

Visão por Computador

(Reconhecimento de Matrículas em Video - Processamento de Imagens)

TP 2 - Grupo 12

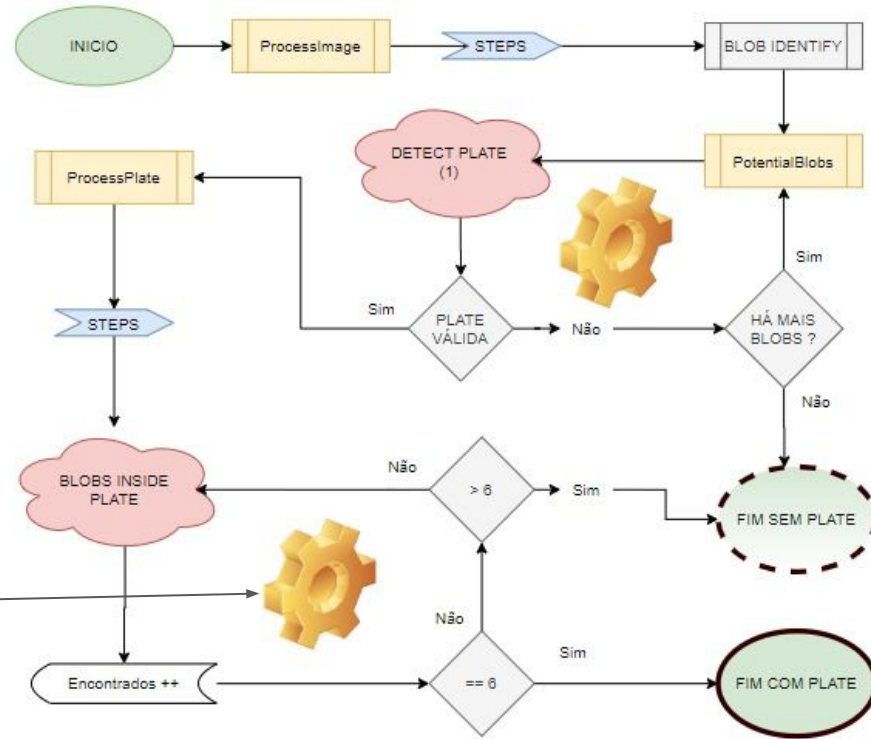
Introdução

Utilizamos praticamente a mesma estrutura do trabalho prático 1, alterando substancialmente a manipulação da imagem de forma a detectar mais frames do vídeo. Contudo e dado a fiabilidade de detecção de uma matrícula ser muito elevada no nosso algoritmo tornou-se muito complicado dadas as características das imagens, reconhecer todas as frames devido a reflexos e não ser termos encontrado uma forma mais dinâmica de manipulação dependendo das características da imagem.

Utilizamos de forma a dinamizar o método de OTSU o qual é usado para calcular thresholds a fim de converter imagens para binário, contudo usamos esse threshold na segmentação HSV com resultados satisfatórios.

O maior desafio foi o reconhecimento dos caracteres, o qual consideramos com 80% de fiabilidade no entanto poderia ser muito mais melhorado, o mesmo assenta nas características de um blog, as quais neste trabalho foram 5 que explicaremos mais à frente, contudo seria muito simples aumentar o número de características no futuro.

Diagrama



NEW

PREDICT()

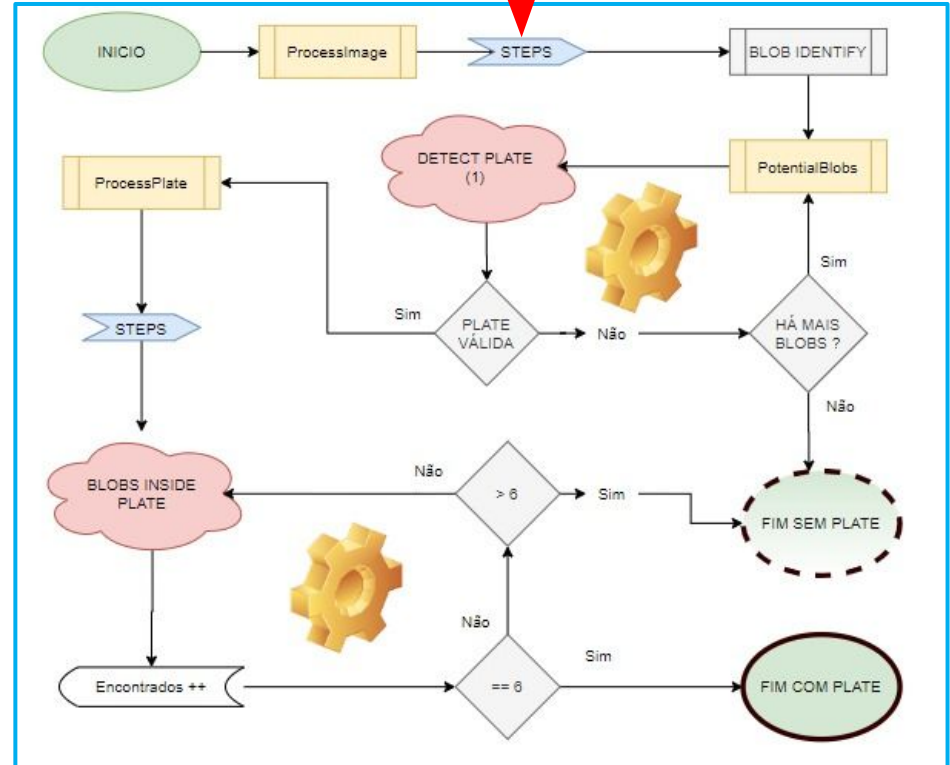
Manipulação da Imagem Geral

- Remoção de Cor TH(12)
- RGB para tons Cinza
- Conversão HSV
- Método OTSU
- Segmentação HSV

(0, 255, 0, 255, **OTSU**, 255)

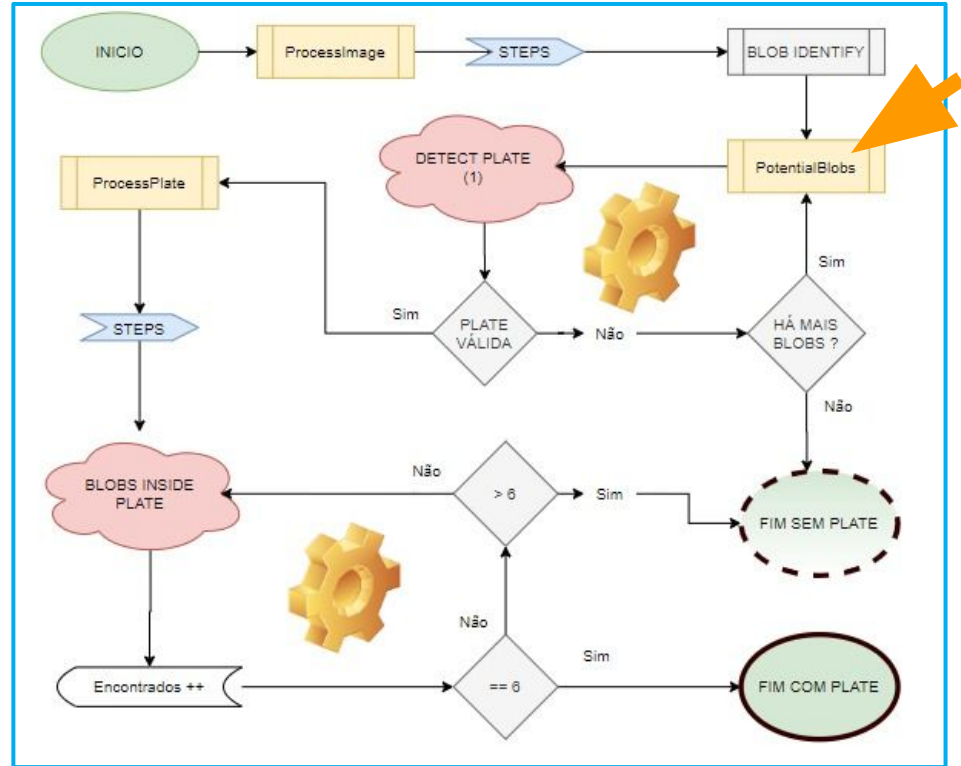


- Identificação de BLOBS



Procura de Potenciais Blobs - PotentialBlobs()

Será dado agora início ao processo iterativo de percorrer todos os blobs encontrados anteriormente de forma a identificar uma matrícula.

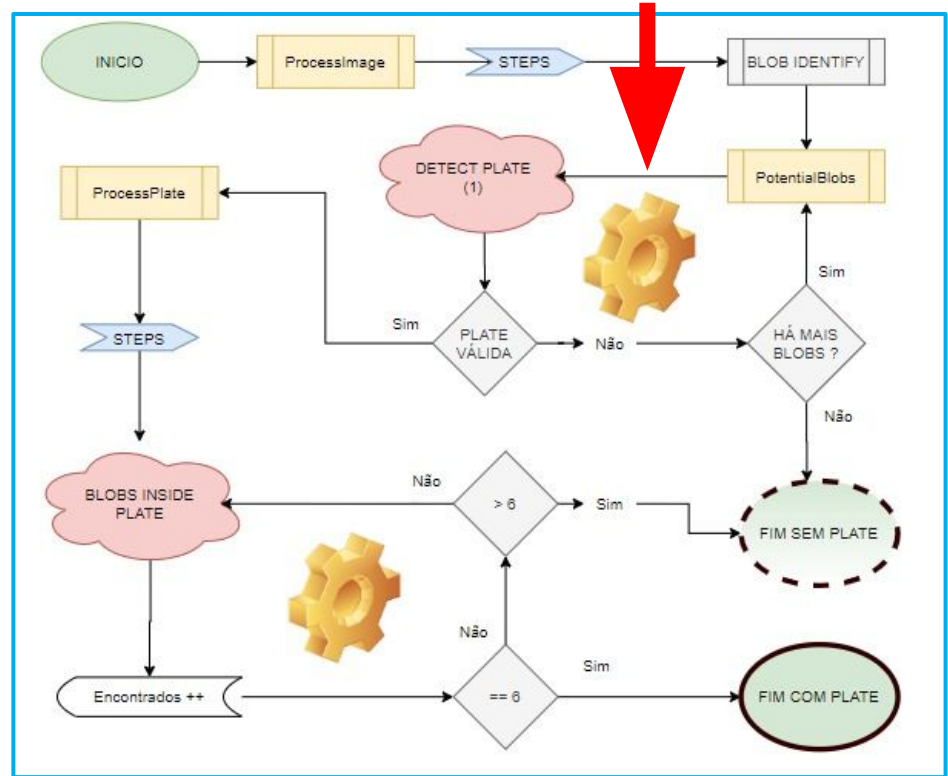


Validações de Blobs (Matrícula)

As condições abaixo cumulativamente verdadeiras são a indicação que estamos perante o blob de uma matrícula.

- Rácio > 4
- Rácio < 5
- 6 Blobs de caracteres dentro

Rácio = Largura/Altura



Manipulação da matrícula / Identificação

É extraído o blob da potencial matrícula para ser trabalhado mediante a manipulação que será apresentada de seguida.

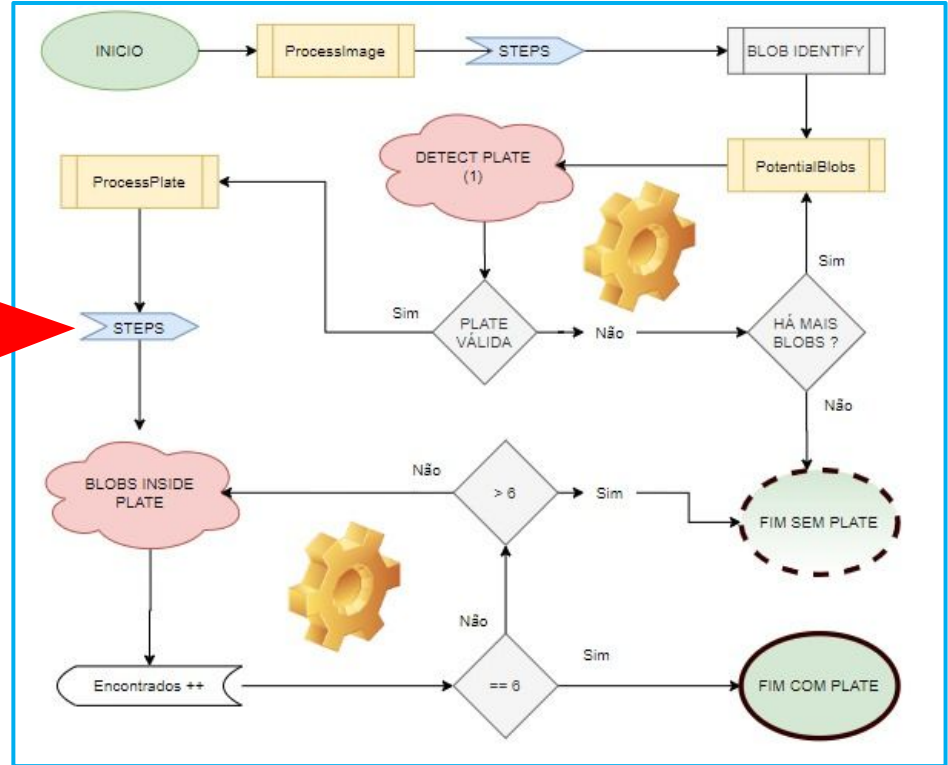


Manipulação da Imagem (Matrícula)

- Conversão HSV
- Segmentação HSV
- Close K=2



- Identificação de BLOBS



Manipulação Matricula

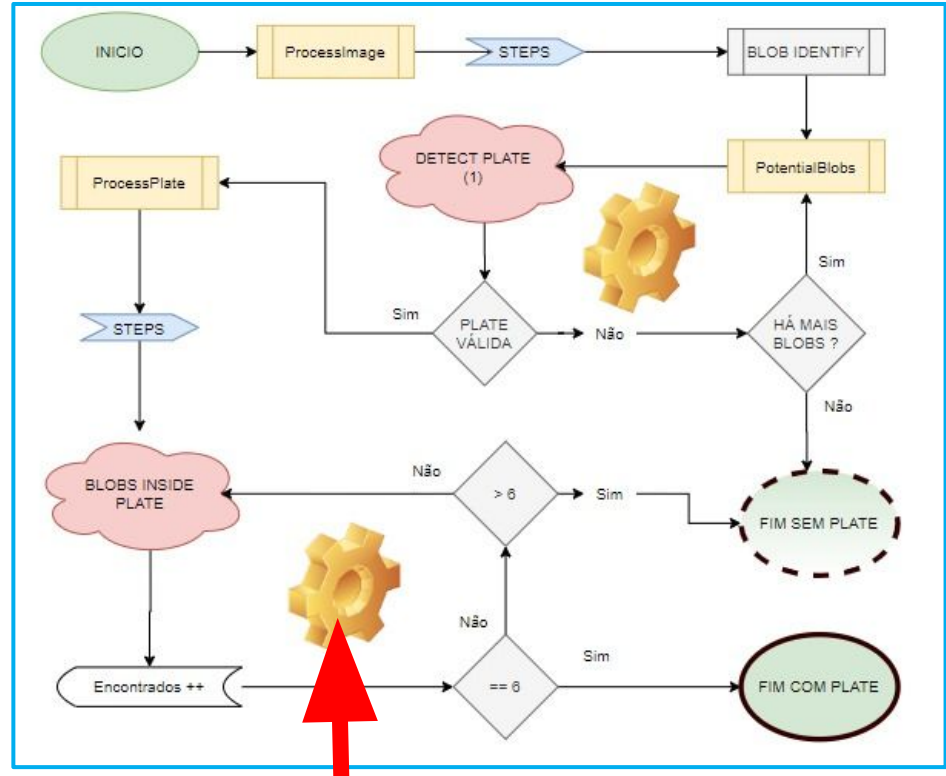


Validações de Blobs (Caracteres da Matrícula)

As condições abaixo cumulativamente verdadeiras são a indicação que estamos perante um blob de um caracter da matrícula.

- $Racio < 1$
- $Altura > 60\%$ da Matrícula

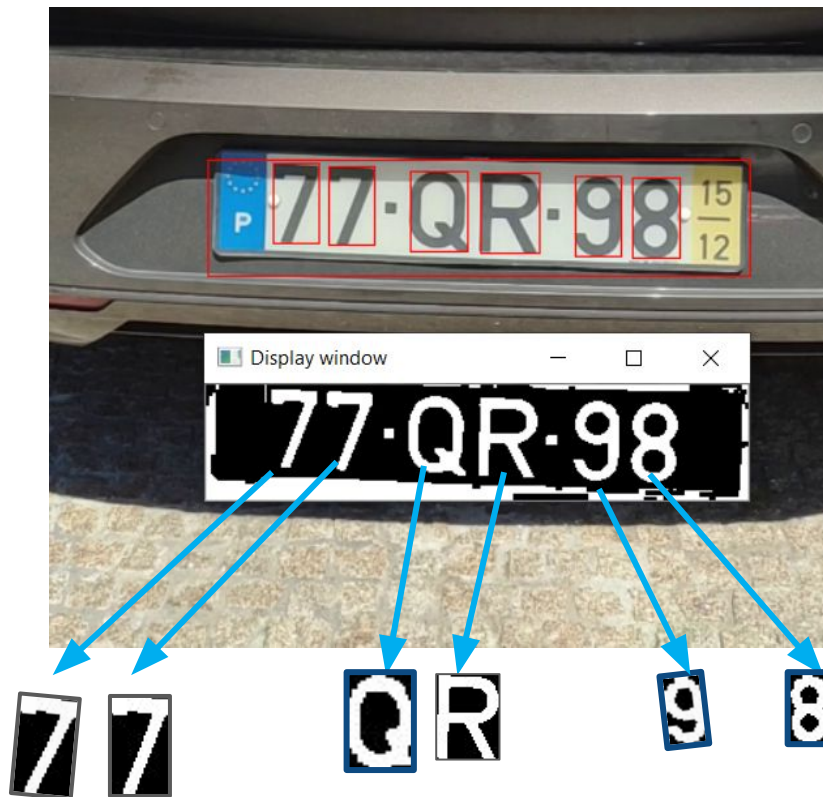
Rácio = Largura/Altura



Identificação / Extração de BLOBS

São identificados os blobs de caracteres e extraídos os mesmos os quais serão reconhecidos usando um método explicado a seguir

Caso sejam identificados blobs de caracteres é retornada a indicação de quais são os blobs e o número de blobs. Caso sejam encontrados mais que 6, a operação devolve que não é válida.



Método de Reconhecimento

`char predict(IVC* image_test)`

Decomposição de imagem em 4 Quadrantes a fim de dar uma matriz ao nosso algoritmo de todas as imagens.

Exemplo do carácter 7

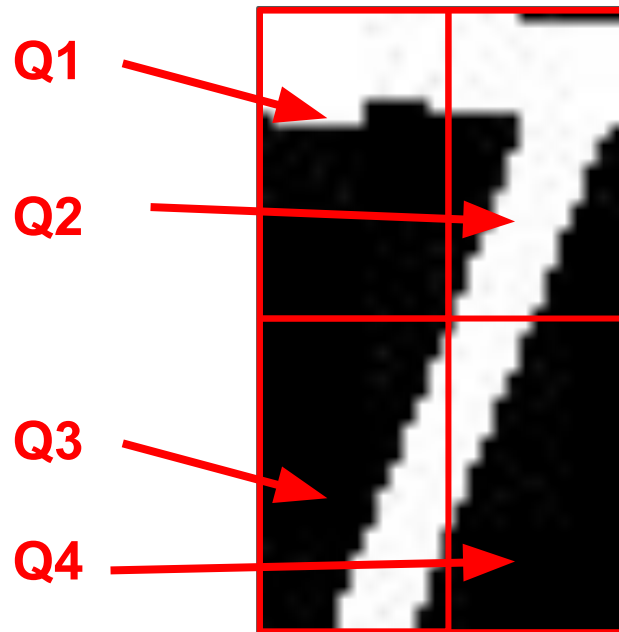
Q1 - 32%

Q2 - 67%

Q3 - 67%

Q4 - 13%

Racio = $0,59 * 100 = 59$



Método de Reconhecimento

Foi então criado uma matriz com as características de cada letra possível.

Q1, Q2, Q3, Q4, Racio

```
int predictions[8][5] =  
{  
  // Q1 Q2 Q3 Q4 RACIO  
  {33, 67, 67, 38, 59},  
  {66, 45, 54, 53, 63},  
  {32, 63, 30, 13, 58},  
  {55, 59, 56, 56, 57},  
  {54, 68, 31, 56, 58},  
  {42, 43, 40, 62, 71},  
  {55, 53, 42, 41, 73},  
  {40, 41, 48, 51, 71},  
};  
char chars[8] = { '2', '6', '7', '8', '9', 'Q', 'R', 'U' };
```

Encontrou a Matrícula

É criada uma nova matriz com as diferenças encontradas na imagem.

No exemplo abaixo podemos ver:

O carácter 8 têm uma soma das diferenças absolutas de 11 unidades e um desvio entre as diferenças de 1.

Logo quando:

O desvio é o de menor valor e a soma absoluta também estamos perante o melhor candidato.

CHAR	Q1	Q2	Q3	Q4	Rc
2	25	9	11	16	1
6	8	13	2	1	5
7	26	5	26	41	0
8	3	1	0	2	1
9	4	10	25	2	0
Q	16	15	16	8	13
R	3	5	14	13	15
U	18	17	8	3	13
55	Desvio 5.000000				
101	Desvio 19.000000				
11	Desvio 1.000000				
57	Desvio 10.000000				
71	Desvio 8.000000				
68	Desvio 7.000000				
59	Desvio 9.000000				



Conclusões sobre este reconhecimento

Apesar de termos tido uma fiabilidade a rondar os 80% no reconhecimento há pontos de melhoria tais como:

- Rotação das imagens de forma a estarem sempre na mesma orientação
- Divisão da imagem em mais quadrantes
- Colocação nas características valores de histograma em linha/coluna/diagonal de forma a perceber melhor a distribuição da cor e de maior número de caracteres.

Conclusões sobre o reconhecimento em Vídeo

É extremamente importante que a manipulação da imagem seja dinâmica, pois em cada frame os valores de manipulação devem ser diferentes devido ao brilho/sombras etc.

A manipulação usando a segmentação HSV torna-se bastante eficiente, no entanto não conseguimos ser tão dinâmicos quanto o esperado tendo a necessidade de haver um valor diferente na segmentação entre ambos os vídeos.

Video1: vc_hsv_segmentation(image6, 0, 255, 0, 20, 0, 50);

Video2: vc_hsv_segmentation(image6, 0, 255, 0, 20, 0, 70);

Conclusões sobre o reconhecimento em Vídeo

A velocidade do vídeo 2 é bastante inferior à do 1 porque o mesmo contém uma resolução muito superior, fazendo com que os ciclos iterativos pelos pixels demorem mais tempo.

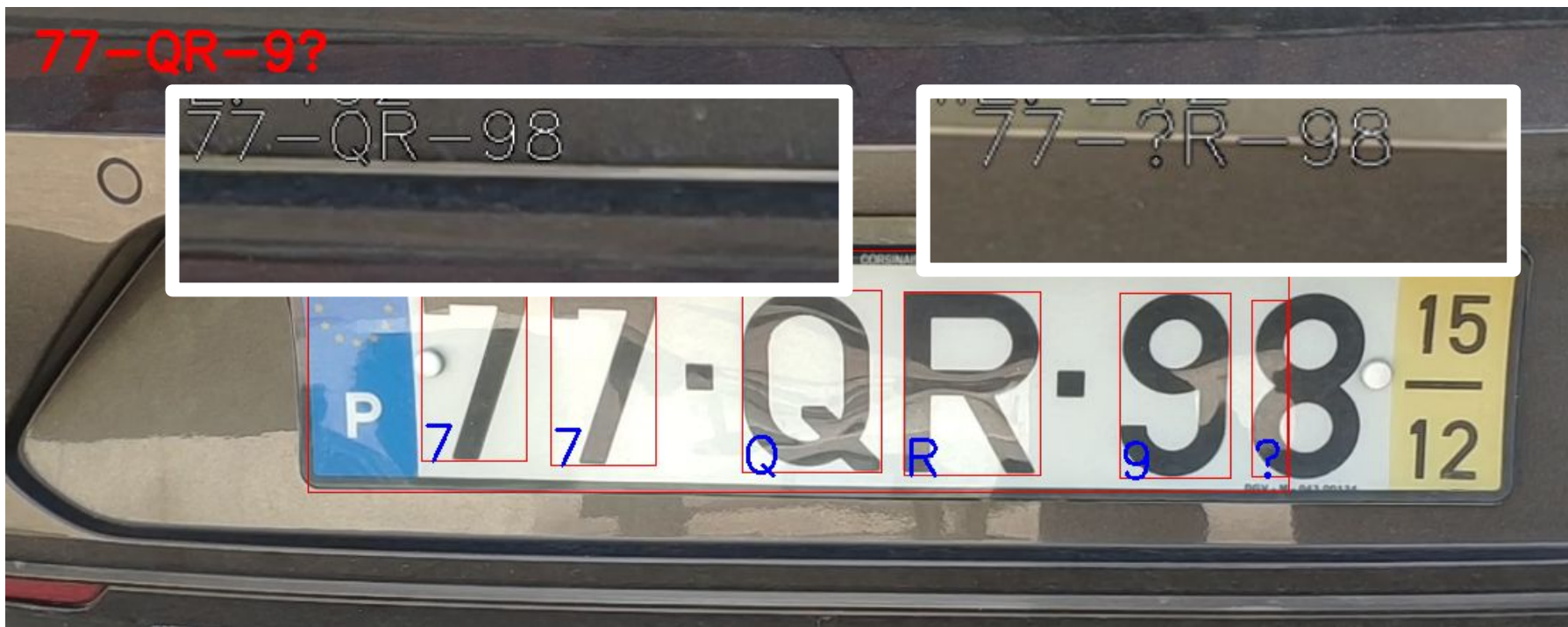
```
Tempo decorrido: 45.1484segundos  
Pressione qualquer tecla para continuar...
```

17 FPS
VIDEO 1

```
Tempo decorrido: 134.061segundos  
Pressione qualquer tecla para continuar...
```

6.7 FPS
VIDEO 2

Conclusões sobre o reconhecimento em Vídeo



Conclusões sobre o reconhecimento em Vídeo



Visão por Computador

FIM

Trabalho realizado por:

a13871@alunos.ipca.pt - Rafael Rocha

a14383@alunos.ipca.pt - Oscar Silva

a17442@alunos.ipca.pt - Daniel Torres