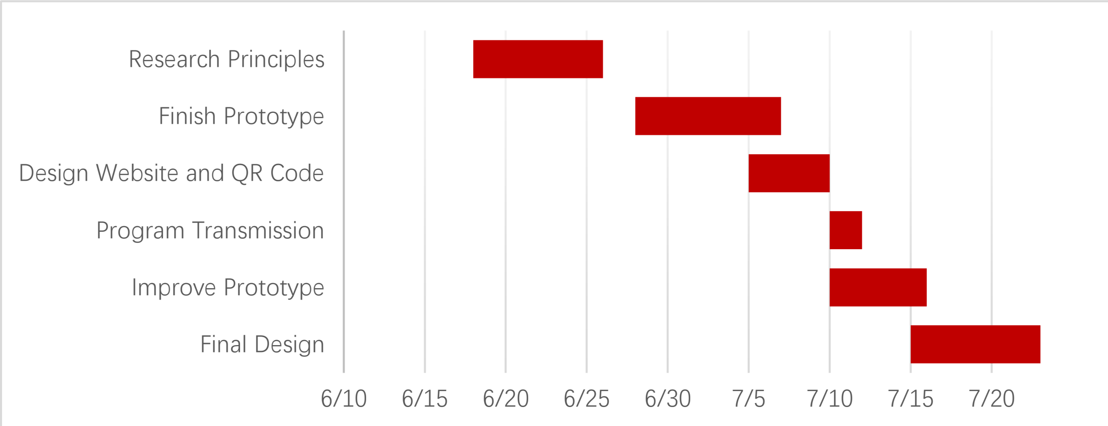


Figure 37. Gantt chart for Task 4

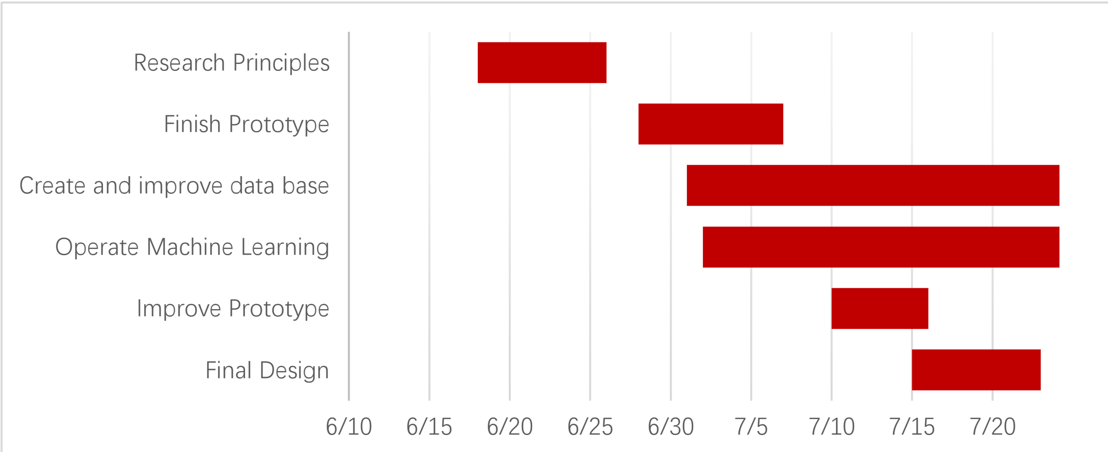


Figure 38. Gantt chart for Task 5

7 Budget

No.	Item	Specification	Quantity	Unit Price (￥)
1	**Camera**	**640×480**	**2**	**19.9**
2	LED tube	0.3m/4W	2	18.8
3	Acrylic board	300mm×250mm	2	11.67
4	Acrylic board	600mm×300mm	1	28
5	Acrylic board	600mm×250mm	2	23.33
6	Dummy model	6cm×6cm×9cm	15	8.4
			Total	301.4

Table 2. Budget

The main tools we use in the project are the 2 cameras, which costs 19.9 yuan each, and below that is the list of materials used to build the testing environment. Our project needs little money and mainly relies on programming. So it is highly economical.

8 Risks

Though we have worked a lot on this project, and our system does have a good performance, there still exists some risks.

First of all is still our greatest concern, accuracy. We do have a data base for this system, but it's only based on our own model carriage and dummies. Besides, our operations and experiments only have a limited amount. We cannot now ensure that we can detect the real-time numbers in real carriages about real people accurately. However, if we have more time and more resources to conduct operations on real metros, we believe that we can obtain real results as good as those from our model.

Another risk is about the light source. The change of light source might disturb our candidate finding process to a considerable extent. Too powerful light causes exposure and too weak light source add to shadows. Both would lower our accuracy of both finding the candidates and verifying them. What we need to do, if we have more time and resources, is to obtain the real data about the light source in the real carriages, so that we can modulate our cameras and the whole system.

9 Results and Potential Improvements

9.1 Results

Overall, our project has been done perfectly in such a short time. It has reach all our original needs, and it is also easy to transplant. It gives users the chance to adjust their own database to the environment. Although we have an original design to solve the problems about metros and their passengers, all kinds of users can also modulate the system so that it can meet the needs of their own. Therefore, it has a huge potential market since any crowded public places can use this method to balance the crowd.

We did the testing work and find out that when we input 20 groups of data of the same kind of environment, our accuracy of number counting reached around 90%. We have made strict algorithms to obtain numbers with relatively high accuracy. Considering the budget and the accuracy, our real-time determination system has a good performance-cost ratio.

Till now our project can find the candidates, determine the number of heads with respect to our model database, and upload the real-time data to our website. Customers can see the real-time data and historical data conveniently with smartphones by scanning our QR code.

9.2 Potential Improvements

1. About growth section, we are using Depth First Search (DFS) now. However, we meet with a problem that if we have too complicated figures or photos obtained from the cameras, those recursions in our code may have a risk to break up because the data obtained from DFS are too large for our PC to operate. If pictures as simple as dummies can cause this kind of problem, the data of real world can be even complicated. Therefore, we can try using Breadth First Search (BFS). This method can help us minimize the data. If we change our machine from PC to Super Computer, we believe our system can deal with all real cases.

2. Linear segmentation can have conflicts with Support Vector Machine, and those parameters must be modulated repeatedly. For time limit, we failed to spent that much time modulating. In the future, We can have further modulation on these parameters, so that we can have a better result.

3. Searching for circles (human heads) by Hough transform has a high demand on parameters and accuracy, but this method is the only one we can refer from now. If we have more time or more resources, we will try finding some new ways, or even create our own ways, to detect human heads more effectively.

4. The light source has a great influence on RGB color space. Though in real world, the light source in metros are steady and bright enough, we still hope that our system can meet the needs of all kinds of light environment. To solve this, we can try applying other forms of color space, like HSV.

5. RPG image de-noising is not that ideal like we have imagined before. With further searching on the Internet, we find another image de-noising method called wavelet de-noising. However, for the limit of time, we have tried using wavelet de-noising in our system, but it is still a very primary practice. We can make it more systematic in the future.

10 Key Personnel

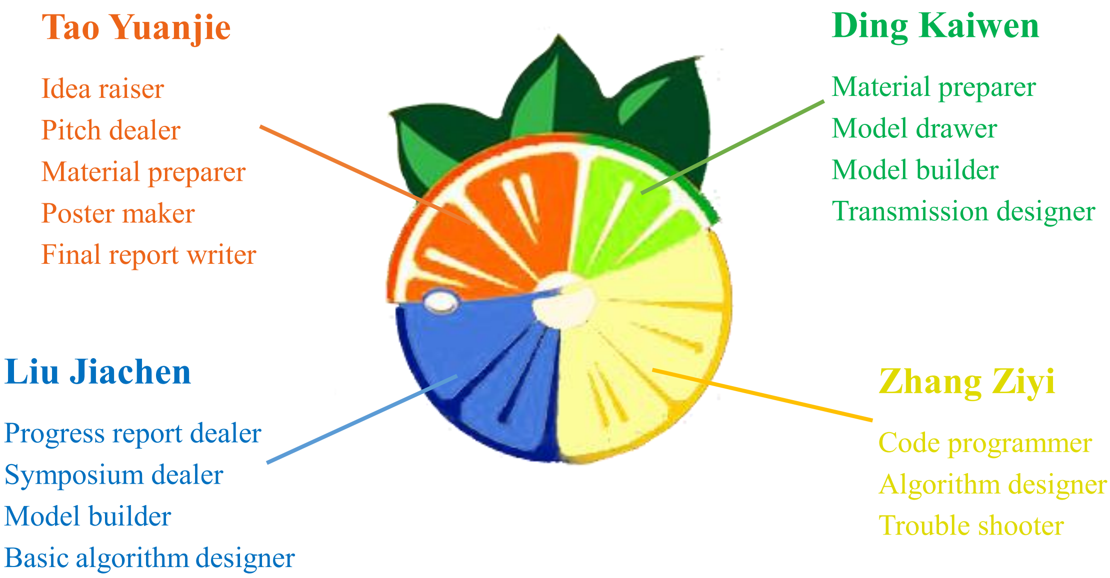


Figure 39. Key personnel

1. Chief Public Relationship Officer (CPRO) **Tao Yuanjie**:
   Idea raiser, Pitch dealer, Material preparer, Poster maker, Final report writer
2. Chief Technology Officer (CTO) **Zhang Ziyi**:
   Code programmer, Algorithm designer, Trouble shooter
3. Chief Design Officer (CDO) **Ding Kaiwen**:
   Material preparer, Model drawer, Model builder, Transmission designer
4. Chief Operation Officer (COO) **Liu Jiachen**:
   Progress report dealer, Symposium dealer, Model builder, Basic algorithm designer

11 References

[1] Chen, Zhaoxia, 2008, "The Application of Hough Transform in Human Head Detection." Journal of Jiamusi University (Natural Science Edition) 26.6, pp. 755-757, accessed at http://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFD2008&filename=JMDB200806009&uid=WEEvREcwSlJHSldRa1FhcEE0NXdoZ1FnWDl2UmcrSjhIWGIwd3pTakVRRT0=$

[2] Lei, Qiong, 2015, "Image segmentation based on Matlab." Electronic Design Engineering 23.21, pp. 167-169, accessed at http://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAST2015&filename=GWDZ201521056&uid=WEEvREcwSlJHSldRa1FhcEE0NXdoZ1FnWDl2UmcrSjhIWGIwd3pTakVRRT0=$

[3] Agoston, Max K, 2005, "Computer Graphics and Geometric Modeling: Implementation and Algorithms." London: Springer. pp. 300–306, accessed at http://www.doc88.com/p-0962950042382.html

[4] MacEvoy, Bruce, 2010, "Color Vision." Handprint.com, accessed at http://www.handprint.com/LS/CVS/color.html

[5] Limberger, F.A. and Oliveira, M.M., 2015, "Real-Time Detection of Planar Regions in Unorganized Point Clouds." Pattern Recognition 48 (6), pp. 2043–2053, accessed at https://www.researchgate.net/publication/272524452_Real-time_detection_of_planar_regions_in_unorganized_point_clouds

[6] Achtert, Elke, Böhm, Christian, David, Jörn, Kröger, Peer, and Zimek, Arthur, 2008, "Global Correlation Clustering Based on the Hough Transfer." Statistical Analysis and Data Mining 1 (3), pp. 111–127, accessed at https://www.researchgate.net/publication/220198488_Global_Correlation_Clustering_Based_on_the_Hough_Transform

[7] Theodoridis, Sergios, and Koutroumbas, Konstantinos, 2009, "Pattern Recognition." 4th Edition, Academic Press, accessed at http://xueshu.baidu.com/s?wd=paperuri%3A%285d0fb0176dc5578ae6f91cf98815c6b2%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fdl.acm.org%2Fcitation.cfm%3Fid%3D1457541&ie=utf-8&sc_us=9575189305164490673

[8] Fradkin, Dmitriy, and Muchnik, Ilya, 2006, "Support Vector Machines for Classification." in Abello, J., and Carmode, G., 2006, "Discrete Methods in Epidemiology.", DIMACS Series in Discrete Mathematics and Theoretical Computer Science, volume 70, pp. 13–20, accessed at https://www.researchgate.net/publication/228915009_Support_vector_machines_for_classification

[9] Forghani, M., Forouzanfar, M., and Teshnehlab, M, 2010, "Parameter optimization of improved fuzzy c-means clustering algorithm for brain MR image segmentation." Engineering Applications of Artificial Intelligence 23 (2), pp. 160–

168, accessed at https://www.researchgate.net/publication/220118867_Parameter_optimization_of_improved_fuzzy_c-means_clustering_algorithm_for_brain_MR_image_segmentation?ev=auth_pub

12 Appendix

12.1 Code

```
%-----------------------------
%           Project 2
%-----------------------------
shed_bg = 0.06;%0.034
SplitShed = 0.001;
SplitShed_denoise = 0.019;
cir_precision = 0.974;
precision_flag = true;
RadiiRatio = 0.2;
WaveletCheck = true;
flag = false;
ImproveMode = false;
MultiCam = true;
CircleInterRatio = 0.1;%0.22
rgb = false;
WebTransfer = true;
presentation = false;
Morecircles = true;
MorecirclesNum = 0.009;
WhetherCut = false;

dirc=[0 1;1 0;0 -1;-1 0];
m=480;
n=640;
for i=1:m
  for j=1:n
    White(i,j,:)=[0;0;0];
  end
end

vid1=videoinput('winvideo',1,'MJPG_640x480');
if MultiCam vid2=videoinput('winvideo',2,'MJPG_640x480');end
disp('Camera activated.');
choice1 = questdlg('Initialize background?','warning','Confirm','Negative','Negative');
switch choice1
 case 'Confirm'
  check1=false;
  check2=false
  while ~(check1==true && check2==true)
  choice2=questdlg('Get snapshot of background.','...','Confirm','Confirm');
  switch choice2
   case 'Confirm'
    if (check1==false)
     background=getsnapshot(vid1);
     background_hsv=rgb2hsv(background);
     background=im2double(background);
    end
    if (MultiCam && check2==false)
     background2=getsnapshot(vid2);
     background2_hsv=rgb2hsv(background2);
     background2=im2double(background2);
    end
    disp('Background saved.');
    %imshow_union(background,background2);
    if (check1==false)
     imshow(background);
     title('background TBD');
     choice3 = questdlg('Background for cam1 qualified?','...','Confirm','Again','Again');
     switch choice3
      case 'Again'
       check1=false;
      case 'Confirm'
       check1=true;
```

```
      end
    end
    if (MultiCam && check2==false)
      imshow(background2);
      title('background TBD');
      choice3 = questdlg('Background for cam2 qualified?','...','Confirm','Again','Again');
      switch choice3
        case 'Again'
          check2=false;
        case 'Confirm'
          check2=true;
      end
    end
  end
  end
 case 'Negative'
  %continue;
end

while (true)
 frame1=getsnapshot(vid1);
 frame1_hsv=rgb2hsv(frame1);
 frame1=im2double(frame1);
 frame1_cut=frame1;

 if MultiCam
  frame2=getsnapshot(vid2);
  if (presentation==false)
   if (WhetherCut)
    subplot(241),imshow(frame1),title('origin frame');
    subplot(245),imshow(frame2),title('origin frame');
   else
    subplot(231),imshow(frame1),title('origin frame');
    subplot(234),imshow(frame2),title('origin frame');
   end
  else
  end
  frame2_hsv=rgb2hsv(frame2);
  frame2=im2double(frame2);
  frame2_cut=frame2;
 else
  subplot(3,3,2),imshow(frame1),title('origin frame');
 end

 for i=1:m
  for j=1:n
   %
   %RGB cut 0.03
   diff=(frame1(i,j,1)-background(i,j,1))^2+(frame1(i,j,2)-background(i,j,2))^2+(frame1(i,j,3)-background(i,j,3))^2;
   if diff<shed_bg
    frame1_cut(i,j,:)=[255;255;255];
   end
  end
 end

 if MultiCam
  for i=1:m
   for j=1:n
    %RGB cut 0.03
    diff=(frame2(i,j,1)-background2(i,j,1))^2+(frame2(i,j,2)-background2(i,j,2))^2+(frame2(i,j,3)-background2(i,j,3))^2;
    if diff<shed_bg
     frame2_cut(i,j,:)=[255;255;255];
    end
   end
  end
  if (WhetherCut)
   subplot(242),imshow(frame1_cut),title('cut');
   subplot(246),imshow(frame2_cut),title('cut');
  end
 else
  subplot(3,3,3),imshow(frame1_cut),title('cut');
 end

 if (MultiCam)
  frame1_cut2=rgb2gray(frame1_cut);
  frame2_cut2=rgb2gray(frame2_cut);
  %d=imdistline;
  if precision_flag==false
   check=false;
   while check==false
    [centers, radii, metric] = imfindcircles(frame1_cut2,[30
```

```
65],'ObjectPolarity','dark','Sensitivity',cir_precision);
    [centers2, radii2, metric2] = imfindcircles(frame2_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
    if (WhetherCut)
    subplot(243),imshow(frame1),viscircles(centers, radii,'EdgeColor','b'),title('candidates');
    subplot(247),imshow(frame2),viscircles(centers2, radii2,'EdgeColor','b'),title('candidates');
    else
    subplot(232),imshow(frame1),viscircles(centers, radii,'EdgeColor','b'),title('candidates');
    subplot(235),imshow(frame2),viscircles(centers2, radii2,'EdgeColor','b'),title('candidates');
    end
    %subplot(3,3,4),imshow(frame1),viscircles(centers, radii,'EdgeColor','b');
    str = sprintf('Circle precision test.(%f)',cir_precision);
    choice2 = questdlg(str,'...','More precision','Less precision','Confirm','Confirm');
    switch choice2
     case 'Confirm'
      check=true;
      precision_flag=true;
     case 'More precision'
      cir_precision=cir_precision-0.001;
     case 'Less precision'
      cir_precision=cir_precision+0.001;
    end
   end
  else
   if (Morecircles==false)
     [centers, radii, metric] = imfindcircles(frame1_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
     [centers2, radii2, metric2] = imfindcircles(frame2_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
     subplot(243),imshow(frame1),viscircles(centers, radii,'EdgeColor','b'),title('candidates');
     subplot(247),imshow(frame2),viscircles(centers2, radii2,'EdgeColor','b'),title('candidates');
   else
     if (WhetherCut)
     [centers, radii, metric] = imfindcircles(frame1_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
     [centers2, radii2, metric2] = imfindcircles(frame2_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
     [centers_s, radii_s, metric_s] = imfindcircles(frame1,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision+MorecirclesNum);
     [centers2_s, radii2_s, metric2_s] = imfindcircles(frame2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision+MorecirclesNum);
     subplot(243),imshow(frame1),viscircles(centers_s, radii_s,'EdgeColor','b'),title('candidates');
     subplot(247),imshow(frame2),viscircles(centers2_s, radii2_s,'EdgeColor','b'),title('candidates');
     else
     [centers, radii, metric] = imfindcircles(frame1_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
     [centers2, radii2, metric2] = imfindcircles(frame2_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
     [centers_s, radii_s, metric_s] = imfindcircles(frame1,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision+MorecirclesNum);
     [centers2_s, radii2_s, metric2_s] = imfindcircles(frame2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision+MorecirclesNum);
     subplot(232),imshow(frame1),viscircles(centers_s, radii_s,'EdgeColor','b'),title('candidates');
     subplot(235),imshow(frame2),viscircles(centers2_s, radii2_s,'EdgeColor','b'),title('candidates');
     end
   end
  end
 else
  frame1_cut2=rgb2gray(frame1_cut);
```

```
  %d=imdistline;
  if precision_flag==false
    check=false;
    while check==false
      [centers, radii, metric] = imfindcircles(frame1_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
      subplot(3,3,4),imshow(frame1),viscircles(centers, radii,'EdgeColor','b');
      title('candidates');
      str = sprintf('Precision test.(%f)',cir_precision);
      choice2 = questdlg(str,'...','More precision','Less precision','Confirm','Confirm');
      switch choice2
        case 'Confirm'
          check=true;
          precision_flag=true;
        case 'More precision'
          cir_precision=cir_precision-0.001;
        case 'Less precision'
          cir_precision=cir_precision+0.001;
      end
    end
  else
    [centers, radii, metric] = imfindcircles(frame1_cut2,[30 65],'ObjectPolarity','dark','Sensitivity',cir_precision);
    subplot(3,3,4),imshow(frame1),viscircles(centers, radii,'EdgeColor','b');
    title('candidates');
  end
 end

 if WaveletCheck==true
  if MultiCam
    frame1_gray=rgb2gray(frame1);
    [x_frame,map]=gray2ind(frame1_gray,256);
    [thr,sorh,keepapp] = ddencmp('den','wv',x_frame);
    xd_frame = wdencmp('gbl',x_frame,'sym4',2,thr,sorh,keepapp);
    sm = size(map,1);

frame1_gray=ind2gray(wcodemat(xd_frame,sm),gray(256));
    frame2_gray=rgb2gray(frame2);
    [x_frame,map]=gray2ind(frame2_gray,256);
    [thr,sorh,keepapp] = ddencmp('den','wv',x_frame);
    xd_frame = wdencmp('gbl',x_frame,'sym4',2,thr,sorh,keepapp);
    sm = size(map,1);

frame2_gray=ind2gray(wcodemat(xd_frame,sm),gray(256));
  else
    frame1_gray=rgb2gray(frame1);
    [x_frame,map]=gray2ind(frame1_gray,256);

    [thr,sorh,keepapp] = ddencmp('den','wv',x_frame);
    xd_frame = wdencmp('gbl',x_frame,'sym4',2,thr,sorh,keepapp);
    sm = size(map,1);

frame1_gray=ind2gray(wcodemat(xd_frame,sm),gray(256));
    subplot(335), imshow(frame1_gray), title('Denoised');
  end
 end

 if (flag==false)
 choice1 = questdlg('Improve database?','...','Confirm','Negative','Negative');
 if strcmp(choice1,'Confirm')==1
  ImproveMode = true;
  fid=fopen('database.txt','a+t');
 else
  ImproveMode = false;
  fid=fopen('database.txt','r+');
 end
 end

 if (ImproveMode==false && flag==false)
  num=0;
  line=fgetl(fid);
  if (~rgb)
    while (~feof(fid))
    if (isempty(line)==true)
      line=fgetl(fid);
```

```
   continue;
  end
  num=num+1;
  [b,c]=strtok(line,' ');
  Group(num)=str2num(b);
  c=strtrim(c);
  [b,c]=strtok(c,' ');
  Pixelnum(num)=str2num(b);
  c=strtrim(c);
  [b,c]=strtok(c,' ');
  Col(num)=str2num(b);
  c=strtrim(c);
  Metric(num)=str2num(c);
  line=fgetl(fid);
  end
  fclose(fid);
 else %read rgb
  while (~feof(fid))
  if (isempty(line)==true)
   line=fgetl(fid);
   continue;
  end
  num=num+1;
  [b,c]=strtok(line,' ');
  Group(num)=str2num(b);
  c=strtrim(c);
  [b,c]=strtok(c,' ');
  Pixelnum(num)=str2num(b);
  c=strtrim(c);
  [b,c]=strtok(c,' ');
  Colr(num)=str2num(b);
  c=strtrim(c);
  [b,c]=strtok(c,' ');
  Colg(num)=str2num(b);
  c=strtrim(c);
  [b,c]=strtok(c,' ');
  Colb(num)=str2num(b);
  c=strtrim(c);
  Metric(num)=str2num(c);
  line=fgetl(fid);
  end
  fclose(fid);
 end

   if (~rgb)
    traindata=[Pixelnum',Col',Metric'];
   else
    traindata=[Pixelnum',Colr',Colg',Colb',Metric'];
   end

SVMModel=fitcsvm(traindata,Group,'Standardize',true,'K
ernelFunction','RBF','KernelScale','auto');
   disp('SVMModel trained.');
  end

  pixelnum=[];
  col=[];col_num=0;
  colr=[];colg=[];colb=[];
  if (flag==false)
   pause(2);
  end

  if MultiCam
   num=1;
   while num<=size(centers,1)
   frame1_gr=frame1;
   pixelnum=[pixelnum 1];
   col=[col 0];col_num=0;
   colr=[colr 0];
   colg=[colg 0];
   colb=[colb 0];
    for i=1:m
     for j=1:n
      CheckMatrix(i,j)=false;
     end
    end
    for                                    i=-
round(radii(num)*RadiiRatio):round(radii(num)*RadiiRatio)
     for                                   j=-
round(radii(num)*RadiiRatio):round(radii(num)*RadiiRatio)
      x = round(centers(num,2))+i;
      y = round(centers(num,1))+j;
      if
(x>0)&&(y>0)&&(x<m+1)&&(y<n+1)&&(CheckMatrix(x,
y)==false)
       listx=[x];
```

```
        listy=[y];
        head=1;
        tail=1;
        x0=x;y0=y;
        while head<=tail
          x=listx(head);
          y=listy(head);
          for k=1:4
            tx=listx(head)+dirc(k,1);
            ty=listy(head)+dirc(k,2);
            if (tx<1)||(ty<1)||(tx>m)||(ty>n)||((tx-x0)^2+(ty-y0)^2>1*radii(num).^2)...
              ||(CheckMatrix(tx,ty)==true) continue;end
            if ((frame1_cut(tx,ty,1)==255)&&(frame1_cut(tx,ty,2)==255)&&(frame1_cut(tx,ty,3)==255)) continue;end

            if WaveletCheck==false
              diff=(frame1(x,y,1)-frame1(tx,ty,1))^2+(frame1(x,y,2)-frame1(tx,ty,2))^2+(frame1(x,y,3)-frame1(tx,ty,3))^2;
              if (diff>SplitShed) continue;end
            else
              diff=abs(frame1_gray(x,y)-frame1_gray(tx,ty));
              if (diff>SplitShed_denoise) continue;end
            end
            CheckMatrix(tx,ty)=true;
            frame1_gr(tx,ty,:)=[255;0;0];
            tail=tail+1;
            listx=[listx tx];
            listy=[listy ty];
            frame1_gr(tx,ty,:)=[255;0;0];
            pixelnum(num)=pixelnum(num)+1;

col(num)=(col(num)*col_num+frame1_cut2(tx,ty))/(col_num+1);

colr(num)=(colr(num)*col_num+frame1(tx,ty,1))/(col_num+1);

colg(num)=(colg(num)*col_num+frame1(tx,ty,2))/(col_num+1);
colb(num)=(colb(num)*col_num+frame1(tx,ty,3))/(col_num+1);
              col_num=col_num+1;
            end
            head=head+1;
          end
        end
      end
    end
    if (ImproveMode==true)
      subplot(248),imshow(White);

subplot(244),imshow(frame1_gr),viscircles(centers(num,:),radii(num),'EdgeColor','b'),title('TBD head');
      choice4 = questdlg('Is this candidate a head?','...','Yes','No','Pass','Pass');
      switch choice4
        case 'Yes'
          group(num)=1;
          if (~rgb)

fprintf(fid,'%d %f %f %f\n',1,pixelnum(num),col(num),metric(num));
          else

fprintf(fid,'%d %f %f %f %f %f\n',1,pixelnum(num),colr(num),colg(num),colb(num),metric(num));
          end
        case 'No'
          group(num)=0;
          if (~rgb)

fprintf(fid,'%d %f %f %f\n',0,pixelnum(num),col(num),metric(num));
          else

fprintf(fid,'%d %f %f %f %f %f\n',0,pixelnum(num),colr(num),colg(num),colb(num),metric(num));
          end
        case 'Pass'
      end
    end
    num=num+1;
```

```
  end
  pixelnum2=[];
  col2=[];col2_num=0;
  colr2=[];
  colg2=[];
  colb2=[];

  num2=1;
  while num2<=size(centers2,1)
  frame2_gr=frame2;
  pixelnum2=[pixelnum2 1];
  col2=[col2 0];col2_num=0;
  colr2=[colr2 0];
  colg2=[colg2 0];
  colb2=[colb2 0];
   for i=1:m
    for j=1:n
     CheckMatrix(i,j)=false;
    end
   end
   for                                              i=-
round(radii2(num2)*RadiiRatio):round(radii2(num2)*Radii
Ratio)
    for                                             j=-
round(radii2(num2)*RadiiRatio):round(radii2(num2)*Radii
Ratio)
     x = round(centers2(num2,2))+i;
     y = round(centers2(num2,1))+j;
     if
(x>0)&&(y>0)&&(x<m+1)&&(y<n+1)&&(CheckMatrix(x,
y)==false)
      listx=[x];
      listy=[y];
      head=1;
      tail=1;
      x0=x;y0=y;
      while head<=tail
       x=listx(head);
       y=listy(head);
       for k=1:4
        tx=listx(head)+dirc(k,1);
        ty=listy(head)+dirc(k,2);
        if   (tx<1)||(ty<1)||(tx>m)||(ty>n)||((tx-x0)^2+(ty-
y0)^2>1*radii2(num2).^2)...
         ||(CheckMatrix(tx,ty)==true) continue;end
        if
((frame2_cut(tx,ty,1)==255)&&(frame2_cut(tx,ty,2)==255)
&&(frame2_cut(tx,ty,3)==255)) continue;end

        if WaveletCheck==false
         diff=(frame2(x,y,1)-
frame2(tx,ty,1))^2+(frame2(x,y,2)-
frame2(tx,ty,2))^2+(frame2(x,y,3)-frame2(tx,ty,3))^2;
         if (diff>SplitShed) continue;end
        else
         diff=abs(frame2_gray(x,y)-frame2_gray(tx,ty));
         if (diff>SplitShed_denoise) continue;end
        end
        CheckMatrix(tx,ty)=true;
        frame2_gr(tx,ty,:)=[255;0;0];
        tail=tail+1;
        listx=[listx tx];
        listy=[listy ty];
        frame2_gr(tx,ty,:)=[255;0;0];
        pixelnum2(num2)=pixelnum2(num2)+1;

col2(num2)=(col2(num2)*col2_num+frame2_cut2(tx,ty))/(
col2_num+1);

colr2(num2)=(colr2(num2)*col2_num+frame2(tx,ty,1))/(co
l2_num+1);

colg2(num2)=(colg2(num2)*col2_num+frame2(tx,ty,2))/(c
ol2_num+1);

colb2(num2)=(colb2(num2)*col2_num+frame2(tx,ty,3))/(c
ol2_num+1);
        col2_num=col2_num+1;
       end
       head=head+1;
      end
     end
    end
   end
   if (ImproveMode==true)
    subplot(244),imshow(White);
```

```
subplot(248),imshow(frame2_gr),viscircles(centers2(num2
,:), radii2(num2),'EdgeColor','b'),title('TBD head');
    choice4 = questdlg('Is this candidate a
head?','...','Yes','No','Pass','Pass');
    switch choice4
     case 'Yes'
      group2(num2)=1;
      if (~rgb)

fprintf(fid,'%d %f %f %f\n',1,pixelnum2(num2),col2(num2),m
etric2(num2));
      else

fprintf(fid,'%d %f %f %f %f %f\n',1,pixelnum2(num2),colr2(nu
m2),colg2(num2),colb2(num2),metric2(num2));
      end
     case 'No'
      group2(num2)=0;
      if (~rgb)

fprintf(fid,'%d %f %f %f\n',0,pixelnum2(num2),col2(num2),m
etric2(num2));
      else

fprintf(fid,'%d %f %f %f %f %f\n',0,pixelnum2(num2),colr2(nu
m2),colg2(num2),colb2(num2),metric2(num2));
      end
     case 'Pass'
    end
   end
   num2=num2+1;
  end
 else
  num=1;
  while num<=size(centers,1)
  frame1_gr=frame1;
  pixelnum=[pixelnum 1];
  col=[col 0];col_num=0;
  colr=[colr 0];
  colg=[colg 0];
  colb=[colb 0];
   for i=1:m
    for j=1:n
     CheckMatrix(i,j)=false;
    end
   end
   for i=-
round(radii(num)*RadiiRatio):round(radii(num)*RadiiRatio)
    for j=-
round(radii(num)*RadiiRatio):round(radii(num)*RadiiRatio)
     x = round(centers(num,2))+i;
     y = round(centers(num,1))+j;
     if
(x>0)&&(y>0)&&(x<m+1)&&(y<n+1)&&(CheckMatrix(x,
y)==false)
      listx=[x];
      listy=[y];
      head=1;
      tail=1;
      x0=x;y0=y;
      while head<=tail
       x=listx(head);
       y=listy(head);
       for k=1:4
        tx=listx(head)+dirc(k,1);
        ty=listy(head)+dirc(k,2);
        if (tx<1)||(ty<1)||(tx>m)||(ty>n)||((tx-x0)^2+(ty-
y0)^2>1*radii(num).^2)...
         ||(CheckMatrix(tx,ty)==true) continue;end
        if
((frame1_cut(tx,ty,1)==255)&&(frame1_cut(tx,ty,2)==255)
&&(frame1_cut(tx,ty,3)==255)) continue;end

        if WaveletCheck==false
         diff=(frame1(x,y,1)-
frame1(tx,ty,1))^2+(frame1(x,y,2)-
frame1(tx,ty,2))^2+(frame1(x,y,3)-frame1(tx,ty,3))^2;
         if (diff>SplitShed) continue;end
        else
         diff=abs(frame1_gray(x,y)-frame1_gray(tx,ty));
         if (diff>SplitShed_denoise) continue;end
        end

        CheckMatrix(tx,ty)=true;
        frame1_gr(tx,ty,:)=[255;0;0];
```

```
            tail=tail+1;
            listx=[listx tx];
            listy=[listy ty];
            frame1_gr(tx,ty,:)=[255;0;0];
            pixelnum(num)=pixelnum(num)+1;

col(num)=(col(num)*col_num+frame1_cut2(tx,ty))/(col_nu
m+1);

colr(num)=(colr(num)*col_num+frame1(tx,ty,1))/(col_num
+1);

colg(num)=(colg(num)*col_num+frame1(tx,ty,2))/(col_nu
m+1);

colb(num)=(colb(num)*col_num+frame1(tx,ty,3))/(col_nu
m+1);
            col_num=col_num+1;
          end
          head=head+1;
        end
      end
    end
  end
  if (ImproveMode==true)

subplot(336),imshow(frame1_gr),viscircles(centers(num,:),
radii(num),'EdgeColor','b');
    title('TBD head');
    choice4 = questdlg('Is this candidate a
head?','...','Yes','No','Pass','Pass');
    switch choice4
      case 'Yes'
        group(num)=1;
        if (~rgb)

fprintf(fid,'%d %f %f %f\n',1,pixelnum(num),col(num),metric(
num));
        else

fprintf(fid,'%d %f %f %f %f %f\n',1,pixelnum(num),colr(num),c
olg(num),colb(num),metric(num));
        end
      case 'No'
        group(num)=0;
        if (~rgb)

fprintf(fid,'%d %f %f %f\n',0,pixelnum(num),col(num),metric(
num));
        else

fprintf(fid,'%d %f %f %f %f %f\n',0,pixelnum(num),colr(num),c
olg(num),colb(num),metric(num));
        end
      case 'Pass'
    end
  end
  num=num+1;
 end
end

ConfirmNum=0;
if MultiCam
  if (ImproveMode==false && ~isempty(metric))
    if (~rgb)
      Testdata=[pixelnum',col',metric];
    else
      Testdata=[pixelnum',colr',colg',colb',metric];
    end
    [group,confd]=predict(SVMModel,Testdata);
    for i=1:length(group)
      if (centers(i,1)<radii(i) || n-centers(i,1)<radii(i) ||
centers(i,2)<radii(i) || m-centers(i,2)<radii(i))
        group(i)=0;
      end
    end

    for i=1:length(group)-1
      for j=i+1:length(group)
        if (group(i)==0 || group(j)==0)
          continue;
        end
        dis=sqrt((centers(i,1)-centers(j,1))^2+(centers(i,2)-
centers(j,2))^2);
        if dis>radii(i)+radii(j) continue;end
        if radii(i)<radii(j) radiiSmall=radii(i); else
```

```
radiiSmall=radii(j);end
        if dis<abs(radii(i)-radii(j))
         if radii(i)<radii(j)
           group(i)=0;
         else
           group(j)=0;
         end
        end
        theta1=acos((radii(i)^2+dis^2-
radii(j)^2)/(2*radii(i)*dis));
        theta2=acos((radii(j)^2+dis^2-
radii(i)^2)/(2*radii(j)*dis));
        Sinter=theta1*radii(i)^2+theta2*radii(j)^2-
0.5*radii(i)^2*sin(2*theta1)-0.5*radii(j)^2*sin(2*theta2);
        if (Sinter/(pi*radiiSmall^2)>CircleInterRatio)
         if abs(confd(i))<abs(confd(j))
           group(i)=0;
         else
           group(j)=0;
         end
        end
      end
    end

    if (WhetherCut)
    subplot(244),imshow(frame1);
    else
    subplot(233),imshow(frame1);
    end
    for i=1:length(group)
      if group(i)==1
        ConfirmNum=ConfirmNum+1;
        viscircles(centers(i,:), radii(i),'EdgeColor','b');
      end
    end
    title('detected');
  end

  % for another cam

  if (ImproveMode==false && ~isempty(metric2))
    if (~rgb)
      Testdata=[pixelnum2',col2',metric2];
    else
      Testdata=[pixelnum2',colr2',colg2',colb2',metric2];
    end
    [group,confd]=predict(SVMModel,Testdata);

    for i=1:length(group)
      if (centers2(i,1)<radii2(i) || n-centers2(i,1)<radii2(i) ||
centers2(i,2)<radii2(i) || m-centers2(i,2)<radii2(i))
        group(i)=0;
      end
    end

    for i=1:length(group)-1
      for j=i+1:length(group)
        if (group(i)==0 || group(j)==0)
          continue;
        end
        dis=sqrt((centers2(i,1)-
centers2(j,1))^2+(centers2(i,2)-centers2(j,2))^2);
        if dis>radii2(i)+radii2(j) continue;end
        if radii2(i)<radii2(j) radiiSmall=radii2(i); else
radiiSmall=radii2(j);end
        if dis<abs(radii2(i)-radii2(j))
         if radii2(i)<radii2(j)
           group(i)=0;
         else
           group(j)=0;
         end
        end
        theta1=acos((radii2(i)^2+dis^2-
radii2(j)^2)/(2*radii2(i)*dis));
        theta2=acos((radii2(j)^2+dis^2-
radii2(i)^2)/(2*radii2(j)*dis));
        Sinter=theta1*radii2(i)^2+theta2*radii2(j)^2-
0.5*radii2(i)^2*sin(2*theta1)-0.5*radii2(j)^2*sin(2*theta2);
        if (Sinter/(pi*radiiSmall^2)>CircleInterRatio)
         if abs(confd(i))<abs(confd(j))
           group(i)=0;
         else
           group(j)=0;
         end
        end
      end
```

```
    end

    if (WhetherCut)
    subplot(248),imshow(frame2);
    else
    subplot(236),imshow(frame2);
    end
    for i=1:length(group)
     if group(i)==1
       ConfirmNum=ConfirmNum+1;
       viscircles(centers2(i,:), radii2(i),'EdgeColor','b');
     end
    end
    title('detected');
   end

 else %multicam split line

   if (ImproveMode==false && ~isempty(metric))
    if (~rgb)
     Testdata=[pixelnum',col',metric];
    else
     Testdata=[pixelnum',colr',colg',colb',metric];
    end
    [group,confd]=predict(SVMModel,Testdata);

    for i=1:length(group)
     if (centers(i,1)<radii(i) || n-centers(i,1)<radii(i) ||
centers(i,2)<radii(i) || m-centers(i,2)<radii(i))
       group(i)=0;
     end
    end

    for i=1:length(group)-1
     for j=i+1:length(group)
       if (group(i)==0 || group(j)==0)
        continue;
       end
       dis=sqrt((centers(i,1)-centers(j,1))^2+(centers(i,2)-
centers(j,2))^2);
       if dis>radii(i)+radii(j) continue;end
       if radii(i)<radii(j) radiiSmall=radii(i); else
radiiSmall=radii(j);end
       if dis<abs(radii(i)-radii(j))
        if radii(i)<radii(j)
         group(i)=0;
        else
         group(j)=0;
        end
       end
       theta1=acos((radii(i)^2+dis^2-
radii(j)^2)/(2*radii(i)*dis));
       theta2=acos((radii(j)^2+dis^2-
radii(i)^2)/(2*radii(j)*dis));
       Sinter=theta1*radii(i)^2+theta2*radii(j)^2-
0.5*radii(i)^2*sin(2*theta1)-0.5*radii(j)^2*sin(2*theta2);
       if (Sinter/(pi*radiiSmall^2)>CircleInterRatio)
        if abs(confd(i))<abs(confd(j))
         group(i)=0;
        else
         group(j)=0;
        end
       end
      end
     end

     subplot(336),imshow(frame1);
     for i=1:length(group)
      if group(i)==1
       ConfirmNum=ConfirmNum+1;
       viscircles(centers(i,:), radii(i),'EdgeColor','b');
      end
     end
     title('detected');
    end
   end %multicam end

   if (ImproveMode==true)
    choice4 = questdlg('One more
frame?','...','Yes','Quit','Quit');
    if strcmp(choice4,'Quit')==1
     fprintf(fid,'\n');
     fclose(fid);
     return;
    end
   end
```

```
 if (WebTransfer==true)
   if (MultiCam==true)
     k=num2str(ConfirmNum);
     url = 'http://vitas.runtianz.cn/?ssid=cbdptbtptpbcptdtptp&data=';
     url = strcat(url,k);
     options = weboptions('RequestMethod', 'post');
     webread(url);
   else
     k=num2str(ConfirmNum);
     url = 'http://vitas.runtianz.cn/?ssid=cbdptbtptpbcptdtptp&data=';
     url = strcat(url,k);
     options = weboptions('RequestMethod', 'post');
     webread(url);
   end
 end

 flag=true;
end

function imshow_union(b1,b2)
 b2=fliplr(b2);
 b2=flipud(b2);
 imshow([b2;b1]);
end
```

12.2 Hyperlinks for Purchasing Materials

1. Camera
   https://item.taobao.com/item.htm?ut_sk=1.V25bBZL7jXkDAAnE8we3eGYa_21380790_1500451173743.ScreenShot.1&id=520872546183&sourceType=item&cpp=1&shareurl=true&spm=a313p.22.2b0.55136909116&short_name=h.NkPyoc&cv=8eZF00DkIYQ&app=chrome
2. LED tube
   https://detail.tmall.com/item.htm?spm=a230r.1.14.6.ebb2eb2z6Q98t&id=520001503981&cm_id=140105335569ed55e27b&abbucket=16&sku_properties=122216547:10122
3. Acrylic board
   https://item.taobao.com/item.htm?spm=a230r.1.14.94.ebb2eb2CByqcr&id=13603291616&ns=1&abbucket=16#detail
4. Dummy model
   https://item.taobao.com/item.htm?ut_sk=1.V25bBZL7jXkDAAnE8we3eGYa_21380790_1500453785496.ScreenShot.1&id=550634855702&sourceType=item&cpp=1&shareurl=true&spm=a313p.22.2je.55146598597&short_name=h.NPZXSA&cv=9VYU00DAW4Q&app=chrome