CSIT 6910A Report

Recycling Symbol Recognition System

Student: CHEN Xutong

Supervisor: Prof. David Rossiter

Table of Contents

1 Introduction

1.1 Overview

In recent years, waste sorting gains growing popularity because of its benefits to the environment and recycling. According to statistics, 63% of the waste in Japan, where waste sorting prevails not only in metropolis but among rural areas, has been recycled and reused through the application of recycling bins. In some other developed countries, waste sorting has been also promoted by the government in various ways, such as booklets, broadcast and TV shows. Nevertheless, people often lack the awareness or knowledge of the right bins, which they should put the rubbish into. With this the camera and the system installed to the recycling bins, people would be guided with ease to the right one. The explicit guidance and automatic system is of great significance to people's motivation to make their contributions for our environment protection. In addition, this system will bring a profound impact on conservation of limited resources so as to realize sustainable development.

With the trend that rubbish sorting and recycling is becoming increasingly popular, this project involves building a system to help people put the rubbish into the right recycling bins. We could set up a camera beside the bins as an independent device or install a camera at the top of the bins. People can scan the recycling symbol image with this camera to find out the right recycling bin into which to put the rubbish.

1.2 Objective

The objective of this project is to develop software, which can help users recognize the recycling symbols and point out the correct garbage to throw the rubbish. The other objective of this project is to help users do study on the images with their features, contours and convex hulls.

2 Preparation

2.1 Development Environment

2.1.1 Operation System

This project is developed on MacOS X 10.8.4, and is also tested in MacOS X 10.7.

2.1.2 Integrated Development Environment & Development Language

This project is developed with two IDEs:

- The backstage interaction is developed with Xcode.
- The GUI is developed with QtCreator.

The main development language is C++.

2.1.3 Framework

- OpenCV: Handling the image processing
- Graphviz: Support graphs on different platforms
- ImageMagick-6.7.6: Support some image features on different platforms

2.2 Core Algorithm

2.2.1 SIFT - Scale Invariant Feature Transforms.

Scale-invariant feature transform (or SIFT) is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by David Lowe in 1999.

SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches. The determination of consistent clusters is performed rapidly by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally the probability that a particular set of features indicates the presence of an object is computed, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

2.2.2 SVM – Support Vector Machine

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories; an SVM training algorithm builds a model that assigns new examples into one category or the other. 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 on.

3 Design

3.1 Structure

The design pattern used by this system is Model – View – Controller. Model–View–Controller is a software architecture pattern that separates the representation of information from the user's interaction with it. The model consists of application data, business rules, logic, and functions. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a bar chart for management and a tabular view for accountants. The controller mediates input, converting it to commands for the model or view. The central ideas behind MVC are code reusability and separation of concerns.

The picture below is the main structure of each mode.

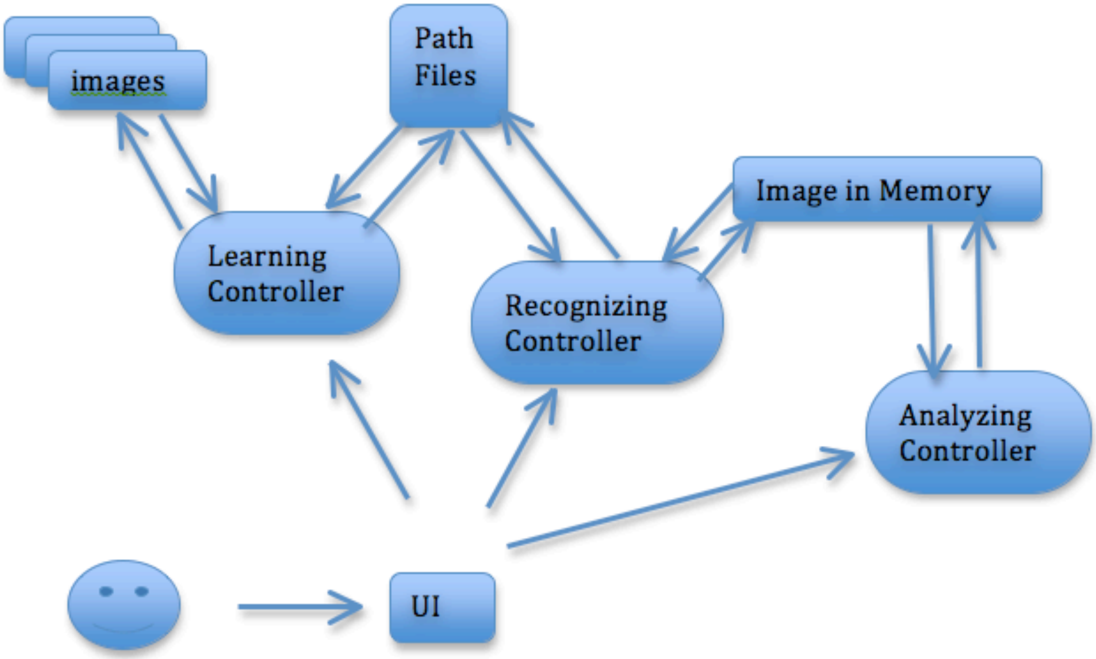


Figure 1 Structure of System

3.2 Modes

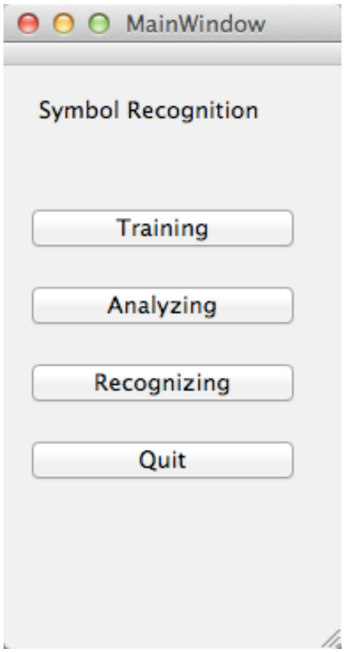


Figure 2 Main Window

There are three modes in the software, recognizing mode, learning mode and analyzing mode. Users can choose the different mode for different purposes. The following is the main window of the software. User can press any button to get into the respective mode view.

3.2.1 Recognizing Mode

There is an exist training set in the recognizing mode. The results of the training set can identify different training symbols, such us the plastic symbols, the paper symbols and Tetra Pak symbols. Users can use this mode to recognize the rubbish with the symbols on it and get the signal to know which garbage to throw it.

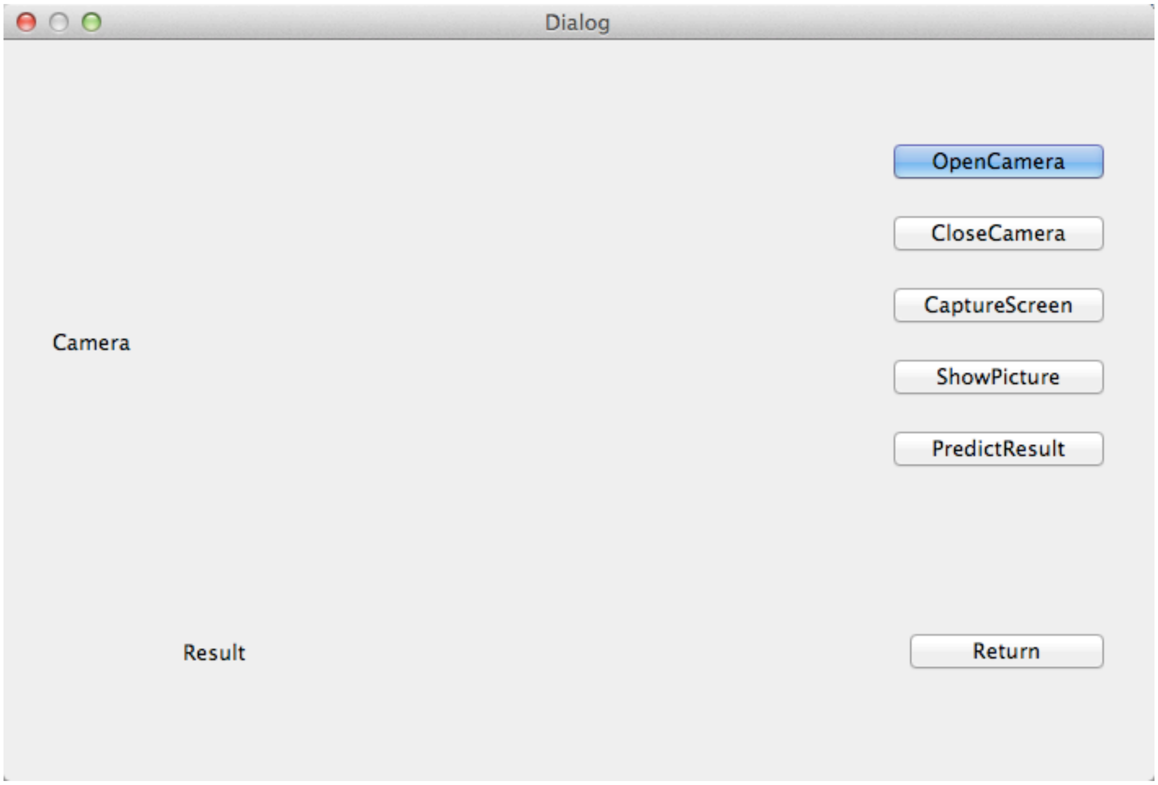


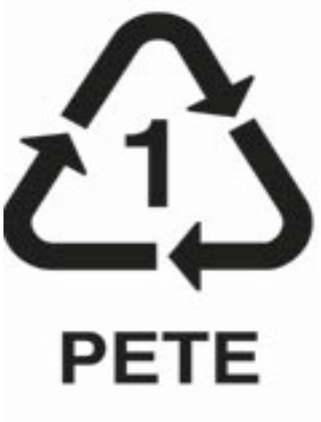
Figure 3 Recognizing Mode

Functions in recognizing mode of different buttons:

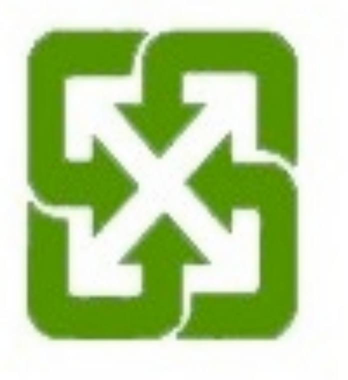
- ❖ Open Camera: Users can open the camera of the computer, and the camera output is at the label camera with real time of the video in camera which is auto size for recognition.
- ❖ Close Camera: Users can close the camera whenever he didn’t use it anymore, but he can press the open camera button to open the camera when he needs it.
- ❖ Capture Screen: Users can capture the image with object and the camera won’t stop to avoid the dissatisfaction of the image. The last one will be stored in the memory.
- ❖ Show Picture: Users can press this button to see if the image captured in the memory is suitable to be predicted.
- ❖ Prediction Result: Users can use this software to predict if the symbol is in the images.

The current training set contains four symbols and the software can identify all of them. These four symbols appear most frequently in the supermarket in HK.

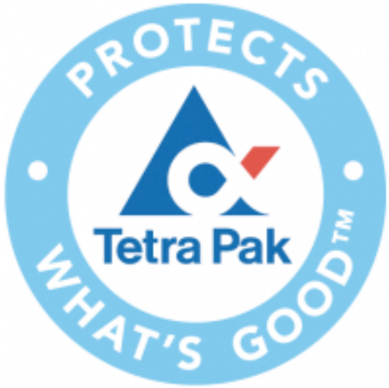
Plastic: Trained Images 17



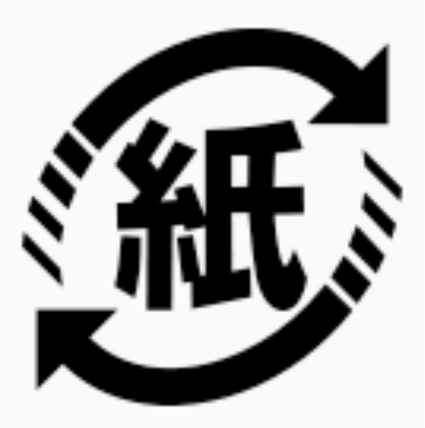
Plastic Cement: Trained Images 19



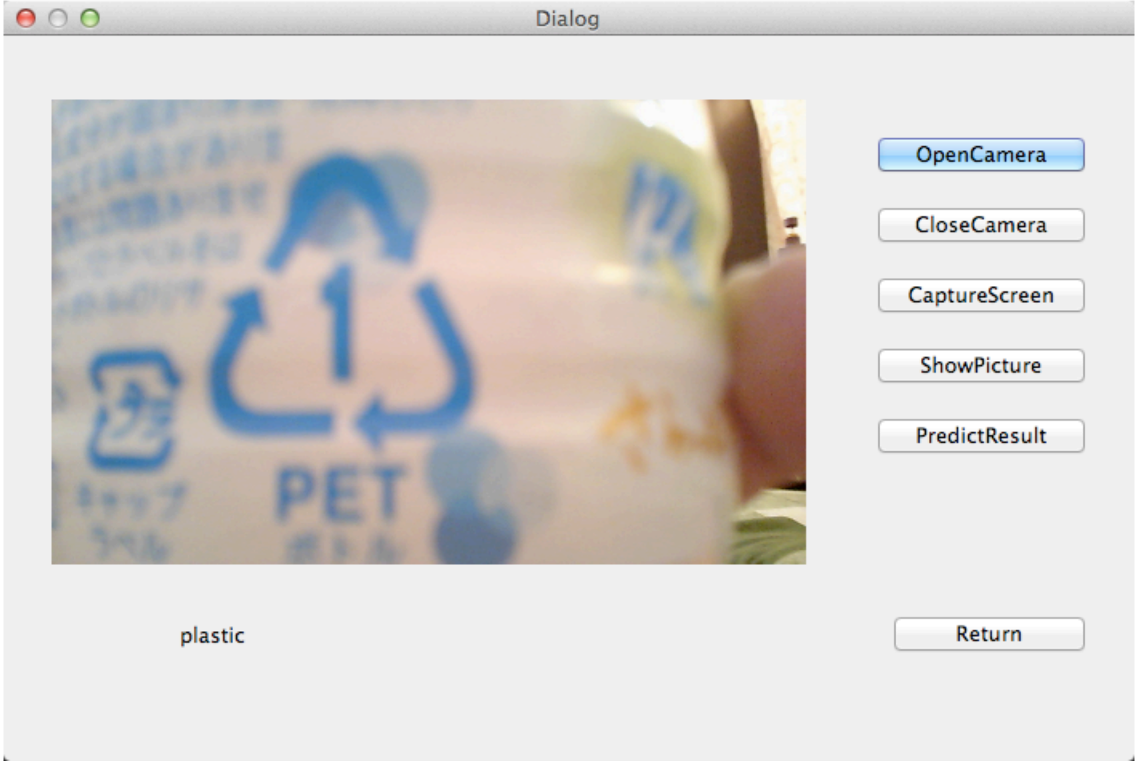
Tetra Pak: Trained Images 28



Paper: Trained Images 12

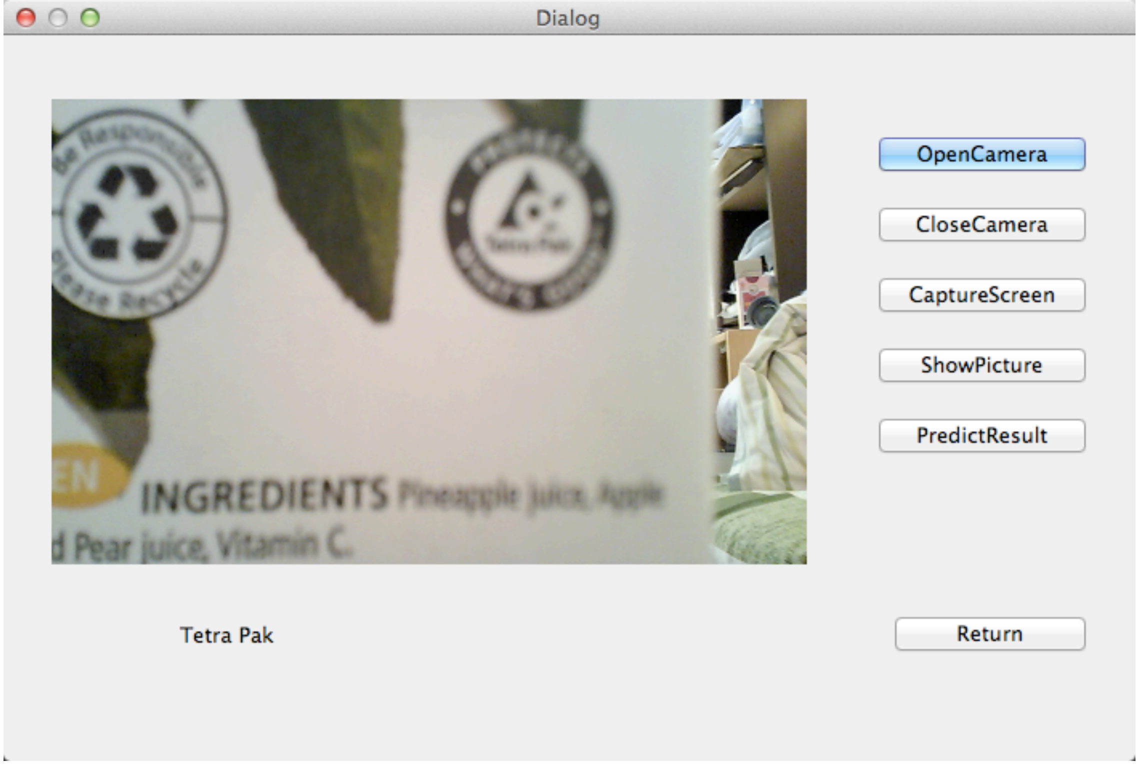


The result of the predicting function is as the picture below:



plastic

Figure 4 Recognition Result Sample I

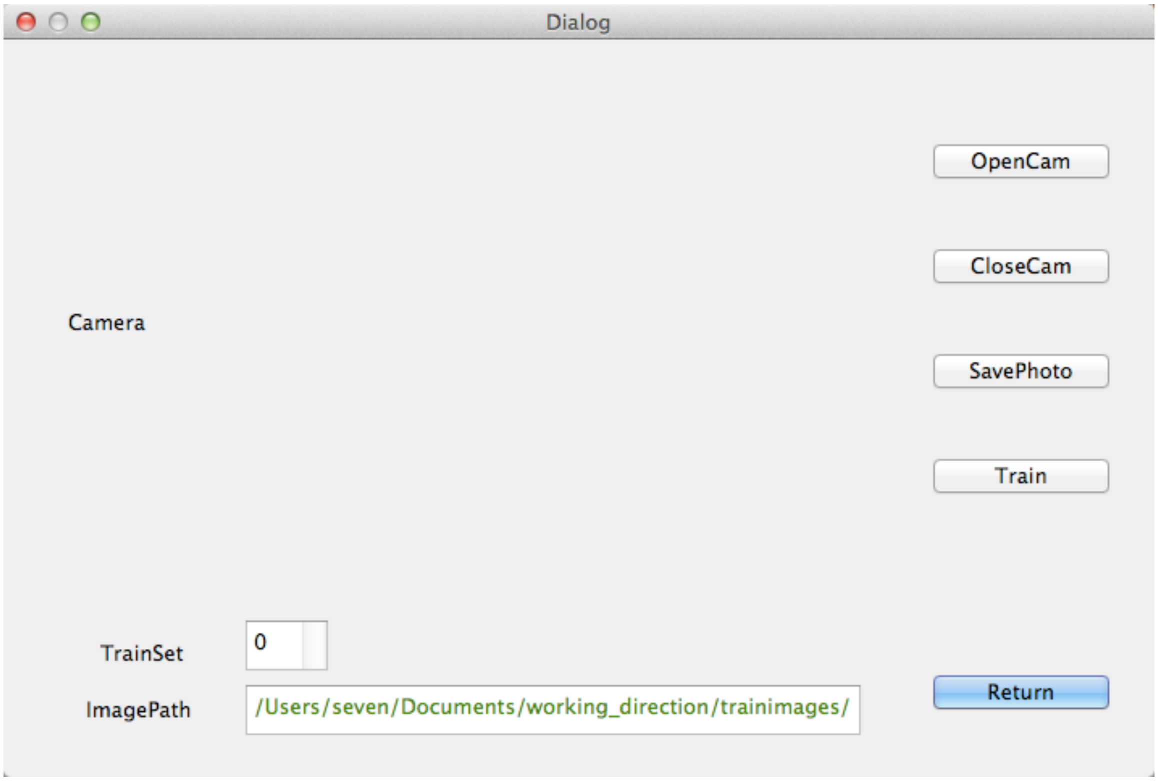


Tetra Pak

Figure 5 Recognition Result Sample II

3.2.2 Learning Mode

The software also set up a learning mechanism. Users can add new images into the database, and you can also set up new databases to train the symbols you want to recognize.



TrainSet 0

ImagePath /Users/seven/Documents/working_direction/trainimages/

Figure 6 Learning Mode

Functions in learning mode of different buttons:

- Open Camera: Users can open the camera of the computer, and the camera output is at the label camera with real time of the video in camera which is auto size for recognition.
- Close Camera: Users can close the camera whenever he didn't use it anymore, but he can press the open camera button to open the camera when he needs it.
- Save Photos: Users can capture the image with object and save the photos in the direction of the text in ImagePath, and the path of the image and the set of the image will be stored in the path file.
- Train: Users can train their own training set with the images captured by themselves. The training xml file will be saved in the direction of the ImagePath.

The new training set will be stored in the direction as following and the information will be stored in different files.

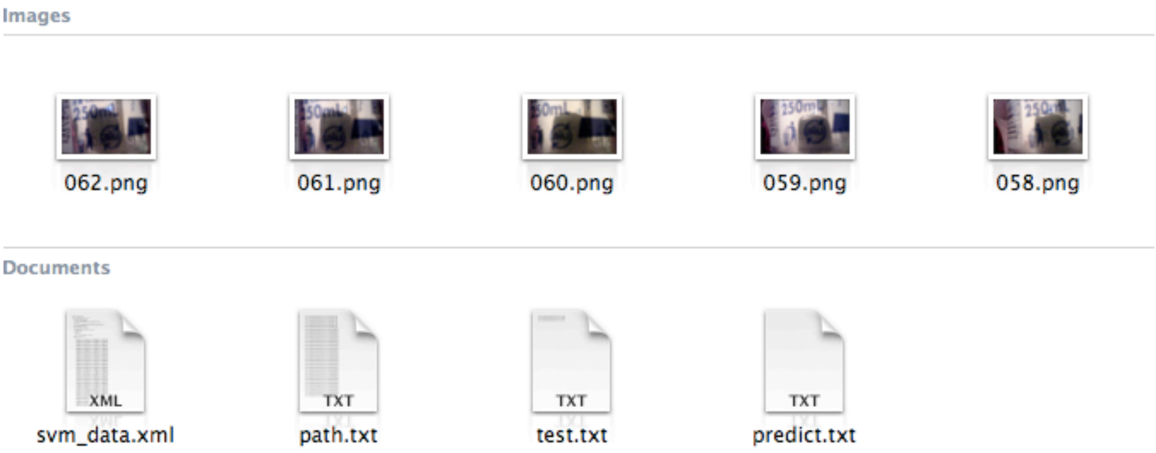


Figure 7 Files in Direction

3.2.3 Analyzing Mode

With this mode, users can know more details of the pictures. Through the changing of the thresholds, users can get more information with the features, contours and convex hulls and this will help a lot to advanced research.

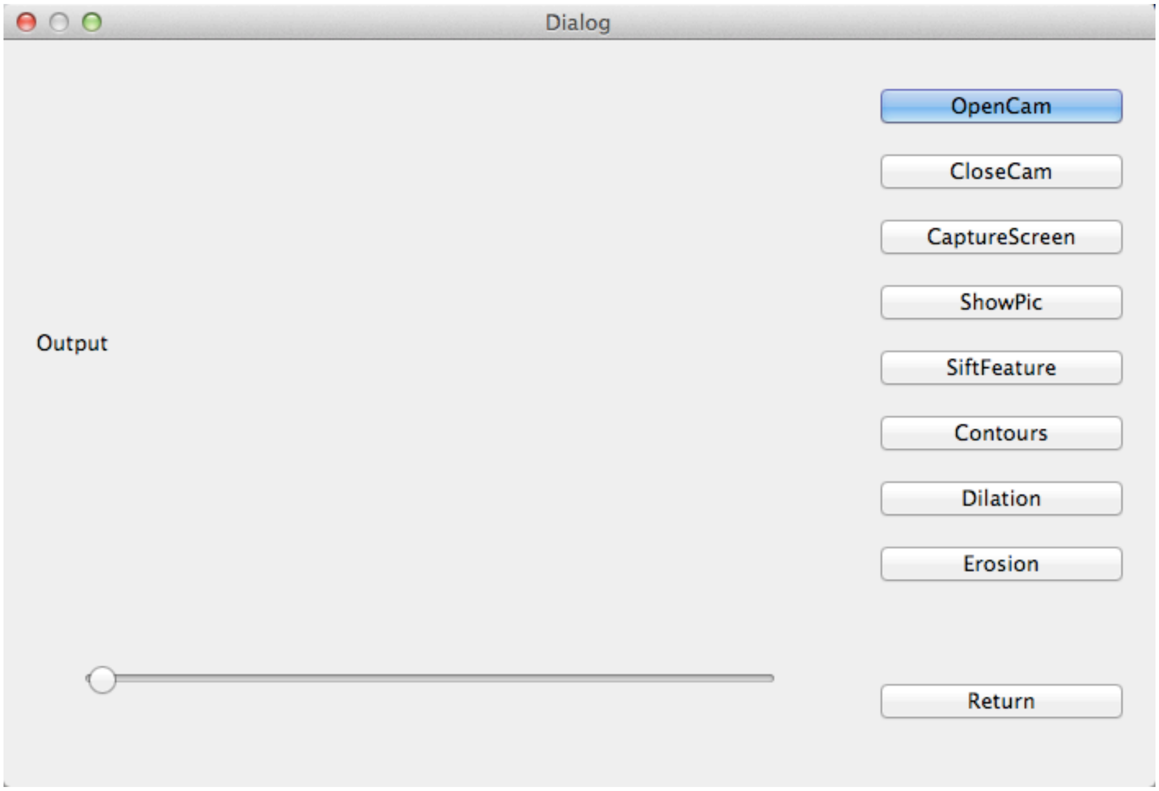


Figure 8 Analyzing Mode

Functions in recognizing mode of different buttons:

- Open Camera: Users can open the camera of the computer, and the camera output is at the label camera with real time of the video in camera which is auto size for recognition.
- Close Camera: Users can close the camera whenever he didn't use it anymore, but he can press the open camera button to open the camera when he needs it.
- Capture Screen: Users can capture the image with object and the camera won't stop to avoid the dissatisfaction of the image. The last one will be stored in the memory.
- Show Picture: Users can press this button to see if the image captured in the memory is suitable to be analyzed.
- Sift Feature: Users can see the sift features of the image drawn on the output screen.

- Contours: Users can see the contours of the image drawn on the output screen. And users can also change the threshold under the output and to get different contours of the image when the threshold is different of the operations.
- Dilation: Users can change the value of the track bar to dilate the picture with different kernel size.
- Erosion: Users can change the value of the track bar to erode the picture with different kernel size.

The results of the function in the analyzing mode are the following pictures.

Sift features:

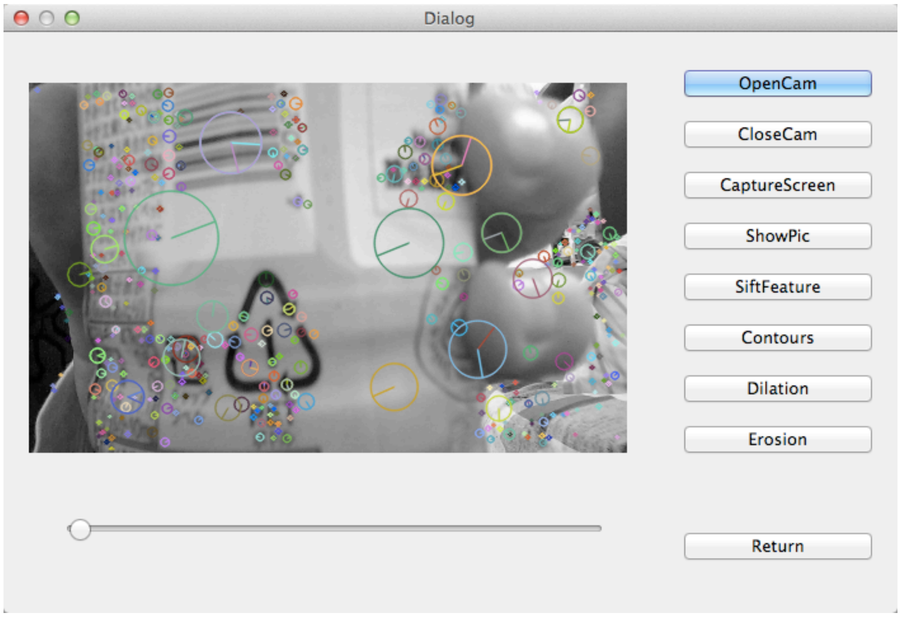


Figure 9 Sift Feature

Contours and Convex hull:

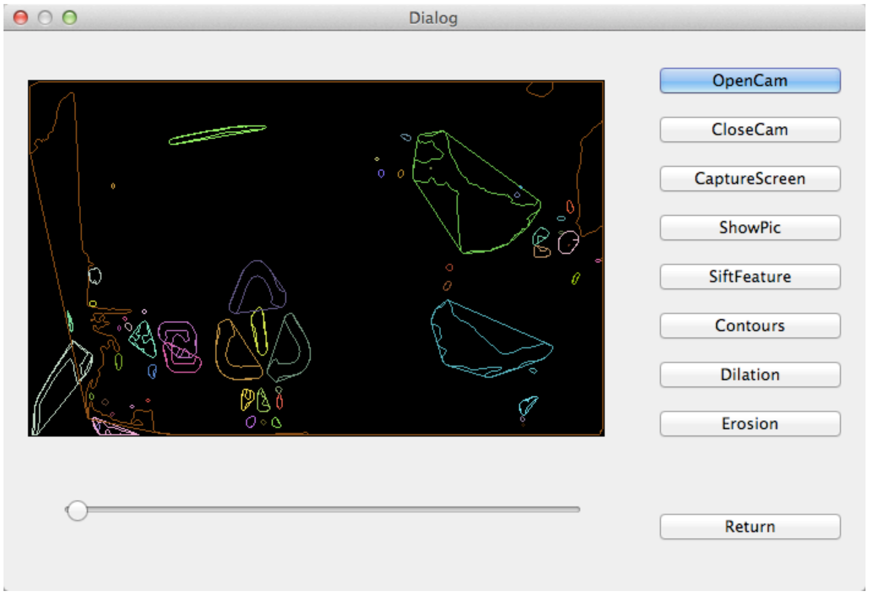


Figure 10 Convex Hull

Dilation and Erosion:

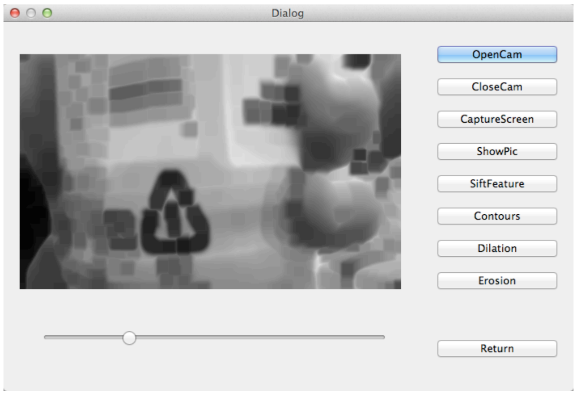


Figure 11 Dilation

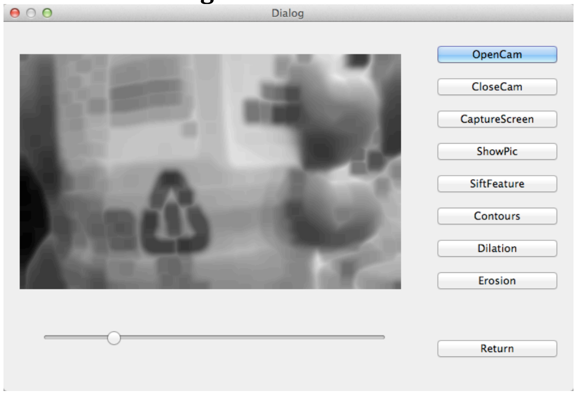
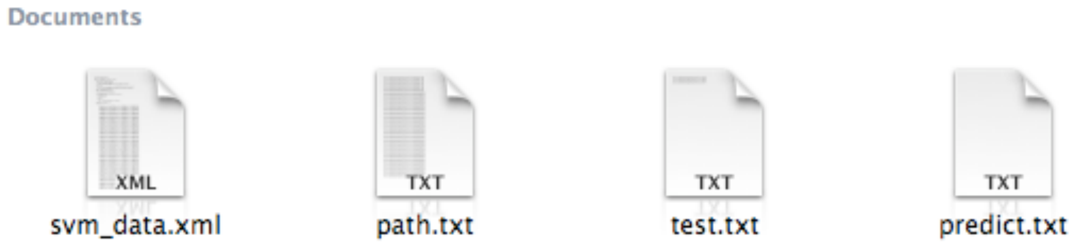


Figure 12 Erosion

4 Implementation

4.1 File I/O

The file I/O is used in the learning mode. Except for the images the information will be stored in four files in the same direction.

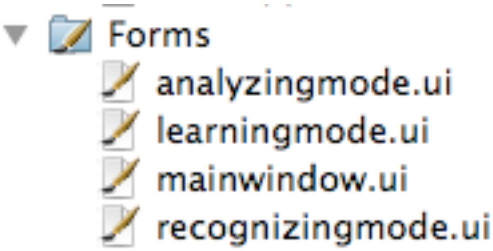


Contents of files:

- Svm_data.xml stores the training result of the new set.
- Path.txt stores the path of the images in this training set.
- Test.txt stores the test set path for this training set.
- Predict.txt stores the class of each images in this training set.

4.2 UI Interaction

The UI of the software uses QT windows to get interacted with the users. And each UI serves a mode.



```
Forms
    analyzingmode.ui
    learningmode.ui
    mainwindow.ui
    recognizingmode.ui
```

4.3 Mode Implementation

The learning mode is based on the class **TrainClass**:

```
class TrainClass {
private:
    string training_class_path;
    int sample_num;
    vector<Mat> training_data;
    vector<int> training_lables;
    vector<string> img_paths;
public:
    bool Initialize(string);
    void setSampleNum(int n){sample_num = n;}
    int getSampleNum(){return sample_num;}
    void traindata();
    void test_data(string);
    int predict_pic(ImageFeature);
};
```

The learning mode is based on the class **ImageFeature**:

```
class ImageFeature {

private:
    std::vector<KeyPoint> keypoints;
    cv::Mat descriptor;
public:
    cv::Mat img;
    cv::Mat ori_img;
    ImageFeature(){}
     std::vector<vector<Point> > contours;
    bool initialize(String);
    bool initialFeatures();
    bool initialContours();
    cv::Mat getDescriptor();
    void setDescriptor(cv::Mat);
    std::vector<cv::KeyPoint> getKeypoint();
    void setKeypoint(std::vector<cv::KeyPoint>);

};
```

The recognition mode is based on the class **RecognitionMode**:

```
namespace Ui {
class RecognizingMode;
}

class RecognizingMode : public QDialog
{
    Q_OBJECT

public:
    explicit RecognizingMode(QWidget *parent = 0);
    ~RecognizingMode();

public:
    Ui::RecognizingMode *ui;

private slots:
        void predict();
        void openCamera();
        void closeCamera();
        void takePicture();
        void readFrame();
        void showPic();
private:
        TrainClass tc;
        ImageFeature imgf;
        QTimer *timer;
    CvCapture *camCapture;
    IplImage* cameraFrame1;
    IplImage* Output_Img1;
    IplImage *cameraFrame;
     IplImage* Output_Img;
};
```

5 Summary

The recycling symbol recognition system has succeeded to recognize four main symbols in the supermarket. And the learning mode of this software can extend its scalability whenever the users want to add new samples in the exist training set or take new photos to make a new training set.

In summary, recycling sorting is of crucial importance to sustainable development. However, the majority of people lack the consciousness and knowledge of waste sorting, and this project will guide people to throw rubbish into the correct bin with indicators. Moreover, the image recognition in the system can be also applied into other projects concerning environmental protection.

6 Appendix

Minutes of the 1st Project Meeting

Date: Monday, 23 Sept 2013

Time: 10:30 AM **Place**: Room 3512

Attending: Prof. Rossiter, CHEN Xutong

Absent: None

Recorder: CHEN Xutong

Approval of minutes

This is first formal group meeting, so there were no minutes to approve.

Report on Progress

CHEN Xutong show the ideas of the algorithm and the way to implement the project.

Discussion Items and Things To Do

- Scope of the project
- Predesign for the project
- Interaction with the users.

Meeting adjournment

The meeting was adjourned at 11:00 AM.

Minutes of the 2nd Project Meeting

Date: Thursday, 17 October 2013

Time: 10:00 AM

Place: Room 3512

Attending: Prof. Rossiter, CHEN Xutong

Absent: None

Recorder: CHEN Xutong

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

CHEN Xutong tried the algorithm in the project and make the algorithm more suitable with finding the best threshold in for the project.

Discussion Items and Things To Do

Try the erosion and dilation for detecting the contours of the images

Meeting adjournment

The meeting was adjourned at 10:30 AM.

Minutes of the 3rd Project Meeting

Date: Thursday, 7 November 2013

Time: 09:45 AM

Place: Room 3512

Attending: Prof. Rossiter, CHEN Xutong

Absent: None

Recorder: CHEN Xutong

Approval of minutes

The erosion and dilation way is not suitable for this project because of the difference among samples, so the project use the training way for the result.

Report on Progress

CHEN Xutong demonstrates that the way in the last meeting is not workable, and find sanother way to get the result.

Discussion Items and Things To Do

Finish the project with a UI to interact with users and set the self-learning mode.

Meeting adjournment

The meeting was adjourned at 10:30 AM.

Minutes of the 4th Project Meeting

Date: Thursday, 4 December 2013

Time: 12:00 PM

Place: Room 3512

Attending: Prof. Rossiter, CHEN XUTONG

Absent: None

Recorder: CHEN XUTONG

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

CHEN XUTONG demonstrated the final result of the project, which meet all the requirements.

Discussion Items and Things To Do

- Finish the report
- Record a demonstration video

Meeting adjournment

The meeting was adjourned at 13:00 PM.