


NHL  
STENDEN  
computer vision  
& data science



## Computer Vision

# License Plate Recognition

Klaas Dijkstra - Jaap van de Loosdrecht

10 April 2018

Copyright © 2001 – 2018 by  
NHL Stenden Hogeschool and Van de Loosdrecht Machine Vision BV  
All rights reserved

j.van.de.loosdrecht@nhl.nl, k.dijkstra.nhl.nl, jaap@vdlmv.nl

### High demand for software engineers with computer vision experience

- **Quality control and quality insurance become more important**
  - Efficient and effective
  - Recall actions
- **Availability of low cost vision solutions**
  - PC + camera
  - Intelligent camera
  - Cellphone
- **Using libraries (No development from scratch)**
- **Structured testing**
- **Evaluation of large datasets**
- **Applications**
  - Industry
  - Gaming
  - Surveillance
  - Augmented reality

27-8-2018 License plate recognition 2

**Demo**

- **Framework**
- **Find License Plate**
- **Find Characters**
- **Match Plate**
- **Lexicon**

27-8-2018 License plate recognition 3

**Overview**

- **Licenseplate recognition competition**
  - **Goal**
  - **Planning**
  - **Example**
  - **Rules**
- **Matching**
  - **Not using a lexicon**
  - **Using a lexicon**
  - **Rejecting classification results**
- **Framework**
  - **VisionLab**
  - **Components**
  - **Finding the license plate**
  - **Finding the characters**
  - **Reading the licenseplate**
- **C# Application**
  - **User interface**
  - **UML()**
- **Appendix**
  - **Generating the pattern matcher file**

27-8-2018 License plate recognition 4

## Goal

### Next week (homework):

- Take 50 photo's of license plates (each student)
- Resolution 1280x960 pixels
- Taken with sensible white balance and exposure
- Different Angles / Lighting conditions
- All Dutch rectangular car license plates (yellow)
- One licenseplate per image (fully visible)
- File format: XXXXXX.jpg example: RVLG20.jpg, FBCG13.jpg, FBCG13-2.jpg, FBCG13-3.jpg etc.

### Before the end of the course (Deliverables)

- 1) C# software for reading license plates
- 2) Report, with focus on the creativity of your solution

27-8-2018

License plate recognition

5

## Planning

### Homework:

- |            |  |
|------------|--|
| Week 1 – 4 | : Theory and assignments   |
| Week 5     | : Finish <i>LicensePlateMatcher.FindPlate()</i> in C# or<br>Finish <i>find_plate.js</i> in VisionLab           |
| Week 6     | : Finish <i>LicensePlateMatcher.FindCharacters()</i> in C# or<br>Finish <i>find_characters.js</i> in VisionLab |
| Week 7     | : Fully functional   |

### Extra

- Questions
- Receive final set
- Receive minimum score

### Midterm week

- Final competition
- Report
- No tweaking possible!

27-8-2018

License plate recognition

6

### Examples



27-8-2018

License plate recognition

7

### Rules

#### Not using the lexicon!

#### Scoring:

- 1 point for each correctly matched licenseplate
- 0 points for each unrecognized licenseplate
- 10 penalty points for each incorrectly recognized licenseplate

#### Grading:

- 1) Based on competencies of the course
- 2) Result of the competition on a selection of the photos
- 3) Minimum number of points will be determined after the selection of photos has been made
- 4) +1 for 1<sup>st</sup> and 2<sup>nd</sup> place (For the whole group)

27-8-2018

License plate recognition

8

**Matching**

27-8-2018 License plate recognition 9

**No Lexicon**

**Approach**

1. Match every character
2. Take character with the lowest error
3. Calculate confidence

**Applications:**

- Toll roads
- Speed camera's

27-8-2018 License plate recognition 10

**No Lexicon**

0 3 H Z H Z

1st	0	0.07	3	0.10	M	0.21	Z	0.08	H	0.58	Z	0.10
2nd	O	0.18	5	0.22	H	0.22	2	0.16	N	0.20	2	0.16
3rd	D	0.20	S	0.24	N	0.28	E	0.23	M	0.27	1	0.24

Green = Correct      Blue = Not matched

Red = Incorrect      Gray = Error

Confidence:

0.61	0.61	0.18	0.59	0.75	0.44
------	------	------	------	------	------

03MZH Z is incorrect !

27-8-2018
License plate recognition
11

**Lexicon**

**Approach**

1. Match every character
2. Match license plate to all possible license plates
3. Take plate with the lowest error for a whole word
4. Calculate confidence

**Applications:**

- Limited entry
- Camp sites
- Car parks

27-8-2018
License plate recognition
12

### Lexicon

0 3 H Z H Z

1st	0	0.07	3	0.10	M	0.21	Z	0.08	H	0.58	Z	0.10
2nd	O	0.18	5	0.22	H	0.22	2	0.16	N	0.20	2	0.16
3rd	D	0.20	S	0.24	N	0.28	E	0.23	M	0.27	1	0.24

Confidence: 0.44

Green = Correct	Blue = Not matched
Red = Incorrect	Gray = Error

03MZH Z is not a word in the lexicon

03HZHZ is chosen instead, which is correct !

27-8-2018
License plate recognition
13

### Rejection

1. Reject confidences below a certain value to detect mismatches.
2. Check *ground-truth* against the *match result*
3. Count instances for each category (TP, FP, FN, TN)
4. Score = True Positives - (10 \* False Positives)

Confusion matrix		Check with ground-truth	
		Correct	Incorrect
Match result	Accept	True Positive	False Positive
	Reject	False Negative	True Negative

Confidence threshold

*Confidence threshold* is a tunable parameter:

- Decreasing will move **False Negatives** to **True Positives** which is good
- Decreasing will move **True Negatives** to **False Positives** which is bad
- The opposite is true for increasing
- When trying to improve the overall performance (**True Positives** and **True Negatives**) the vision method or parameters have to be improved

27-8-2018
License plate recognition
14

## Framework

27-8-2018

License plate recognition

15

## VisionLab

VisionLab is used for image processing ([www.vdlmv.nl](http://www.vdlmv.nl)):

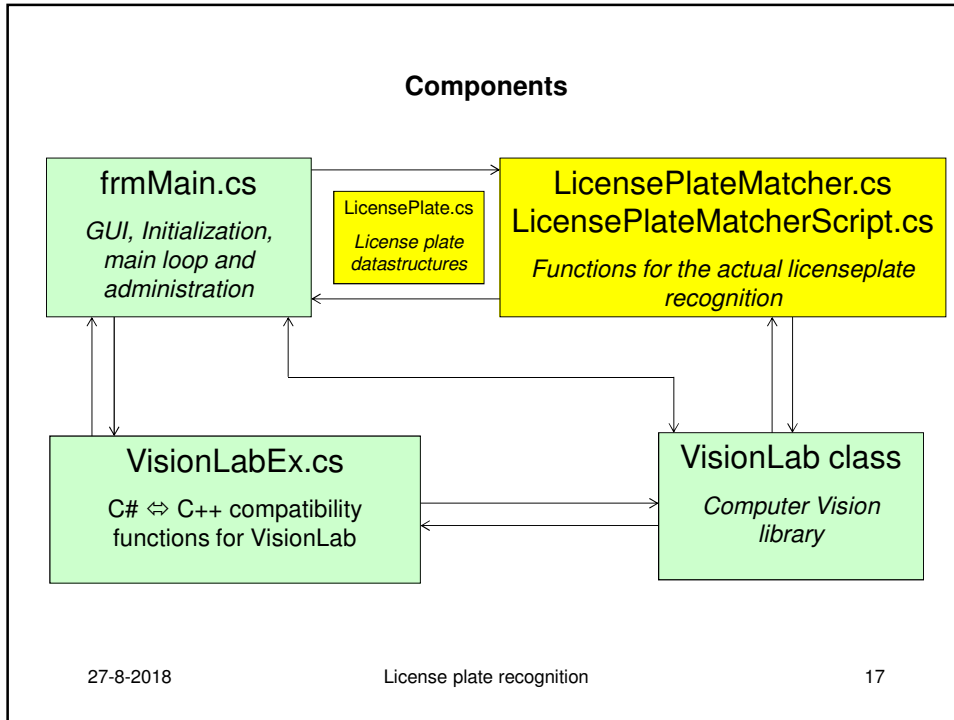
- Image processing algorithms
- Pattern matching
- Neural networks
- Genetic algorithms
- Algorithms written in ANSI C++
- OpenMP
- OpenCL
- Portable software:
  - Windows, Linux and Android
  - x86, x64, ARM and PowerPC

27-8-2018

License plate recognition

16





### LicensePlate.cs

```

public class LicenseCharacter {
    public LicenseCharacter(string character, double error, double confidence)
    public string character()
    public double error()
    public double confidence()
    public new string ToString()
}

public class LicensePlate {
    public LicensePlate()
    public double confidence
    public List<LicenseCharacter> characters
    public string getLicensePlateErrorsString()
    public string getLicensePlateString()
    public new string ToString()
}
  
```

27-8-2018 License plate recognition 18

```

LicensePlateMatcher.cs
    Functions for finding and reading license plate using VisionLab C# library

public class LicensePlateMatcher {

    public static bool FindPlate(           RGB888Image platelImage,
                                           ref Int32Image binaryPlatelImage)

    public static bool FindCharacters(     RGB888Image platelImage,
                                           Int32Image binaryPlatelImage,
                                           ref Int32Image labeledRectifiedPlatelImage)

    public static bool MatchPlate(        Int32Image binaryRectifiedPlatelImage,
                                           BlobMatcher_Int32 matcher,
                                           ClassLexicon lexicon,
                                           ref LicensePlate result,
                                           ref LicensePlate lexiconResult)

}

27-8-2018                               License plate recognition                               19

```

```

LicensePlateMatcherScript.cs
    Functions for finding and reading license plate using VisionLab .JL scripts

public class LicensePlateMatcher {

    public static bool Init()              // Read scripts and pattern matcher
                                           // initialize command interpreter

    public static bool FindPlate(         RGB888Image platelImage,
                                           ref Int32Image binaryPlatelImage)

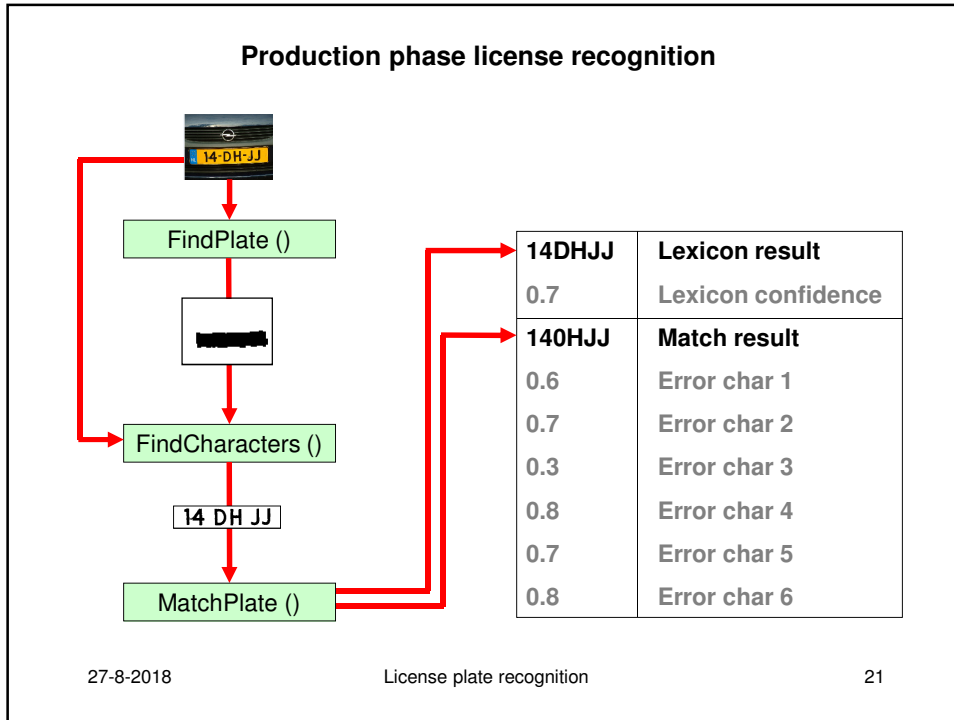
    public static bool FindCharacters(     RGB888Image platelImage,
                                           Int32Image binaryPlatelImage,
                                           ref Int32Image labeledRectifiedPlatelImage)

    public static bool MatchPlate(        Int32Image binaryRectifiedPlatelImage
                                           ref LicensePlate result)

}

27-8-2018                               License plate recognition                               20

```



### public static bool FindPlate ()

#### Description:

Find the largest license plate in the image

1. Segment using ThresholdHSVchannels
2. Remove blobs which are not license plates

#### Input:

//Original image  
RGB888Image platelImage

#### Output:

//Segmented license plate  
ref Int32Image binaryPlatelImage

#### Return:

//License plate found?  
bool



27-8-2018

License plate recognition

23

### IsServerScript

```
$ServerScript = IsServerScript

if $ServerScript == false
  //Copy script selected image (F6)
  copy %currentimage OriginalImage
else
  //Copy first image passed by C#
  copy %p1 OriginalImage
endif
```

```
if $ServerScript == false
  display LicensePlateBin
else
  //Copy result image back to C#
  copy LicensePlateBin %p2
endif
```

If IsServerScript returns *true*, the script is being executed from C#.

If IsServerScript returns *false*, the script is being executed from the VisionLab Client.

This can be used to autodetect if images are passed by C# or if selected images in VisionLab should be used. This is convenient for debugging scripts.

27-8-2018

License plate recognition

24

## find\_plate.jls

```

$ServerScript = IsServerScript
if $ServerScript == false
  //Copy script selected image (F6)
  copy %currentImage OriginalImage
else
  //Copy first image passed by C#
  copy %p1 OriginalImage
endif

//Convert from RGB888Image to HSV888Image
FasterRGBToHSV OriginalImage OriginalImage

//*****
//*** Exercise: Find license plate ***
//*****

//Threshold HSV
ThresholdHSVchannels OriginalImage LicensePlateBin Int32Image 21 50 100 255 100 255
//Remove small blobs
RemoveBlobs LicensePlateBin EightConnected Area 1 5000 UseX
//Fill up characters
FillHoles LicensePlateBin FourConnected

if $ServerScript == false
  display LicensePlateBin
else
  //Copy result image back to C#
  copy LicensePlateBin %p2
endif

//Return true, if pixels found
$sum = SumIntPixels LicensePlateBin
if $sum > 0 then
  return true
else
  return false
endif

```

## LicensePlateMatcher.FindPlate()

```

public static bool FindPlate( RGB888Image plateImage, ref Int32Image binaryPlateImage ){

  const int c_threshold_h_min = 21;
  const int c_threshold_h_max = 50;
  const int c_threshold_s_min = 100;
  const int c_threshold_s_max = 255;
  const int c_threshold_v_min = 100;
  const int c_threshold_v_max = 255;
  const int c_remove_blobs_min = 1;
  const int c_remove_blobs_max = 5000;

  HSV888Image plateImageHSV = new HSV888Image();
  //Convert to RGB to HSV
  VisionLab.Convert(plateImage, plateImageHSV);

  //*****
  //*** Exercise: ***
  //*** adjust licenseplate ***
  //*** segmentation ***
  //*****

  //Threshold HSV image
  VisionLab.Threshold3Channels( plateImageHSV, binaryPlateImage,
                              c_threshold_h_min, c_threshold_h_max,
                              c_threshold_s_min, c_threshold_s_max,
                              c_threshold_v_min, c_threshold_v_max

  //Remove blobs with small areas
  VisionLab.RemoveBlobs( binaryPlateImage, Connected.EightConnected,
                        BlobAnalyse.BA_Area,
                        c_remove_blobs_min, c_remove_blobs_max);

  //Fill up characters
  VisionLab.FillHoles( binaryPlateImage, Connected.FourConnected);

  plateImageHSV.Dispose();
  //Return true, if pixels found
  return (VisionLab.SumIntPixels(binaryPlateImage) > 0);
}

```

27-8-2018

License plate recognition

26

### LicensePlateMatcherScript.FindPlate()

```
public static bool FindPlate( RGB888Image plateImage, ref Int32Image binaryPlateImage ){
    //Upload image to VisionLab command interpreter
    VisionLab.SetRGB888Image(cmdInt, "plateImage", plateImage);

    //Execute scrip using the "icall" command
    String result = StripTime(
        cmdInt.ExecRequest("icall FindPlate plateImage, binaryPlateImage")
    );

    //Download result image from VisionLab command interpreter
    VisionLab.GetInt32Image(cmdInt, "binaryPlateImage", binaryPlateImage);

    //Return if the script returned "true" or "false"
    return (result == "true");
}
```

27-8-2018

License plate recognition

27

### Solutions brainstorm for finding the license plate

1. Tune *ThresholdHSVChannels*  
Find darkest and brightest yellow license plates  
Analyse HSV values, apply values, test values
2. Tune *RemoveBlobs*  
Find smallest and largest licensplate  
Analyse *Area*, apply criteria, test criteria
3. Add additional criteria  
Add *RemoveBlobs* line using *LengthBreadthRatio* as a feature  
Add *RemoveBlobs* using additional features
4. Add additional segmentation functions  
Use a different alternative *Threshold* values
5. Etcetera  
This will score points



27-8-2018

License plate recognition

28

## Finding the characters

27-8-2018

License plate recognition

29

### **public static bool** FindCharacters ()

#### Description:

Locates the characters of the license plate

- Warp image (Rectify)
- Segment characters
- Remove blobs which are too small (Lines between characters)

#### Input:

```
//Original image
RGB888Image platelImage
//Segmented license plate
Int32Image binaryPlatelImage
```

#### Output:

```
//Image containing binary six characters
ref Int32Image binaryCharacterImage
```

#### Return:

```
//Function executed successfully
bool
```



27-8-2018

License plate recognition

30

## find\_characters.jls

```

$ServerScript = IsServerScript
if $ServerScript == false
  //copy script selected image (F6) and second selected image (F5)
  copy %currentimage OriginalImage
  copy %secondimage LicensePlateBin
else
  //Copy first and second image passed by C#
  copy %p1 OriginalImage
  copy %p2 LicensePlateBin
endif

//Find corner points of the license plate
$found = FindCornersRectangle LicensePlateBin EightConnected 0.5 Landscape %$tab
if $found then
  //Warp (rectify) licenseplate
  Warp OriginalImage binaryCharacterImage ForwardT $tab[0] $tab[1] $tab[2] $tab[3]
  100 470 0 NearestPixelInterpolation

  //***** Exercise: Find license plate characters *****/
  //*****

  Convert binaryCharacterImage binaryCharacterImage Int32Image
  //Automatic threshold finds black letters
  ThresholdIsodata binaryCharacterImage DarkObject
  //Remove all blobs connected to the border
  RemoveborderBlobs binaryCharacterImage EightConnected AllBorders
  //Remove all blobs with an area less than 400
  RemoveBlobs binaryCharacterImage EightConnected Area 1 400 UseX

  if $ServerScript == false
    display BinaryCharacterImage
  else
    //Copy result image back to C#
    copy BinaryCharacterImage %p3
  endif
  return true
else
  return false
endif

```

## Solutions brainstorm for finding characters

1. Try *FindCornersRectangleSq* operator/function  
Instead of *FindCornersRectangle*
2. Tune *RemoveBlobs*  
Find smallest and largest licenseplate  
Analyse *Area*, apply criteria, test criteria
3. Add additional criteria  
Add *RemoveBlobs* using additional features
4. Add additional segmentation functions  
Use a few different *Threshold* values and methods (manual vs. automatic)
5. Use a background correction  
Subtract the background
6. Use a different color space  
Use HSV
7. Use binary morphological filters to dilate or erode blobs  
Try making the blobs more like the characters in the .pm file
8. Etcetera  
This will score points



27-8-2018

License plate recognition

32



## Reading the license plate

27-8-2018

License plate recognition

33

### public static bool MatchPlate ()

#### Description:

Read the license plate

#### Input:

//Rectified license plate image containing six characters

Int32Image labeledRectifiedPlateImage

BlobMatcher\_Int32 matcher //initialized blobmatcher

ClassLexicon lexicon //initialized lexicon

#### Output:

//Result by the blob matcher

ref LicensePlate result

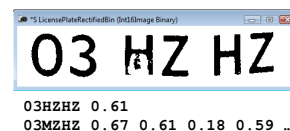
//Result by the lexicon

ref LicensePlate lexiconResult

#### Return:

//six characters found?

bool



27-8-2018

License plate recognition

34

### match\_plate.jls (Part I)

```

$ServerScript = IsServerScript
if $ServerScript == false
  //Copy third selected image (Operator->Select 3rd)
  copy %thirdimage LicensePlateRectifiedBin
  $lpwd = lpwd
  cwd $lpwd
  PM_ReadFromFile PatternMatcher ../VL/lic_fonts.pm //Read pattern matcher file
else
  copy %pl LicensePlateRectifiedBin
endif

//Analyse blobs locations
copy LicensePlateRectifiedBin LicensePlateRectifiedLabel
labelblobs LicensePlateRectifiedLabel EightConnected
$maxBlobIndex = BlobAnalysisArray LicensePlateRectifiedLabel &$tab SortDown TopLeft UseX Height
TopLeft Width
if $maxBlobIndex != 5 then //Check if 5 characters were found
  return false
endif

...

```

### match\_plate.jls (Part II)

```

...

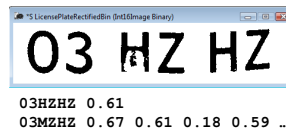
//Begin matching in a for loop
SetSizeArray &$errors 0
$matches = ""
$confidence = 1000
for $i = 0 to $maxBlobIndex do
  VarToArray &$tab[$i] &$elm
  $label = $elm[0]
  $h = $elm[1]
  $t1 = $elm[2]
  $w = $elm[3]
  $x = getnthfromvector 1 $t1
  $y = getnthfromvector 2 $t1
  ROI LicensePlateRectifiedBin LicensePlateRectifiedBinROI $x $y $h $w
  $bestMatch = PM_BestMatch LicensePlateRectifiedBinROI PatternMatcher -0.5 0.5 //Match
  $patternId = GetNthWord 1 $bestMatch //Get pattern ID
  $conf = GetNthWord 2 $bestMatch //Get confidence
  $error = GetNthWord 3 $bestMatch //Get error
  $patternName = PM_PatternName PatternMatcher $patternId //Convert pattern ID to confidence
  if $conf < $confidence
    //Keep lowest character confidence as license plate confidence
    $confidence = $conf
  endif
  $errors[$i] = $error
  $matches = $matches . $patternName
endifor

//Return results
ArrayToVar &$errors &$errors
$result = concat $matches $confidence $errors
return $result

```

### Solutions brainstorm for reading the license plate

1. **Adapt pattern matcher (.pm file)**  
Analyze which characters occur in real license plates  
Use a different font in the .pm file
2. **Check or correct license plate grammar**  
D3-HF-BR is not very likely to be a license plate, while 03-HF-BR is
3. **Use separate matchers for numbers and letters**  
Use two .pm files
4. **Etcetera**  
This will score points



27-8-2018

License plate recognition

37

### Framework

27-8-2018

License plate recognition

38

The screenshot shows the LicensePlateMatcher application. Key elements are annotated with red boxes and numbers:

- 1: File list showing images to be processed.
- 2: Current score (4).
- 3: Process button to begin processing.
- 4: License plate image.
- 5: Confidence threshold (35).
- 6: Results table showing match results for various license plates.
- 7: Export files button for true positives.
- 8: Export files button for false positives.
- 9: Export files button for rejected items.
- 10: Export files button for false positives (high confidence, wrong match).

Results Table:

Result	Confidence	Error 1	Error 2	Error 3	Error 4	Error 5	Error 6
13NJZV	0,491	0,05	0,1	0,12	0,13	0,05	0,06

True Positives (with lexicon):

- 13NJZV.JPG
- 13VTRG.JPG
- 35PN30EMO.JPG
- 3VFS241.JPG

FindPlate() Returns false:

- ZVNB06.JPG
- 54SLT1.JPG

FindCharacters() Returns false:

- 08GVGG.JPG
- 29TRNG.JPG

MatchPlate() Returns false:

- 3DNZ2H.JPG
- 36DZHT.JPG

Reject:

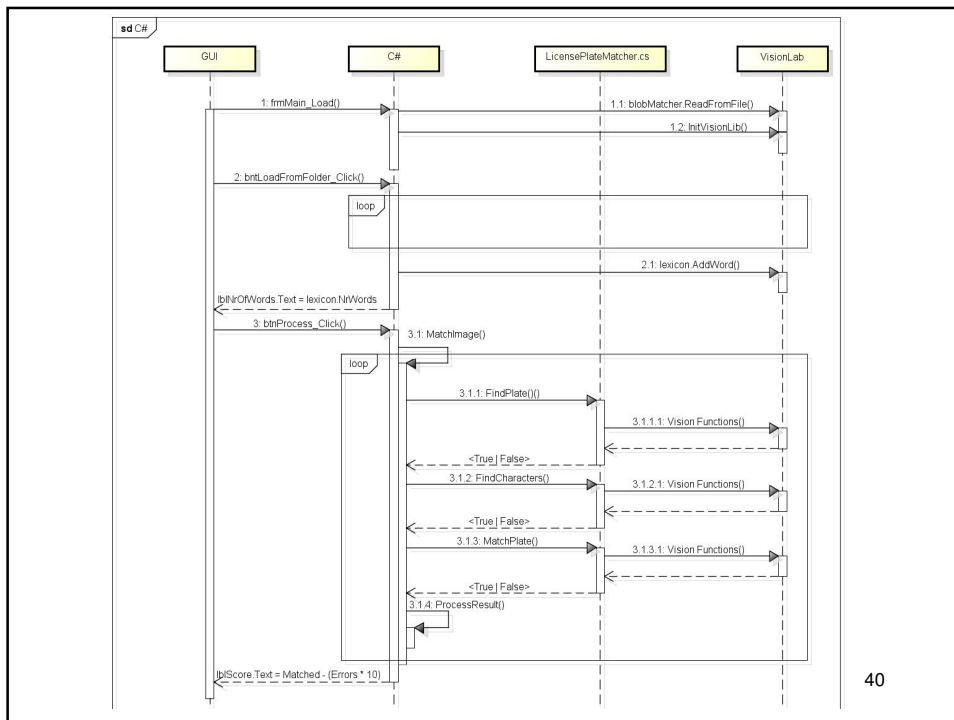
- 03HFB8.JPG
- 03HZHZ-3.JPG
- 13VTRG.JPG
- 21PNVU.JPG

False Positives:

- None listed.

27-8-2018 License plate recognition 39

1. Images to be processed
2. Current score
3. Begin processing
4. Output images from LicensePlateMatcher (Script)
5. Reject below this confidence divided by 10
6. Match result from current licenseplate
7. Functions return values for these licenseplates
8. True Positive (Correct match and high confidence)
9. Reject (Confidence is too low)
10. False Positive (Confidence is high, but match is wrong)



## Appendices

27-8-2018

License plate recognition

41

### public void frmMain.DisplayBlobs ()

**Description:**

Display features of the blob to the debugging output of C#

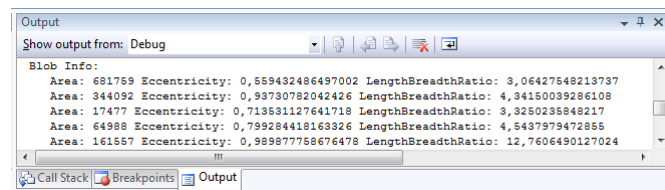
**Input:**

//Binary image containing the blobs

Image binaryImage

**Effect:**

//Blob features to the output

**\* Note: only works in debug mode**

27-8-2018

License plate recognition

42

```

public void DisplayBlobs(Image binaryImage)
{
    vector_BlobAnalyse ba = new vector_BlobAnalyse();
    vector_Blob blobs = new vector_Blob();
    ba.Add(BlobAnalyse.BA_Area);
    ba.Add(BlobAnalyse.BA_Eccentricity);
    ba.Add(BlobAnalyse.BA_LengthBreadthRatio);
    VisionLabEx.GetBlobsInfo(binaryImage, ba, ref blobs);
    System.Diagnostics.Debug.WriteLine("Blob Info: ");
    foreach (Blob b in blobs)
    {
        System.Diagnostics.Debug.WriteLine( "   Area: " +
                                             b.Area().ToString() +
                                             " Eccentricity: " +
                                             b.Eccentricity().ToString() +
                                             " LengthBreadthRatio: " +
                                             b.LengthBreadthRatio().ToString());
    }
    ba.Dispose();
    blobs.Dispose();
}

```

27-8-2018

License plate recognition

43

### gen\_lic\_pm.jls (VisionLab)

**Idea:**

Generate the license plate *pattern matcher*

**Input:**

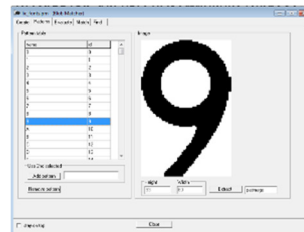
lic\_fonts.jl => Image containing the characters

**Return:**

lic\_fonts.pm => Pattern matcher containing the characters

0123456789 ABCDE

Lic\_fonts.jl



Lic\_fonts.pm

27-8-2018

License plate recognition

44

```
PM_CreateBlobMatcher pm Int16Image 60 1 20 0
$names = 0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
VarToArray &$names &$nameTab
lread allPats lic_fonts.jl
copy allPats allPatsB
ThresholdIsodata allPatsB DarkObject
$nrNums = LabelBlobs allPatsB EightConnected
$maxBlob = BlobAnalysisArray allPatsB &$tab SortDown TopLeft UseX Height TopLeft Width
for $i = 0 to $maxBlob do
  VarToArray &$tab[$i] &$elm
  $label = $elm[0]
  $h = $elm[1]
  $t1 = $elm[2]
  $w = $elm[3]
  $x = getnthfromvector 1 $t1
  $y = getnthfromvector 2 $t1
  ROI allPats roi $x $y $h $w
  copy roi roiB
  Threshold roiB 0 100
  PM_AddPattern roiB pm $nameTab[$i]
endfor
PM_WriteToFile pm lic_fonts.pm
PM_Delete pm
```

27-8-2018

License plate recognition

45