

Deteção de matrículas utilizando a IPT do MATLAB

(Car plate detection using MATLAB IPT)

Diogo Silva, up201809213@fe.up.pt. Francisco Terra, up201604468@fe.up.pt.
Miguel Ramos, up201808904@fe.up.pt. Paulo Silva up201909556@fe.up.pt

Resumo — Este documento apresenta o resultado do trabalho desenvolvido para a detecção de matrículas de carros tendo como ferramenta o MATLAB, mais concretamente a *Image Processing Tool* (IPT) que este apresenta. Este trabalho está dividido em duas tarefas principais, a primeira cai sobre a detecção da localização da matrícula numa imagem e a segunda trata de identificar quais as letras e números presentes em cada matrícula. Existe ainda uma terceira tarefa que consiste na junção das duas anteriores, mas aplicada a imagens com elevado grau de dificuldade.

Palavras-chave — *Processamento de imagem; detecção de objetos; template matching; identificação de letras e números.*

I. INTRODUÇÃO

A utilização de *software* para processamento de imagem é uma prática cada vez mais recorrente na atualidade. A ferramenta de processamento de imagem do MATLAB (IPT) é um dos muitos exemplos de ferramentas existentes e que permitem realizar uma vasta gama de operações.

A essência do trabalho que se pretende realizar consiste em, dada uma imagem de um carro, onde se encontre presente uma matrícula, ao submeter esta imagem a diferentes tipos de processamentos e técnicas se obtenha no final um conjunto de letras e números correspondentes à devida matrícula. Para realizar isto pretende-se recorrer ao uso de técnicas de pré processamento, morfologia matemática, segmentação de imagem, *template matching* e análise de imagem.

Por fim relativamente à estrutura do trabalho em si este estará dividido por três tarefas (*tasks*), onde as duas primeiras são as principais para o trabalho. Na primeira tarefa pretende-se, através de algumas dos tipos de técnicas anteriormente referidas, localizar na imagem a matrícula e seleccionar apenas essa parte. Para a segunda tarefa o objetivo é através das imagens das matrículas, fazendo uso de outro tipo de técnicas conseguir identificar quais as letras e números nelas presentes e guardar essa informação. Por fim numa terceira tarefa, caso o tempo o permita o objetivo é melhorar a aplicabilidade dos algoritmos desenvolvidos nas duas primeiras tarefas para imagens que apresentam um alto nível de dificuldade (fraca luminosidade, ângulos complexos, desfocagem, etc...).

II. PRIMEIRA TAREFA

A primeira tarefa que se realizou neste trabalho consistiu em encontrar um conjunto de passos envolvendo as técnicas pretendidas que quando aplicados às 40 imagens de teste (imagens de frente e trás de carros com as devidas matrículas), se conseguisse identificar e recortar as mesmas.

A. Processamento

O processamento utilizado nesta parte consiste, inicialmente, em ler a imagem e de seguida convertê-la para uma escala passível de ser trabalhada, neste caso *greyscale*, seguido do seu redimensionamento para a mesma gama de valores a ser aplicada em todas as imagens. Depois é aplicado um filtro gaussiano para suavização da imagem e para reduzir o ruído presente na mesma.

Após esta etapa é realizada uma operação que permite a eliminação da metade superior da imagem, isto porque se constatou que todas as matrículas se encontravam na secção inferior da imagem (Figura 1) e deste modo reduz-se a probabilidade de ocorrerem falhas nas outras partes do processo.



Figura 1 - Imagem com a zona de interesse reduzida à parte inferior da mesma, após ser convertida para grayscale

O passo seguinte foi a aplicação de um *top-hat* (Figura 2) de modo a se realçar os brancos da imagem, neste caso com o intuito de se realçar a parte branca das matrículas, seguido depois da aplicação da função *edge* com o método de *canny* para se poder obter as bordas da imagem com o parâmetro *threshold* de 0.2 (Figura 3), visto que a imagem está dimensionada para valores de intensidade entre 0 e 1



Figura 2 - Resultado da aplicação do top-hat

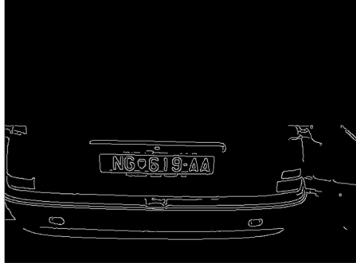


Figura 3 - Orlas obtidas através do canny

Posto isto, efetuam-se mais alguns recortes nas imagens até se obter uma área de imagem mais pequena, mas mantendo sempre a informação completa da matrícula em causa.

O processo seguinte consiste numa dilatação horizontal seguida de uma vertical, de modo a reforçar as linhas previamente detectadas e também de modo a se poder unir certos pontos que não fossem contínuos. Após este passo é aplicada a função *imfill* (Figura 4) com o objetivo de que a parte da matrícula fica completamente preenchida a branco.



Figura 4 - Imagem com os buracos preenchidos

A imagem resultante deste processo ainda não define unicamente a área da matrícula, por isso é realizado um *open* na imagem recorrendo a um elemento estruturante vertical que elimina todos os elementos que tenham uma altura inferior à da matrícula. É também utilizado um elemento estruturante horizontal, no entanto este não corresponde ao comprimento da matrícula porque, caso contrário, este eliminaria as matrículas que se encontram ligeiramente na diagonal. No final obtém-se a área correspondente à matrícula (Figura 5).



Figura 5 - Resultado final

B. Resultados e Considerações

Dado todo o processamento anterior obtiveram-se resultados relativamente razoáveis. Para a avaliação destes foi utilizada tanto a função jaccard em comparação com o *ground-truth* fornecido, como também a observação crítica e objetiva dos resultados. Tendo em conta o jaccard obteve-se como valor médio global 0.8508, sendo o pior resultado de 0.5997 que corresponde à imagem 39 (Figura 6) de teste e o melhor de 0.9827 (Figura 7) que corresponde à imagem 3.



Figura 6 - Matricula 39



Figura 7 - Matricula 3

O algoritmo apresenta piores resultados quando existe uma região fechada maior que a matrícula e que inclua a mesma. Visto que depois de usar a função *imfill* torna-se difícil separar as regiões que não nos interessam da matrícula.

III. SEGUNDA TAREFA

A segunda tarefa que se realizou consistiu em dadas apenas as imagens contendo a matrícula, resultantes da primeira tarefa, ser capaz de identificar as letras e números e obter a sua combinação no fim do processamento.

A. Processamento

Nesta tarefa o processamento começa por redimensionar a imagem dada para um tamanho padrão de modo a generalizar o processo para todas as imagens, obtendo assim uma imagem inicial como a da Figura 8.



Figura 8 - Matricula redimensionada

Visto não ser relevante ter a imagem em RGB (apenas é necessário identificar as letras) a imagem é convertida para *greyscale*. Depois disso aplica-se um filtro gaussiano para eliminar algum ruído presente na imagem, aplica-se uma equalização de histograma para realçar o contraste e normaliza-se a imagem para intensidades entre $[0,1]$, obtendo-se assim uma imagem como a da Figura 9.



Figura 9 - Matricula em greyscale com filtro e equalização

De seguida para uma melhor identificação das letras nos seguintes processos a imagem é convertida para binário. Inicialmente testou-se utilizando o método de Otsu, porém visto, só interessar as letras e sabendo que estas são pretas, ou seja o valor da sua intensidade é próximo de zero, utilizou-se então um *threshold* fixo de 0.2, visto que a imagem está dimensionada para valores de intensidade entre 0 e 1. Após isso faz-se um *close* para se eliminar espaços que não interessam e que podem interferir com o algoritmo, resultado na Figura 10.

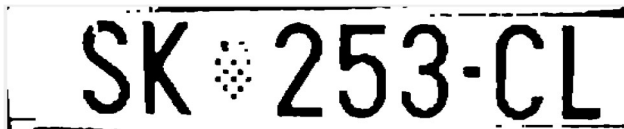


Figura 10 - Matricula Binarizada

Após isto faz-se uma negação da imagem para que as letras fiquem brancas, ficando assim de acordo com o template. Em alguns casos ao fazer a binarização da imagem, algumas letras não ficam ligadas, como por exemplo o 'S'. Para corrigir isso é aplicada uma dilatação com o elemento estruturante *line* com orientação horizontal e depois vertical, assim unindo a letra. Depois disso elimina-se as secções que se encontram nas extremidades da imagem, como também todas as secções com área abaixo de 800 pixels. Ficando assim com um resultado como o da Figura 11.



Figura 11 - Primeira iteração de processamento da imagem final

Para algumas imagens isto seria o bastante, porém, como é o caso da Figura 11, algumas letras não ficam completamente unidas, mesmo depois da primeira dilatação. Para resolver estes casos são realizadas mais algumas dilatações seguidas de

erosões e eliminações por área, resolvendo assim quase na totalidade os casos deste tipo.

Posto isto devido a alguns destes processos existiam situações em que partes significativas da imagem eram eliminados e por isso decidiu-se então utilizar uma quantificação usando 4 valores de *threshold* (5 classes), escolhendo depois os valores de intensidade mais baixos que correspondem às letras. De seguida é feita uma soma das duas imagens (Figura 12) finalizando assim a parte de processamento da imagem antes da detecção das letras e números em si.



Figura 12 - Última iteração de processamento da imagem final

Para a parte seguinte da segunda tarefa, usando a função *bwlabel* obtém-se a matriz que contém objetos ligados em 8-*neighbourhood*. Na matriz retornada utiliza-se a função *regionprops* com propriedade *boundingbox*, conseguindo-se obter blocos que delimitam as regiões de interesse à semelhança da Figura 13.



Figura 13 - Identificação dos objetos

Recortando os blocos previamente selecionados e descartando os que não tenham as dimensões dentro de um limite definido, consegue-se obter as letras para usar nas funções de identificação. Temos como exemplo a Figura 14, onde se pode ver o número 3 recortado e redimensionado para 24x42.



Figura 14 - Exemplo de objeto a identificar

Comparando esta imagem com o template de todas as letras existentes (usando a função *corr2*), obtém-se um índice de correlação para cada uma. Usando também o método de *euler* para detecção de buracos, pega-se na letra para o qual a semelhança é maior e verifica-se se o número de buracos é igual. Se o número de buracos da letra com maior semelhança e da letra a identificar for igual então a letra a identificar é tida como a letra com maior semelhança. Caso o número de buracos não seja igual testa-se com a letra com a segunda maior semelhança, e assim sucessivamente, até se obter a matrícula.

B. Resultados e Considerações

Com a utilização deste algoritmo conseguiu-se identificar corretamente 37 das 40 matrículas propostas para o trabalho. Duas das matrículas não identificadas deveu-se à não distinção da letra “O” pelo número “0”.

Estas matrículas seguem um padrão de ‘LL NNN - LL’, logo se o hífen fosse identificado podíamos fazer uma fácil distinção entre os dois. Porém como no processamento elimina-se o hífen, isto torna a identificação por esta maneira impossível.

Outro problema foi o não preenchimento da letra mesmo após todo aquele processamento o que leva a casos como o que se segue como o da Figura 15.



Figura 15 - Letra não unida

Como se pode verificar na figura o 5 não está unido, tornando a sua identificação impossível, este problema poderia ser resolvido detetando que o centroide dos dois blocos eram relativamente próximos no eixo das abcissas, logo pertenciam à mesma letra e assim unia-se os dois.

Em suma para esta parte do trabalho considera-se que com uma melhoria no processamento da imagem os resultados seriam bastante melhores. Dever-se-ia ter usado uma abordagem diferente, não apenas usando operadores morfológicos, o que não permite a grande generalização do algoritmo. Para preenchimento e melhoramento das letras foram usadas algumas dilatações e erosões, mais uma vez generalizando em muito o trabalho para estes casos.

IV. TERCEIRA TAREFA

Para a terceira e última tarefa o objetivo foi testar a robustez dos algoritmos desenvolvidos para as duas primeiras tarefas em imagens que apresentam má qualidade.

A. Resultados e Considerações

Aplicando, em conjunto, o algoritmo da primeira tarefa e o algoritmo da segunda, esperava-se identificar corretamente as matrículas desta tarefa, porém os resultados obtidos foram não satisfatórios e apenas foi possível identificar com sucesso uma matrícula na primeira tarefa e nenhuma na segunda tarefa.

Isto deve-se à má qualidade das imagens a analisar, o que significa que para serem obtidos melhores resultados seria necessário, eventualmente, uma reestruturação do código base, de forma a este ser ainda mais robusto e conseguir lidar com situações menos favoráveis.

V. CONCLUSÕES

Com a realização deste projeto, o grupo foi capaz de aprofundar os conhecimentos adquiridos durante as aulas teóricas. O relacionamento entre os conhecimentos teóricos e as

ferramentas da “Image Processing Toolbox” do matlab, permitiram a aprendizagem de uma nova ferramenta e, também, um entendimento mais profundo dos conceitos teóricos, nomeadamente, como é que estes se relacionam entre si e que implicações têm quando combinados.

De uma forma geral, o balanço deste trabalho é positivo uma vez que das três das tarefas realizadas, a terceira foi a única que originou resultados menos satisfatórios. Na primeira tarefa obteve-se um valor médio de 0.8508 tendo em conta o índice de jaccard e na segunda tarefa foi possível identificar 37 das 40 matrículas. Na terceira tarefa, conforme referido, apenas foi possível identificar com sucesso uma matrícula no total das duas tarefas.

Embora os resultados tenham sido positivos, consideramos que o trabalho realizado possui uma margem de evolução, margem essa que poderia ser alcançada através da combinação entre uma reestruturação do código base para melhorar o processamento/tratamento de imagem e uma definição mais objetiva dos processos de distinção de letras/números semelhantes, de forma a reduzir mais ainda a ambiguidade entre ambos.

Em suma, consideramos que a realização deste projeto contribuiu de forma significativa para uma melhor aprendizagem dos conteúdos lecionados, tal como, para uma noção bastante prática das funcionalidades do matlab no tratamento de imagens e quais os fatores mais determinantes no tratamento das mesmas.

REFERÊNCIAS

- [1] Mendonça, A.M, Slides da Disciplina de SBVI.
- [2] The Mathworks, Inc, Matlab.