



ADEETC
Processamento de Imagem e Visão
1º Semestre 2017/2018

Docente: Pedro Mendes Jorge

Relatório do Trabalho Prático Nº1

Rui Santos nº 39286

5 de Novembro de 2017

Contents

1	Introdução	3
2	Desenvolvimento	4
2.1	Leitura da Imagem	4
2.2	Tratamento da Imagem	5
2.2.1	Aplicação de filtro	5
2.2.2	Binarização da Imagem	6
2.2.3	Erosão e Dilatação	7
2.2.4	Contornos	8
2.3	Classificação de moedas	9
3	Conclusão	10

List of Figures

1	Níveis RGB	4
2	Canal R.	4
3	Canal B.	4
4	Imagem com filtro	5
5	Imagem Binarizada	6
6	Imagem de erro	6
7	Erodida	7
8	Original	7
9	Contornos da imagem	8
10	Contornos	9
11	Classificação	9

1 Introdução

Este trabalho destina-se a aprofundar conhecimentos adquiridos na unidade curricular de PIV.

Desenvolver algoritmo de visão por computador, capaz de contar automaticamente a quantia em dinheiro (moedas), colocado em cima de uma mesa;

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)

Descrição

Pretende-se desenvolver um algoritmo para contagem da quantia em dinheiro (moedas de euro), colocado em cima de uma mesa de superfície homogénea e clara, observada por uma câmara, montada num tripé, ajustado de modo a que o plano do sensor é paralelo ao plano da mesa.

O algoritmo deverá possuir alguma robustez relativamente às seguintes perturbações: (i) presença de objectos, diferentes de moedas, no campo de visão; (ii) existência de pequenas sombras; (iii) eventual contacto dos objectos.

Serão fornecidos exemplos de imagens de treino que podem ser usadas para o desenvolvimento do algoritmo.

O algoritmo será avaliado usando um conjunto de imagens de teste, diferentes das de treino, mas adquiridas nas mesmas condições.

2 Desenvolvimento

2.1 Leitura da Imagem

Para a realização de experiências que confirmem as necessidades do algoritmo a implementar, foram fornecidas 9 imagens diferentes. De forma a conseguir efetuar os cálculos de reconhecimento de padrões, é necessário a escolha de quais os melhores valores presentes na imagem a analisar, valores esse que são provenientes dos canais RGB.

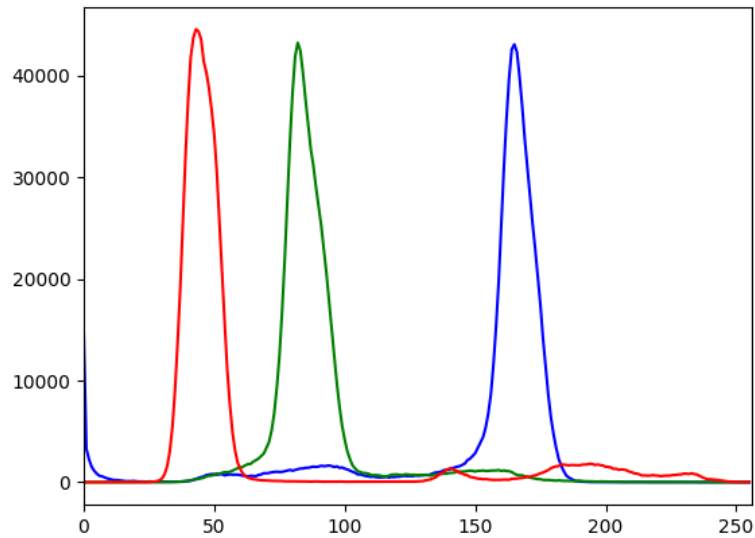


Figure 1: Níveis RGB

Podemos reparar que os níveis de azul são bastante elevados, pois existe um factor em comum em todas as imagens fornecidas, a existência de um fundo azul, logo para uma análise mais precisa sobre os contornos dos objectos na imagem iremos utilizar o canal Red. Em seguida um exemplo de uma imagem do canal R e outra do canal B.



Figure 2: Canal R.

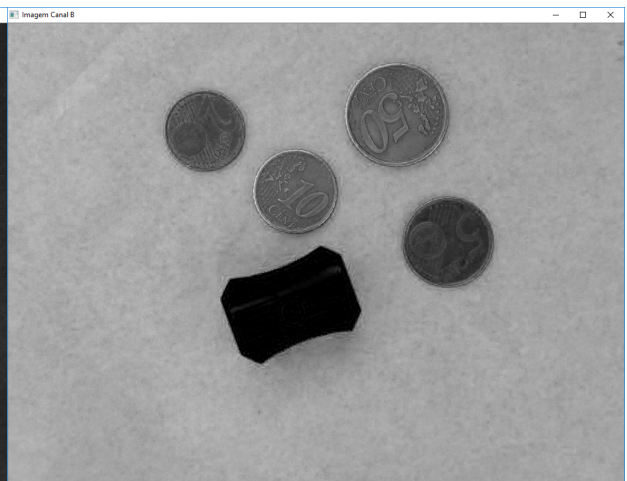


Figure 3: Canal B.

2.2 Tratamento da Imagem

2.2.1 Aplicação de filtro

Tendo em conta as imagens anteriormente analisadas, podemos concluir que nos processos que se seguem, iremos ter alguns problemas caso as imagens tenha algumas imperfeições, por isso iremos aplicar um filtro de forma a aplicar um certo "smooth" para assim a descoberto dos contornos seja mais fácil.

Em seguida um exemplo do filtro aplicado:



Figure 4: Imagem com filtro

Ao aplicar o smooth nem sempre resolvemos todos os problemas que possivelmente poderemos ter. O facto de os níveis de qualquer canal RGB que queiramos analisar poderem ter valores que na seguinte fase de tratamento da imagem, podem induzir ao algoritmo de binarização em erro e assim obtermos resultados não esperados.

Um desses resultados passa por, a moeda poder ter uma parte mais escura e assim ser classificado como ausência de luz.

Assim sendo, aplicamos um *BrightEnhance* em que damos mais brilho à imagem, e assim os valores serem mais sensíveis ao algoritmo.

2.2.2 Binarização da Imagem

Através da operação anterior de aplicação de filtro, podemos finalmente binarizar a imagem, em que a consiste na atribuição de valores 0 e 1 baseando nos valores que neste caso o canal R apresenta. O fundo aparece preto, pois como dito anteriormente, existe uns níveis de azul bastante elevados no que toca ao fundo da imagem, portanto se estamos analisar os níveis do canal R sabemos que iremos ter valores muito próximos de 0, fazendo com que o algoritmo de binarização *OTSU* o classifique como 0.

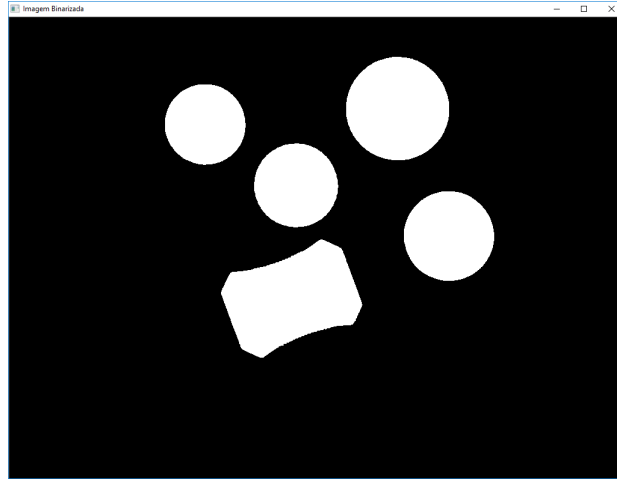


Figure 5: Imagem Binarizada

No entanto, mesmo tendo objectivo de reduzir os problemas com a aplicação do *BrightEnhance*, podem surgir outros problemas. Um deles passa por haver moedas ou outros objectos que estejam muito próximos um do outro, ou até mesmo a tocarem-se. Tendo isto em consideração, sabemos que podemos ter imagens enganadoras, tais como a seguinte:

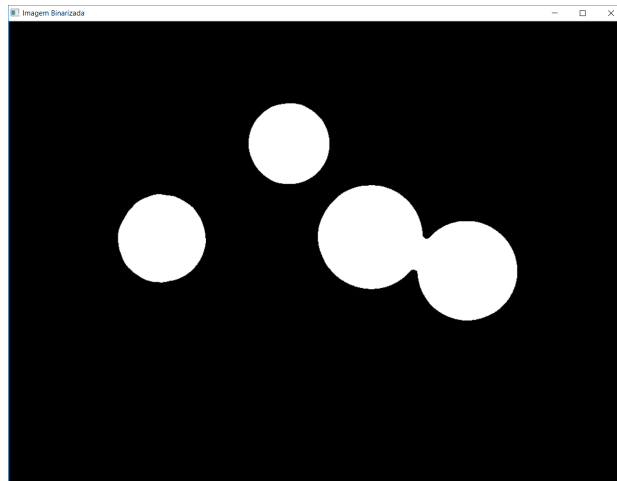


Figure 6: Imagem de erro

E por causa de estas ocasiões que existe a fase seguinte de tratamento da imagem.

2.2.3 Erosão e Dilatação

Como referido na fase anterior, o nosso filtro nem sempre consegue identificar como dois objectos diferentes tornando assim mais tarde a classificação de moedas um processo complicado, pois iremos chegar a uma fase que queremos descobrir contornos das partes que estão em branco, e se as duas moedas estão juntas como na imagem anterior, irão ser classificadas como apenas um objecto quando na verdade são duas moedas distintas. Para evitar esse problema existem duas operações morfológicas, *Erosão* e *Dilatação*, em que ambas consistem em usar um *Elemento Estruturante* a cada pixel de contorno, sendo que a dilatação da imagem, neste caso, iria aumentar o numero de pixeis a branco, e a erosão da imagem pixeis a preto.

Para resoluções no nosso algoritmo, erodimos em algumas iterações e com um elemento estruturante duas vezes superior ao elemento estruturante de dilatação, para assim ficarmos com tamanhos possiveis de classificação e as moedas não irão estar mais juntas.

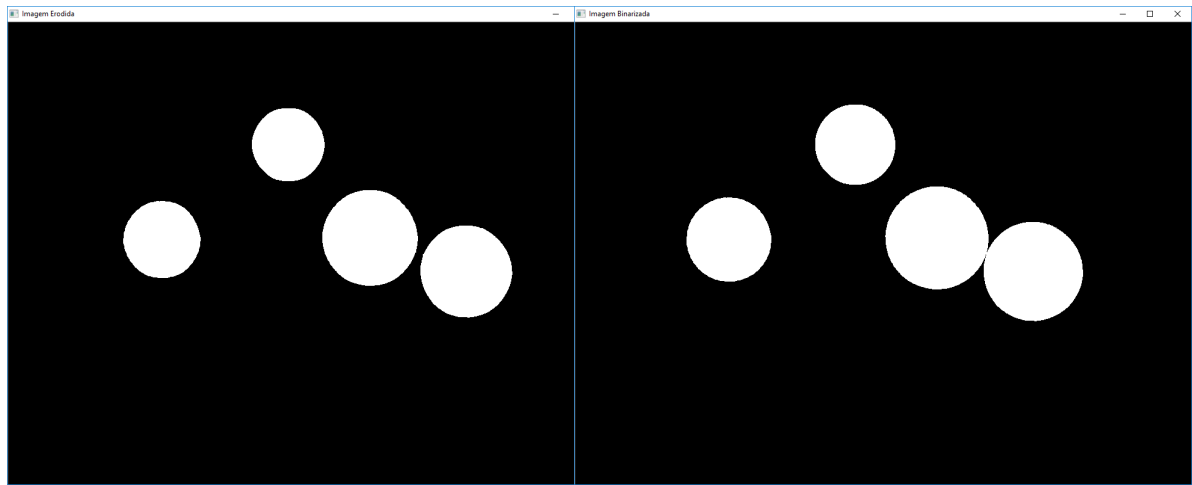


Figure 7: Erodida

Figure 8: Original

Este elemento estruturante que se falou, pode ter varias formas, neste caso foi escolhida uma elipse pois é a forma que faz sentido ter neste trabalho.

2.2.4 Contornos

Como falado anteriormente, este é o processo mais importante no algoritmo a desenvolver, pois é nesta parte que realmente encontramos as áreas de contorno dos objectos que queremos classificar como moedas ou não.



Figure 9: Contornos da imagem

Podemos verificar que encontramos um contorno de um objecto que não nos interessa para a classificação de moedas.

O de forma a conseguirmos determinar se o objecto que encontramos é de facto um círculo correspondente a uma moeda, temos de descobrir o seu centroide e verificar se todos os pontos que o algoritmo determinou pertencem aquele objecto são todos equidistantes ao centro.

Em primeira instância com a ajuda de uma função que descobre a área do objecto e consequentemente, dá-nos o valor em falta para o calculo do raio, podendo após este calculo se possivel uma comparação com a distância do ponto do objecto ao seu centroide, se todos os pontos se verificarem dentro de um limite fixo, considera-se que é um círculo.

No entanto mesmo após esta operação sobre os pontos que constituem os contornos achados, eles mesmo ainda podem ser considerados como moedas, pois estamos a verificar o valor das mesmas através das suas áreas. Para isso iremos utilizar uma expressão que calcula a circularidade de um objecto.

$$circularidade = (4 * \pi * Area) / Perimetro^2$$

Caso o valor esteja muito proximo de 1, podemos admitir de que se trata de um objecto circular, caso contrário é descartado.

2.3 Classificação de moedas

Esta classificação torna-se um processo fácil após todas as operações anteriores. É delimitado por nós um intervalo de áreas correspondente a cada moeda, sendo apenas preciso o cálculo da área de cada objecto considerado válido. Em baixo podemos ver um exemplo do output da nossa consola e a imagem analisada.

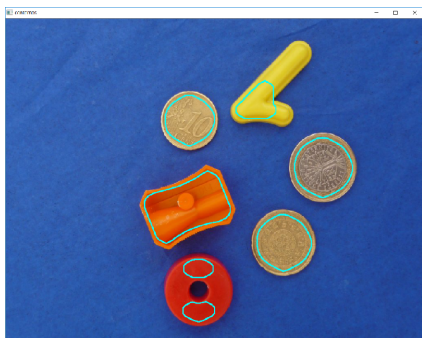


Figure 10: Contornos

```
C:\Users\Denga\Desktop\ISEL\PIV\TP1>python tp1.py  
Valor final: 1.3
```

Figure 11: Classificação

3 Conclusão

Neste trabalho foram encontradas algumas dificuldades na compreensão de certa operações necessárias para que o classificador ficasse com as caracterisiticas necessárias. Ainda que todos os processos fossem respeitados pela minha parte, notou-se que em algumas imagens nem sempre fora excluida alguns contornos indesejados, induzindo assim falsas contagens e não permitindo que a avaliação final fosse a correta.

Foram encontrados alguns problemas sobre que valores utilizar, ao inicio fora experimnentada a leitura em nvieis de cinzento, mas verificou-se ser dificil a classificação de moedas dando valores bastante discrepantes do numeros de moedas que se verificava na imagem, por isso meso se optou por utilizar os valores do canal R da imagem.

Em primeira instância tambem se verificou que tinha que ser elimanada a opção de contornos interiores pois interferiam com a classificação final das moedas, e por isso recorreu-se a um dos retornos da função que descobria os contornos, Hierarquia, que através de um valor específico (-1) indicava quando é que estava presente na imagem um contorno interior ou não.