Comprehensive Experiment Report on Digital Image Processing and Pattern Recognition

Project Name: Workpiece Classification

Faculty: School of Artificial Intelligence and Automation
Class: Automation Class 1606
Name: Chen Mingxiang
Student ID: U2016 14343.
Instructors: Ma Jie, Zheng Dingfu

School of Artificial Intelligence and Automation
February 23, 2019

Table of contents

一、Overall scheme design

1、 Design purpose

1） Learn and master the image acquisition device and its usage.

2） Apply theory to practice and enhance the ability of innovation, self-study and independent thinking.

3） Master VC digital image programming skills.

4） In the independent design at the same time, the teacher to assist, faster and better understanding of the essence of design.

5） Using image processing and pattern recognition related theoretical knowledge and methods, the task of workpiece detection based on computer vision is realized, which lays the foundation for future study and work.

2、 Design requirements

1） Photo shooting requirements

It requires the use of SLR cameras to collect correct exposure and clear digital images under various complex conditions.

For example, take a clear image when the object is moving fast.

Expose correctly to obtain a clear image in a dark environment.

Put the object in the highlight area and the low light area respectively to get a clear image.

Take an image with very small and very large depth of field.

Adjust the camera parameters in real time to get clearer pictures.

2） Mission requirements

It is required to measure the size of the workpiece and classify the workpiece by programming through the real-time image.

In the process of realizing this task, the following points need to be done:

- Installation and debugging of image acquisition equipment

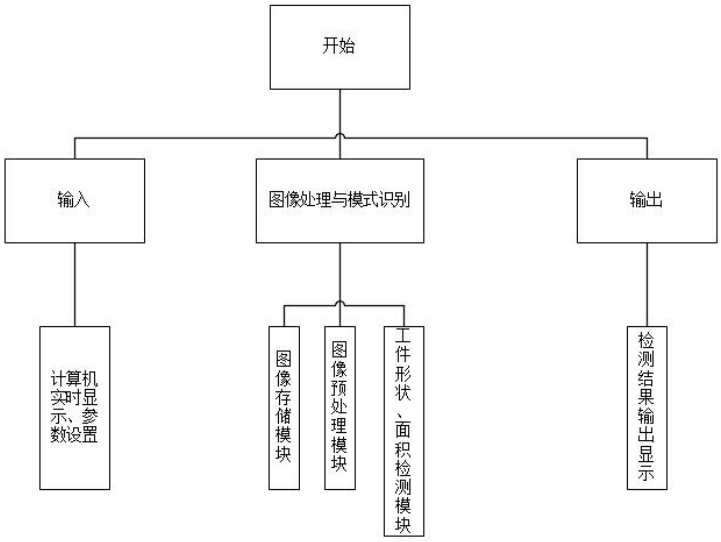
Complete the establishment of the VC digital image programming environment and the establishment of the programming framework.

- Successful acquisition of correctly exposed, clear images is required
- Realize the classification and size measurement of the workpiece in the picture by learning theoretical knowledge and programming
- System commissioning, timely and complete report writing

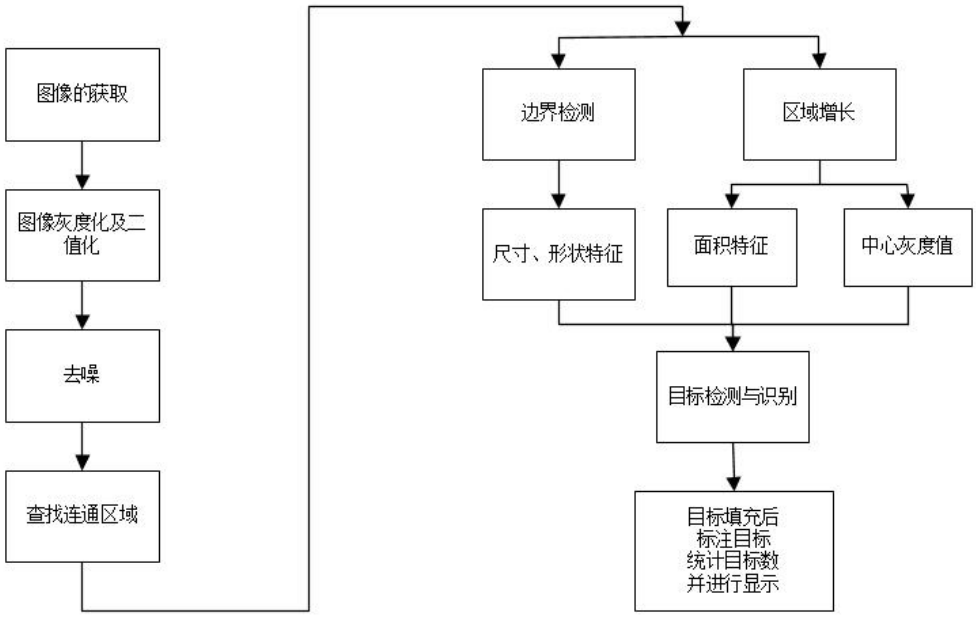
3） Error requirements

It is required that the workpiece classification and measurement can still work stably under various unexpected circumstances, with strong anti-interference ability, and the identification and measurement errors can be minimized.

3、 Overall implementation block diagram



4、 Program flow chart of image processing and pattern recognition module



5、 Division of labor among team members

Chen Mingxiang: Responsible for completing all tasks.

二、Image acquisition system design and source program design

1、 Image Acquisition Technology and Design of Image Acquisition System

Using MV-E & em series of Gigabit Ethernet industrial cameras, the experimental camera requires Win7 environment to run normally. In the process of connecting the camera, it is necessary to confirm that the computer protection software and firewall are closed, and the network card is a gigabit network card. Install the camera driver and set the network card and IP, then run it, pop up the main interface of Demo, and then you can collect images.In the process of image acquisition, the brightness of the environment is changed by adjusting the parameters to improve the image clarity. The larger the gain is, the larger the image noise point is, so the gain is set to the minimum in this experiment. In this experiment, the camera automatically adjusts the exposure time value. When the image brightness is adjusted to the target brightness, the automatic exposure time is set to off. At the same time, this experiment uses red light or natural light from bottom to top, and the specific light used depends on the environmental conditions and specific error requirements.

2、 Software programming and debugging technology

1） Image graying and binarization

Convert the true color image to a grayscale image using the formula: Gray = R * 0.3 + G * 0.59 + B * 0.11.

Converting the grayscale image into a binary image, using the formula: threshold = (float) (Max _ gray * K _ guss + min _ gray * (1 − K _ guss)), where K _ guss is a preset threshold coefficient, and the preset value is 0.5. The gray value of the point larger than threshold in the image is set as 255, which is pure white, and the gray value of the point smaller than threshold is set as 0, which is pure black, and then the binary image is obtained.

2） Image preprocessing (denoising)

• Scan upper, lower, left and right neighborhoods of each black point in the binarized image. If there is no black point, the current point is recognized as noise and set it as a white point.

· Set the black points at the edge of the image as white points for subsequent search.

• Clear image boundaries for processing. The first connected region found in the image is considered the boundary of the image.

The dilatation-erosion method has some denoising ability, but I did not use this method this time.

3) Find the connected area

In this experiment, the pointer of pre-allocated memory is used to set the corresponding connected area and find its boundary. When looking later, you only need to look in its boundary area.

The search order is bottom to top and left to right. The searching method is as follows: according to the searching order, the first point of a class is found and set as the value of the current class, and at the same time, the black points in the four neighborhoods are also set. And then repeat back

to that current row to reseat the bits in ord. Whenever a black dot is found to the left of a current class dot, it retreats to the initial point of the row; Each time a black dot is found below a class dot, the row initial point of the lower row is returned. Update boundaries as you search.After searching the whole graph, add one to the class value, and continue to search upward from the lower boundary of the previous class until there is no unset black dot in the graph.

3） Target identification

The targets identified in this experiment are screws, nuts and coins, and the rest are invalid targets. If you want to recognize other targets, you only need to change part of the code in the pattern recognition section.

Identification method: center gray value discrimination, the gray value of the screw center is consistent with the target gray value, and the gray value of the nut center is consistent with the background gray value. The ratio of the area of the screw to the area of the boundary of the screw rectangle is greater than ratio of the area of the nut to the area of the boundary of the nut rectangle. Area discrimination, if there is a coin, it can be judged by the area whether it is a nut or a coin. Coins were not identified in this experiment. Boundary length-width ratio judgment

4） Target mark filling and number statistics

In this experiment, the nut is filled with green, the screw is filled with red, and the rest are invalid targets. The target pixel value is marked next to the target, and the number of targets of the same type is marked at the top left of the screen.

III. Operation results

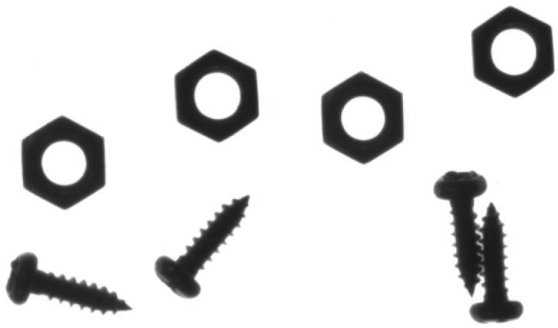


Figure 1: Original figure Figure 2: Original figure of adhesion between screw and screw

Figure 3: Original drawing of external interference

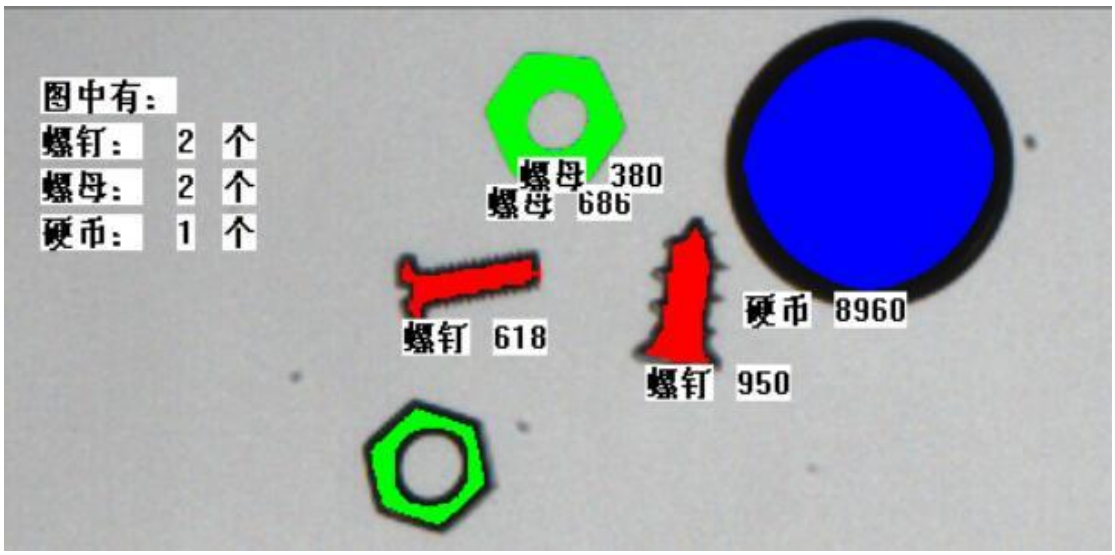


Figure 4: Non-adhesion results of target recognition

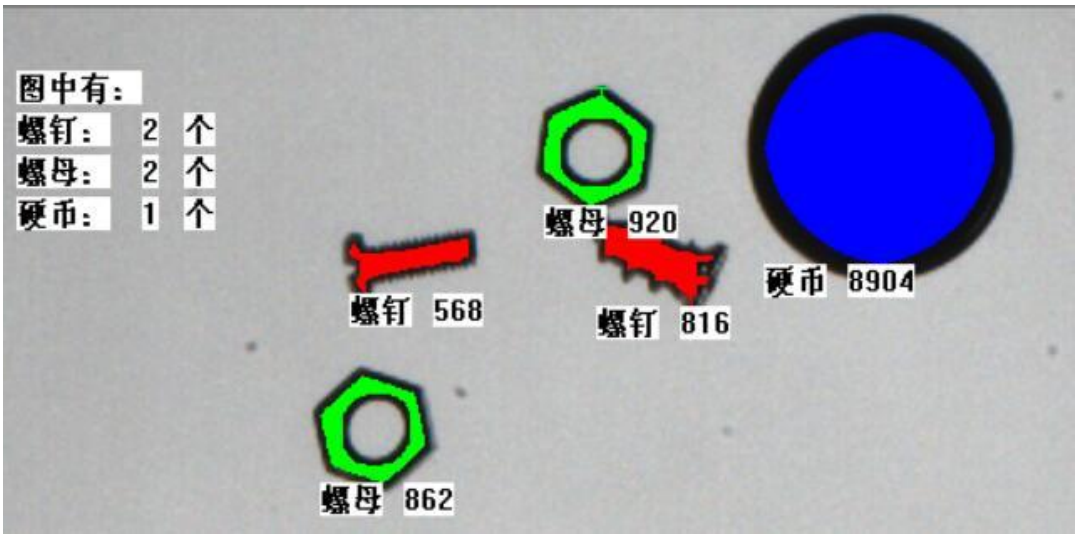


Figure 5: Result of target recognition with very few adhesions

IV. Analysis and Summary of Experimental Results

- Deficiencies in the experimental results:

When the adhesion condition shown in Figure 2 is encountered, the two screws that are adhered cannot be identified.

When there is external interference as shown in Figure 3, the finger is misrecognized as a coin.

The nut position shown in Fig. 4 is marked incorrectly, and it can be marked correctly after a period of time.

After running the program, the picture circularly moves to the left, and the target label position is wrong and becomes more.

- Analysis:

The two screws cannot be identified when they are stuck together. Because the corrosion procedure I used this time is not the expansion corrosion technology mentioned by the teacher in class, but a relatively simple corrosion, the effect of corrosion to remove adhesion is not good.

The finger was mistaken for a coin. The analysis found that it was the part of target recognition, and only the central gray value and area size were used to distinguish coins. The area and center gray value of the finger just meet the requirements of the coin, so it is wrongly recognized as a coin.

The nut position is marked incorrectly. This is not a single fixed image for target recognition, but a series of image sequences. Occasionally, because the camera aperture and focal length are not adjusted properly, or other external factors, some of the images collected cause errors in recognition. At the same time, because the target recognition program does not extract enough features of the target when it recognizes a target, it misrecognizes the target.

The picture moves cyclically to the left. After analysis, it is because the camera parameters are adjusted incorrectly, and the camera parameters are displayed and identified normally after adjustment.

- Summary: This experiment has completed the basic task well, from the collection of pictures to target recognition, the recognition effect is good. However, there is no innovation, and the efficiency of the program is not very high, and the occasional error is relatively large, which needs

further improvement and improvement.

V. Gains and recommendations

• Gains: Through this experiment, I initially became familiar with the image acquisition operation and the adjustment of camera parameters, learned to consult my classmates and teachers modestly, and learned to apply theory to practice. Most importantly, I learned the importance of preparing ahead of time and making plans.At the beginning, I didn't know how to start because I didn't know how to store pictures. Because of this, I looked up information and asked my teachers and classmates for help. In the process, I learned the importance of self-study and diligence. At the same time, I understood that if I didn't understand, I would ask, not to do it alone, not to persist in getting into a dead end.After figuring out how to realize the graying, binarization and image erosion of the image, I found that as long as I dare to challenge and work hard, it will not be so difficult.

• Reflection: Because my programming ability is not very good, at the beginning of the team, I was worried that everyone would dislike me, and at the same time, I was afraid of dragging everyone down, so I never found someone to form a team, so that in the later process, no one discussed with me, many times, I was very stubborn in doing things, did not want the teacher to help, I thought I could finish all the things independently, and later. I'm not keeping up with everyone.Before doing this experiment, I didn't make any plans. I worked hard every day. In fact, as long as I could learn to communicate and learn to ask the teacher for help modestly if I didn't understand, my progress would not be so slow. At the beginning, I was afraid that I couldn't do it, so I didn't dare to choose a difficult topic. Later, I found that as long as I dared to choose a difficult topic, dared to challenge myself, worked harder, prepared in advance, and made a good plan, then nothing could not be done.

• Suggestion: If possible, it is hoped that students can have a little more freedom in class time. It would be better if they can use electronic check-in instead of paper check-in, and students can choose their most efficient time to study. It would be nice if classes could be scheduled in the first half of the semester instead of near the end.Most of the time, the programming ability of girls is not so strong, when forming a team, they are afraid to involve others, so they dare not form a team with boys, and the number of girls is just odd, and finally they have to form a team alone, if the last remaining girl can join the full team of girls, so that in the process of doing, there will be discussions, and the progress will be a little faster.

VI. References

Lu Zongqi, Visual C + +.NET Image Processing Programming, Tsinghua University Press, 2006;
[Eckel] Bruce Eckel, Thinking in C + +, China Machine Press, 2000

VII. Source Code

Description: All core code has been placed in the text box

- Macro definition and image structure definition

```
#define WHITEPIXEL 255
#define BLACKPIXEL 0
#define UP 0
#define DOWN 1
#define LEFT 2
#define RIGHT 3
# define COIN 2//COIN
# define NUT 1//NUT
# define SCREW 0//screw
# define INVALID 3//Invalid target

typedef struct ClassInFomation{
  unsigned int centerX; //Abscissa of the geometric center of the class
  unsigned int centerY; //Ordinate of the geometric center of the class
  unsigned int length; //The length of the area occupied by this class
  unsigned int wide; //The width of the area occupied by this class
  BYTE grayOfCenter; //the pixel value of the geometric center position of the class
  float classDuty; //Area duty ratio of the element in the area
  float lengthToWide; //The aspect ratio of the area occupied by this class
  float membershipGrade; //The membership degree is 0 for nuts and 1 for screws;
}ClassInfo;
```

• 目标识别全体代码人口

```
//********************************************** 以 下 为 自 己 的 部 分
**********************************************************//
        if (recognitionFlag)                                                    //标志位:按下
Recognition 进入目标识别
        {
                g_imageWidth = m_pBmpInfo->bmiHeader.biWidth;          //图像宽度
               g_imageHeight = m_pBmpInfo->bmiHeader.biHeight;       //图像高度
                g_imageRow = m_pBmpInfo->bmiHeader.biWidth;            //一行图像像素个数
                ImageProcess(pDC);                                                  //图像处
```

• 图像灰度化及二值化

```
//////////////////////////////////////
//函数名:GrayOrRGBTo2
//输入参数:无
//输出参数:无
//功能描述:对图像进行灰度化及二值化
//////////////////////////////////////

void CMainFrame::GrayOrRGBTo2()
{
        unsigned int                x = 0;
        unsigned int                y = 0;
      unsigned int       max_gray = 0;
        unsigned int    min_gray = 255;
        unsigned int      threshold = 0;              //阈值
        unsigned int bwith = g_imageWidth;
        unsigned int bhight = g_imageHeight;

        unsigned int H[256] = {0};                  //灰度直方图
        double sum0 = 0;                                    //前景灰度和
        double sum1 = 0;                                    //背景灰度和

        double N0 = 0;                                      //前景像素数
        double N1 = 0;                                    //背景像素数
```

• 图像预处理

```
//////////////////////////////////////
//函数名:PreProcess
//输入参数:无
//输出参数:无
//功能描述:图像预处理，用于去噪
//////////////////////////////////////
void CMainFrame::PreProcess()
{
        unsigned int                    x = 0;
        unsigned int                    y = 0;
        unsigned int              width = 0;
        unsigned int             height = 0;
        unsigned int     pixelCounter = 0;      //邻域内黑点个数累计


        for (x=0;x<g_imageWidth;x++)
```

• 连通域的查找

```
//////////////////////////////////////
//函数名:FindClass
//输入参数:无
//输出参数:无
//功能描述:将图像中各个连通的区域找出来
//说明：遍历顺序：从左至右，从下至上
//////////////////////////////////////
void CMainFrame::FindClass()
{
        unsigned int pixelClass = 0;
      unsigned int                x = 0;
        unsigned int                y = 0;
```

• 图像腐蚀，去粘连

```
//////////////////////////////////////
//函数名:UpdateBound
//输入参数:无
//输出参数:无
//功能描述:统计个数、判断并腐蚀图像（去粘连）
//////////////////////////////////////
void CMainFrame::corrode()
{
        unsigned int             width = 0;
        unsigned int            height = 0;
```

• 目标识别

```
////////////////////////////////////////
//函数名:ImageRecognize
//输入参数:无
//输出参数:无
//功能描述:目标识别
////////////////////////////////////////
void CMainFrame::ImageRecognize()
{
        unsigned int              width = 0;
        unsigned int             height = 0;
    unsigned int      pixelClass = 0;
```

• 目标填充与显示

```
////////////////////////////////////////////
//函数名:PourObject
//输入参数:无
//输出参数:无
//功能描述:在原图像中对已确定的目标进行填充
////////////////////////////////////////////
void CMainFrame::PourObject()
{
    unsigned int pixelClass = 0;
    unsigned int           count = 0;
      unsigned int           width = 0;
      unsigned int          height = 0;

      for (pixelClass=2;pixelClass<=g_imageClass[0];pixelClass++)
      {
            /*  搜寻螺钉并将其改成绿色*/
        if (SCREW == g_classCategory[pixelClass])
            {
             for
(width=g_classCoordinate[pixelClass][LEFT];width<=g_classCoordinate[pixelClass][RIGHT];widt
h++)
                    {
                        for
(height=g_classCoordinate[pixelClass][DOWN];height<=g_classCoordinate[pixelClass][UP];height
```