

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema de Identificação de Veículos Desaparecidos Baseado em Dispositivos Móveis

José Miguel de Oliveira Pinho Barbosa

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Dr. Miguel Velhote Correira

Co-orientador: Prof. Dr. Jorge Barbosa

15 de Abril de 2015

A Dissertação intitulada

“Sistema de Identificação de Veículos Desaparecidos Baseado em Dispositivos Móveis”

foi aprovada em provas realizadas em 27-03-2015

o júri



Presidente Professora Doutora Ana Cristina Costa Aguiar
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Rui Silva Moreira
Professor Associado da Faculdade de Ciências e Tecnologia da Universidade
Fernando Pessoa



Professor Doutor Miguel Fernando Paiva Velhote Correia
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - José Miguel de Oliveira Pinho Barbosa

Resumo

O furto e desaparecimento de veículos, apesar de ser um problema bem conhecido da sociedade, tem vindo a apresentar um constante incremento nos últimos anos. Estes veículos são, na sua maioria, utilizados em atividades ilícitas, circulando abertamente na via pública sem que seja possível reconhecê-los como furtados. E acabando, por norma, por serem abandonados na via pública, semanas ou meses sem que ninguém se aperceba.

Por outro lado, o exponencial crescimento do mercado de *Smartphones*, faz destes um dos objetos mais comuns na sociedade, que aliado a um apreciável poder de processamento e disponibilidade de um leque avançado de sensores, torna estes equipamentos possíveis ferramentas muito para lá das “simples” chamadas.

Com esta dissertação, é pretendido realizar a ponte entre estas duas realidades, investigando metodologias e possíveis implementações de um sistema de reconhecimento de matrículas para dispositivos móveis.

Abstract

The auto theft and the overall disappearance of vehicles, despite being a well-known problem in society, has been showing a steady increase in recent years. These vehicles are mostly used in doubtful activities, circulating freely in the public road without any fear of being recognized. And eventually, to be abandoned on the public road, weeks or months without anyone noticing.

On the other hand, the exponential growth of the smartphones market, makes these one of the most common objects in society, which combined with a significant processing power and availability of an advanced range of sensors, makes these a amazing tool, far beyond a "simple" call machine.

With this dissertation, is intended to make the bridge between these two realities, investigating methodologies and possible implementations of an licence plate recognition system for mobile devices.

Agradecimentos

Aos Professores Orientadores Miguel Velhote e Jorge Barbosa por terem aceite esta ideia e por toda a orientação no projecto e ajuda na focalização do que era essencial.

Ao Professor Doutor Armando Padilha pelo incentivo e indicações.

Aos meus pais e irmão por todo o apoio e incentivo que sempre me deram.

À minha namorada por toda a paciência, compreensão e força que sempre me deu.

A todos os meus amigos que direta ou indiretamente me ajudaram ao longo desta etapa.

E em memória da minha avó Lurdes, da minha tia Elisa e do meu avô Barbosa

José Barbosa

*“Flying is learning how to throw
yourself at the ground and miss.”*

Douglas Adams

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	2
1.3	Objetivo	2
1.4	Estrutura da Dissertação	2
2	Conceitos Fundamentais	3
2.1	Visão Computacional	3
2.2	Sistema de Reconhecimento de Matrículas	3
2.3	Dispositivos Móveis	4
3	Revisão Bibliográfica	7
3.1	Sistemas de Reconhecimento de Matrículas	7
3.1.1	Localização e segmentação da matrícula na imagem	8
3.1.2	Segmentação dos Caracteres da Matricula	14
3.1.3	Reconhecimento dos caracteres (OCR – <i>Optical Character Recognition</i>)	15
4	Implementação e Discussão	17
4.1	Localização e Segmentação da Área da Matricula	18
4.1.1	Método das janelas concêntricas deslizantes (SCW)	18
4.1.2	Método inspirado em Análise de Orlas e Projeções	19
4.2	Segmentação dos Caracteres da Matricula	23
4.3	<i>Pattern/Template Matching</i>	24
4.4	Case Study	25
4.4.1	Método das Janelas Deslizantes	25
4.4.2	Método Inspirado em Análise de Orlas e Projeções	27
4.5	Conclusões	28
5	Dispositivo Móvel	29
5.1	Plataforma Móvel	29
5.2	Aplicação Móvel	30
5.2.1	Conceito	30
5.2.2	Requisitos da aplicação	31
5.2.3	Implementação	31
6	Conclusão	35
6.1	Conclusões	35
6.2	Trabalho Futuro	36

A Anexos	37
A.1 Código SCW	37
A.2 Código para threshold de Savola	39
A.3 Código para extração da matricula por projeções	40
A.4 Código para segmentação dos caracteres	42
A.5 Código para segmentação dos caracteres	44
Referências	47

Lista de Figuras

2.1	(Numero de utilizadores de dispositivos móveis) Adaptada da página web de <i>Pew Research Center</i>	4
3.1	Etapas de um LPR	8
3.2	(a) Ilustração das janelas concêntricas b) Ilustração do processo de deslizamento das janelas (adaptado de [2])	9
3.3	Flowchart que ilustra o algoritmo de localização da matricula (Adaptado de [2])	10
3.4	a)imagem original. b) máscara resultado de SCW. c)resultado da aplicação da mascara à imagem original. d)localização da matricula (Adaptado de [2])	11
3.5	Ilustração de equalização de histograma	12
3.6	Da esquerda para a direita: Operador de Sobel aplicado nos dois sentidos; Aplicado Horizontalmente; Aplicado Verticalmente (Adaptado de [4])	13
3.7	Ilustração de uma projeção vertical (Adaptado de [4])	13
3.8	Ilustração do corte da imagem por uma banda horizontal	14
3.9	Segmentação por projeções (Adaptado de [3])	14
3.10	Resultado do método CCA	15
3.11	Seguimento de contorno (Adaptado de [19])	15
3.12	Exemplo de caracteres para OCR	16
4.1	Exemplos de imagens de teste	17
4.2	Etapas da segmentação por SCW	18
4.3	Exemplo de uma possível imagem capturada pela aplicação final	20
4.4	Primeiro passo - corte	20
4.5	Após binarização	21
4.6	Da esquerda para a direita: Imagem após binarização; Imagem resultante da aplicação do filtro; Resultado da subtração	21
4.7	Imagem pre processada e respectiva projeção Horizontal	22
4.8	Faixa cortada e respectiva projeção vertical	22
4.9	Matricula recortada	22
4.10	Caracteres encontrados por CCA	23
4.11	Caracteres reconhecidos	24
4.12	Caso 1 - Resultado Positivo	25
4.13	Caso 2 - Resultado Negativo	26
4.14	Caso 3 - Resultado Positivo	27
4.15	Caso 4 - Resultado Negativo	27
5.1	Ilustração do conceito de utilização da aplicação	31
5.2	Ilustração do sistema implementado em <i>Matlab Simulink</i>	32
5.3	Aplicação móvel em funcionamento	33

Lista de Tabelas

5.1	Comparação entre plataformas	29
-----	--	----

Abreviaturas e Símbolos

GPS	Global Positioning System
ALPR	Automatic Licence Plate Recognition System
LPR	Licence Plate Recognition
ITS	Intelligent Transportation System
NFC	Near Field Communication
SCW	Sliding Concentric Window
CCA	Connected Component Analysis
ROI	Region of Interest
OCR	Optical Character Recognition
px	Pixéis
OS	Operating System
ASCII	American Standard Code for Information Interchange

Capítulo 1

Introdução

1.1 Enquadramento

Cada vez mais ouvimos, sabemos e até sentimos em primeira mão a problemática que é o furto e desaparecimento de veículos. Segundo as autoridades policiais, cerca de 3/4 de todas as viaturas desaparecidas conseguem ser recuperadas. Estes veículos aparecem estacionados, bem estacionados, em locais que não levantam suspeitas, em zonas residenciais, como se tratasse apenas de mais um carro, sendo que a sua descoberta se deve, normalmente, a dois factores: ao acaso ou à sorte de uma patrulha Policial se cruzar com o veículo desaparecido e de, efetivamente, o reconhecer como desaparecido, através de uma pesquisa na base de dados proprietária das forças de segurança, após o avistamento; ou através de denúncias de cidadãos, que, ao verem uma viatura que não reconhecem estacionada por vários dias, comunicam o facto às autoridades. Esta denúncia é, por regra, rara, pois apesar de muitos cidadãos terem conhecimento de veículos que se encontram desaparecidos, através de amigos, redes sociais, etc, por norma não são capazes de os identificar na via publica. Isto deve-se, resumidamente, a dois fatores: ao simples facto de não andarem, ativamente, à procura; ou simplesmente porque não foram capazes de memorizar a matrícula ou até características estéticas do veículo.

A ideia e conseqüente tema para esta dissertação nasceu do problema supramencionado. Avaliar a possibilidade de criação de “algo”, simples, fácil e prático, que retire da equação as variáveis de memorizar e procurar, pela parte dos cidadãos.

Por sua vez, o desenvolvimento e massiva disseminação de equipamentos de comunicação móvel pessoais, tais como *Smartphones*, que possuem, hoje em dia, um apreciável poder de processamento aliado a um avançado leque de sensores, como câmaras fotográficas digitais, sensores GPS, acelerómetros, etc, torna-os em plataformas de desenvolvimento preferenciais e com possibilidade de uso em grande escala.

1.2 Motivação

Com o aparecimento e proliferação do conceito de Sistemas de Transporte Inteligente (ITSs), tem vindo a aumentar também o interesse por sistemas de reconhecimentos de matrículas. É já possível observar estes sistemas espalhados um pouco por todo o lado, desde parques de estacionamento para controlo de entradas e saídas, postos fronteiriços para controlo e segurança ou ainda em autoestradas para cobrança automática de portagens.

A par disto, o número de utilizadores de dispositivos móveis no mundo têm aumentado exponencialmente, o que, considerando toda a tecnologia que eles carregam, os tornam ferramentas poderosas muito para além do antigo conceito de telefone.

Daqui nasceu esta ideia, de tentar utilizar dispositivos móveis para, recorrendo a um sistema de reconhecimento de matrículas, tentar ajudar a solucionar um dos problemas com que a sociedade se depara cada vez mais, o furto de veículos.

1.3 Objetivo

A presente dissertação tem como objetivo central a exposição, estudo e divulgação de um conceito original, a utilização de dispositivos móveis como ferramentas na detecção de veículos furtados.

Para isso é proposta pesquisa, implementação e avaliação dos vários constituintes de um sistema desta natureza. Para que no fim, a ideia possa ser portada para uma plataforma móvel.

1.4 Estrutura da Dissertação

Para além do presente capítulo, esta dissertação é composta por mais 6, sendo eles:

- **Capítulo 2 - Conceitos Fundamentais** — Neste capítulo são apresentados os conceitos de Visão Computacional, Sistema de Reconhecimento de Matrículas, e Dispositivos Móveis.
- **Capítulo 3 - Revisão Bibliográfica** — Neste capítulo é exposta a revisão bibliográfica para os diferentes passos de um ALPR.
- **Capítulo 4 - Implementação e Discussão** — Neste capítulo são expostos os métodos implementados, para cada etapa do sistema, assim como uma discussão e comparação entre os mesmos.
- **Capítulo 5 - Dispositivo Móvel** — Neste capítulo é feita uma breve discussão sobre as possíveis plataformas móveis e qual a mais adequada para a criação da aplicação móvel. Assim como uma descrição do conceito e de como a aplicação deverá operar.
- **Capítulo 6 - Conclusões** — Neste capítulo é feita uma análise final ao trabalho realizado e são apresentadas propostas de melhorias para próximas investigação.

Capítulo 2

Conceitos Fundamentais

Com o objectivo de expor alguma informação sobre os temas tratados nesta dissertação, é realizada, numa primeira fase, a análise dos conceitos fundamentais dos mesmos.

2.1 Visão Computacional

“Visão computacional é a ciência de dotar, computadores e outras máquinas, com o poder da visão, ou a habilidade de ver” [1]

Tal como indicado em [1], “ver”, ou ter visão, não se resume ao simples e direto ato de gravar luz por forma a poder ser revisto, tal como é feito, por exemplo, por uma máquina de filmar ou fotografar. Visão é definida como o ato de captar uma imagem e, acima de tudo, interpreta-la por forma a serem tomadas ações ou decisões. Visão enquanto componente central, mas não fundamental, de um sistema que se considere inteligente, terá que seguir as mesmas linhas da própria definição, filosófica, de inteligência. Tal como esta, existem diversos componentes inerente à visão, incluindo, mas não limitado a memória, raciocínio, recuperação, estimação, reconhecimento e coordenação com outros sentidos. Não será, no entanto, correto afirmar que seriam necessários todos estes atributos para se considerar que um sistema teria algum grau de visão; mas por outro lado, um sistema que apenas contenha um destes não poderá ser considerado apto a ver. Até um certo nível, o ser Humano define visão a par do próprio sistema de visão Humano/animal, levando por isso, por vezes, a um juízo subjetivo por comparação a um sistema de visão computacional.

2.2 Sistema de Reconhecimento de Matrículas

Tal como referido em [2], nos últimos anos, os Sistemas de Transporte Inteligente (ITS - Intelligent Transportation Systems) têm tido um impacto grande e muito positivo nas vidas da população em geral. Estes sistemas procuram, com recurso há tecnologia, tornar o transporte de pessoas mais seguro, prático e inteligente. Tal como descrito em [2] e [3], estes sistemas encontram-se divididos em duas grandes categorias de sistemas: infraestruturas inteligentes e veículos inteligentes. Sendo que os Sistemas de Reconhecimento de Matrículas (LPR), estão tidos como centrais nos

sistemas de infraestruturas inteligentes, ao serem utilizados em diversas situações como o controlo de acessos a um edifício, gestão de acessos e pagamentos em parques de estacionamento, cobrança de portagens, gestão de tráfego e segurança fronteiriça, entre outras neste campo. Assim como referido em [4], o alastramento exponencial dos Sistemas de Informação, levou a que, também os veículos, sejam processados enquanto recursos conceptuais. Este processamento cria a necessidade de algo que faça a transformação entre o mundo real e os ITSs. Todos os veículos têm a possibilidade real de serem muito distintos uns dos outros; cor, forma, tamanho, entre outros, variam muito, mesmo dentro da mesma cidade. Sendo que a placa de matrícula, ainda que divergindo de país para país e em alguns caso de cidade para cidade [5], é efetivamente o elemento mais controlado e standard a partir do qual é possível identificar um veículo, tornando os LPRs instrumentos de extrema importância agora e no futuro.

2.3 Dispositivos Móveis

Segundo um estudo estatístico realizado pelo *Pew Research Center* intitulado *Pew Internet Project's research related to mobile technology* realizado a uma amostra da população dos Estados Unidos da América, o número de utilizadores de dispositivos móveis tem visto um aumento constante desde o início da década.

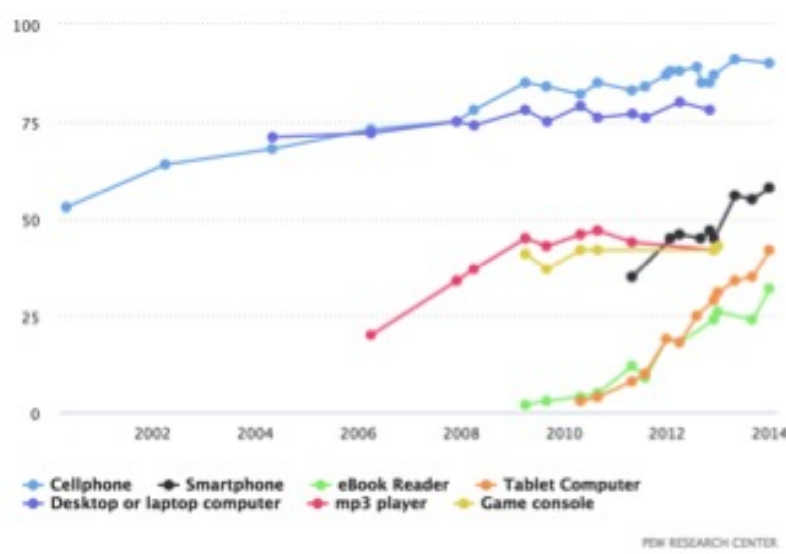


Figura 2.1: (Número de utilizadores de dispositivos móveis) Adaptada da página web de *Pew Research Center*

Tal como se pode observar pela figura 2.1, cerca de 90% da população tem um telemóvel, assim como 58% afirmam possuir um smartphone. Estes números levam a concluir que estes dispositivos tornaram-se, de facto, instrumentos valiosos para os utilizadores. Deixando de ser apenas

um equipamento para realizar e receber chamadas e mensagens, e passando a ser um dispositivo multiusos equipado com um leque de sensores para o dia a dia dos utilizadores. A existência de alguns desses sensores abriu a porta a um número alargado de aplicações especializadas, como sendo aplicações de GPS (*Global Position System*), pagamentos por NFC (*Near Field Communication*), aplicações de edição e partilha de imagens, entre outros. Será, então, possível concluir que os smartphones e outros dispositivos móveis são, atualmente, uma plataforma poderosa e valiosa para a criação ou adaptação de novas tecnologias.

Capítulo 3

Revisão Bibliográfica

Neste capítulo será exposta, sucintamente, alguma da pesquisa bibliográfica realizada e tida em consideração no decorrer da dissertação. De notar que, a pesquisa a seguir apresentada, explora apenas uma porção de toda a informação existente neste campo de estudo; pois tratando-se de uma área que desperta, atualmente, um grande interesse e que por isso se encontra em constante evolução e crescimento [6], e onde as ideias e soluções existentes divergem bastante, mesmo para problemas semelhantes, foi necessária uma pré seleção da informação a estudar com mais profundidade, ou seja, foram revistos mais métodos e soluções que não foram considerados para a elaboração deste documento.

3.1 Sistemas de Reconhecimento de Matrículas

Apesar de ser possível observar estes sistemas em funcionamento espalhados um pouco por todo o lado e em diversas aplicações. Não existe contudo, partindo da pesquisa bibliográfica, uma solução genérica para sistemas de reconhecimento de matrículas. Por observação e tal como mencionado em [5], é possível compreender uma das razões que poderá levar a esta variedade, pois, por norma, os sistemas de reconhecimento de matrículas encontram-se estacionários, numa posição conhecida e bem determinada e onde é também conhecida, à priori, a posição aproximada da matrícula na imagem captada pelo sensor. Tornando, com isto, o reconhecimento mais simples e sem necessidade de utilização de algoritmos ou soluções complexas, garantindo, também, uma boa taxa de sucesso, no reconhecimento.

Dada a natureza da aplicação proposta, foi necessária uma alargada pesquisa sobre o tema, onde foi por mais notória a diversidade de soluções propostas para as mais variadas situações de aplicação de um sistema desta natureza.

Como referido em [7], ao longo dos anos, muitas soluções distintas foram apresentadas, tendo por base técnicas e algoritmos bastante distintos, ainda que seguindo a mesma sequência de etapas.

Por norma, um sistema de reconhecimento de matrículas encontra-se dividido em 3 passos [6],[7], sendo eles: detecção da área da matrícula na imagem, segmentação dos caracteres, e reconhecimento dos caracteres.

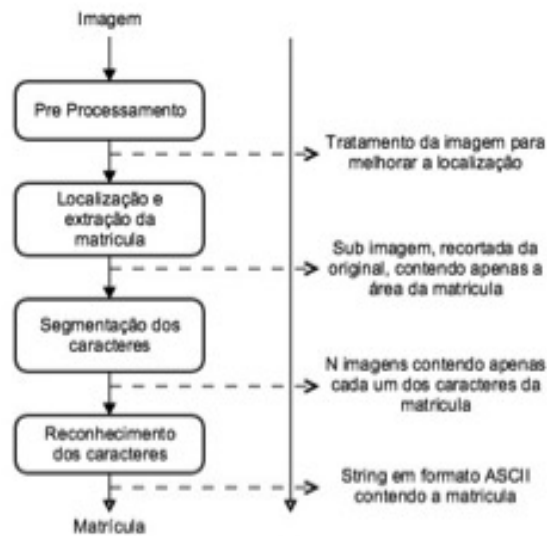


Figura 3.1: Etapas de um LPR

De todos estes passos, e segundo [6],[7], [8], etc, o mais importante e que envolve maior complexidade é a detecção da matrícula na imagem. Isto deve-se ao facto de, nesta fase, o algoritmo ter que ser capaz de ultrapassar todo o tipo de adversidades, como sendo o movimento da câmara, condições de iluminação adversas, tamanho efetivo da matrícula na imagem, rotação da mesma, distancia, ou ainda o número de outros objetos presentes na imagem. Sendo, também, nesta etapa que é possível observar a maior disparidade e variedade de métodos e soluções.

Entre elas, mas não exclusivamente: combinação de análise de orlas com operações morfológicas [9] , [10] e [11], baseadas em redes neuronais [12], baseadas em cor e lógica Fuzzy [13] e [14], baseadas em *Wavelets* [15], baseadas em análise da transformada de *Gabor* [16], entre outras.

3.1.1 Localização e segmentação da matrícula na imagem

Como já referido anteriormente, as diferentes técnicas desenvolvidas têm, por norma, uma aplicação física conhecida. Onde são também conhecidas e aceites as limitações do sistema, e sabendo-se, de ante mão, que alterando as condições físicas, para as quais o sistema de reconhecimento foi concebido, os resultados irão piorar.

Tendo em conta a ideia proposta neste trabalho, tais condições não são conhecidas à priori, o que acrescenta um elevado grau de dificuldade à sua concepção.

Esta dificuldade torna-se, por demais evidente, quando se analisam imagem recolhidas de fontes não controladas, como por exemplo, imagens capturadas por alguém através do seu telemóvel. Nesta situação, e sobre a qual este trabalho assenta, é virtualmente impossível controlar, à priori, aspetos como: distancia entre o sensor e o objeto; condições de iluminação; ângulo do sensor face à posição horizontal; a posição do objeto no enquadramento da imagem; ou ainda o número de

objetos presentes na imagem. Sendo por isso necessário procurar uma solução que minimize toda esta aleatoriedade.

3.1.1.1 Método das Janelas Concêntricas Deslizantes (SCW)

Este método foi proposto em [2], como sendo uma possível solução para situações onde a imagem é capturada em ambiente não controlado, ou seja, no exterior e com as variáveis já expostas para a aplicação idealizada. Para isso, descreve, hipoteticamente, que uma placa de matrícula numa imagem, pode ser vista como sendo uma irregularidade na textura dessa imagem. E como tal, alterações abruptas na característica local da imagem, poderão significar a probabilidade de existência de uma matrícula. Partindo desta definição, o método propõe então uma técnica para a segmentação da imagem baseada em janelas concêntricas deslizantes (SCW-*Sliding Concentric Windows*) e análise de componentes ligados (CCA – *Connected Components Analysis*) para procurar e identificar as, já definidas, irregularidades.

O algoritmo proposto segue, então, os seguintes passos, partindo de uma imagem em níveis de cinza $I(x,y)$:

- Criação de duas janelas rectangulares concêntricas A e B, de tamanhos $(2X_1) \times (2Y_1)$ e $(2X_2) \times (2Y_2)$, respectivamente, no primeiro pixel (canto superior esquerdo) da imagem, como ilustrado na imagem 3.2.
- Calcular o rácio $(M_B)/(M_A)$ do valor estatístico, média ou variância, das respetivas janelas.
- Aplicar a regra de segmentação.
- Seguir para o pixel seguinte, utilizando um movimento como representado na figura 3.2.

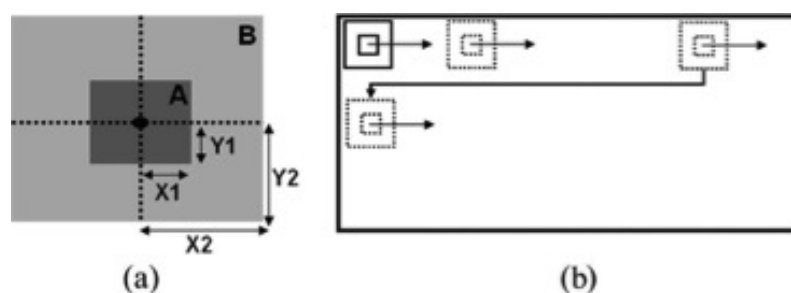


Figura 3.2: (a) Ilustração das janelas concêntricas b) Ilustração do processo de deslizamento das janelas (adaptado de [2])

Onde as dimensões e rácios (altura x largura) das janelas devem ser ajustadas face aos objetos de interesse, o que, e segundo descrito no método, implica conhecer as dimensões físicas, aproximadas, das matrículas em pesquisa. No sistema original proposto pelo autor, as matrículas em estudo eram Gregas com um rácio de aproximadamente $1/3$. Algo que, segundo a autoridade europeia *EUR-Lex*, será o rácio padrão para matrículas dentro do espaço europeu.

A regra de segmentação pode ser resumida da seguinte forma: se o rácio do valor estatístico medido (média ou variância) exceder um valor threshold especificado pelo utilizador, então o pixel central é considerado como pertencente a uma região de interesse (ROI – *Region of Interest*), ou seja, considerando (x,y) como sendo as coordenadas do pixel central das janelas, no correspondente pixel de uma nova imagem $I_{and}(x,y)$ será colocado o valor 0 se não pertencer a uma ROI ou o valor 1 caso pertença, seguindo a seguinte regra:

$$Em I(x,y) \rightarrow \begin{cases} I_{AND}(x,y) = 0 & , \frac{M_B}{M_A} \leq Threshold \\ I_{AND}(x,y) = 1 & , \frac{M_B}{M_A} > Threshold \end{cases} \quad (3.1)$$

Seguidamente é apresentado um *flowchart* do algoritmo de localização da matricula, por forma a ser mais perceptível os restantes passos:

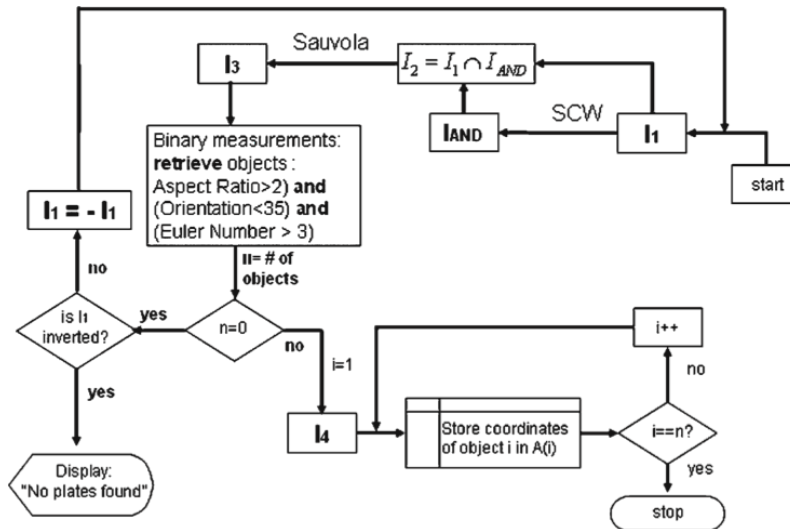


Figura 3.3: Flowchart que ilustra o algoritmo de localização da matricula (Adaptado de [2])

Durante o processo de SCW, é criada uma nova imagem binária $I_{and}(x,y)$, onde os pixels dos locais onde ocorrem as irregularidades (matriculas), têm o valor 1 e tudo o resto tem valor 0. Sendo esta considerada uma mascara, contendo as ROIs. Seguidamente é realizada uma operação lógica de adição *AND* entre a imagem original e a mascara criada: $I_2 = I_1 \wedge I_{and}$, resultando assim uma imagem, I_2 , em níveis de cinza, onde apenas aparecem as zonas de interesse, ROIs.

O próximo passo, como descrito no método, será a binarização da imagem I_2 . Para isso o autor propõem a utilização do método de threshold adaptativo local de *Sauvola* [17]. Este diverge dos métodos de binarização por threshold globais pois, em vez de utilizar um valor fixo de threshold para toda a imagem, este adapta-se às possíveis variâncias de luminosidade e contraste, calculando e alterando o valor de threshold para cada pixel, assegurando assim uma binarização mais correta.

Este constante ajuste, é efetuado percorrendo a imagem com uma janela $b * b$, e calculando os valores de média, $m(i,j)$, e desvio padrão $\sigma(i,j)$, por forma a calcular o valor de threshold,

$T(x,y)$, da seguinte forma:

$$T(x,y) = m(x,y) + \left[1 + k \cdot \left(\frac{\sigma(i,j)}{R} - 1 \right) \right] \quad (3.2)$$

E, como sugerido em [17], e utilizado no método, os valores para $k=0.5$, $R=128$ e $b=10$. Sendo ainda importante notar que a formula, juntamente com o valor atribuído a R possibilita que a contribuição, efetiva, do desvio padrão para o calculo do threshold seja também ela adaptativa, possibilitando que zonas onde a iluminação seja mais fraca, o valor de threshold seja também ele mais baixo.

Realizada a binarização, segue-se a etiquetagem dos objetos que permaneceram na imagem. Para isso é proposta a utilização de um método já vastamente documentado, (CCA - *Connected Components Analysis*), que resumidamente varre a imagem, procurando pixéis que poderão estar, de alguma forma, conectados entre si e podendo, esta conectividade, ser de 4 (conectividade vertical e horizontal) ou de 8 (conectividade vertical, horizontal e diagonal), e tendo sido escolhida conectividade de 8 para este método. Esta conectividade foi escolhida tendo em consideração 2 factores: ser a mais utilizada ao longo da bibliografia estudada; e tendo em atenção os objetos de interesse, que sendo nesta aplicação, caracteres, podem conter contornos arredondados, algo que apenas utilizando conectividade 8, é possível fazer uma correta aglomeração de objetos. Após o agrupamento dos pixéis, todos estes que pertencerem a um dado objeto são identificados com o mesmo valor numérico.

Feita a etiquetagem, e seguindo o esquema da figura 3.3, é feita uma filtragem de entre os objetos encontrados. Sendo, para isso, considerados 3 aspectos, que, até certo nível, serão únicos de um caractere: orientação $< 35^\circ$, $2 < \text{Excentricidade} < 6$ e número de Euler > 3 . Sendo que as coordenadas dos objetos que cumpram estes requisitos são então guardadas numa matriz $A(i)$, com (i) objetos/caracteres, para posterior reconhecimento.

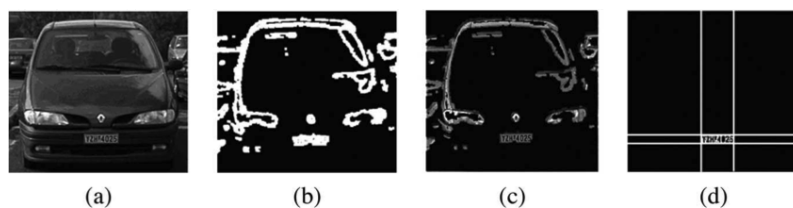


Figura 3.4: a)imagem original. b) máscara resultado de SCW. c)resultado da aplicação da mascara à imagem original. d)localização da matricula (Adaptado de [2])

3.1.1.2 Combinação de análise de orlas com operações morfológicas

Tal como proposto em [4], [18] e [11], este é um método que se baseia na classificação de uma matricula numa imagem como sendo: uma área rectangular onde a ocorrência de orlas hori-

zontais e verticais será superior ao resto da imagem. Utilizando esta premissa, é então concebido um algoritmo de pesquisa e contagem de orlas por forma a se descobrir possíveis existências de matrículas na imagem.

Inicialmente, e dependendo de aplicação para aplicação, é realizado algum tipo de pre processamento há imagem. Este passo consiste numa série de operações cujo objetivo será acentuar a definição das orlas da matrícula e eliminar algum ruído na imagem, como por exemplo, uma ferramenta que possibilita o aumento do contraste geral da imagem: equalização de histograma.



Figura 3.5: Ilustração de equalização de histograma

Feito este passo, será então necessário passar à pesquisa e análise de orlas. Para isso, e como proposto em [4], é utilizado o algoritmo de detecção de orlas de *Sobel*. Este algoritmo utiliza duas mascaras, uma horizontal e uma vertical, que, quando aplicadas à imagem, por convolução, fazem sobressair, de forma notória, as orlas dos objetos presentes nelas. Sendo as máscaras as seguintes:

$$G_V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} ; G_H = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & 1 \end{bmatrix} \quad (3.3)$$

Seguidamente, são calculadas as projeções horizontais e verticais das respectivas imagens criadas pela aplicação da detecção de orlas. Este processo, e seguindo a definição de matrícula feita anteriormente, permite encontrar o local na imagem onde estão concentradas o maior número de orlas, ou seja, a matrícula. E é calculado somando ao longo de, linhas no caso da projeção vertical e colunas na projeção horizontal, o número de pixels brancos (valor binário 1), ou seja orlas, existentes. Ou seja, definindo-se as projeções vertical e horizontal da seguinte forma:



Figura 3.6: Da esquerda para a direita: Operador de Sobel aplicado nos dois sentidos; Aplicado Horizontalmente; Aplicado Verticalmente (Adaptado de [4])

$$P_H(x) = \sum_{j=1}^M f(x, j) ; P_V(y) = \sum_{i=1}^N f(i, y) \quad (3.4)$$

onde x e y são colunas e linhas respectivamente, e M e N são o tamanho da imagem (M – número total de linhas e N - número total de colunas)

Feito este passo, é então possível utilizar a projeção vertical para encontrar a banda (faixa horizontal) onde existe maior probabilidade de estar presente a matricula. Para isso é encontrado o valor máximo desta projeção e a respetiva linha onde ocorre, e é recortada uma faixa (banda) em torno dessa linha, utilizando, no caso mais simples, um valor fixo predefinido ou ainda um valor threshold variável.

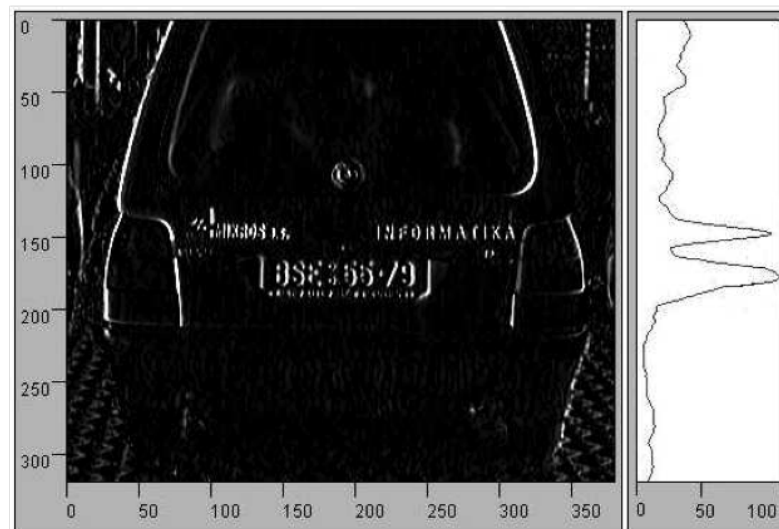


Figura 3.7: Ilustração de uma projeção vertical (Adaptado de [4])

Feito isto, é então realizado de novo o processo, desta vez para a projeção horizontal, e tendo em consideração as informações já recolhidas no passo anterior. Resulta assim uma janela, retirada da imagem original, onde existe a maior probabilidade de se encontrar a matricula.



Figura 3.8: Ilustração do corte da imagem por uma banda horizontal

3.1.2 Segmentação dos Caracteres da Matricula

Feita a localização e extração da matricula da imagem original, é necessário agora proceder à segmentação ou separação dos caracteres da matricula. Para isso, e como indicado em [3], foram já desenvolvidas uma considerável quantidade de técnicas. Das quais se enumeram apenas algumas:

- Algoritmos baseados em projeções [4], onde, à figura de um método já explanado anteriormente, a ideia é, partindo de uma imagem binária da matricula, obter a soma de pixels brancos, por linha e por coluna, criando assim uma projeção. Com esta, é possível encontrar os mínimos, que representarão os espaços entre caracteres, para assim os segmentar.

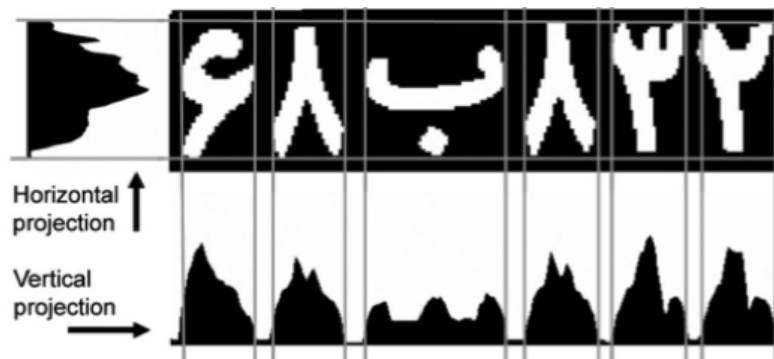


Figura 3.9: Segmentação por projeções (Adaptado de [3])

- Algoritmos baseados em Análise de Componentes Ligados - CCA (*Connected Component Analysis*), onde cada objeto binário presente na imagem é procurado e etiquetado como um valor único, sendo depois possível filtrar e separá-los face a características padrão dos caracteres, como rácio altura/largura, área, tamanho, orientação, alongamento, etc.
- Algoritmos baseados em seguimento de contornos e “*bug following*” [19], onde a imagem binária é percorrida, pixel a pixel até ser encontrado um pixel pertencente a um objeto. Encontrado este, o algoritmo altera o seu funcionamento e passa a seguir o contorno do objeto,

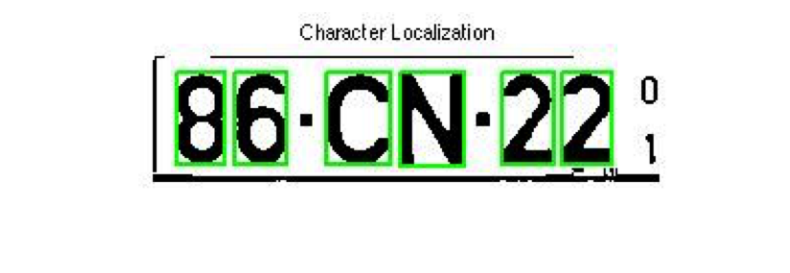


Figura 3.10: Resultado do método CCA

procurando o pixel seguinte da orla, e guardando informação do mesmo, até regressar ao ponto de partida.

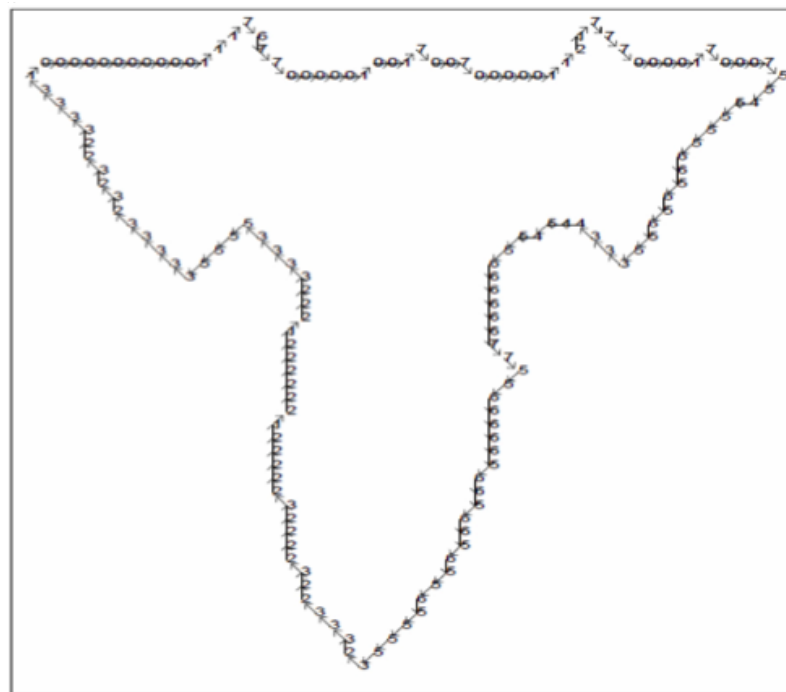


Figura 3.11: Seguimento de contorno (Adaptado de [19])

- Entre outros.

3.1.3 Reconhecimento dos caracteres (OCR – *Optical Character Recognition*)

Extraídos os caracteres da matricula, faltará a ultima fase do modelo, que passa então por identificar e traduzir as pequenas imagens binárias, resultantes de toda a segmentação efetuada, em texto legível (*ASCII*). Para isso, e como referido em [3], também aqui existem diversas soluções e algoritmos propostos, sendo que, a partir da bibliografia revista e mencionado em [3], é possível compreender 3 principais grupos de soluções:

Classificadores estatísticos, Arquiteturas de Inteligência Computacional e *Pattern-matching* (correspondência de padrões). Segundo [3], os resultados obtidos de soluções de Arquiteturas de

Inteligência Computacional como redes neuronais, obtêm, na sua grande maioria, excelentes resultados no reconhecimento, porém é também referido que é necessária uma muito extensa biblioteca de amostras para que a rede neuronal consiga, efetivamente, aprender a reconhecer caracteres.

Por outro lado, é ainda indicado que os sistemas de OCR baseados em *Pattern-matching*, têm vindo a evoluir positivamente com o tempo, sendo cada vez mais precisas e com menos ambiguidades no reconhecimento. Referindo-se ainda que será muito mais vantajoso para este meio, a continua melhora deste método, do que redesenhar ou treinar sistemas de reconhecimento do zero.

3.1.3.1 *Pattern/Template-Matching*

Trata-se de um método clássico em reconhecimento de padrões (Pattern Recognition), e baseia-se na ideia de comparação. [18] É, efetivamente, passível de ser usado nesta aplicação devido há padronização das fontes tipográficas utilizadas nas matriculas Portuguesas e Europeias.

Primeiramente é necessária a criação de uma base de caracteres (templates) para a comparação. Como não foi possível verificar qual a fonte tipográfica exata das matriculas Portuguesas, foram realizados alguns testes que levaram a concluir, com algum grau de certeza, que esta se assemelhará a um tipo de letra padrão – *Arial*.

Sendo assim, é então criada uma pequena imagem de 42 por 24 pixéis para cada caractere do alfabeto Português (A-Z) e o mesmo para os números de 1 a 9, o que totaliza 36 caracteres possíveis.



Figura 3.12: Exemplo de caracteres para OCR

Visto este método ser dependente do tamanho (altura x largura) das imagens que compara e os templates serem 42x24, é necessário proceder a um ajuste do tamanho, normalmente redução, dos caracteres segmentados.

Tendo todos os caracteres guardados num vector, são então comparados, um a um, com cada caractere da matricula. Esta comparação é realizada calculando-se o coeficiente de correlação entre o objeto retirado da matricula e todos os caracteres presentes na base de dados, sendo que a o algoritmo que obtiver maior coeficiente de correlação é considerado como encontrado e o respectivo código *ASCII* é guardado.

Capítulo 4

Implementação e Discussão

Neste capítulo irão ser expostos os métodos implementados para cada etapa descrita anteriormente, ou seja, localização e segmentação da área da matrícula, segmentação dos caracteres, e reconhecimento dos mesmos. Assim como uma breve discussão sobre as diferenças entre eles, os seus pontos fortes e fracos.

Sendo ainda necessário referir que foram implementados outros métodos e ideias para além dos a seguir apresentados, mas que por não terem, no geral, sido obtidos resultados satisfatórios, foram descartados. Tendo sido esta implementação realizada em Software Matlab, e utilizando como imagens de testes um conjunto de fotografias captadas por um smartphone Android, com uma resolução nativa de 2448x3264 pixéis (aprox. 8 Mpx) RGB. Estas imagens de teste foram capturadas tendo em conta possíveis situações com as quais o sistema se poderá deparar, ou seja, com distintas distâncias entre o sensor e a matrícula, diferentes localizações da mesma na imagem, com mais e menos objetos e de diferentes tamanhos, e ainda em várias situações distintas de luminosidade, como por exemplo: matrícula bem e uniformemente iluminada, matrícula não uniformemente iluminada, e ainda fracamente iluminada. Sendo possível observar alguns exemplos na figura 4.1.



Figura 4.1: Exemplos de imagens de teste

4.1 Localização e Segmentação da Área da Matricula

Neste ponto foram implementados os dois métodos descritos com mais ênfase anteriormente, ou seja, o método das janelas concêntricas deslizantes e um método inspirado em a análise de orlas e projeções, com algumas modificações pensadas pelo autor.

4.1.1 Método das janelas concêntricas deslizantes (SCW)

Este método foi selecionado para implementação devido à ideia, bastante diferente do resto da bibliografia, que apresenta. Que juntamente com as premissas sobre as quais ele foi desenvolvido, ou seja, o funcionamento com imagens naturais complexas, fizeram dele um bom candidato.

Tendo sido implementado tal como descrito anteriormente e baseado em [2]. E cujo algoritmo se encontra nos anexos.

Os resultados da segmentação foram, no geral, muito satisfatórios, dentro de certos parâmetros e limitações. Conseguindo em 10 imagens distintas, tanto em termos de distancia como de condições de iluminação e ainda de numero de objetos presentes na imagem, ainda que respeitando algumas limitações principalmente em termos de distancia, conseguido identificar a matricula em 7 delas.



Figura 4.2: Etapas da segmentação por SCW

A imagem utilizada na figura 4.2, foi escolhida como imagem base para os testes, pois apresenta uma distancia à matricula que se considera razoável, juntamente com uma distribuição de brilhos não uniforme, sombras e reflexos e ainda objetos muito distintos ao longo da imagem.

E sobre a qual este método funcionou corretamente.

Por observação dos resultados que foram sendo obtidos, utilizando esta técnica, foi possível retirar algumas conclusões:

- Apesar de, segundo os autores do método, ser afirmado que o método apresenta tempos de processamento muito baixos (inferiores a 1 segundo) para imagens capturadas com uma resolução de 786432px (1024x768), o melhor tempo registado pelo algoritmo recriado, e utilizando imagens ajustadas para, aproximadamente, a mesma resolução foi, em média, de 120 segundos.
- Tal como referido no artigo original, este método torna-se inconsistente com a variação de luminosidade, ou seja, imagem capturadas onde a matricula se encontra bem iluminada obtêm um bom resultado, enquanto matriculas que estejam na sombra não.
- É ainda bastante dependente da distancia entre o sensor e a matricula, pois para distancias superiores a 6/7 metros, este começa a encontrar problemas.
- Como se baseia na premissa que uma matricula pode ser vista como uma irregularidade na imagem, imagens que contenham mais objetos com notórias variações de luminosidade correm o risco de ter várias detecções falsas.
- O sucesso do método variou e dependeu bastante do valor de threshold escolhido aquando da comparação entre as duas janelas, sendo o valor indicado em [2] de 0.7

4.1.2 Método inspirado em Análise de Orlas e Projeções

Este método foi baseado no estudo das projeções vertical e horizontal da imagem binária resultante do pre processamento. Mas em vez de estas projeções representarem a distribuição de orlas na imagem, representam sim o número total de pixéis, na imagem binária, com valor 1.

Para esta ideia funcionar, é necessário existir um sistema de pre processamento forte, por forma a eliminar o máximo de objetos indesejados da imagem.

4.1.2.1 Pre-Processamento

A primeira fase pensada para o pre processamento, baseou-se em observação do ambiente e condições em que a imagem iria ser capturada pela aplicação móvel. Para isso é necessário observar em que posição e condições o equipamento móvel irá ser utilizado durante a procura de matriculas.

Observando a figura 4.3, que representa uma possível imagem capturada durante a utilização da aplicação móvel, é possível concluir que a grande maioria dos objetos de interesse se encontram localizados na parte inferior da imagem. Sendo assim foi pensada uma técnica que tira partido desse facto, dividindo a imagem em 3 zonas, como demonstrado na figura 4.3. Sendo dada, ao utilizador, a opção de seleccionar a zona antes do inicio da procura, ou ainda de não utilizar qualquer



Figura 4.3: Exemplo de uma possível imagem capturada pela aplicação final

divisão e utilizar a imagem por inteiro.

Feita, ou não, esta seleção, segue-se a limpeza e binarização da imagem. Para isso foi pensado um método que tira proveito do facto de a matrícula ter um fundo branco e letras pretas, e implementado da seguinte forma:

1 – Leitura da imagem original, capturada com uma resolução 3264x2448, seguida de uma redução para uma resolução de 1152x868 e efetuado o corte da mesma baseado na seleção da zona 3:



Figura 4.4: Primeiro passo - corte

2 – Conversão da imagem para níveis de cinza e binarização da mesma, utilizando o método de binarização por threshold de *Otsu*:



Figura 4.5: Após binarização

3 – Seguidamente é criada uma nova imagem binária, resultante da convolução da imagem anterior com um filtro de mediana rectangular de tamanho 20x50 px. Sendo depois realizada uma operação de subtração entre esta nova imagem e a imagem anterior:

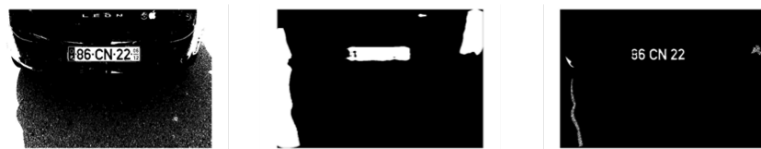


Figura 4.6: Da esquerda para a direita: Imagem após binarização; Imagem resultante da aplicação do filtro; Resultado da subtração

Com esta subtração, foi então possível tirar partido do aspeto físico de uma matrícula para a fazer sobressair.

Em seguida, para a extração da matrícula em si, foram utilizadas as projeções verticais e horizontais, tal como já exposto numa secção anterior deste relatório.

Para isso é inicialmente calculada a projeção horizontal:

Daqui, é procurado o valor máximo da projeção e retirada uma faixa horizontal onde esta presente a matrícula. Esta faixa é calculada segundo um valor *threshold* que pode ser ajustado, ou como uma percentagem do valor máximo.

Seguidamente é efectuado o mesmo processo mas desta vez utilizando a projeção vertical e apenas sobre a faixa anteriormente cortada. Descobrendo-se assim uma janela, aproximada, onde estará a matrícula.

Deste processo resulta então uma janela, cujas coordenadas indicam o local mais provável para a existência de uma matrícula.

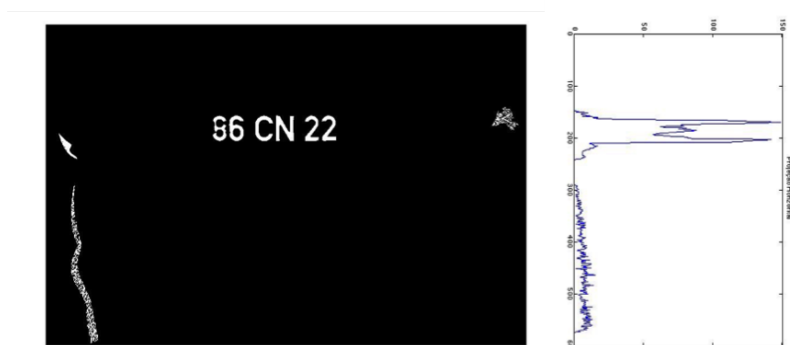


Figura 4.7: Imagem pre processada e respectiva projeção Horizontal

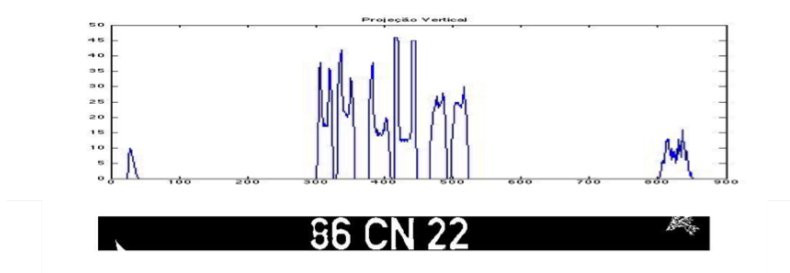


Figura 4.8: Faixa cortada e respectiva projeção vertical

Utilizando esta informação, é então transposta a janela para a imagem em níveis de cinza original. Este processo é feito por forma a minimizar os possíveis estragos que o pre processamento possa ter causado aos caracteres.



Figura 4.9: Matricula recortada

Este método, ainda que limitado, provou ser muito rápido e com um custo computacional bastante baixo, algo que seria de esperar por comparação a outros que seguem a mesma topologia [3], [10], e que, nas condições certas, bastante bom em termos de reconhecimentos conseguidos.

Tal como no método anterior, também neste são possíveis enumerar alguns problemas e limitações que foram observadas, sendo eles:

- A matricula deverá ter uma presença proeminente na imagem
- A inclinação da matricula não deverá exceder aproximadamente 20 graus

- No caso de existirem vários objetos com significativa alteração de brilho na imagem, a matricula deverá ser o objeto mais próximo
- A matricula deverá estar bem e uniformemente iluminada
- A matricula não deverá estar obstruída por sujidade ou objetos que a ela não pertençam
- Bastante rápido e simples, se comparado com o anterior, tendo o programa completo (usando este método) de reconhecimento demorado 3 segundos até identificar e traduzir os caracteres

Ainda assim, e tendo em conta as condições em que o produto final irá, preferencialmente, operar, foi considerado uma boa plataforma para a criação da aplicação móvel.

4.2 Segmentação dos Caracteres da Matricula

Nesta secção foi implementado um algoritmo de CCA (*Connected Components Analysis*) tal como já exposto anteriormente.

Este recebe a imagem, já recortada, da matricula em níveis de cinza, sobre a qual são efectuadas operações de limpeza de ruído, binarização, e remoção de pequenos objetos que possam existir.

Posteriormente são então etiquetados todos os objetos ainda presentes e é efectuada uma filtragem dos mesmos, por forma a serem seleccionados apenas os caracteres. Esta filtragem leva em consideração o tamanho de cada objeto face à média de tamanhos do total de objetos e o rácio Altura/Largura do objeto, que sendo caracteres, são bem definidos.

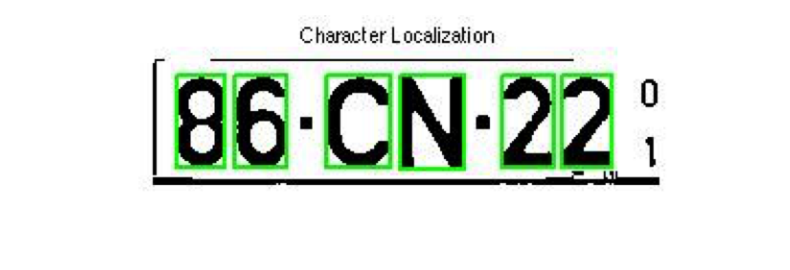


Figura 4.10: Caracteres encontrados por CCA

Este método, ainda que simples face a outras opções, demonstrou ser muito eficaz na segmentação dos caracteres, encontrando corretamente quase a totalidade de matriculas testadas. Ainda de referir que as matriculas para chegarem a esta etapa, já tiveram que ultrapassar as limitações das etapas anteriores.

4.3 *Pattern/Template Matching*

Esta etapa final no sistema de reconhecimento foi implementada segundo a descrição efectuada anteriormente no documento. Recorrendo à criação de uma biblioteca de caracteres alfanuméricos com uma resolução de 42x24 px, compreendida pelas letras de A a Z e os algarismos de 1 a 9.

Visto que, o calculo do coeficiente de correlação é altamente dependente do tamanho dos dois objetos, foi necessário acrescentar ao programa uma ferramenta que padronize, ou seja, reduza, os caracteres segmentados para a mesma resolução dos algarismos da biblioteca.

Foi possível observar, e como também mencionado em várias fontes como [3], que este método não é totalmente preciso e que pode, efetivamente, errar no reconhecimento, pois os caracteres segmentados terão que ser o mais perfeitos em aparência, ou seja, com curvas e rectas bem definidas, para assim evitar ambiguidades.

Tal como seria de esperar, e comprovado durante os testes, estes erros derivados a ambiguidades são mais frequentemente em caracteres alfabéticos que se assemelham a caracteres numéricos, como por exemplo a letra i (I) e o algarismo 1, ou a letra o(O) e o algarismo 0.



Figura 4.11: Caracteres reconhecidos

4.4 Case Study

Seguidamente serão apresentados mais casos de sucesso e insucesso na localização da matrícula, tanto para o método das janelas deslizantes como para o método inspirado em análise de orlas e projeções.

4.4.1 Método das Janelas Deslizantes



Figura 4.12: Caso 1 - Resultado Positivo

Neste caso, e apesar de as condições de iluminação não serem as ideais, o método conseguiu localizar os caracteres com sucesso, isto dever-se-á à distância entre a câmara e a matrícula, que a torna de facto um objeto de destaque na imagem.

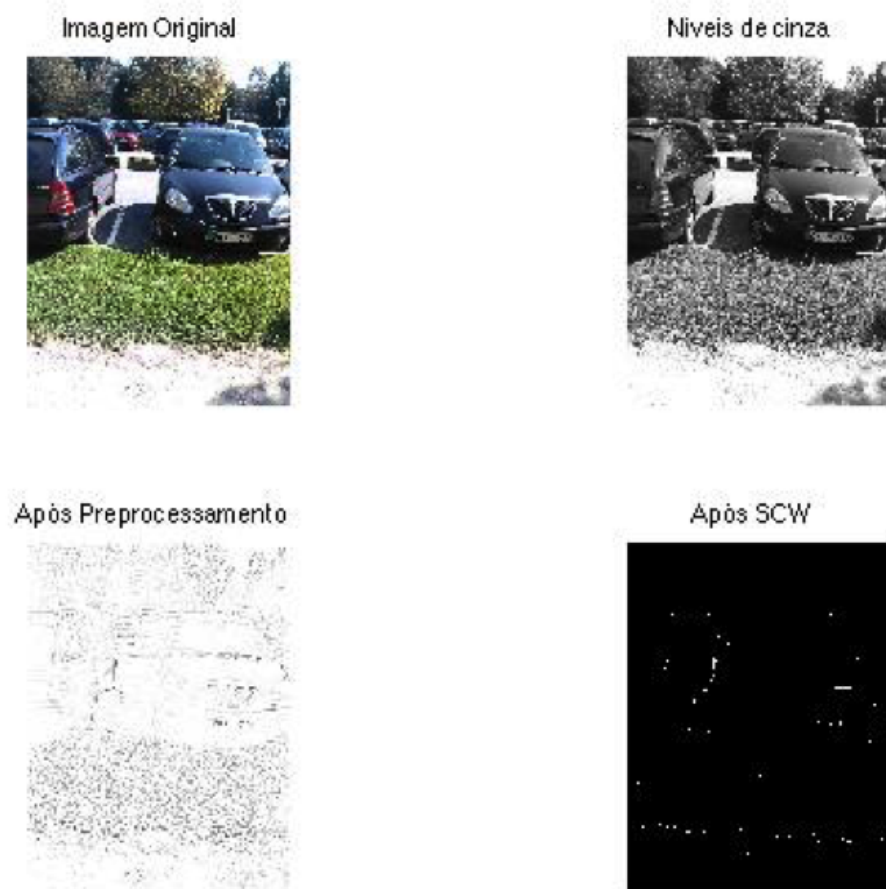


Figura 4.13: Caso 2 - Resultado Negativo

Neste caso o método não obteve um resultado positivo, podendo justificar-se não só pela distância da matrícula, mas principalmente pela existência de vários objetos que apresentam transições entre zonas claras e escuras bastante acentuadas.

4.4.2 Método Inspirado em Análise de Orlas e Projeções



Figura 4.14: Caso 3 - Resultado Positivo

Neste caso, como é possível observar na imagem 4.14, o método foi capaz de localizar com sucesso a matrícula e os caracteres, pois apesar de esta não se encontrar horizontal, aparece como um objeto de destaque na imagem e com uma iluminação bastante uniforme.



Figura 4.15: Caso 4 - Resultado Negativo

Tal como é possível observar na imagem 4.15, neste caso o método não foi capaz de extrair a matrícula, sendo o principal motivo da falha a fraca iluminação da matrícula face aos restantes objetos da imagem. Sendo que nesta situação o método elimina a matrícula na fase de processamento e extração.

4.5 Conclusões

Após bastante pesquisa e experimentação, e tendo em consideração o conceito da aplicação móvel que daqui poderá advir, foi considerado que o conjunto de métodos mais simples, ou seja, análise de projeções, segmentação por CCA e OCR por *pattern/template matching* seriam a melhor plataforma para a implementação em ambiente móvel. Pois tudo considerado, e tendo em atenção o utilizador final, estes métodos mais simples implicam um menor peso em termos de processamento no equipamento móvel, o que implicitamente significará um menor gasto energético e uma aplicação mais eficiente.

Adicionado a estes fatores, será ainda importante considerar o número de dispositivos a operar, isto porque atendendo há ideia de pesquisa distribuída, ou seja, vários equipamentos a operarem ao mesmo tempo, será interessante explorar uma temática de “o número faz a força”, ou seja, o facto de a taxa de sucesso de um equipamento poder ser mais baixa por utilizar um método mais simples de detecção, será compensado pela quantidade de equipamentos na busca.

Capítulo 5

Dispositivo Móvel

Neste capítulo irá ser feita uma breve discussão sobre as possíveis plataformas móveis e qual a mais adequada para a criação da aplicação móvel. Assim como uma descrição do conceito e de como a aplicação deverá operar.

5.1 Plataforma Móvel

Tal como já referido anteriormente, a plataforma/dispositivo móvel preferencial para o lançamento deste sistema centra-se, mas não se limita, aos smartphones.

Estes por sua vez divergem, grandemente, não só de marca para marca, mas também no sistema operativo (OS) que usam. Sendo assim tornou-se necessário realizar um estudo das plataformas/OS mais utilizadas e escolher a possível plataforma de lançamento.

Este estudo debruçou-se em dois grandes campos: número de utilizadores ativos e o custo, relativo, de lançar a aplicação ao público. Dado isto, foi então possível reduzir a lista de plataformas às três mais famosas e consequentemente mais utilizadas: Google Android OS, Apple iOS e Microsoft Windows Phone.

Sendo assim, e tendo por base as informações recolhidas nas respectivas páginas WEB acerca dos preços de publicação, assim como recorrendo a uma análise de mercado realizada pela empresa Techcrunch cerca da distribuição das várias plataformas pelo mercado, foi possível condensar a seguinte tabela:

Tabela 5.1: Comparação entre plataformas

Plataforma	Custo Associado	Numero de Utilizadores, 1º Trimestre 2014
Google Android / AOSP	25\$ (aprox. 18 Euro)	240,777,252
Apple IOS	99\$/Ano (aprox. 75 Euro)	43,719,000
Microsoft Windows Phone	19\$/Ano (aprox. 14 Euro)	13,274,030

Feito isto foi então possível decidir a plataforma Android OS como sendo a mais promissora para uma primeira migração, pois será a que apresenta uma relação (custo de publicação)/(número

de potenciais utilizadores) menor. Será ainda importante referir que com os avanços na área do processamento de imagem móvel, em particular com a migração de bibliotecas de visão computacional *Open CV* para esta plataforma, assim como o ambiente Open Source que a mesma apresenta, faz desta a melhor candidata para a criação da aplicação móvel deste sistema.

5.2 Aplicação Móvel

5.2.1 Conceito

Como já referido a quando da introdução, o objetivo da aplicação será procurar matriculas de veículos desaparecidos, e gravar a sua localização para futuro processamento.

Para isso foram pensadas duas situações diferentes: o smartphone estar colocado num suporte no vidro, com a câmara virada para a frente do veículo; e uma outra onde o utilizador poderia apenas apontar o equipamento para um veículo estacionado em via publica. O objetivo de ambas é o mesmo, localizar e identificar a matricula, mas o conceito que a aplicação teria de seguir é bastante diferente. Para a primeira situação, a aplicação seria iniciada, seria feito o descarregamento da lista de matriculas a procurar, e funcionaria de forma totalmente autónoma, para assim não causar distrações ao condutor, capturando e processando uma imagem a uma predeterminada cadência. Sendo que, segundo os testes realizados, esta opção será a que apresenta maiores dificuldades ao sistema de reconhecimento, pois terá de ser capaz de ultrapassar problemas como: movimento na imagem, distâncias, iluminação, número de objetos, etc.

Para a segunda situação as condições serão mais controladas, pois o utilizador não deverá estar a uma grande distancia do veículo e terá a câmara diretamente apontada para o mesmo. O que segundo os testes realizados, levará a um maior sucesso na detecção e reconhecimento.

Para qualquer dos casos, o passo final será o mesmo. Quando detectada uma matricula assinalada como desaparecida, o equipamento deverá guardar a localização em que se encontra, e enviar uma notificação para o servidor central indicando a matrícula descoberta, as coordenadas da mesma, assim como a imagem que despoletou a descoberta para posterior confirmação. Um ponto muito importante, e também pensado, trata da problemática da legalidade e defesa da privacidade dos cidadãos, sendo que a solução equacionada baseia-se no bloqueio de acesso, por parte do utilizador da aplicação, às fotografias que a mesma captura e analisa, ou seja, o utilizador não terá acesso nem poderá guardar para si as imagem capturadas. Sendo que, após o processamento pela aplicação, as imagens serão automaticamente eliminadas do dispositivo.

Outro ponto ponderado foi a possibilidade de ter todo o sistema de reconhecimento implementado num servidor remoto, e onde a aplicação móvel serviria apenas como sensor remoto e interface com o utilizador. Esta ideia implementada através da criação de uma simples aplicação para sistema Android, que captura uma imagem e a envia para um servidor. Sendo que, não tendo sido possível realizar testes práticos de campo com a mesma, serve apenas de conceito para um trabalho futuro.

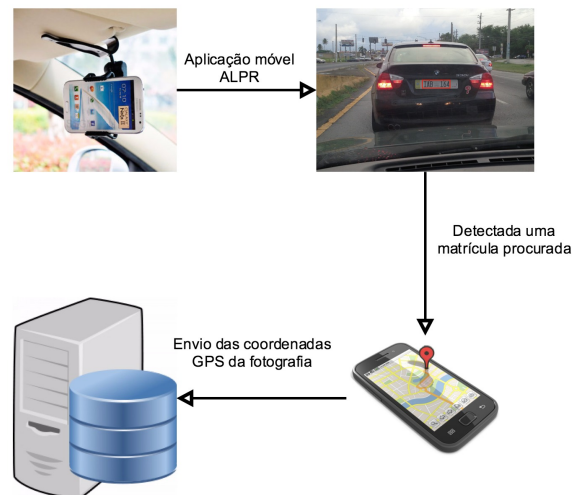


Figura 5.1: Ilustração do conceito de utilização da aplicação

Sendo possível resumir a ideia e conseqüente conceito a um funcionamento em comunidade, baseado numa busca distribuída, onde o sucesso do conceito depende, vastamente, do número de utilizadores ativos.

5.2.2 Requisitos da aplicação

Seguidamente são listados alguns requisitos funcionais e não funcionais definidos para a aplicação:

- Sistema operativo Android OS
- Sensor de câmara com 1 Mpx
- Sensor de posicionamento GPS
- Interface simples e de rápido manuseio
- Depois de iniciado o processo de pesquisa de matrículas, não deve necessitar de manuseio por parte do operador
- Optimizada para um gasto de bateria o mais baixo possível
- Incitar a menor distração possível ao utilizador

5.2.3 Implementação

Delineado o conceito geral e alguns requisitos, foram então criadas duas aplicações distintas para teste. A primeira aplicação criada tirou partido da mais recente ferramenta fornecida pelo software *Matlab Simulink*, que resumidamente é capaz de compilar um sistema de blocos de

Simulink numa aplicação Android. Sendo assim, foi então necessário migrar todo o sistema, previamente escrito em *Matlab*, para ambiente *Simulink*, algo que é possível observar na figura 5.2. Foram usados, sempre que possível, os blocos predefinidos pelo programa, como o redimensionamento, possível recorte, binarização por *threshold* adaptativo, filtragem e operações matemáticas. Mas ainda assim foi necessário recorrer a funções definidas pelo utilizador para a definição de algumas funções mais específicas como um bloco para a extração da matrícula, que irá calcular as projeções horizontal e vertical da imagem e realizar o recorte da matrícula, ou ainda o reconhecimento de caracteres realizado por *template matching*.

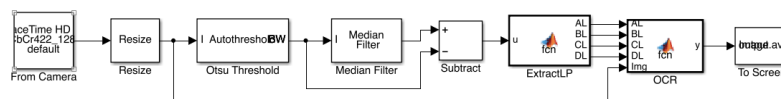


Figura 5.2: Ilustração do sistema implementado em *Matlab Simulink*

Feita esta migração e compilada a aplicação, foi então instalada no smartphone, e feitos alguns testes, foi então possível retirar algumas conclusões:

- Após execução da aplicação no equipamento, não é possível qualquer interação com a mesma, ou seja, não possui qualquer interface entre a aplicação e o utilizador. Podendo, este, apenas iniciá-la e terminá-la
- A necessidade de instalação de software e bibliotecas adicionais para a execução da aplicação
- Extremamente lenta na sua execução, o que leva a concluir ser muito pouco eficiente
- Quando implementado o sistema na sua totalidade, não foi possível obter nenhum resultado minimamente satisfatório. Apenas se obtiveram resultados positivos quando migrada apenas a parte inicial, de pre processamento, do sistema

Em conclusão, apesar de, inicialmente, se ter demonstrado como uma solução promissora para uma primeira migração, acabou por não o ser. Não tendo sido obtidos quaisquer resultados satisfatórios com a mesma.

A segunda aplicação criada assenta na ideia já exposta, baseada em processamento remoto (*Cloud Computing*), onde a aplicação servirá como interface entre o utilizador e o sistema, que estará alojado numa máquina remota.

Para o efeito foi então criada uma aplicação simples, com recurso a uma ferramenta de criação de aplicações Android disponibilizada pelo MIT (*Massachusetts Institute of Technology*) intitulada *MIT App Inventor*, e recorrendo a alguns tutoriais disponibilizados na página WEB *Pura Vida Apps*. Esta aplicação permite ao utilizador captar uma imagem com a câmara do smartphone, ou ainda escolher uma imagem guardada no dispositivo, e envia-la para um servidor para posterior processamento, tendo sido concebida com uma interface simples e de rápido manuseio.



Figura 5.3: Aplicação móvel em funcionamento

Apesar de ainda serem necessários testes adicionais a esta solução de processamento remoto, a aplicação demonstrou ser efetivamente rápida e simples, cumprindo corretamente a sua função, sendo que testes futuros deverão ter em conta uma análise custo/benefício entre esta solução e uma aplicação cujo processamento se realiza offline, ou seja, todo o sistema implementado e executado no dispositivo.

Capítulo 6

Conclusão

Neste capítulo serão apresentadas as principais conclusões, assim como algumas sugestões de trabalho a desenvolver em investigações futuras.

6.1 Conclusões

No final de bastante pesquisa, implementação, testes, erros e correções, pensa-se que foi realizada uma boa revisão há possibilidade de implementação de um sistema de reconhecimento de matriculas num dispositivo móvel e dos benefícios que tal implementação possa trazer. Foram analisadas e compreendidas as diferentes etapas que compõem um sistema desta natureza, tendo-se chegado a algumas conclusões específicas sobre a possível implementação noutro sistema. No que toca há fase mais crucial de um LPR, ou seja, a localização e segmentação da matricula, foram analisados mais profundamente dois métodos muito diferentes, e compreendidas as limitações e pontos fortes de cada um, sugerindo-se que para uma primeira migração do sistema, seja adoptado o método mais simples, e que por revisão da implementação feita, existe ainda a possibilidade de tornar o método mais complexo, rápido o suficiente para que possa também ele ser implementado noutra plataforma. Foi ainda analisada a aplicação móvel, onde foi realizado um comparativo entre as varias plataformas disponíveis, e quais as vantagens e desvantagens de cada uma, tendo por fim sido aconselhada a plataforma Android para uma primeira migração. Sobre a qual ainda foi criada uma pequena aplicação de captura de imagens de teste.

6.2 Trabalho Futuro

Esta dissertação cobriu a primeira etapa da ideia de utilizar dispositivos móveis como sistemas de reconhecimento de matrículas, ou seja, o estudo e compreensão dos vários métodos e a proposta de uma possível solução para a implementação.

Tentou-se, com este trabalho, dar o primeiro passo na concepção da solução final, criando e implantando a ideia, e sugerindo os passos a seguir.

Sendo que o trabalho futuro passará pela migração do sistema para a plataforma Android ou ainda para uma outra plataforma móvel, o estudo e análise do custo/benefício entre as soluções de processamento remoto e processamento offline, assim como a conexão entre os dispositivos móveis e a plataforma e base de dados que conterà as matrículas desaparecidas.

Anexo A

Anexos

A.1 Código SCW

*O código de criação das janelas concêntricas deslizantes foi inspirado e adaptado de <https://lebgeeks.com/forums/viewtopic.php?id=7786> pelo utilizador *mesa177**

```
function Iand = SCWSegmentation(Img, M1, N1, TS)

I1=Img;

I1_ext = horzcat(I1(:,1:12), I1, I1(:,N1-10:N1));
I1_ext = vertcat(I1_ext(1:4,:), I1_ext, I1_ext(M1-2:M1,:));

[M1_ext, N1_ext] = size(I1_ext);

I1_SCW_ext=ones(size(I1_ext));

for i = 5:M1_ext-3
    for j = 13:N1_ext-11
        A_SCW = I1_ext(i-2:i+1, j-6:j+5);
        B_SCW = I1_ext(i-4:i+3, j-12:j+11);
        mA = mean(mean(A_SCW));
        mB = mean(mean(B_SCW));
        if (mA/mB < TS)
            I1_SCW_ext(i, j) = 1;
        else
            I1_SCW_ext(i, j) = 0;
        end
        clear A_SCW B_SCW mA mB
    end
end

I1_SCW = I1_SCW_ext(5:M1_ext-3, 13:N1_ext-11);
Iand=I1_SCW;
```

end

A.2 Código para threshold de Savola

*O código de criação das janelas concêntricas deslizantes foi inspirado e adaptado de <https://lebbeeks.com/forums/viewtopic.php?id=7786> pelo utilizador *mesa177**

```
function ISAU=sauvola (Img)
k=0.5;
R=128;
b=10;
I2=Img;
[Ms,Ns]=size(I2);

I2_ext = horzcat(I2(:,1:round(b/2)-1),I2,I2(:,Ns-round(b/2):Ns));
I2_ext = vertcat(I2_ext(1:round(b/2)-1,:),I2_ext,I2_ext(Ms-round(b/2):Ms,:));

[M1_ext,N1_ext] = size(I2_ext);

I2_SCW_ext=ones(size(I2_ext));

for i = 5:M1_ext-5
    for j = 5:N1_ext-5
        A2_Sau = I2_ext(i-4:i+5,j-4:j+5);
        mA = mean(A2_Sau(:));
        vA = var(A2_Sau(:));
        th = mA.*(1 + k * (vA / R-1));

        if (I2_ext(i,j) >= th)
            I2_SCW_ext(i,j) = 0;
        else
            I2_SCW_ext(i,j) = 1;
        end
        clear A_Sau mA Va th
    end
end
I2_SCW = I2_SCW_ext(5:M1_ext-5,5:N1_ext-5);

ISAU=I2_SCW;

end
```

A.3 Código para extração da matrícula por projeções

```

function [Imat, AL, BL, CL, DL] = extractLP(ImBW,Ilg)

Ilf=ImBW;
horizontalProjection = sum(Ilf, 2);
[~,maxHorzIndexValue]=max(horizontalProjection);

top = maxHorzIndexValue;
lower = maxHorzIndexValue;
threshold =std(horizontalProjection);% max_horz*k;

while(horizontalProjection(top,1)>threshold)
    top = top-1;
end
while(horizontalProjection(lower,1)>threshold)
    lower = lower+1;
end

lpSqPlus=0.05; %percentage of increase of the LP square area
horizontalBand = (round(top-top*lpSqPlus):round(lower+lower*lpSqPlus));
bwBand = Ilf(top:lower,:);
imagem=Ilg(top:lower,:);

verticalProjection = sum(bwBand, 1);

[~,maxVertIndexValue]=max(verticalProjection);

auxleft=maxVertIndexValue;
auxright=maxVertIndexValue;

while(sum(verticalProjection(1,(auxleft-50):auxleft))>50)
    auxleft=auxleft-1;
end
while(sum(verticalProjection(1,auxright:(auxright+50)))>50)
    auxright = auxright+1;
end

verticalBand = (round(auxleft-auxleft*lpSqPlus):round(auxright+auxright*lpSqPlus));

Imat=Ilg(horizontalBand,verticalBand);

```

```
AL = [top,auxleft];  
BL = [top,auxright];  
CL = [lower,auxright];  
DL = [lower,auxleft];  
  
end
```

A.4 Código para segmentação dos caracteres

```

function [LP,NoCh] = myocr(Imat)

Imat1=Imat;

threshold = graythresh(Imat1);
Imat1=medfilt2(Imat1,[3 3]);
Imat1 =~im2bw(Imat1,threshold);
Imat1 = bwareaopen(Imat1,35);
L=logical(Imat1);
rpmat=regionprops(L,'BoundingBox');
BBsizes=0;
for n=1:size(rpmat,1)
    if rpmat(n).BoundingBox(3)/rpmat(n).BoundingBox(4)<1
        auxsize=rpmat(n).BoundingBox(3)*rpmat(n).BoundingBox(4);
        BBsizes=BBsizes+auxsize;
    end
    clear auxsize
end
AvgSize=round(BBsizes/size(rpmat,1));
hold on
ii=1;
lpch=struct;
for n=1:size(rpmat,1)
    if (rpmat(n).BoundingBox(3)*rpmat(n).BoundingBox(4) > AvgSize) && (rpmat(n).BoundingBox(4) > AvgSize)
        rectangle('Position',rpmat(n).BoundingBox,'EdgeColor','g','LineWidth',2)

        lpch(ii).ch = Imat1(round(rpmat(n).BoundingBox(2):rpmat(n).BoundingBox(2)+rpmat(n).BoundingBox(3)),round(rpmat(n).BoundingBox(1):rpmat(n).BoundingBox(1)+rpmat(n).BoundingBox(3))); %save
        ii=ii+1;
    %    else
    %        rectangle('Position',rpmat(n).BoundingBox,'EdgeColor','r','LineWidth',2)
    end
    clear ch
end
hold off
LP=('');
for i=1:size(lpch,2)
    ch=lpch(i).ch;
    ch=imresize(ch,[42 24]);
    letter=read_letter(~ch);
    LP(i)=letter;
end

```

```
NoCh=size(lpch,2);  
end
```

A.5 Código para segmentação dos caracteres

```

function letter=read_letter(imagn)%#codegen

%Size of image must be 42 x 24 pixels

comp=[];
load templates
for n=1:36
    sem=corr2(templates{1,n},imagn);
    comp=[comp sem];
end
vd=find(comp==max(comp));
%*****
if vd==1
    letter='A';
elseif vd==2
    letter='B';
elseif vd==3
    letter='C';
elseif vd==4
    letter='D';
elseif vd==5
    letter='E';
elseif vd==6
    letter='F';
elseif vd==7
    letter='G';
elseif vd==8
    letter='H';
elseif vd==9
    letter='I';
elseif vd==10
    letter='J';
elseif vd==11
    letter='K';
elseif vd==12
    letter='L';
elseif vd==13
    letter='M';
elseif vd==14
    letter='N';
elseif vd==15
    letter='O';
elseif vd==16

```

```
        letter='P';
elseif vd==17
        letter='Q';
elseif vd==18
        letter='R';
elseif vd==19
        letter='S';
elseif vd==20
        letter='T';
elseif vd==21
        letter='U';
elseif vd==22
        letter='V';
elseif vd==23
        letter='W';
elseif vd==24
        letter='X';
elseif vd==25
        letter='Y';
elseif vd==26
        letter='Z';
        %*-*-*-*-*
elseif vd==27
        letter='1';
elseif vd==28
        letter='2';
elseif vd==29
        letter='3';
elseif vd==30
        letter='4';
elseif vd==31
        letter='5';
elseif vd==32
        letter='6';
elseif vd==33
        letter='7';
elseif vd==34
        letter='8';
elseif vd==35
        letter='9';
else
        letter='0';
end

end
```


Referências

- [1] Christos Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Vassili Loumos, e Eleftherios Kayafas. A license plate-recognition algorithm for intelligent transportation system applications. *7:377–392*, 2006.
- [2] Faculty Of e Information Technology. Principles of automatic number plate.
- [3] Christos-nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, e Eleftherios Kayafas. License plate recognition from still images and video sequences : A survey. *9:377–391*, 2008.
- [4] Theoretical Background. Border tracing algorithm. *Image Processing*, páginas 6–10.
- [5] Mohammadreza Asghari Oskoei. Introduction to computer vision & robotics. páginas 1–25, 2014.
- [6] Satadal Saha, Subhadip Basu, Mita Nasipuri, e Dipak Kumar Basu. License plate localization from vehicle images : An edge based multi-stage approach. *1(1):284–288*, 2009.
- [7] Hamid Mahini, Shohreh Kasaei, P O Box, Faezeh Dorri, e Fatemeh Dorri. An efficient features – based license plate localization method. páginas 18–21, 2006.
- [8] Vahid Abolghasemi e Alireza Ahmadyfard. Local enhancement of car image for license plate detection. *European Signal Processing Conference*, páginas 2179–2183, 2007.
- [9] Songke Li e Yixian Chen. License plate recognition. 2011.
- [10] Rafiqul Islam Ahmad Suleiman Othman Khalifa, Sheroz Khan. Malaysian vehicle licence plate recognition. *The International Arab Journal of Information Technology*, 4(4), 2007.
- [11] Lin Luo, Hao Sun, Weiping Zhou, e Limin Luo. An efficient method of license plate location, 2009.
- [12] Bai Hongliang Bai Hongliang e Liu Changping Liu Changping. A hybrid license plate extraction method based on edge statistics and morphology. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2:2–5, 2004.
- [13] Sorin Draghici. A neural network based artificial vision system for licence plate recognition. *International Journal of Neural Systems*, 8:113–126, 1997.
- [14] Yun-Chung Chung Shyang-Lih Chang, Li-Shien Chen e Sei-Wan Chen. Automatic license plate recognition. *IEEE Int. Conf Intelligent Transportation Systems*.
- [15] W. Zhao X. Shi e Y. Shen. Automatic license plate recognition system based on color image processing. *Lecture Notes on Computer Science.*, 3483:1159–1168, 2005.

- [16] Sunil Kumar, Rajat Gupta, Nitin Khanna, Santanu Chaudhury, e Shiv Dutt Joshi. Text extraction and document image segmentation using matched wavelets and mrf model. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 16:2117–2128, 2007.
- [17] B. Kurt F. Kahraman e M. Gökmen. License plate character segmentation based on the gabor transform and vector quantization. *Lecture Notes on Computer Science*, 2869:381–388, 2003.
- [18] J. Sauvola e M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.
- [19] Suprokash Dey. An efficient technique to recognize license plate using morphological edge detection and character matching algorithm. páginas 36–41, 2014.