

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES, RAMO: TELECOMUNICAÇÕES, ELETRÓNICA E COMPUTADORES, ESPECIALIZAÇÃO: REDES E SERVIÇOS DE COMUNICAÇÃO

|  |
| --- |
| PROJETO DE PROCESSAMENTO DE IMAGEM E CLASSIFICAÇÃO DE VEÍCULOS - CARVANA |

**Unidade Curricular de Sistemas Baseados em Visão**

**Professora Ana Maria Mendonça**

**Professora Teresa Araújo**

**Professor Guilherme Aresta**

Anabela Machado Reigoto, 201405662

André Duarte Correia de Oliveira, 201405639

Baltasar de Vasconcelos Dias Aroso, 201404125

4MIEEC\_T\_RC

**Dezembro 2017**

Índice

[1. Sumário 4](#_Toc500586720)

[2. Introdução 5](#_Toc500586721)

[2. Tarefa 1 - Segmentação 6](#_Toc500586722)

[2.1. A 6](#_Toc500586723)

[3. Tarefa 2 – Classificação dos veículos nas 4 classes: compacto, pick-up, sedan e SUV 7](#_Toc500586724)

[4. Tarefa 3 – Classificação dos veículos nas 7 marcas: Mini, BMW, Chevrolet, Ford, Honda, Jaguar e Audi 8](#_Toc500586725)

[5. Caracterização Estatística da Eficiência 9](#_Toc500586726)

[9.1. Variar FER (Frame Error Rate) 9](#_Toc500586727)

[9.2. Variar o tempo de propagação 10](#_Toc500586728)

[9.3. Variar C (capacidade de ligação, bit/s) 11](#_Toc500586729)

[9.4. Variar o tamanho da trama 12](#_Toc500586730)

[10. Conclusões 13](#_Toc500586731)

[10. Anexos 14](#_Toc500586732)

[I. Suporte da Caracterização Estatística de Eficiência 14](#_Toc500586733)

[I.1. Variar FER (Frame Error Rate) 14](#_Toc500586734)

[II. Código Fonte 15](#_Toc500586735)

1. 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 projecto final da unidade curricular Sistemas Baseados em Visão.

2. Introdução

Este 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 código 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 *Sørensen–Dice*.

Numa segunda fase, era pedido a classificação dos veículos nas suas possíveis classes: compacto, pick-up, sedan ou SUV; através das suas máscaras *ground-truth*. Para a realização desta tarefa tínhamos livre-arbítrio para escolhermos qual a melhor disposição do carro para que o algoritmo fosse mais eficiente. Posto isto e 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.

2. Tarefa 1 - Segmentação

A

2.1. A

A

3. Tarefa 2 – Classificação dos veículos nas 4 classes

A

4. Tarefa 3 – Classificação dos veículos nas 7 marcas

A

5. Caracterização Estatística da Eficiência

5.1. Variar FER (Frame Error Rate)

Para tal foi criada a função ***randomError()*** que é chamada sempre que é lida uma trama válida, portanto na função ***llread()***. Esta introduz, tendo em conta uma percentagem FER pré-definida, aleatoriamente, um erro na trama em qualquer campo, exceto nas flags que sinalizam o início e o fim de trama.

**Valores pré-definidos:**

**Baudrate:** 115200; **Tamanho da trama:** 256; **Ficheiro:** penguin.gif

Figura 1 - Eficiência versus Frame Error Rate

void randomError **(**unsigned char **\***buffer**,** int buffer\_size**)** **{**

int indice **=** 0**,** err **=** 0**;**

struct timeval micros**;**

gettimeofday**(&**micros**,** **NULL);**

srand**(**micros**.**tv\_usec**);**

err **=** rand**()** **%** 101**;** //n de 0 a 100 que corresponde a percentagem de erro

**if** **(**err **<** FER**)** **{**

**do** **{**

gettimeofday**(&**micros**,** **NULL);**

srand**(**micros**.**tv\_usec**);**

indice **=** rand**()** **%** **(**buffer\_size **-** 3**)** **+** 1**;**

**}** **while(**buffer**[**indice**]** **==** 0x7D **||** buffer**[**indice**]** **==** 0x7E **||** buffer**[**indice**]** **==** 0x5D **||** buffer**[**indice**]** **==** 0x5E**);**

buffer**[**indice**]** **=** 0x00**;**

**}**

**}** //randomError()

5.2. Variar o tempo de propagação

Para tal foi usada a função ***usleep()*** inserida em ***llread()*** imediatamente após a leitura da trama da porta. Esta função tem como objetivo aumentar o tempo de propagação simulando o aumento do cabo de ligação da porta série aos computadores. Os valores usados foram passados por parâmetro à função em microssegundos.

**Valores pré-definidos:**

**Baudrate:** 115200; **Tamanho da trama:** 1024; **Ficheiro:** penguin.gif

Figura 2 - Eficiência versus Tempo de Propagação

10. Conclusões

Em suma, os objetivos pretendidos foram atingidos com sucesso. O grau de exigência deste trabalho face aos conhecimentos que tínhamos para a sua execução foi compatível com a nossa vontade de aprender e a proatividade demonstrada nos Elementos de Valorização. A extensa pormenorização do nosso trabalho neste relatório, onde fazemos questão de explicar todos os passos realizados, deve-se às limitações que nos foram aparecendo à medida que íamos testando e limando infindáveis vezes o nosso programa, pelo se julga que se pode afirmar que o protocolo de transmissão de dados implementado ficou bem compreendido. Os 99,7% de eficiência apresentados na Caracterização Estatística da Eficiência corroboram esse progresso académico e contribuem para a nossa satisfação na conclusão deste projeto.

10. Anexos

I. Suporte da Caracterização Estatística de Eficiência

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **i** | **FER**  **Teórico** | **FER**  **Real** | **L\_TX (bytes)** | **L\_RX (bytes)** | **L\_total (bytes)** | **Time DataLink (ns)** | **Time DataLink (s)** | **S(1)** | **S(i)** |
| **1** | 0,00% | 0,0000% | 230 | 11631 | 11861 | 1061413751 | 1,0614 | 0,9512 | 0,9700 |
| **3** | 10,00% | 4,1667% | 240 | 11971 | 12211 | 1094556581 | 1,0946 | 0,9224 | 0,9684 |
| **4** | 10,00% | 6,1224% | 245 | 12420 | 12665 | 1134834048 | 1,1348 | 0,8897 | 0,9688 |
| **2** | 10,00% | 8,0000% | 250 | 12562 | 12812 | 1148626166 | 1,1486 | 0,8790 | 0,9682 |
| **6** | 20,00% | 19,2982% | 285 | 14528 | 14813 | 1318030524 | 1,3180 | 0,7660 | 0,9756 |
| **7** | 20,00% | 22,0339% | 295 | 14865 | 15160 | 1357424921 | 1,3574 | 0,7438 | 0,9695 |
| **5** | 20,00% | 23,3333% | 300 | 15329 | 15629 | 1399166542 | 1,3992 | 0,7216 | 0,9696 |
| **9** | 30,00% | 25,8064% | 310 | 15852 | 16162 | 1448795477 | 1,4488 | 0,6969 | 0,9684 |
| **8** | 30,00% | 28,1250% | 320 | 16370 | 16690 | 1496629006 | 1,4966 | 0,6746 | 0,9680 |
| **10** | 30,00% | 33,3333% | 345 | 17586 | 17931 | 1609575631 | 1,6096 | 0,6273 | 0,9670 |
| **13** | 40,00% | 37,8378% | 370 | 19014 | 19384 | 1738442930 | 1,7384 | 0,5808 | 0,9679 |
| **12** | 40,00% | 41,0256% | 390 | 19628 | 20018 | 1794727251 | 1,7947 | 0,5626 | 0,9682 |
| **11** | 40,00% | 46,5116% | 430 | 21994 | 22424 | 2011354725 | 2,0114 | 0,5020 | 0,9678 |
| **14** | 50,00% | 47,7273% | 440 | 22597 | 23037 | 2064738678 | 2,0647 | 0,4890 | 0,9685 |
| **16** | 50,00% | 51,5789% | 475 | 24165 | 24640 | 2207184316 | 2,2072 | 0,4574 | 0,9691 |
| **15** | 50,00% | 52,5773% | 485 | 24529 | 25014 | 2240724116 | 2,2407 | 0,4506 | 0,9690 |
| **17** | 60,00% | 53,5354% | 495 | 23940 | 24435 | 2192928025 | 2,1929 | 0,4604 | 0,9672 |
| **18** | 60,00% | 57,0093% | 535 | 27164 | 27699 | 2484573284 | 2,4846 | 0,4064 | 0,9677 |
| **19** | 60,00% | 61,0169% | 590 | 29693 | 30283 | 2712208374 | 2,7122 | 0,3723 | 0,9692 |
| **21** | 70,00% | 65,1515% | 660 | 33731 | 34391 | 3079672282 | 3,0797 | 0,3278 | 0,9694 |
| **22** | 70,00% | 68,9189% | 740 | 37515 | 38255 | 3431342562 | 3,4313 | 0,2942 | 0,9678 |
| **20** | 70,00% | 76,6497% | 985 | 51009 | 51994 | 4662649140 | 4,6626 | 0,2165 | 0,9680 |
| **23** | 80,00% | 77,6699% | 1030 | 51906 | 52936 | 4743385523 | 4,7434 | 0,2129 | 0,9687 |
| **24** | 80,00% | 79,1855% | 1105 | 57441 | 58546 | 5243947432 | 5,2439 | 0,1925 | 0,9691 |
| **25** | 80,00% | 80,2575% | 1165 | 60383 | 61548 | 5502511944 | 5,5025 | 0,1835 | 0,9710 |
| **27** | 90,00% | 85,6698% | 1605 | 83139 | 84744 | 7585836369 | 7,5858 | 0,1331 | 0,9697 |
| **26** | 90,00% | 87,9896% | 1915 | 100139 | 102054 | 9135415749 | 9,1354 | 0,1105 | 0,9697 |
| **28** | 90,00% | 90,0000% | 2300 | 119964 | 122264 | 10950911077 | 10,9509 | 0,0922 | 0,9692 |
| **29** | 95,00% | 94,2999% | 4035 | 208761 | 212796 | 19052641187 | 19,0526 | 0,0530 | 0,9695 |
| **30** | 100,00% | 96,9313% | 7495 | 393479 | 400974 | 35923401853 | 35,9234 | 0,0281 | 0,9689 |
| **31** | 100,00% | 97,6410% | 9750 | 507983 | 517733 | 46379998807 | 46,3800 | 0,0218 | 0,9690 |

I.1. Variar FER (Frame Error Rate)

II. Código Fonte