Anabela Reigoto, *Estudante, FEUP*, André Oliveira, *Estudante, FEUP* e Baltasar Aroso, *Estudante, FEUP*

[[1]](#footnote-1)

Projeto de Processamento de Imagem e Classificação de Veículos - Carvana

*Sumário* - Este projeto consistiu no desenvolvimento e teste da manipulação de imagens fornecidas por um stand de automóveis, Carvana, através de métodos de pré-processamento de imagem, segmentação, análise e reconhecimento e classificação para o projeto final da unidade curricular Sistemas Baseados em Visão.

*Palavras-Chave* – Segmentação, Classificação, Reconhecimento.

# Introdução

E

ste relatório tem como objetivo complementar o código efetuado para o projeto final da Unidade Curricular Sistemas Baseados em Visão, elucidando a sua natureza mais teórica, visto que esta poderá não ter sido bem detalhada na apresentação.

Este trabalho tinha como primeiro objetivo a implementação de um algoritmo em MATLAB capaz de segmentar as imagens apresentadas para que se assemelhassem às mascaras *ground-truth* fornecidas, sendo o seu termo de comparação o cálculo da eficiência através do coeficiente de *Sørensen–*Dice.

Numa segunda fase, era pedida a classificação dos veículos nas suas possíveis classes: Compacto, Pick-up, Sedan ou SUV; sendo possível recorrer, para este efeito, às suas máscaras *ground-truth*. Para a realização desta tarefa havia livre-arbítrio para escolher qual a melhor disposição do carro para que o algoritmo fosse mais eficiente. Posto isto, após uma análise das imagens fornecidas escolhemos a vista lateral.

A última tarefa tinha como objetivo a classificação dos veículos na sua marca correspondente, dentro das 7 marcas possíveis: Mini, BMW, Chevrolet, Ford, Honda, Jaguar e Audi. Para a realização deste algoritmo escolhemos a disposição traseira dos automóveis, com a exceção de um Mini *Clubman* cujo símbolo da marca se encontrava no canto inferior esquerdo da traseira, o que impossibilitava a sistematização do código que procurava o símbolo na parte central da imagem, tendo sido por isso usada a disposição frontal neste caso.

# Segmentação

Numa primeira abordagem, tentou-se remover o letreiro CARVANA recorrendo a uma operação de **close** com um elemento estruturante “linha”, horizontal, grande o suficiente para que estas desaparecessem completamente (150), tentando-se de seguida proceder à *edge detection* (deteção de orla) com a função **edge**, usando o algoritmo de *Canny* e um *threshold* adequado. Os resultados ficaram aquém dos esperados, visto que, após a remoção das letras, o fundo apresentava bastantes mudanças súbitas na intensidade da cor, semelhante à imagem ter sido esboçada perfeitamente na horizontal, devido ao elemento estruturante escolhido.

Numa segunda abordagem, e percebendo desta vez que a imagem requeria algum pré-processamento mais elaborado antes de se tentar a deteção de orla, decidiu-se aplicar a operação de **close**, desta vez com uma linha ligeiramente mais curta (120), de modo a remover em grande parte as letras, mas não danificando em demasia as orlas do carro. De seguida, após aplicar um filtro gaussiano para remoção de algum ruído, efetuou-se a soma de todas as intensidades, em cada coluna da matriz obtida da leitura da imagem.[[2]](#footnote-2) Isto teve como objetivo a análise de máximos e mínimos para que se pudesse tentar encontrar as coordenadas limite do veículo, seja esse qual for. Este procedimento procura “encontrar” um retângulo que limite o veículo, tornando tudo à volta deste completamente uniforme (escolheu-se a cor cinza).

Após uma primeira observação de máximos e mínimos, encontrou-se um problema: mesmo que se recorresse a uma remoção quase completa das letras (elemento estruturante ‘linha’ de comprimento maior), as transições abruptas do fundo branco para as letras escuras permaneciam. A solução que se provou bastante eficaz consistiu numa medição das larguras de letras relevantes (‘C’, ‘R’ e ‘N’: letras que possuem fortes traços verticais, ou quase verticais), ainda na análise de máximos e mínimos. Verificou-se que, mesmo para fotografias com diferente distância à câmara fotográfica, essas larguras se encontravam em 2 intervalos restritos[[3]](#footnote-3) e facilmente detetáveis, uma vez que correspondiam sempre a um máximo precedido de um mínimo (claro; escuro; claro). A sua remoção torna-se, portanto, trivial: procuram-se os máximos e mínimos (guardados separadamente) e, sempre que houver um máximo com um mínimo precedido a uma distância aproximadamente igual à largura de uma letra, ambos são removidos. São ainda removidos todos os máximos e mínimos abaixo de um certo nível, relativo ao maior máximo. Neste momento, o algoritmo encontra-se pronto para determinar o limite esquerdo e o limite direito do veículo através do respetivo pico (máximo ou mínimo) extremo. Para encontrar os limites superior e inferior usou-se um procedimento semelhante, sendo que para estes foi possível desprezar o impacto das letras fazendo a operação de **close** com o elemento estruturante ‘linha’, comprimento 150, e aplicando um filtro gaussiano mais potente.

Tendo agora as dimensões da caixa que delimita o carro, aproximadamente, e aplicando-lhe uma margem de segurança obtida através de experimentação, colocou-se o fundo exterior a este retângulo todo a cinza. Aplica-se um algoritmo de deteção de orla de *Canny* e remove-se a orla detetada em volta do retângulo (visto que é indesejada e facilmente removível). Através da transformada de *Hough*, podemos detetar segmentos de reta na imagem correspondentes à orla das letras e a traços verticais irrelevantes, tendo em conta a sua inclinação e definindo um tamanho bom o suficiente para não apanhar traços importantes do carro, mas que apanhe orlas de letra, dentro de um limite inferior razoável. Utilizou-se o tamanho 50 e os ângulos descritíveis nos vetores [-40:-34 34:40 -26:-18 18:26] (diagonais) e [-3:3] (verticais).[[4]](#footnote-4)

No fim, com vista a “limpar” e preencher o segmento do veículo obtido, aplicou-se uma operação binária de **thin**, seguida de um **close**, depois **fill**, e por fim um **open**.

O procedimento acima descrito encontra-se dentro do script **task1.m**, na secção *Segmentation*. Esta secção efetua a segmentação de todos os veículos em modo automático, sendo que no final nos indica a eficiência média obtida através da média dos coeficientes de *Sørensen–Dice* individuais. O script faz uso de 3 funções: **startup.m** (recebe as imagens de todos os carros em escala de cinza), **getGroundTruths.m** (recebe todos os ground-truths) e **bresenham.m**. Esta última foi obtida através da biblioteca pública de funções MATLAB e retorna todos os pontos de um segmento de reta limitado pelos pontos passados como argumentos, usando uma versão do algoritmo de *Bresenham* otimizada para MATLAB (sem *loops*).

Para o cálculo dos coeficientes de *Sørensen–Dice* individuais usou-se:

(1)

Em (1), *TP* são os verdadeiros positivos, *FP* os falsos positivos e *FN* os falsos negativos.

Em anexo encontram-se os valores obtidos para a eficiência e gráficos que permitem uma boa análise quantitativa detalhada da mesma. A eficiência média obtida foi de PERCENTAGEM%.

# Classificação dos veículos nas suas Classes

Os veículos da base de dados apresentada encontram-se separados em quatro classes: compactos, pick-ups, sedans e SUVs. A classificação dos veículos nas suas classes rege-se essencialmente pelas suas características dimensionais, sendo do senso comum normal de analisar que um veículo das classes pick-up e SUV é maior do que um veículo das outras duas classes. Porém, essas classificações são inconsistentes e ambíguas. Pela análise das imagens da base de dados podemos verificar que a distância dos veículos à câmara de onde foi tirada a fotografia é variável. Posto isto e para garantir uma correta classificação dos dados, fez-se uma análise pormenorizada de todas as características que, na nossa opinião, seriam preponderantes para a completa caracterização dos veículos.

Numa primeira fase, os parâmetros analisados foram: a altura, a largura, o perímetro dado pelo retângulo que delimitava a altura e a largura do veículo e a área desse mesmo retângulo (box[[5]](#footnote-5)). Foi calculado o valor máximo, mínimo e médio de todos estes parâmetros, como é apresentado na Tabela 1.

Perante a análise visual das imagens e dos resultados tabelas foi escolhida apenas a vista lateral para a classificação dos veículos.

Primeiramente procurou-se distinguir os veículos com base nas características mais intrínsecas à sua classe, como por exemplo a distinção das Pick-ups pela altura da sua suspensão ou a separação dos SUVs pela extensão da sua “cúpula” face à sua largura. Para isto foram usadas as funções **suspension.m** e **decision\_suspension.m** para a distinção das pick-ups e a função **decision\_dome.m** para a distinção dos SUVs.

A função **suspension.m** calcula o tamanho da suspensão mínima e máxima de cada classe considerando a distância entre o último pixel branco no eixo central da imagem e a altura delimitada pelos pneus do automóvel. Após a análise dos resultados apresentados na Tabela 2 verificou-se que o tamanho mínimo da suspensão dos Pick-ups era superior a todos os outros e que havia uma grande discrepância entre o tamanho da suspensão dos Pick-ups e SUVs face aos Compactos e Sedans. Posto isto, o algoritmo já conseguia separar primeiro as Pick-ups e depois as SUVs. Sendo o último parâmetro de desempate entre os Compacto se os Sedans determinado pela relação dos seus eixos vertical e horizontal, altura e largura, respetivamente. Portanto, pela divisão da altura pela sua largura observou-se, como seria de esperar dado o extenso comprimento dos Sedans, que o valor mínimo desta divisão na classe Compactos era superior ao valor máximo na classe Sedans, concluindo desta maneira a classificação de todos os veículos nas quatro classes.

Porém e devido ao facto dos valores utilizados para a separação do SUV usando o **decision\_dome.m** terem sido inseridos com base em tentativas e a margem de distinção ser muito pequena, uma outra opção para separar as SUVs seria o cálculo da sua área, usando a função do MATLAB **regionprops**, e inserindo como argumento ‘*Area’*.

Após a utilização e exploração da função **regionprops** e tendo em conta que os métodos que estavam a ser usados eram muito restritos à precisão da segmentação, procurou-se alternativas. Para isso aumentou-se a análise dos dados como é apresentado na Tabela 3.

Depois de uma análise detalhada dos dados obtidos e procurando criar uma árvore de decisão robusta aos erros da segmentação foram calculadas as margens de decisão tendo em conta os seis parâmetros apresentados na Tabela 4.

Perante as combinações nessa tabela assinaladas criou-se quatro algoritmos diferentes para a determinação da classe dos veículos. Todos os algoritmos apresentaram uma eficiência de 100% usando as imagens *ground-truth* na vista lateral.

Na utilização do algoritmo para a classificação das imagens obtidas com a segmentação da tarefa 1 obtivemos as eficiências apresentadas na Tabela 5.

# MATH

## Equations

(1)

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (*T* might refer to temperature, but T is the unit tesla). Refer to “(1),” not “Eq. (1)” or “equation (1),” except at the beginning of a sentence: “Equation (1) is ... .”

# Units

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). This applies to papers in data storage**.** For example, write “15 Gb/cm2 (100 Gb/in2).” An exception is when English units are used as identifiers in trade, such as “3½-in disk drive.” Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength *H* is A/m. However, if you wish to use units of T, either refer to magnetic flux density *B* or magnetic field strength symbolized as µ0*H*. Use the center dot to separate compound units, e.g., “A·m2.”

# Some Common Mistakes

The word “data” is plural, not singular. The subscript for the permeability of vacuum µ0 is zero, not a lowercase letter “o.” The term for residual magnetization is “remanence”; the adjective is “remanent”; do not write “remnance” or “remnant.” Use the word “micrometer” instead of “micron.” A graph within a graph is an “inset,” not an “insert.” The word “alternatively” is preferred to the word “alternately” (unless you really mean something that alternates). Use the word “whereas” instead of “while” (unless you are referring to simultaneous events). Do not use the word “essentially” to mean “approximately” or “effectively.” Do not use the word “issue” as a euphemism for “problem.” When compositions are not specified, separate chemical symbols by en-dashes; for example, “NiMn” indicates the intermetallic compound Ni0.5Mn0.5 whereas “Ni–Mn” indicates an alloy of some composition NixMn1-x.

Be aware of the different meanings of the homophones “affect” (usually a verb) and “effect” (usually a noun), “complement” and “compliment,” “discreet” and “discrete,” “principal” (e.g., “principal investigator”) and “principle” (e.g., “principle of measurement”). Do not confuse “imply” and “infer.”

Prefixes such as “non,” “sub,” “micro,” “multi,” and “ultra” are not independent words; they should be joined to the words they modify, usually without a hyphen. There is no period after the “et” in the Latin abbreviation “*et al.*” (it is also italicized). The abbreviation “i.e.,” means “that is,” and the abbreviation “e.g.,” means “for example” (these abbreviations are not italicized).

A general IEEE styleguide is available at [www.ieee.org/authortools](http://www.ieee.org/authortools).



Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

TABLE I

Units for Magnetic Properties

|  |  |  |
| --- | --- | --- |
| Symbol | Quantity | Conversion from Gaussian and  CGS EMU to SI a |
| Φ | magnetic flux | 1 Mx → 10−8 Wb = 10−8 V·s |
| *B* | magnetic flux density,  magnetic induction | 1 G → 10−4 T = 10−4 Wb/m2 |
| *H* | magnetic field strength | 1 Oe → 103/(4π) A/m |
| *m* | magnetic moment | 1 erg/G = 1 emu  → 10−3 A·m2 = 10−3 J/T |
| *M* | magnetization | 1 erg/(G·cm3) = 1 emu/cm3  → 103 A/m |
| 4π*M* | magnetization | 1 G → 103/(4π) A/m |
| σ | specific magnetization | 1 erg/(G·g) = 1 emu/g → 1 A·m2/kg |
| *j* | magnetic dipole  moment | 1 erg/G = 1 emu  → 4π × 10−10 Wb·m |
| *J* | magnetic polarization | 1 erg/(G·cm3) = 1 emu/cm3  → 4π × 10−4 T |
| χ*,* κ | susceptibility | 1 → 4π |
| χρ | mass susceptibility | 1 cm3/g → 4π × 10−3 m3/kg |
| μ | permeability | 1 → 4π × 10−7 H/m  = 4π × 10−7 Wb/(A·m) |
| μr | relative permeability | μ → μr |
| *w, W* | energy density | 1 erg/cm3 → 10−1 J/m3 |
| *N, D* | demagnetizing factor | 1 → 1/(4π) |

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

aGaussian units are the same as cg emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

# Guidelines for Graphics Preparation and Submission

## Types of Graphics

The following list outlines the different types of graphics published in IEEE journals. They are categorized based on their construction, and use of color / shades of gray:

### *Color/Grayscale figures*

### Figures that are meant to appear in color, or shades of black/gray. Such figures may include photographs, illustrations, multicolor graphs, and flowcharts.

### *Line Art figures*

### Figures that are composed of only black lines and shapes. These figures should have no shades or half-tones of gray, only black and white.

### *Author photos*

### Head and shoulders shots of authors that appear at the end of our papers.

### *Tables* Data charts which are typically black and white, but sometimes include color.

## Multipart figures

Figures compiled of more than one sub-figure presented side-by-side, or stacked. If a multipart figure is made up of multiple figure types (one part is lineart, and another is grayscale or color) the figure should meet the stricter guidelines.

## File Formats For Graphics

Format and save your graphics using a suitable graphics processing program that will allow you to create the images as PostScript (PS), Encapsulated PostScript (.EPS), Tagged Image File Format (.TIFF), Portable Document Format (.PDF), or Portable Network Graphics (.PNG) sizes them, and adjusts the resolution settings. If you created your source files in one of the following programs you will be able to submit the graphics without converting to a PS, EPS, TIFF, PDF, or PNG file: Microsoft Word, Microsoft PowerPoint, or Microsoft Excel. Though it is not required, it is strongly recommended that these files be saved in PDF format rather than DOC, XLS, or PPT. Doing so will protect your figures from common font and arrow stroke issues that occur when working on the files across multiple platforms. When submitting your final paper, your graphics should all be submitted individually in one of these formats along with the manuscript.

## Sizing of Graphics

Most charts, graphs, and tables are one column wide (3.5 inches / 88 millimeters / 21 picas) or page wide (7.16 inches / 181 millimeters / 43 picas). The maximum depth a graphic can be is 8.5 inches (216 millimeters / 54 picas). When choosing the depth of a graphic, please allow space for a caption. Figures can be sized between column and page widths if the author chooses, however it is recommended that figures are not sized less than column width unless when necessary.

There is currently one publication with column measurements that do not coincide with those listed above. Proceedings of the IEEE has a column measurement of 3.25 inches (82.5 millimeters / 19.5 picas).

The final printed size of author photographs is exactly   
1 inch wide by 1.25 inches tall (25.4 millimeters x 31.75 millimeters / 6 picas x 7.5 picas). Author photos printed in editorials measure 1.59 inches wide by 2 inches tall (40 millimeters x 50 millimeters / 9.5 picas x 12 picas).

## Resolution

The proper resolution of your figures will depend on the type of figure it is as defined in the “Types of Figures” section. Author photographs, color, and grayscale figures should be at least 300dpi. Line art, including tables should be a minimum of 600dpi.

## Vector Art

In order to preserve the figures’ integrity across multiple computer platforms, we accept files in the following formats: .EPS/.PDF/.PS. All fonts must be embedded or text converted to outlines in order to achieve the best-quality results.

## Color Space

The term color space refers to the entire sum of colors that can be represented within the said medium. For our purposes, the three main color spaces are Grayscale, RGB (red/green/blue) and CMYK (cyan/magenta/yellow/black). RGB is generally used with on-screen graphics, whereas CMYK is used for printing purposes.

All color figures should be generated in RGB or CMYK color space. Grayscale images should be submitted in Grayscale color space. Line art may be provided in grayscale OR bitmap colorspace. Note that “bitmap colorspace” and “bitmap file format” are not the same thing. When bitmap color space is selected, .TIF/.TIFF/.PNG are the recommended file formats.

## Accepted Fonts Within Figures

When preparing your graphics IEEE suggests that you use of one of the following Open Type fonts: Times New Roman, Helvetica, Arial, Cambria, and Symbol. If you are supplying EPS, PS, or PDF files all fonts must be embedded. Some fonts may only be native to your operating system; without the fonts embedded, parts of the graphic may be distorted or missing.

A safe option when finalizing your figures is to strip out the fonts before you save the files, creating “outline” type. This converts fonts to artwork what will appear uniformly on any screen.

## Using Labels Within Figures

### Figure Axis labels

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity “Magnetization,” or “Magnetization *M*,” not just “*M*.” Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write “Magnetization (A/m)” or “Magnetization (Am−1),” not just “A/m.” Do not label axes with a ratio of quantities and units. For example, write “Temperature (K),” not “Temperature/K.”

Multipliers can be especially confusing. Write “Magnetization (kA/m)” or “Magnetization (103 A/m).” Do not write “Magnetization (A/m) × 1000” because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 10 point type.

### Subfigure Labels in Multipart Figures and Tables

Multipart figures should be combined and labeled before final submission. Labels should appear centered below each subfigure in 8 point Times New Roman font in the format of (a) (b) (c).

## File Naming

Figures (line artwork or photographs) should be named starting with the first 5 letters of the author’s last name. The next characters in the filename should be the number that represents the sequential location of this image in your article. For example, in author “Anderson’s” paper, the first three figures would be named ander1.tif, ander2.tif, and ander3.ps.

Tables should contain only the body of the table (not the caption) and should be named similarly to figures, except that ‘.t’ is inserted in-between the author’s name and the table number. For example, author Anderson’s first three tables would be named ander.t1.tif, ander.t2.ps, ander.t3.eps.

Author photographs should be named using the first five characters of the pictured author’s last name. For example, four author photographs for a paper may be named: oppen.ps, moshc.tif, chen.eps, and duran.pdf.

If two authors or more have the same last name, their first initial(s) can be substituted for the fifth, fourth, third... letters of their surname until the degree where there is differentiation. For example, two authors Michael and Monica Oppenheimer’s photos would be named oppmi.tif, and oppmo.eps.

## Referencing a Figure or Table Within Your Paper

When referencing your figures and tables within your paper, use the abbreviation “Fig.” even at the beginning of a sentence. Do not abbreviate “Table.” Tables should be numbered with Roman Numerals.

## Checking Your Figures: The IEEE Graphics Analyzer

The IEEE Graphics Analyzer enables authors to pre-screen their graphics for compliance with IEEE Transactions and Journals standards before submission. The online tool, located at <http://graphicsqc.ieee.org/>, allows authors to upload their graphics in order to check that each file is the correct file format, resolution, size and colorspace; that no fonts are missing or corrupt; that figures are not compiled in layers or have transparency, and that they are named according to the IEEE Transactions and Journals naming convention. At the end of this automated process, authors are provided with a detailed report on each graphic within the web applet, as well as by email.

For more information on using the Graphics Analyzer   
or any other graphics related topic, contact the IEEE Graphics Help Desk by e-mail at [graphics@ieee.org](mailto:graphics@ieee.org).

## Submitting Your Graphics

Because IEEE will do the final formatting of your paper,   
you do not need to position figures and tables at the top and bottom of each column. In fact, all figures, figure captions, and tables can be placed at the end of your paper. In addition to, or even in lieu of submitting figures within your final manuscript, figures should be submitted individually, separate from the manuscript in one of the file formats listed above in section VI-J. Place figure captions below the figures; place table titles above the tables. Please do not include captions as part of the figures, or put them in “text boxes” linked to the figures. Also, do not place borders around the outside of your figures.

# Conclusion

## A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Appendix

Appendixes, if needed, appear before the acknowledgment.

References

*Basic format for books:*

J. K. Author, “Title of chapter in the book,” in *Title of His Published Book, x*th ed. City of Publisher, (only U.S. State), Country: Abbrev. of Publisher, year, ch. *x*, sec. *x*, pp. *xxx–xxx.*

*Examples:*

1. G. O. Young, “Synthetic structure of industrial plastics,” in *Plastics,* 2nd ed., vol. 3, J. Peters, Ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.
2. W.-K. Chen, *Linear Networks and Systems.* Belmont, CA, USA: Wadsworth, 1993, pp. 123–135.

*Basic format for computer programs and electronic documents (when available online):*

Legislative body. Number of Congress, Session. (year, month day). *Number of bill or resolution*, *Title*. [Type of medium]. Available: site/path/file

***NOTE:*** ISO recommends that capitalization follow the accepted practice for the language or script in which the information is given.

*Example:*

1. U.S. House. 102nd Congress, 1st Session. (1991, Jan. 11). *H. Con. Res. 1, Sense of the Congress on Approval of Military Action*. [Online]. Available: LEXIS Library: GENFED File: BILLS

*Basic format for the most common types of unpublished references:*

a) J. K. Author, private communication, Abbrev. Month, year.

b) J. K. Author, “Title of paper,” unpublished.

c) J. K. Author, “Title of paper,” to be published.

*Examples:*

1. A. Harrison, private communication, May 1995.
2. B. Smith, “An approach to graphs of linear forms,” unpublished.
3. A. Brahms, “Representation error for real numbers in binary computer arithmetic,” IEEE Computer Group Repository, Paper R-67-85.

1. [↑](#footnote-ref-1)
2. Todas as imagens lidas pelo algoritmo em MATLAB foram, aquando da sua leitura, convertidas de RGB para escala de cinza (rgb2gray) e para dupla precisão (im2double). [↑](#footnote-ref-2)
3. Presentes no código MATLAB nas variáveis *lw\_sup* e *lw\_inf*, que contêm, respetivamente, os limites superiores e inferiores desses intervalos. Os valores que serviram de guia para esses intervalos encontram-se acima dessas variáveis, comentados. [↑](#footnote-ref-3)
4. Os ângulos das letras foram medidos com recurso à ferramenta de visualização de imagem do MATLAB e através de conhecimento básico matemático, analisando pontos dessas mesmas letras. [↑](#footnote-ref-4)
5. Este será o nome utilizado para a área do retângulo calculada pelo produto da altura com a largura em contraste com a área posteriormente mencionada como a soma de todos os pixéis brancos na imagem do *ground-truth*. [↑](#footnote-ref-5)