



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia Eletrónica
e Telecomunicações e de Computadores

Processamento de Imagem e Visão

Trabalho Prático 1

André Rodrigues, nº39085



Docente:

Professor Pedro M. Jorge

Novembro de 2018

Índice

1.	Introdução.....	5
1.1.	Objectivos	6
2.	Desenvolvimento.....	7
2.1.	Segmentação	7
2.2.	Operadores Morfológicos	9
2.3.	Extracção de componentes conexos	10
2.4.	Extracção de características	10
2.5.	Classificação	11
3.	Resultados do conjunto de teste	12
4.	Conclusões	14
5.	Bibliografia.....	15
6.	Anexos.....	16

Índice de Figuras

2-1 Imagem de teste	7
2-2 Histograma de uma imagem de teste	7
2-3 Imagem depois da erosão	9
2-4 Imagem original.....	9
2-5 Imagem depois da dilatação	9
2-6 Imagem etiquetada.....	10
2-7 Imagem classificada	11

Índice de Tabelas

Tabela 1 Áreas reais das moedas.....	12
Tabela 2 Áreas das moedas do anexo 1.....	12
Tabela 3 Áreas das moedas do anexo 2.....	12
Tabela 4 Áreas das moedas do anexo 3.....	12
Tabela 5 Áreas das moedas do anexo 4.....	13
Tabela 6 Áreas das moedas do anexo 5.....	13
Tabela 7 Áreas das moedas do anexo 6.....	13
Tabela 8 Áreas das moedas do anexo 7.....	13
Tabela 9 Áreas das moedas do anexo 8.....	13
Tabela 10 Áreas das moedas do anexo 9.....	13

1. Introdução

Nos dias de hoje, a correlação entre os 3 componentes *RGB* torna o processamento de imagens pesado que é proporcional ao tamanho da imagem. A Binarização é meramente um factor de simplificação do problema, em que consiste na tradução de valores de pixel descritos por um conjunto de componentes, *RGB*, *GrayScale*, entre outros, em apenas dois valores, 0 (Preto) e 255 (branco).

O uso de operadores morfológicos numa determinada imagem é uma ferramenta bastante interessante aplicada a uma imagem que serve para modificar a informação de modo a apresentar a informação mais importante para o processamento futuro. Existem vários operadores morfológicos, mas todos eles derivam apenas de dois: erosão e dilatação. Estas operações são implementadas com recurso a um elemento estruturante. Este elemento é representado por uma imagem binária com a forma desejada para melhor servir o propósito da operação.

Depois da identificação dos objetos, é possível extrair as suas características. Estas características são propriedades geométricas como a área, centroide, perímetro, circularidade e muitos mais. Através deste tipo de características é possível identificar e distinguir objetos.

Assim que se saibam as características dos objetos, então é aplicada a classificação dos objetos, que serve para identificar elementos de interesse de uma imagem. Este classificador é tanto melhor quanto mais exato for na separação e identificação.

No presente trabalho prático pretende-se implementar um algoritmo que seja capaz de contar automaticamente a quantia de dinheiro colocada num plano. O resultado é obtido através de análises de imagens de moedas que estão dispostas numa determinada superfície, sob condições semelhantes. Para a construção deste algoritmo é necessário o conhecimento da biblioteca *OpenCV*. Para a realização deste algoritmo, foi fornecido um conjunto de imagens de vários tipos de moedas, em cima de uma superfície homogénea e clara. Este algoritmo deve ser robusto o suficiente para que se consiga lidar com a presença de objetos diferentes de moedas.

1.1. Objectivos

- a. Desenvolver algoritmo de visão por computador, capaz de contar automaticamente a quantia em dinheiro (moedas), colocado em cima de uma mesa;
- b. Familiarização com a biblioteca de funções *OpenCV* (*Open Source Computer Vision*) para programação de aplicações de visão por computador em tempo real (para linguagem de programação *Python*)

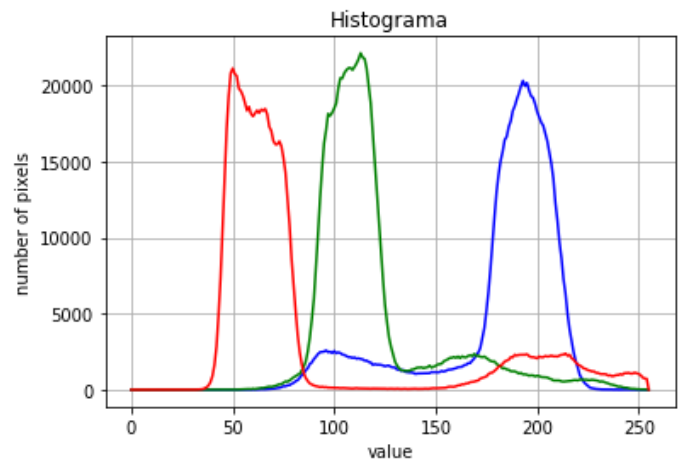
2. Desenvolvimento

2.1. Segmentação

Para começar, verificou-se qual o melhor componente RGB, e chegou-se à conclusão que a melhor componente a trabalhar seria a componente “Red”, apesar de termos mais pixels azuis, existem mais pixels do canal “Red” activos, tendo assim uma linha com dois picos mais separados, o que leva a uma melhor diferenciação dos objectos com o fundo. Esta decisão foi tomada apartir dos histogramas das diferentes imagens originais.



2-1 Imagem de teste



2-2 Histograma de uma imagem de teste

Primeiro foi feita uma suavização da imagem com um filtro gaussiano, “cv2.GaussianBlur()”, que é um filtro passa-baixo que remove os componentes de alta frequência. E depois foi então utilizada a função threshold. O threshold é o método de segmentação mais simples. Separa as regiões de uma imagem correspondente aos objetos que queremos analisar. Essa separação é baseada na variação de intensidade entre os *pixels* do objeto e os *pixels* do plano de fundo. Para diferenciar os *pixels* dos quais estamos interessados (os quais serão eventualmente rejeitados), é realizada uma comparação de cada valor de intensidade de pixel em relação a um limiar.

Uma vez que tenhamos separado adequadamente os pixels importantes, é possível defini-los com um determinado valor para identificá-los (ou seja, é possível atribuir-lhes um valor de preto, branco ou qualquer valor).

Como temos uma imagem imagem bimodal, que é uma imagem cujo histograma tem dois picos, é possível obter aproximadamente um valor no meio desses picos como limiar, e isso é o que a binarização Otsu faz. Ou seja, ele calcula automaticamente o valor(limiar) de separação das duas classes do histograma da imagem para uma imagem bimodal.

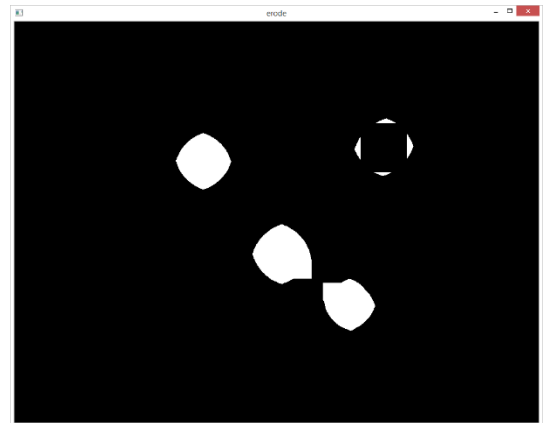
Nesta parte também se poderia ter optado por colocar a imagem em tons de cinzento, mas não era tão assertivo pois as moedas mais pequenas acabavam por desaparecer porque tinha ocorrido muita erosão. Ou seja, acabava-se por ter que se ajustar os operadores morfológicos às novas características da imagem.

2.2. Operadores Morfológicos

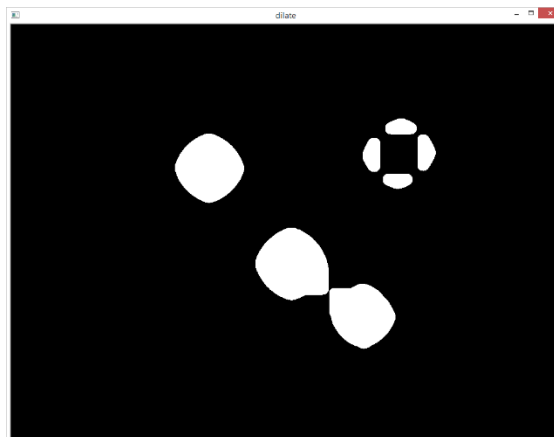
Após a realização da Binarização, foram então aplicados os operadores morfológicos de forma a delimitar melhor cada objecto, dado que em algumas regiões referentes a moedas diferentes se encontram conexas. Na Figura 2-4 **Erro! A origem da referência não foi encontrada.** será possível verificar que dois objectos que estavam unidos, ficaram bem delimitados. Para tal, aplicou-se uma erosão com um determinado elemento estruturante MORPH_CLOSE, Figura 2-3isto para ser possível saber se a região perto do centro do objecto seria a moeda e a região que estivesse mais longe do objecto seria o fundo, seguido de uma dilatação (Figura 2-5 Imagem depois da dilatação) com outro elemento estruturante MORPH_ELLIPSE pelo facto da moeda se aproximar mais de uma elipse, para tornar o elemento um pouco maior, uma vez que a erosão o tornou mais pequeno.



2-4 Imagem original



2-3 Imagem depois da erosão



2-5 Imagem depois da dilatação

2.3. Extracção de componentes conexos

Para a realização da classificação de objectos, é necessário analisar de modo a diferenciá-los uns dos outros. Para obter os componentes conexos usa-se uma função designada por *findContours*, sendo a sua função a recolha de informação sobre todos os contornos. A Figura 2-6 mostra a Figura 2-4 depois das regiões serem etiquetadas.



2-6 Imagem etiquetada

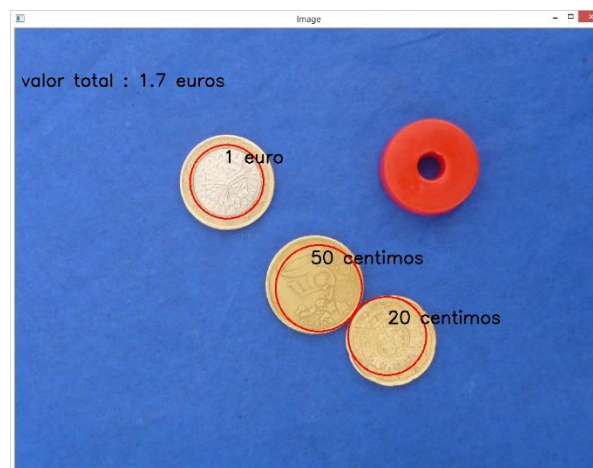
2.4. Extracção de características

Após a análise dos resultados obtidos, foi possível obter características como a área, perímetro, circularidade e os momentos. Mas toda a classificação acabou por ser feita apenas a partir da área.

2.5. Classificação

Depois da obtenção das propriedades geométricas, e com dos resultados obtidos foi possível classificar cada objecto. Foi implementado um dicionário onde contém os nomes das moedas e as respectivas áreas que foram obtidas, através do conjunto de imagens de treino. Para saber qual a moeda em questão foi implementado um método designado por *desc_moeda*, que recebe como parâmetro a área do objecto e verifica no dicionário se essa área está contida na estrutura de dados. Caso se verificar a existência dessa moeda, devolve o nome da mesma.

E por fim a quantia final é representada na imagem como mostra a Figura 2-7.



2-7 Imagem classificada

3. Resultados do conjunto de teste

Na tabela 1 temos os valor de refência das moedas na realidade, e na tabelas de 2 a 5 temos os resultados para todas as imagens de teste. É possível observar que de imagem para imagem a diferença é pequena, no entanto deverá ser possível uma melhor acertividade na característica da área.

Tabela 1 Áreas reais das moedas

Propriedade	2€	1€	0.50€	0.20€	0.10€	0.05€	0.02€	0.01€
Área(pixéis)	14997.88	12034.48	13092.09	11021.66	8683.93	10053.07	7826.74	5878.77

Tabela 2 Áreas das moedas do anexo 1

Propriedade	1€	0.50€	0.20€	0.10€	0.05€	0.02€	0.01€
Área(pixéis)	11311.0	12800.0	9872.5/10274.0	7079.0	8714.5	6363.5	3855.0

Tabela 3 Áreas das moedas do anexo 2

Propriedade	1€	0.20€	0.10€	0.02€	0.01€
Área(pixéis)	11322.0	10031.0	7125.0	6276.5	3721.0

Tabela 4 Áreas das moedas do anexo 3

Propriedade	0.50€	0.20€	0.05€	0.02€	0.01€
Área(pixéis)	12916.0	10035.5	8589.5	6230.0	3712.5

Tabela 5 Áreas das moedas do anexo 4

Propriedade	0.50€	0.10€	0.05€	0.02€
Área(pixéis)	12879.0	7098.5	8552.0	6222.5

Tabela 6 Áreas das moedas do anexo 5

Propriedade	1€	0.50€	0.10€	0.02€
Área(pixéis)	11345.0	12661.0	6821.5	5754

Tabela 7 Áreas das moedas do anexo 6

Propriedade	1€	0.50€	0.10€	0.05€	0.02€
Área(pixéis)	11450.5	12901.5	7260.5	9112.5	6687.5

Tabela 8 Áreas das moedas do anexo 7

Propriedade	1€	0.20€	0.10€
Área(pixéis)	11432.5	9952.0	7155.0

Tabela 9 Áreas das moedas do anexo 8

Propriedade	1€	0.20€	0.10€
Área(pixéis)	11427.5	9842.0	7162.0

Tabela 10 Áreas das moedas do anexo 9

Propriedade	1€	0.50€	0.20€
Área(pixéis)	11576.0	13555.0	10795.0

4. Conclusões

Ao longo da realização deste trabalho, apareceram alguns problemas a nível de operadores morfológicos, pois quando existiam moedas colocadas era preciso alterar alguns parâmetros dos operadores. Por vezes os valores introduzidos na dilatação e na erosão eram suficientes para chegar a bons resultados nas imagens com as moedas soltas, no entanto acabava por arruinar a classificação nas imagens quando elas estavam juntas.

Outro aspecto importante é o uso exclusivo do *threshold*, sendo que se não fossem usufruídos operadores morfológicos, isto é, usar unicamente o threshold (binarização), a imagem ficava com bastante ruído, o que originava um resultado bastante diferente do pretendido.

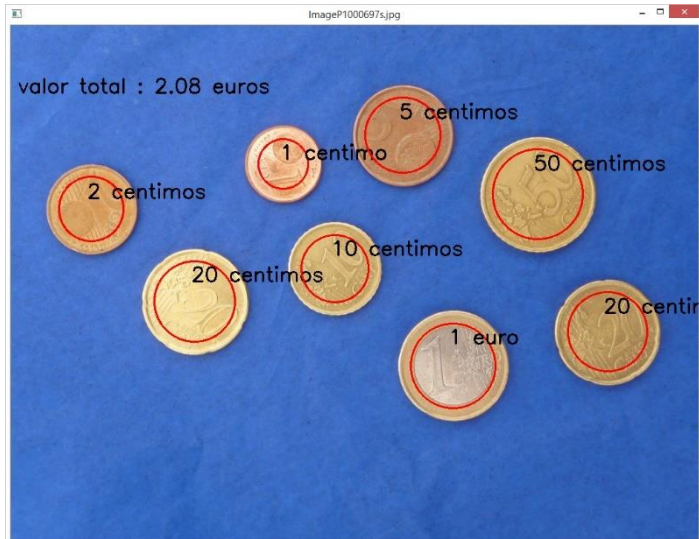
É de salientar que a implementação deste algoritmo está dependente das condições de obtenção da imagem sobre o qual se faz a extracção de características. Este algoritmo não é robusto ao nível de alterações de luminosidade e posição da câmara.

5. Bibliografia

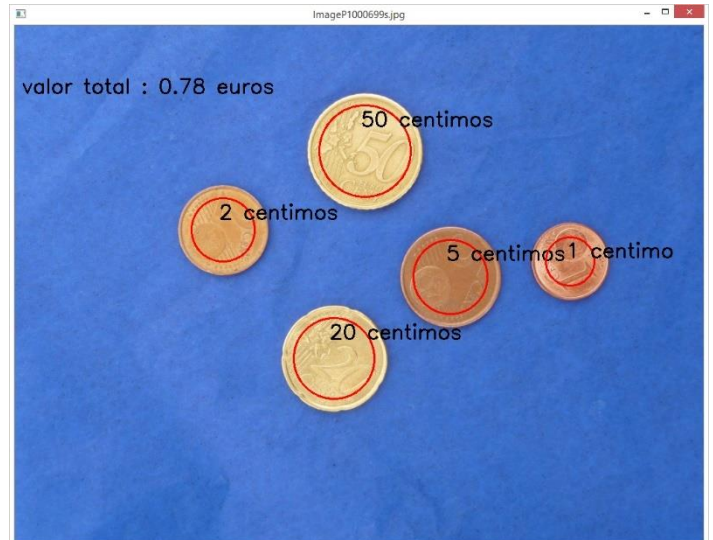
- <http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>
- http://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html
- <http://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html>
- <http://stackoverflow.com/questions/29810128/opencv-python-set-background-colour>
- Slides fornecidos para Unidade Curricular

6. Anexos

Anexo 1



Anexo 3



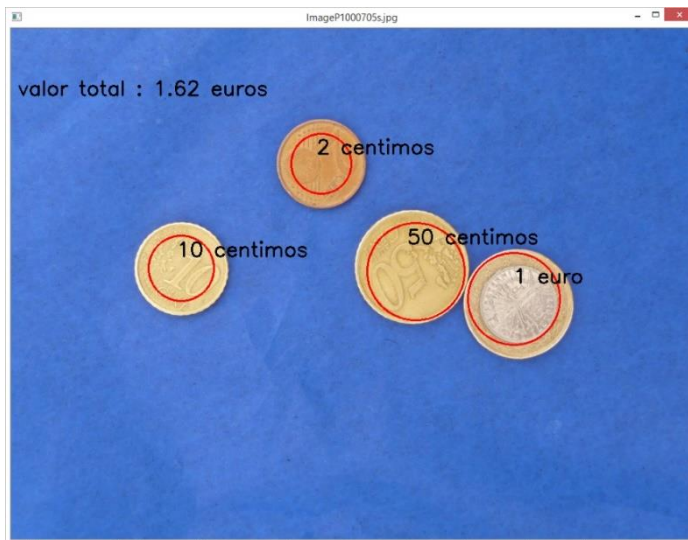
Anexo 2



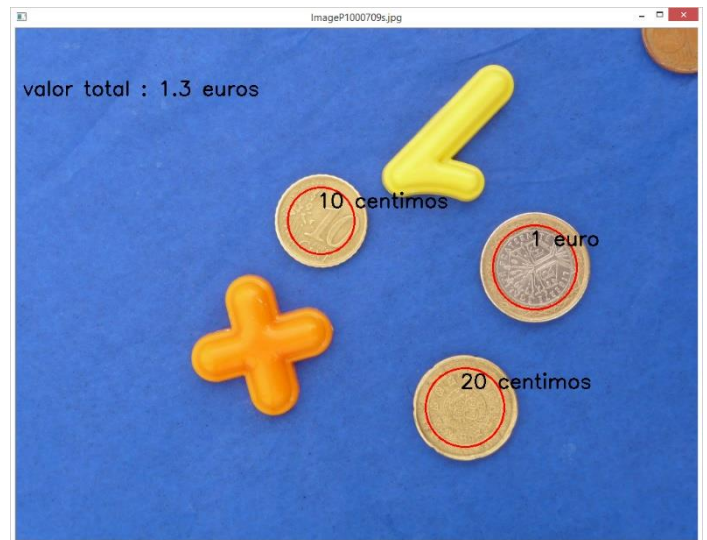
Anexo 4



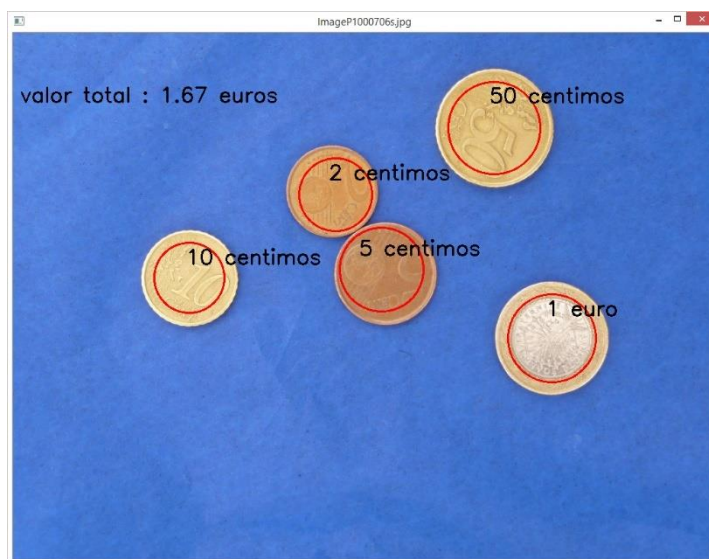
Anexo 5



Anexo 7



Anexo 6



Anexo 8



Anexo 9

