

# Sistemas Sensoriais

## 1º Trabalho Laboratorial – Detecção e Reconhecimento de matrículas

Nº Aluno	Nome	Turno
53579	Eduardo Dias	P6
52853	Fábio Pires	P6
52755	Filipe Marques	P6

Docente das aulas práticas: André Mora

Ano letivo: 2021/2022

Data: 10-1-2022

## Índice

1- Introdução.....	3
2- Requisitos da aplicação.....	4
3- Abordagem ao projeto.....	5
4- Reconhecimento de caracteres .....	7
5- Separação dos caracteres da matrícula .....	8
6- Localização da matrícula.....	10
6.1- Localização horizontal da matrícula .....	10
6.2- Localização vertical da matrícula – limite esquerdo.....	12
6.3- Localização vertical da matrícula – limite direito .....	12
7- Aplicação dos três métodos.....	14
8- Resultados obtidos e considerações.....	16
9- Conclusões .....	17
10- Código fonte utilizado.....	17

## **1- Introdução**

Este relatório tem como objetivo o reconhecimento de matrículas bem como os seus caracteres. Este reconhecimento vai ser feito através de imagens de veículos portugueses.

Cada imagem apresentada irá ser acompanhada de um nível de dificuldade de onde está localizada a matrícula.

Para este projeto irá ser utilizado o funções acondicionamento da imagem e binarização da imagem, como filtros de media, mediana, binarização pelo método de OTSU e projeções quer verticais, quer horizontais da imagem.

Para o teste de resultados utilizou-se o EvalForm e as imagens utilizadas foram disponibilizadas pelos docentes da cadeira.

Ao longo deste relatório irão ser mostradas bocados de código para demonstrar a execução do programa, mas em anexo irão estar disponibilizados os códigos fontes das funções principais utilizadas.

## **2- Requisitos da aplicação**

A aplicação a ser desenvolvida terá de se guiar por um número de requisitos, sendo as mais importantes as referidas em baixo:

1. O programa tem de disponibilizar uma aplicação onde numa imagem se pode aplicar as funções desenvolvidas nas aulas prática
2. O reconhecimento dos caracteres individuais deverá ser feito por análise de semelhança com uma lista de imagens de caracteres alfanuméricos previamente guardada.
3. Para a localização da matrículas poderá ser utilizado um algoritmo de deteção de contornos e um algoritmo de projeções verticais ou horizontais na imagem.
4. As matrículas aceites são do tipo A, matrículas antigas com hífen de dois em dois caracteres, ou do tipo B, matrículas recentes, sem hífen.
5. Todo o processamento deverá ser feito através de uma função de nome LP\_Recognition avaliada através do EvalForm.
6. Existirá uma matrícula por imagem e a sua dimensão poderá variar bem como a iluminação.

Antes da inicialização deste projeto, foram disponibilizados o EvalForm e o cabeçalho da função LP\_Recognition que permite a realização dos requisitos 1 e 5.

As várias imagens disponibilizadas pelos docentes correspondem a vários níveis de dificuldade, sendo que o nível 1 corresponde ao reconhecimento da matrícula isolada, o nível 2 à localização e reconhecimento de matrícula, ou seja, uma imagem com a matrícula e alguns elementos há volta. Por fim o nível 3 é a localização e reconhecimento da matrícula onde a imagem apresenta o carro inteiro.

Ainda dentro de cada nível de dificuldade, as imagens podem variara de acordo com a iluminação, ou a sujidade da matrícula.

### 3- Abordagem ao projeto

Para começar a escrever o algoritmo para a aplicação, foi utilizado uma abordagem *bottom-up*, ou seja, começou-se a desenvolver algoritmos dos requisitos mais simples para os mais complexos.

Começou-se a desenvolver o algoritmos de reconhecimento de caracteres, depois um algoritmo para a separação de caracteres e por fim um algoritmo para a localização da matrícula na imagem.

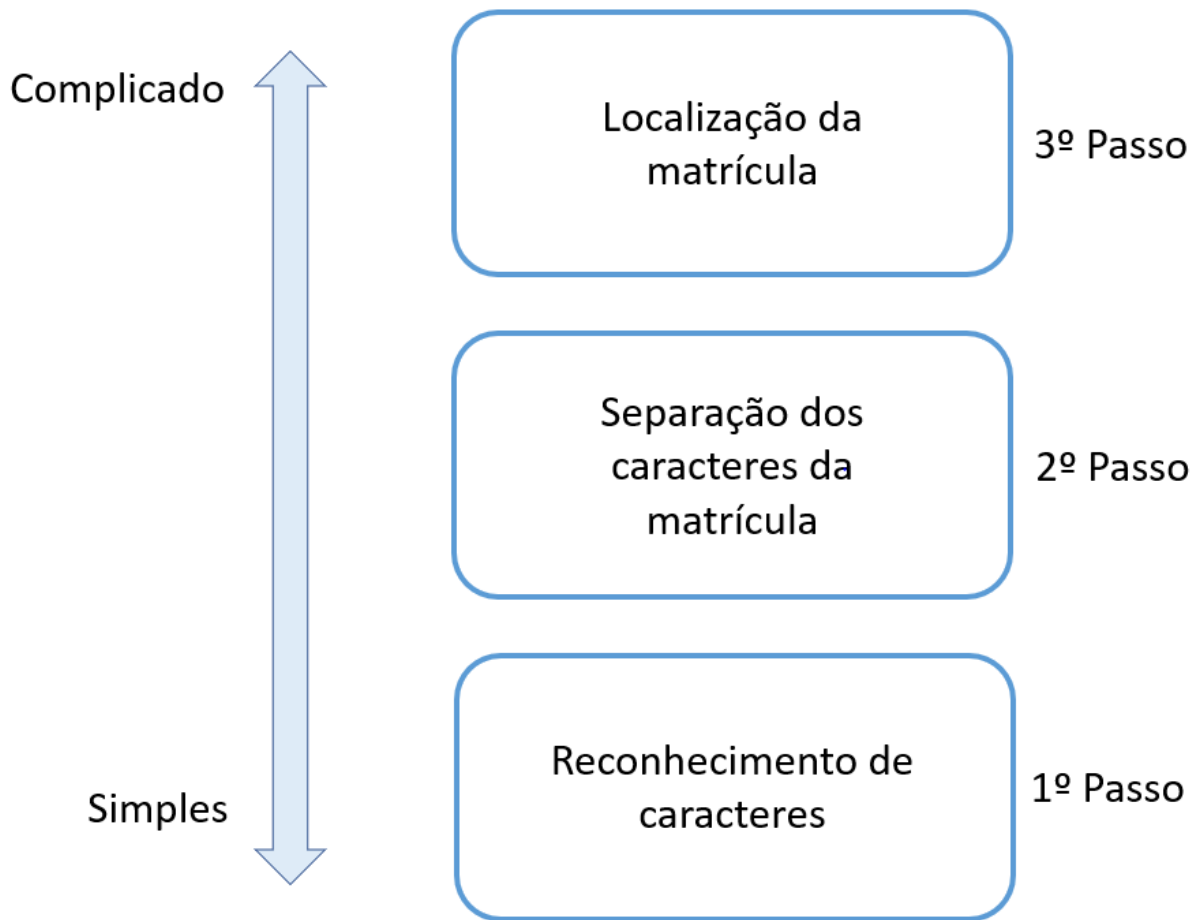


Figura 1-Abordagem ao projeto

Para cada um dos passos foi criado uma função principal complementadas por outras funções, estas funções principais são elas:

FUNÇÃO	OBJETIVO DA FUNÇÃO
LP_RECOGNITION	Função principal da aplicação
REC_CHAR	Função do reconhecimento de caracteres
LOC_CHAR	Função para separação dos caracteres
LOC_PLATE	Função para localização da matrícula na imagem

Ao longo deste relatório vão ser apresentadas figuras de cada passo para demonstrar o bom funcionamento de código, a imagem original usada é apresentada em baixo



*Figura 2-Imagem a ser trabalhada*

## 4- Reconhecimento de caracteres

Para o desenvolvimento deste passo, considerou-se que o grupo disponibilizava dos caracteres já individualizados da matrícula.

Foi disponibilizado ao grupo pelos docentes uma pasta que continha ficheiros das várias letras do alfabeto e números de 0 a 9. Com este ficheiro é possível fazer a comparação com os caracteres da matrícula para os identificar.

Procedeu-se a criar um vetor de imagens onde se colocariam os ficheiros de comparação.

Foi criado então o algoritmo de comparação de caracteres denominado `rec_char`, que tem como parâmetros de entrada o nosso caracter matrícula e o vetor de imagens com os ficheiros de comparação. Para a comparação ser mais viável houve um redimensionamento da nossa imagem do caracter e dos vetor de imagens para o mesmo tamanho.

Na figura 1 é possível observar uma parte importante da função `rec_char`.

```
for (y = 0; y < height; y++)
{
    for (x = 0; x < width; x++)
    {
        if (dataPtr[0] == dataPtr_BD[0])    //ver porque o 0 blue, 1 red e 2 green
            p++;

        dataPtr += nChan;
        dataPtr_BD += nChan_BD;
    }
    dataPtr += nChan;
    dataPtr_BD += nChan_BD;
}
Console.WriteLine(char_name[i] + ": " + p);
if (p > p_max)
{
    p_max = p;
    LP_Char = char_name[i];
}

return LP_Char;
```

*Figura 3-Comparação dos caracteres*

O código em cima funciona do seguinte modo:

1. A todas as imagens é aplicada uma binarização pelo método de OTSU.
2. A começar pelo início da imagem do caracter da matrícula e pelo início da primeira imagem no vetor de imagens, vai haver um varrimento da imagem e sempre que o valor da cor do pixel é correspondente nas duas imagens existe um incremento de um contador.
3. No fim deste código percorreu-se todas as imagens do vetor de imagens e aquela que apresentar um contador de maior valor é a imagem correspondente que será retornada pela variável `LP_Char`.

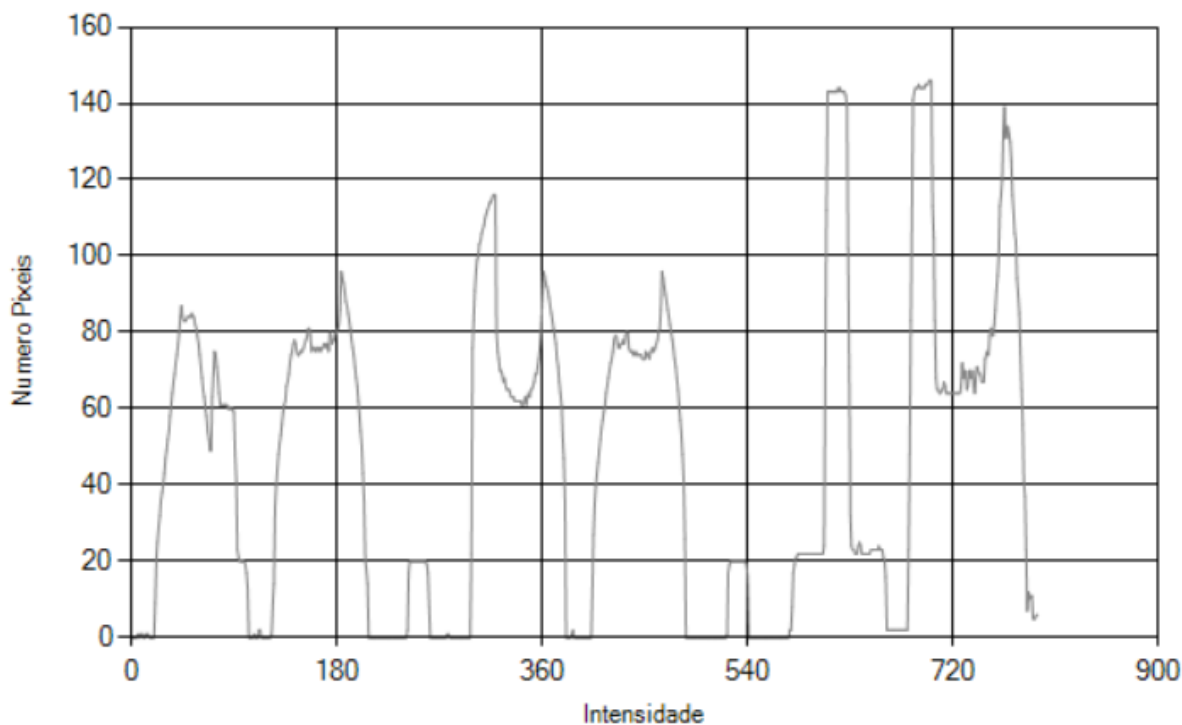
## 5- Separação dos caracteres da matrícula

Para o desenvolvimento deste passo, considerou-se que o grupo disponibilizava de uma imagem em que apresenta só a matrícula.

Neste algoritmo há imagem foi aplicado uma binarização por método de OTSU e de seguida fez-se projeções verticais e horizontais separadamente na imagem, onde esses resultados são guardados em vetores.

Estas projeções dependendo de ser for vertical ou horizontal fazem respetivamente um varrimento em x e em y guardando a intensidade de níveis de preto nas respetivas coordenadas x e y da imagem.

A figura 3 apresenta uma projeção vertical de uma imagem de uma matrícula.



*Figura 4-Projeção vertical de pretos*

Por fim para a separação de caracteres foi utilizada a função `loc_char` que recebe como parâmetro uma imagem, uma cópia da imagem e as variáveis do tipo `rectangle` para a localização dos caracteres.

Abaixo apresenta-se uma parte do código para a separação vertical dos caracteres.



```

{
    if (vector_vert[x] > 6 && x_finish == 0) //vê quando começa a haver pixels pretos
    {
        x_indice_inicial = x;
        x_finish = 1;
    }
    else if (vector_vert[x] < threshold && x_finish == 1 || x == width - 1)
    {
        x_indice_final = x;

        if (x_max > (val_max_string / 2.2))
        {
            Console.WriteLine(x_indice_inicial);
            Console.WriteLine(x_indice_final);
            X_vert[num_x] = x_indice_inicial;
            X_vert[num_x + 1] = x_indice_final;
            num_x = num_x + 2;
        }
        x_finish = 0;
        x_max = 0;
    }

    if (x_finish == 1)
    {
        if (x_max < vector_vert[x])
        {

```

Figura 5-Separação vertical da matrícula

Para a separação vertical, sabe-se que uma região que começa num valor  $x$  de intensidade de pretos e que ao correr o vetor, essa intensidade passa a 0, será um carácter da matrícula. O código acima faz essa análise e guarda os índices de início e fim do bloco de intensidade de pretos, até guardar 12 índices correspondentes aos 6 caracteres.

Um problema encontrado foi nas matrículas de tipo A onde, pela figura 3, se pode observar que os blocos de preto com maior intensidade são os caracteres, mas existe no meio blocos de intensidade de preto correspondentes ao hífen da matrícula. Uma solução encontrada foi, antes desta análise, encontrar o pico de intensidade máxima do vetor de intensidades e ao analisar cada bloco, se o seu pico for maior que  $(pico\ máximo)/2.2$  então considera-se esse bloco como um carácter.

De seguida utilizou-se o mesmo processo para a projecção vertical da matrícula, para distinguir os caracteres de qualquer interferências de letras ou números que não dizem respeito há matrícula.



Figura 6-Caracter isolado

A figura 5 demonstra o resultado da separação do carácter 6 da matrícula.

## 6- Localização da matrícula

Neste capítulo irá ser abordado o método utilizado para fazer a localização da matrícula numa imagem, esta função só irá servir para matrículas de nível 2 e 3. A função onde este processo ocorre é a Loc\_plate

### 6.1- Localização horizontal da matrícula

Para se localizar a matrícula é necessário começar pela localização horizontal na imagem, mas o grupo deparou-se logo com o problema do ruído presente. Para mitigar este efeito é utilizado um filtro de mediana de tamanho 7 para eliminar os ruídos de alto valor.

O passo seguinte foi aplicar um algoritmo de contornos na matrícula e de seguida uma binarização com o método de OTSU. O algoritmo de contornos usado foi um de diferenciação, mas só na horizontal.

Estes dois passos são importantes pois permite a distinção da matrícula do que a rodeia, pois, a matrícula apresenta um elevado valor de branco. Também é uma maneira de distinguir e remover algum tipo de sombras na imagem.

Por fim é aplicada a esta imagem uma projeção horizontal, mas de intensidade de brancos.

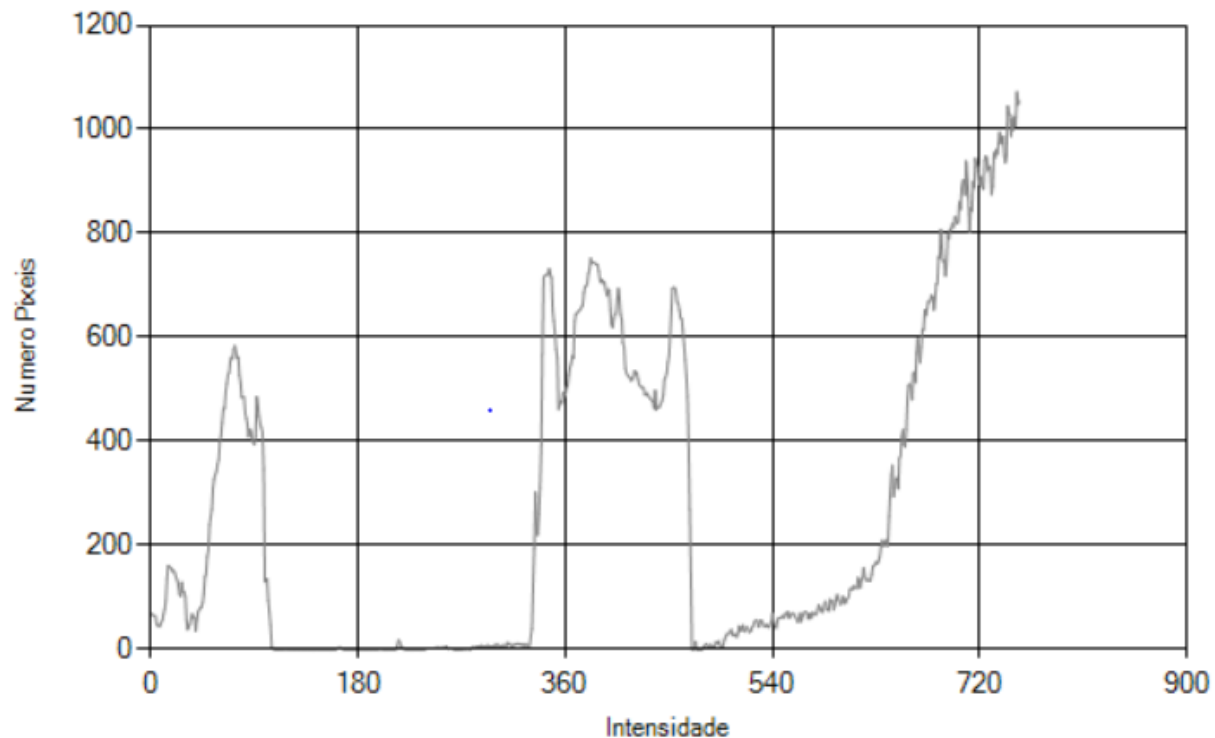


Figura 7-Projeção horizontal de brancos

Como podemos observar na figura 5 existe muitos locais da imagem onde a quantidade de brancos é elevada. Para fazermos a distinção e encontrarmos o limite superior e inferior na matrícula utilizou-se o código abaixo

```
for (y=0; y<height;y++)
{
    if (vector[y] > width * 0.008 && vector[y] < width * 0.07)
    {
        if (i == 0)
        {
            vector_y_test[0] = y;
            i = 1;
        }
        contador_test++;
    }
    else if (i == 1)
    {
        vector_y_test[1] = y;
        if (contador < contador_test)
        {
            contador = contador_test;
            vector_y[0] = vector_y_test[0];
            vector_y[1] = vector_y_test[1];
        }
        contador_test = 0;
        i = 0;
    }
}
```

Figura 8-Código para localizar limites da matrícula

A técnica para a distinguir a matrícula na imagem é se inferirmos que a quantidade de brancos de uma matrícula é sempre maior que  $0.008 * largura da imagem$  e menor que  $0.07 * largura da imagem$ .

Um outro problema que esta técnica encontra é se existir picos de ruídos dentro desta gama de valores. A solução encontrada é saber entre esses picos todos encontrada qual tem um intervalo de intensidades maior, e esse intervalo será o intervalo horizontal da matrícula.

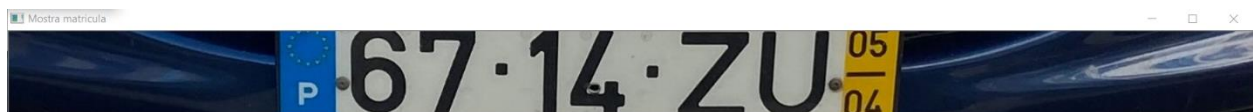


Figura 9-Imagem cortada horizontalmente

## 6.2- Localização vertical da matrícula – limite esquerdo

Obtendo-se a localização do limite superior e inferior da matrícula, tem-se que agora arranjar os seus limites verticais.

Para isso, é utilizada uma imagem igual há da figura 9 onde é aplicada um algoritmo de contornos, diferenciação, e binariza-se a imagem normalmente com um threshold escolhido de 50.

E é aplicado uma projeção vertical das intensidades de branco na imagem.

Para simplificar o processo da descoberta dos limites verticais da imagem o grupo observou que no gráfico de intensidades um valor alto iria corresponder ao início da matrícula, então foi definido que se houvesse um pico de intensidades superior a  $0.45 * (\text{altura da imagem})$ , esse índice do vetor de intensidades correspondente é o limite esquerdo da matrícula.



Figura 10-Imagem com 1ª corte vertical

## 6.3- Localização vertical da matrícula – limite direito

Neste ponto falta só descobrir o limite vertical direito da imagem.

O código desenvolvido começa por aplicar um filtro de mediana de tamanho 7 para eliminar qualquer ruído e aplicar uma binarização com um threshold de 80.

De seguida é aplicada uma projeção vertical de intensidade de pretos.

```
for (int s = 2; s < width_img1_2; s++)
{
    if ((vector_x_teste[s] < (height_img1_2 * 0.1)) && entry == 0)
    {
        entry = 1;
        maximo = 0;
    }
    if (entry == 1)
    {
        if (vector_x_teste[s] > height_img1_2 * 0.1)
        {
            if (cont == 0)
            {
                x_inicial_teste = s;
                cont = 1;
            }
            max_teste = vector_x_teste[s];
            //contador_2++;
            if (maximo < max_teste)
                maximo = max_teste;
        }
    }
    else if (vector_x_teste[s] < height_img1_2 * 0.1)
    {
        if (threshold < maximo)
        {
            contador++;
        }
        else
        {
            cont = 0;
        }
        if (contador == 1)
        {
            x_inicial = x_inicial_teste - 4;
        }
        if (contador == 6)
        {
            x_final = s + 4;
            break;
        }
        maximo = 0;
    }
}
```

Figura 11-Código do corte vertical do lado direito

Este pedaço de código percorre o vetor de intensidades de pretos e funciona como a função `loc_char` do capítulo 5, em que encontra blocos de intensidade de pretos e cada bloco corresponde a cada carácter, quando deteta seis blocos significa que chegou ao fim da matrícula e esse índice final corresponde ao limite vertical direito da matrícula.

Existe um problema se só se fizer assim a deteção da matrícula que é na generalidade das matrículas, existe uma área com uma forte intensidade de pretos que corresponde ao sítio da letra que indica o país de onde a matrícula é proveniente. Para resolver esse problema, o contador usado só aceita blocos de preto quando encontrar um valor muito baixo e próximo de 0 de intensidade de pretos.

Na figura abaixo pode-se observar o resultado deste código.



*Figura 12-Matrícula Final*

Por fim obteve-se os limites horizontais e verticais da matrícula numa imagem.

## 7- Aplicação dos três métodos

A função principal que junta estes algoritmos desenvolvidos é a função LP\_Recognition que tem como parâmetros de entrada:

- Imagem original e a sua cópia
- O nível de dificuldade das imagens
- Uma string com o tipo de matrícula

Tem também parâmetros de saída que são eles:

- Retângulo com o tamanho da matrícula e sua localização na imagem
- 6 retângulos com o tamanho de cada caracter e sua localização
- 6 strings com o valor do caracter correspondente

No início desta função foram criadas para além da imagem principal e da sua cópia outras duas cópias denominadas img2 e img3.

O primeiro passo é identificar o nível de dificuldade da imagem porque para imagens de dificuldade maior que 1 é necessário a função Loc\_plate. Esta função retorna um retângulo com a localização da matrícula.

Na figura 13 apresenta-se parte do código que invoca a função Loc\_plate

```
if (difficultyLevel > 1)
{
    vector_return = Loc_plate(img, img);
    LP_Location = new Rectangle(vector_return[0], vector_return[1], vector_return[2], vector_return[3]);
    imgCopy = imgCopy.Copy(LP_Location);
    img3 = img3.Copy(LP_Location);
}
else
    LP_Location = new Rectangle(0, 0, m.width, m.height);
```

*Figura 13-Obtenção da localização da matrícula*

De seguida é chamada a função `loc_char` onde é enviado a imagem cortada pela localização da matrícula, o nível de dificuldade da matrícula e os vetores de localização da matrícula. Esta função retorna seis retângulos com a localização dos seis caracteres na imagem

Por fim os ficheiros da diretoria onde se encontram as imagens para comparação são adicionadas num vetor de imagens e a função `rec_char` é invocada.

```
LP_C1 = rec_char(img2.Copy(LP_Chr1).Resize(10, 20, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR), char_BD, char_names);  
LP_C2 = rec_char(img2.Copy(LP_Chr2).Resize(10, 20, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR), char_BD, char_names);  
LP_C3 = rec_char(img2.Copy(LP_Chr3).Resize(10, 20, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR), char_BD, char_names);  
LP_C4 = rec_char(img2.Copy(LP_Chr4).Resize(10, 20, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR), char_BD, char_names);  
LP_C5 = rec_char(img2.Copy(LP_Chr5).Resize(10, 20, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR), char_BD, char_names);  
LP_C6 = rec_char(img2.Copy(LP_Chr6).Resize(10, 20, Emgu.CV.CvEnum.INTER.CV_INTER_LINEAR), char_BD, char_names);
```

*Figura 14-Invocação da função `rec_char`*

Esta função `rec_char` vai retornar uma string com o valor do caracter.

Umas considerações do grupo é que ao chamar a função `Loc_plate` e `loc_char` os índices dos retângulos que retornariam estariam desajustados face há imagem original, para isso esses índices vão sendo guardados ao longo do código na variável `vetor_return`.

## 8- Resultados obtidos e considerações

Para testar os resultados da nossa aplicação foi utilizado o EvalForm onde são selecionadas várias imagens e são todas analisadas.

Na figura 15 e 16 são apresentados os resultados obtidos no EvalForm.

Run	Filename	is_LP	Level	Type	Loc_Check	C1_Check	C2_Check	C3_Check	C4_Check	C5_Check	C6_Check	CTotal_Check
<input checked="" type="checkbox"/>	A(1)(1).jpg	<input checked="" type="checkbox"/>	1	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(1)(2).jpg	<input checked="" type="checkbox"/>	1	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(1)(3).jpg	<input checked="" type="checkbox"/>	1	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(1)(4).jpg	<input checked="" type="checkbox"/>	1	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(2)(1).jpg	<input checked="" type="checkbox"/>	2	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.8333333
<input checked="" type="checkbox"/>	A(2)(2).jpg	<input checked="" type="checkbox"/>	2	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.5
<input checked="" type="checkbox"/>	A(2)(3).jpg	<input checked="" type="checkbox"/>	2	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(3)(1).jpg	<input checked="" type="checkbox"/>	3	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.8333333
<input checked="" type="checkbox"/>	A(3)(2).jpg	<input checked="" type="checkbox"/>	3	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(3)(3).jpg	<input checked="" type="checkbox"/>	3	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(3)(4).jpg	<input checked="" type="checkbox"/>	3	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	A(4)(1).jpg	<input checked="" type="checkbox"/>	4	A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input type="checkbox"/>	A(4)(2).jpg	<input checked="" type="checkbox"/>	4	A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Figura 15-Resultados das matrículas tipo A

<input checked="" type="checkbox"/>	B(1)(1).jpg	<input checked="" type="checkbox"/>	1	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input type="checkbox"/>	B(1)(2).jpg	<input checked="" type="checkbox"/>	1	B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
<input checked="" type="checkbox"/>	B(1)(3).jpg	<input checked="" type="checkbox"/>	1	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	B(2)(1).jpg	<input checked="" type="checkbox"/>	2	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	B(2)(2).jpg	<input checked="" type="checkbox"/>	2	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.8333333
<input checked="" type="checkbox"/>	B(2)(3).jpg	<input checked="" type="checkbox"/>	2	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	B(3)(1).jpg	<input checked="" type="checkbox"/>	3	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	B(3)(2).jpg	<input checked="" type="checkbox"/>	3	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	B(3)(3).jpg	<input checked="" type="checkbox"/>	3	B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input type="checkbox"/>	B(4)(1).jpg	<input checked="" type="checkbox"/>	4	B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="checkbox"/>	B(4)(2).jpg	<input checked="" type="checkbox"/>	4	B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

Figura 16-Resultados das matrículas tipo B

Como se pode observar a grande maioria das imagens de nível 1, 2 e 3 funcionam corretamente, as poucas exceções são como a A(2)(1), A(3)(1) e B(2)(2) existe um caracter que não faz bem o reconhecimento pois as suas formas são muito parecidas, como o número 6 e a letra G ou a letra F e a letra E.

A imagem B(1)(2) encontra o problema de matrícula ter um fundo muito azul e a sua binarização não ser a ideal.



## **9- Conclusões**

As grandes dificuldades encontradas pelo grupo ao desenvolver a aplicação foram conseguir distinguir, usando projeções na imagem, a matrícula de picos de ruído que existem, o que por vezes levou a que a localização na matrícula não seja a ideal para o algoritmo de reconhecimento de caracteres.

No entanto a grande maioria das imagens testadas funcionaram com sucesso em que as exceções têm na origem o mau reconhecimento do carácter pela parecença de letras e números existentes.

O grupo realizou todos os requisitos pedidos no capítulo 2, um requisito extra era analisar com sucesso imagens de dificuldade 4, infelizmente das imagens de nível 4 disponibilizadas só foi possível ao grupo analisar com sucesso uma delas, sendo que o problema das outras imagens é a elevada luminosidade e reflexo na imagem.

Para um melhor resultado deste trabalho o grupo sugere experimentar com outras técnicas de diferenciação e binarização da imagem para melhorar a localização da matrícula e assim um maior sucesso no reconhecimento de caracteres.

## **10- Código fonte utilizado**