OCR Robot Coding Notes

MUCH2020 Studio

Tian'Yi Zheng

2020. 02.24

Context

Software idea and target

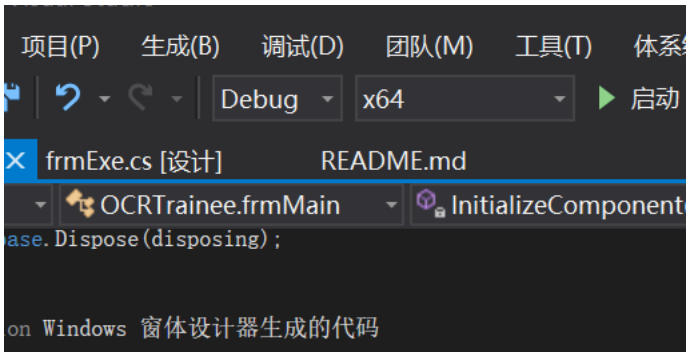
1. Visual Studio work with Halcon operators;
2. Connected with 2D camera for OCR for handwrite letters and print letters, for civil and industry use;
3. Adjustment is fully flexible for user;
4. Adjustment parameters can be save and loaded project by project, for large OCR volume operation, including OCR classifier, picture segment, character region choose, manual and automatic OCR;
5. ShapeModel function for system to automatically capture region for OCR;
6. OCR classifier can be uploaded by user;
7. More OCR classifier will be uploaded periodically by MUCH2020 Studio.

UI Introduction

For UI info, please check "OCR Robot Manual".

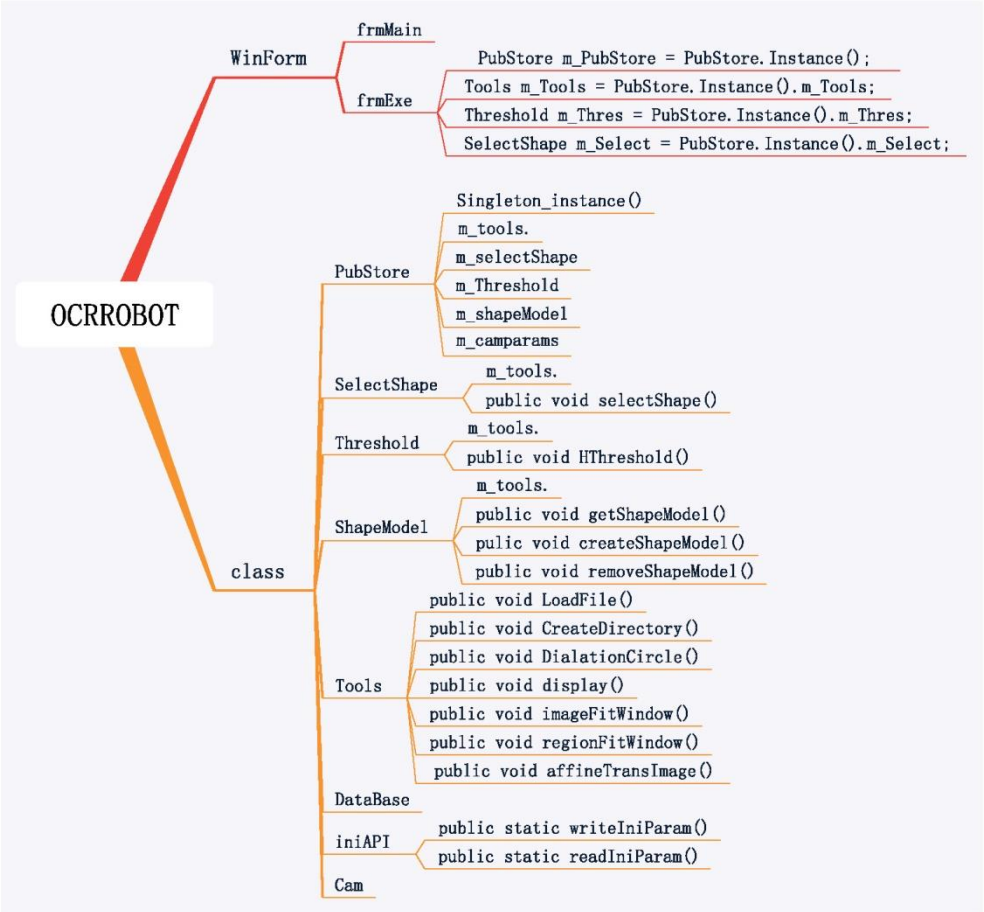
C# Code Introduction

Window X64



Architecture

Class



Singleton Pattern

For class PubStore

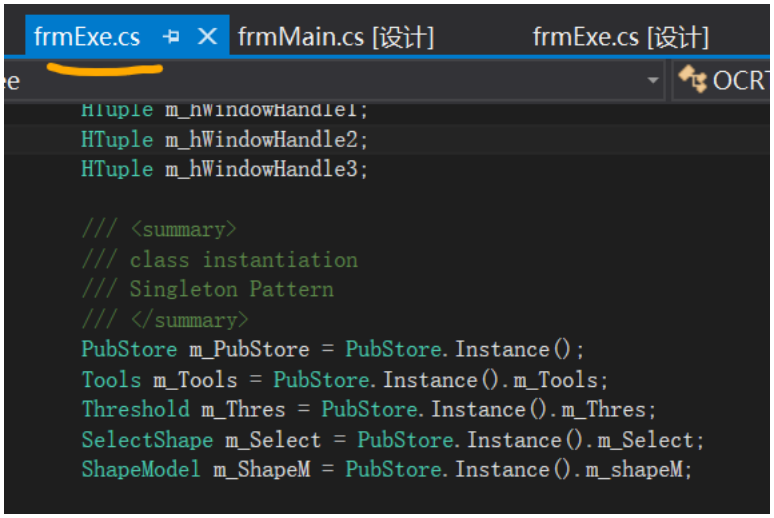
```
#region PubStore Singleton

        private static PubStore _instance = null;

        public static PubStore Instance()
        {
            if (_instance == null)
            {
                _instance = new PubStore();
            }
            return _instance;
        }
        #endregion

        /// <summary>
        /// PubStore Singleton Pattern
        /// </summary>
        public Tools m_Tools = new Tools();
        public CamParam m_camParam = new CamParam();
        public Threshold m_Thres = new Threshold();
        public SelectShape m_Select = new SelectShape();
        public ShapeModel m_shapeM = new ShapeModel();
```

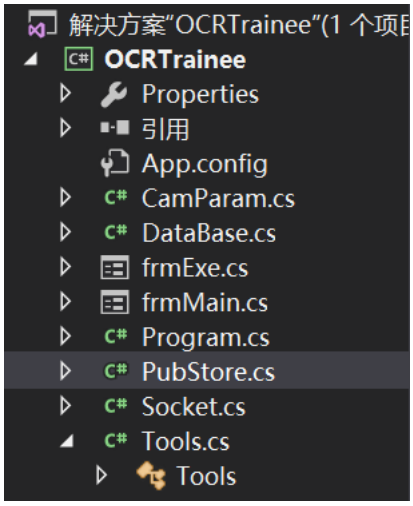
Under frmExe



Cross WinForm delivery

Steps:

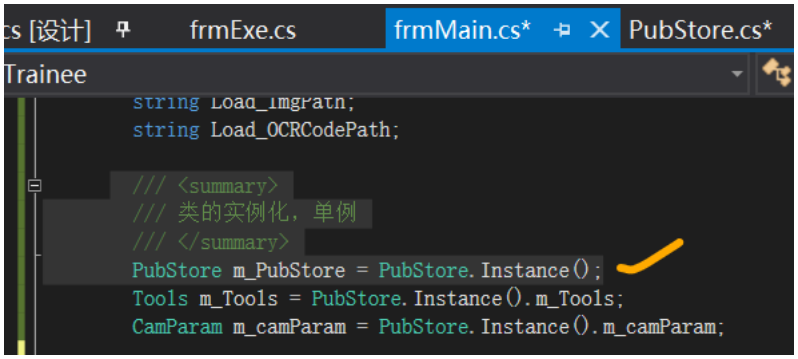
1. Create a public class, name as "PubStore"



2. Make a transfer variable m_type, and a singleton

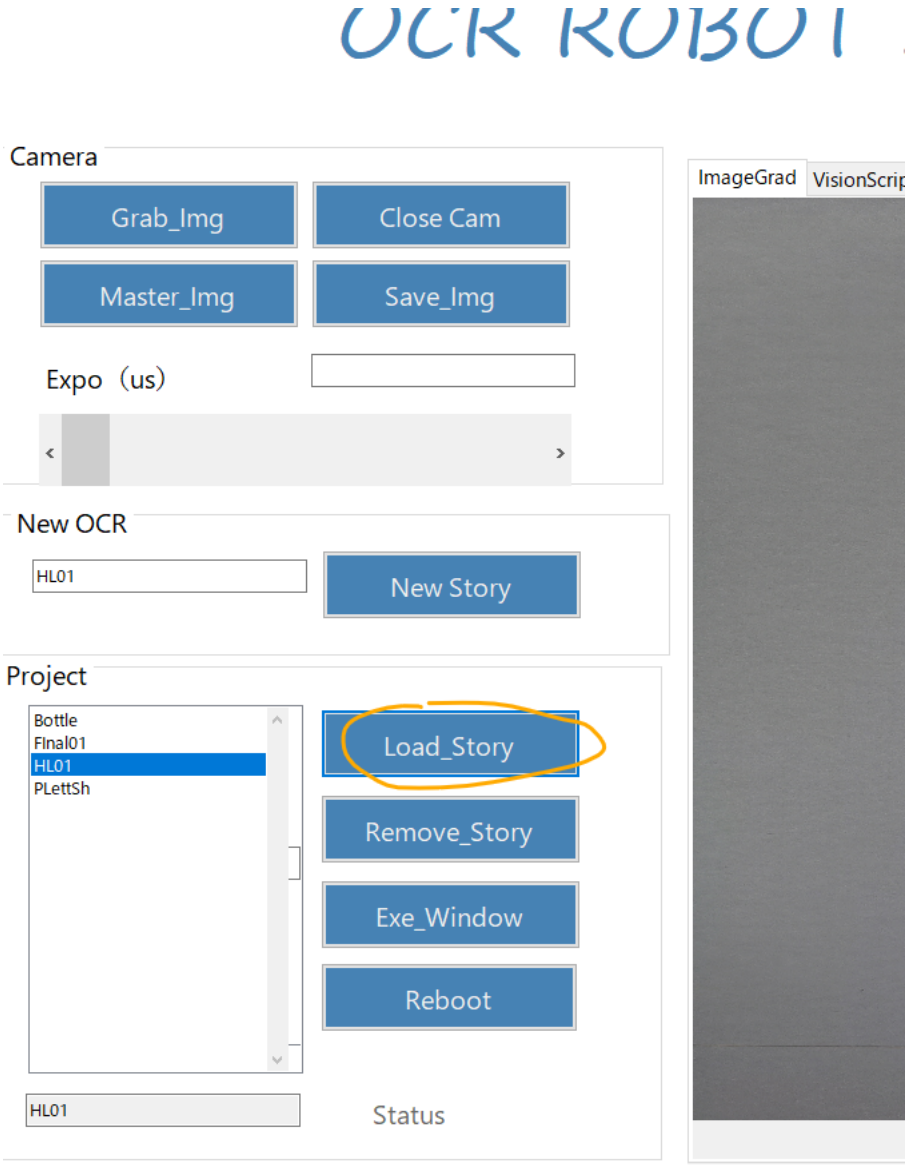
```
7
8
9   namespace OCRTrainee
10  {
        13 个引用
11      public class PubStore
12      {
13          #region variable
14
15          /// <summary>
16          /// transfer variable
17          /// </summary>
18          public string m_type; //bridge from frmMain to frmExe
19
20          #endregion
21
22          #region PubStore Singleton
23
24          private static PubStore _instance = null;
```

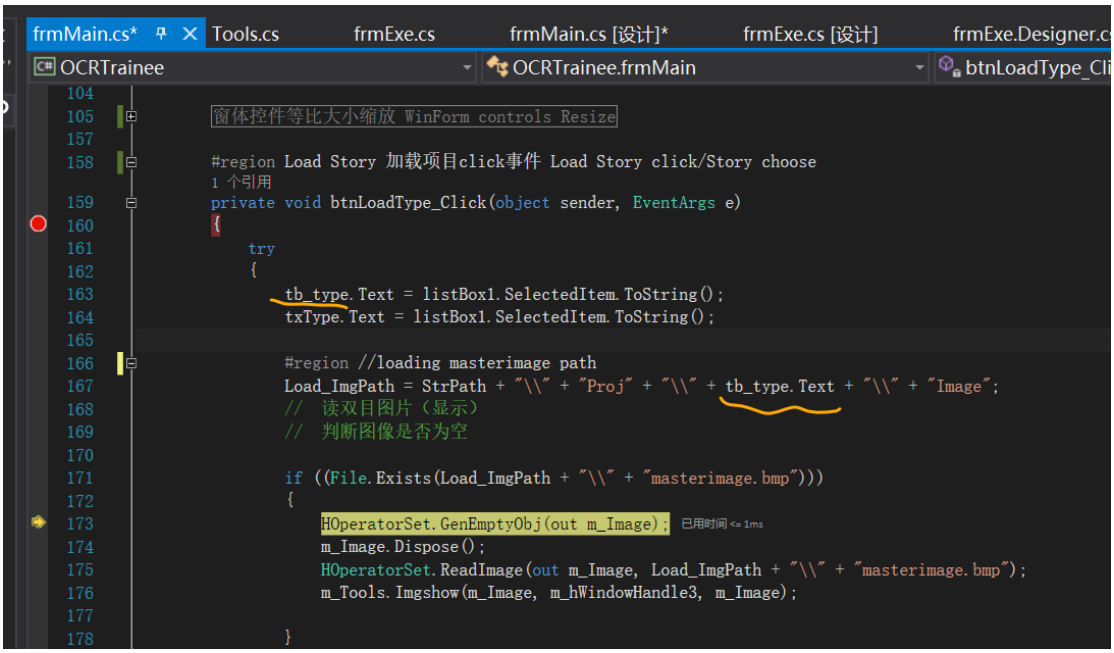
3. Under frmMain, make class PubStore instantiation with singleton

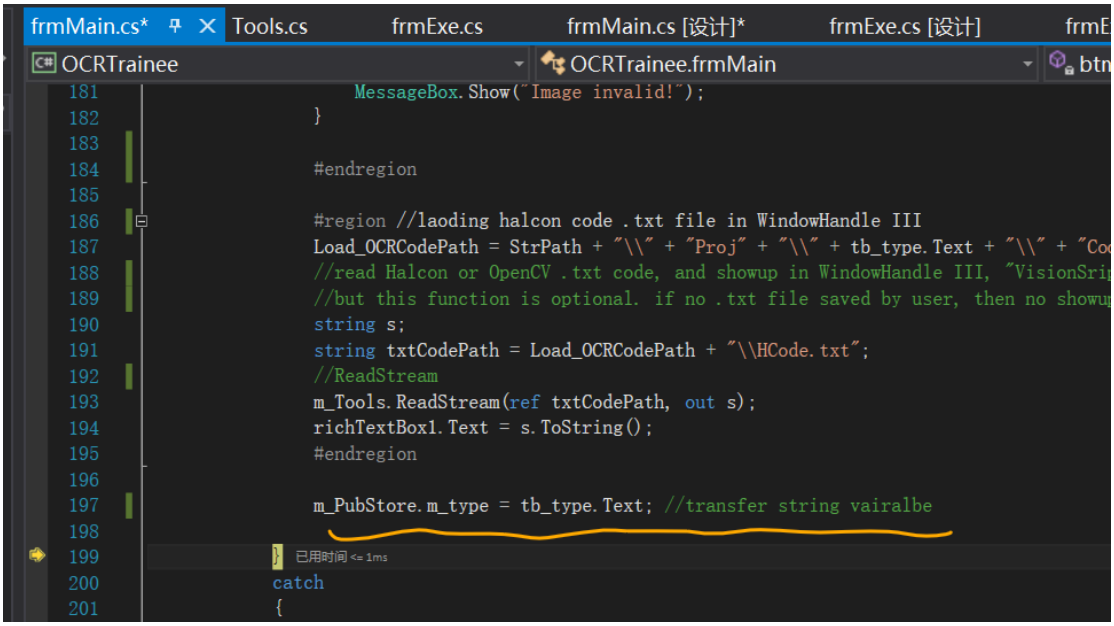


4. Steps of this transfer variable running-----m_PubStore.m_type

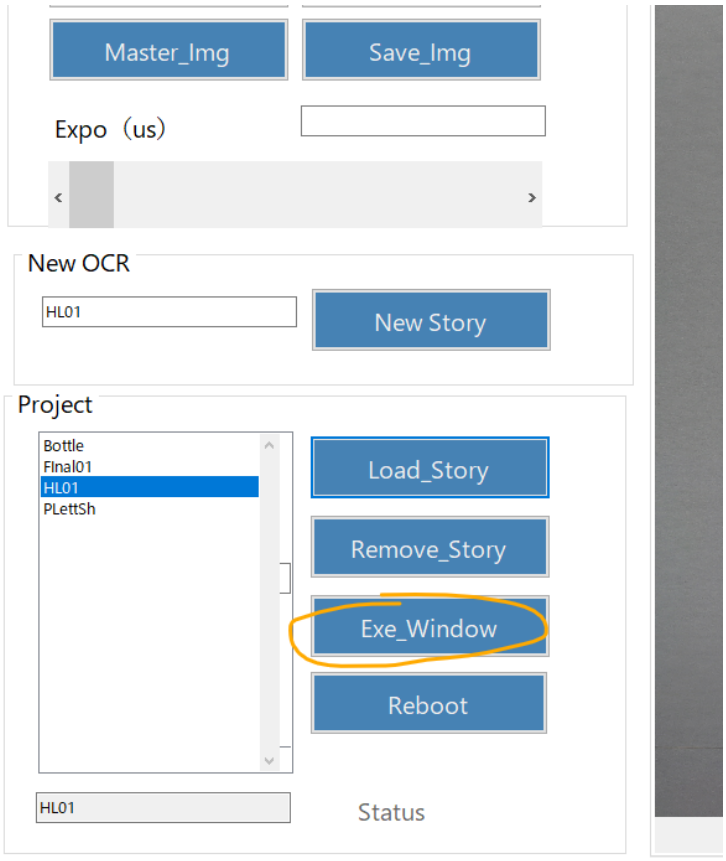
frmMain

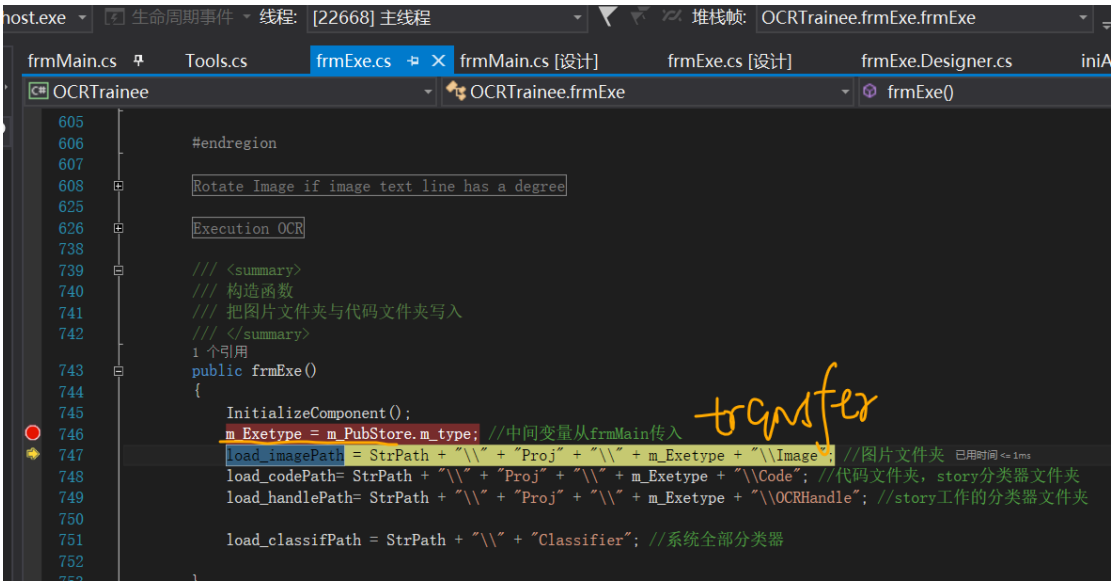


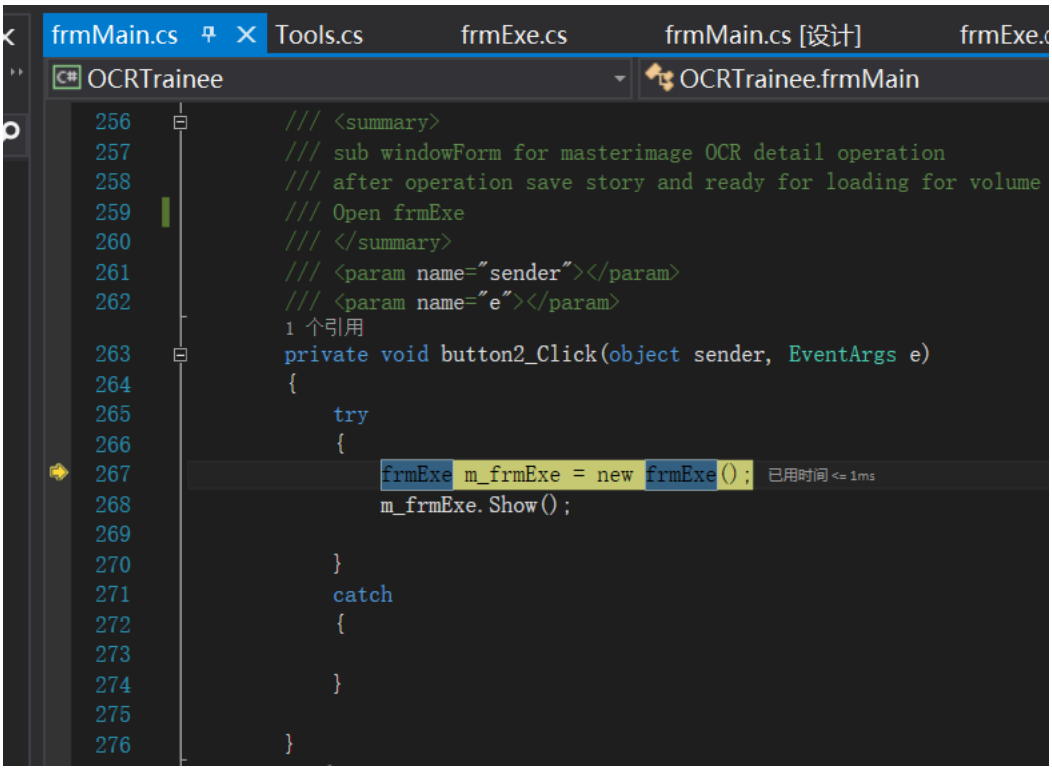

frmMain.cs*
Tools.cs
frmExe.cs
frmMain.cs [设计]*
frmExe.cs [设计]
frmExe.Designer.cs
OCRTrainee
OCRTrainee.frmMain
btnLoadType_Cli
104
105 窗体控件等比大小缩放 WinForm controls Resize
157
158 #region Load Story 加载项目click事件 Load Story click/Story choose
1 个引用
159 private void btnLoadType_Click(object sender, EventArgs e)
160 {
161 try
162 {
163 tb_type.Text = listBox1.SelectedItem.ToString();
164 txType.Text = listBox1.SelectedItem.ToString();
165
166 #region //loading masterimage path
167 Load_ImgPath = StrPath + "\\" + "Proj" + "\\" + tb_type.Text + "\\" + "Image";
168 // 读双目图片（显示）
169 // 判断图像是否为空
170
171 if ((File.Exists(Load_ImgPath + "\\" + "masterimage.bmp")))
172 {
173 HOperatorSet.GenEmptyObj(out m_Image); 已用时间 <= 1ms
174 m_Image.Dispose();
175 HOperatorSet.ReadImage(out m_Image, Load_ImgPath + "\\" + "masterimage.bmp");
176 m_Tools.Imgshow(m_Image, m_hWindowHandle3, m_Image);
177
178 }


frmMain.cs*
Tools.cs
frmExe.cs
frmMain.cs [设计]*
frmExe.cs [设计]
OCRTrainee
OCRTrainee.frmMain
181 MessageBox.Show("Image invalid!");
182 }
183
184 #endregion
185
186 #region //laoding halcon code .txt file in WindowHandle III
187 Load_OCRCodePath = StrPath + "\\" + "Proj" + "\\" + tb_type.Text + "\\" + "Co
188 //read Halcon or OpenCV .txt code, and showup in WindowHandle III, "VisionSri
189 //but this function is optional. if no .txt file saved by user, then no showu
190 string s;
191 string txtCodePath = Load_OCRCodePath + "\\HCode.txt";
192 //ReadStream
193 m_Tools.ReadStream(ref txtCodePath, out s);
194 richTextBox1.Text = s.ToString();
195 #endregion
196
197 m_PubStore.m_type = tb_type.Text; //transfer string vairalbe
198
199 } 已用时间 <= 1ms
200 catch
201 {

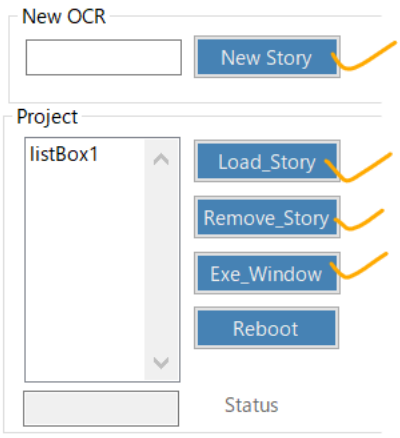
frmExe:


Master_Img
Save_Img
Expo（us）
New OCR
HL01
New Story
Project
Bottle
FInal01
HL01
PLettSh
Load_Story
Remove_Story
Exe_Window
Reboot
HL01
Status


host.exe
生命周期事件
线程: [22668] 主线程
堆栈帧: OCRTrainee.frmExe.frmExe
frmMain.cs
Tools.cs
frmExe.cs
frmMain.cs [设计]
frmExe.cs [设计]
frmExe.Designer.cs
OCRTrainee
OCRTrainee.frmExe
frmExe()
#endregion
Rotate Image if image text line has a degree
Execution OCR
/// <summary>
/// 构造函数
/// 把图片文件夹与代码文件夹写入
/// </summary>
1 个引用
public frmExe()
{
InitializeComponent();
m_Exetype = m_PubStore.m_type; //中间变量从frmMain传入
load_imagePath = StrPath + "\\" + "Proj" + "\\" + m_Exetype + "\\Image"; //图片文件夹
load_codePath= StrPath + "\\" + "Proj" + "\\" + m_Exetype + "\\Code"; //代码文件夹，story分类器文件夹
load_handlePath= StrPath + "\\" + "Proj" + "\\" + m_Exetype + "\\OCRHandle"; //story工作的分类器文件夹
load_classifPath = StrPath + "\\" + "Classifier"; //系统全部分类器
transfer



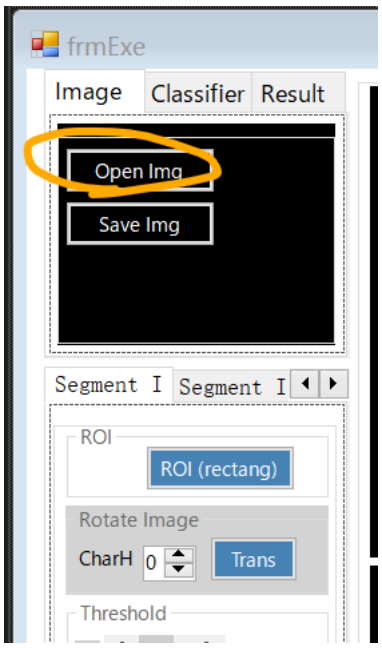
frmMain



frmExe

This is the execution winform for detail OCR operation.

Open image---m_Image




```
private void btnOpenImage_Click(object sender, EventArgs e)
{
    ofdImage.Filter = "(*.bmp,*.png;*.jpg;*.jpeg;*.tif)|*.bmp;*.png;*.jpg;*.tif";
    ofdImage.Multiselect = false;
    if (ofdImage.ShowDialog() == DialogResult.OK)
    {
        try
        {
            HOperatorSet.GenEmptyObj(out m_Image);
            m_Image.Dispose();
            HOperatorSet.ReadImage(out m_Image, ofdImage.FileName);

            m_Tools.Imgshow(m_Image, m_hWindowHandle1, m_Image);
        }
        catch (Exception)
        {
            MessageBox.Show("Image Invalid!");
        }
        finally
        {
            ofdImage.Dispose();
        }
```

Then get new m_Image.

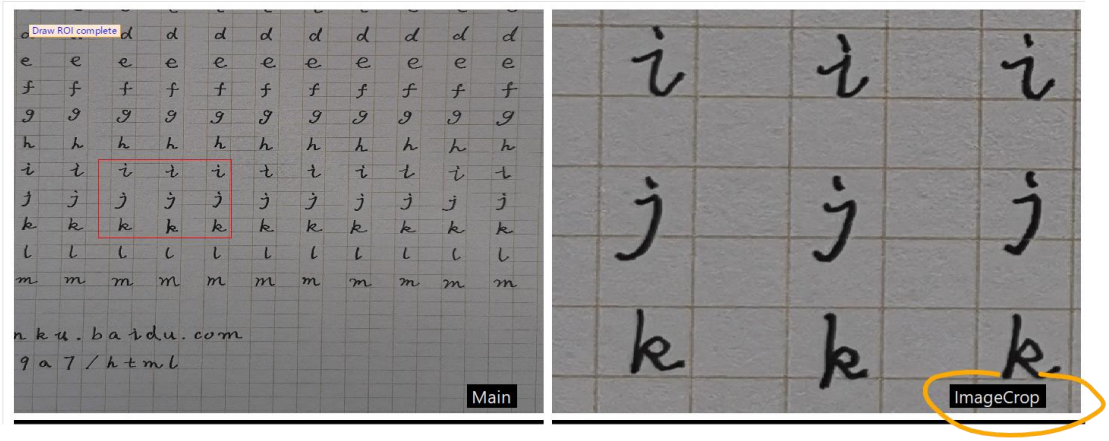
Halcon code accordingly

```
8 TrainFile := FontName+'Training.trf'
9 read_image (Image1,FontName+'train8.bmp')
```

ImageCropped for Display

ImageReduced is for real execution for OCR, and ImageCropped is only for display on the big windowHandle.

Example for ROI button




```
#region ROI rectangular
1 个引用
private void button5_Click(object sender, EventArgs e)
{
    try
    {
        HOperatorSet.GenEmptyObj(out m_Region);
        HOperatorSet.GenEmptyObj(out m_imageReduced);
        HOperatorSet.GenEmptyObj(out m_imageCroped);
        m_Region.Dispose();
        m_imageReduced.Dispose();
        m_imageCroped.Dispose(); //display for windowHandle2

        m_Tools.DrawRectangel1(hWindowControl1, m_Image, m_hWindowHandle1, ref m_Region, ref m_imageReduced,ref m_imageCroped);
        m_Tools.ImageFitWindow(m_imageCroped,m_hWindowHandle2);  //display for windowHandle2
        m_Region.Dispose();

        //Show execute result
        label1.Text = "ROI done";
    }
    catch
    {

    }
}
#endregion
```

Appendix

Halcon Code Example

OCR Story: PrintLetter

Create OCR TrainingDocument

a a a a a a a a a a a a
b b b b b b b b b b b b
c c c c c c c c c c c c
d d d d d d d d d d d d
e e e e e e e e e e e e
f f f f f f f f f f f f
g g g g g g g g g g g g
h h h h h h h h h h h h
...

A A A A A A A A A A
B B B B B B B B B B
C C C C C C C C C C
D D D D D D D D D D
E E E E E E E E E E
F F F F F F F F F F
G G G G G G G G G G
H H H H H H H H H H
I I I I I I I I I I
J J J J J J J J J J
K K K K K K K K K K
L L L L L L L L L L
M M M M M M M M M M
N N N N N N N N N N
O O O O O O O O O O
...

, , , , , , , ,
! ! ! ! ! ! ! !
– – – – – – – –
' ' ' ' ' ' ' '
- - - - - - - -
...

** ----------

Use append_ocr_trainf() collect more and more letters into training file

```
68
69 *TrainingFileName
70 TrainingFileName := FontName + 'Training.trf'
71 sort_region (RegionIntersection, Characters, 'character', 'true', 'row')
72 shape_trans (Characters, RegionTrans, 'rectangle1')
73 area_center (RegionTrans, Area, Row, Column)
74 for I := 0 to |Classes| -1by 1
75     select_obj (Characters, CharaterRegions, I+1)
76     shape_trans (CharaterRegions, RegionTrans, 'rectangle1')
77     area_center (RegionTrans, Area, Row, Column)
78     append_ocr_trainf (CharaterRegions, Image, Classes[I], TrainingFileName)
79     disp_message (WindowHandle, Classes[I], 'image', Row, Column, 'blue', 'true')
80 endfor
81 stop()
82
83 * Create the classifier.  We read out the classes from the train file.
84 * Therefore, the training part of the program is generic and can be
85 * used to train any OCR classifier.
86 *读入TrainFile，为训练做准备
87 read_ocr_trainf_names (TrainingFileName, CharacterNames, CharacterCount)
88 *利用MLP（多层感知器）创建一个新的OCR分级器
89 * NumHidden 隐藏层20层
90 create_ocr_class_mlp (8, 10, 'constant', 'default', CharacterNames, 20, 'normalization', 26, 42, OCRHandle)
91 * Train the classifier 训练分类器
92 trainf_ocr_class_mlp (OCRHandle, TrainingFileName, 100, 0.01, 0.01, Error, ErrorLog)
93 TrainingDocument:=FontName+'Classifier.omc'
94 write_ocr_class_mlp (OCRHandle, TrainingDocument)
95 stop ()
96 ************************************************
```

So the training file including a to z, A to Z, and

. . .

, , ,

! ! !

_ _ _

' ' '

- - -

Classifier Sample

...

For more info of *.trf pls see code uploaded

Do OCR

```
* Now test the classifier on the whole training image
*Prepare
FontName := 'E:/Halcon/OCR/OCRRobot1/'
*.trf _path
*TrainFile := FontName+'Training.trf'

dev_update_off ()
dev_get_window (WindowHandle)

*Read image
*ROI
read_image (Image1,FontName+'train8.bmp')
gen_rectangle1 (ROI_0, 960.708, 866.167, 1509.97, 2386.17)
reduce_domain (Image1, ROI_0, ImageReduced)

* Segment characters the same way as before---1.threshold or 2.further select_shape
******option1 just get Region directly
*binary_threshold (ImageReduced, Region, 'max_separability', 'dark', UsedThreshold1)
threshold (ImageReduced, Region, 0, 100)
******option2 also get Region
connection (Region, ConnectedRegions)
select_shape (ConnectedRegions, SelectedRegions, 'area', 'and', 127.49, 5000)
union1 (SelectedRegions, Region)
******
**Dilation
dilation_circle (Region, RegionDilation, 5.5)
connection (RegionDilation, ConnectedRegions)
intersection (ConnectedRegions, Region, RegionIntersection)
sort_region (RegionIntersection, Characters, 'character', 'true', 'row')
**show sort region result
count_obj (Characters, Number)
for i:=1 to Number by 1
    select_obj (Characters, ObjectSelected_i, i)
    area_center (ObjectSelected_i, Area_i, Row_i, Column_i)
    disp_message (WindowHandle, i, 'image', Row_i+60, Column_i, 'black', 'true')
endfor

* Classification
**read OCRHandle from file
read_ocr_class_mlp (FontName+'KenHLett01.omc', OCRHandle)
```

**do OCR
do_ocr_multi_class_mlp (Characters, Image1, OCRHandle, Class, Confidence)

* Display results
area_center (Characters, Area, Row, Column)
dev_display (Image1)
set_display_font (WindowHandle, 16, 'sans', 'true', 'false')
disp_message (WindowHandle, Class, 'image', Row +36, Column + 8, 'blue', 'false')
*set_display_font (WindowHandle, 16, 'mono', 'true', 'false')
disp_message (WindowHandle, 'Classification done', 'window', 12, 12, 'black', 'true')