

# Visão por Computador

## Tutorial 2

João Pimentel (a80874)

Carolina Cunha (a80142)

Janeiro de 2020

Universidade do Minho  
Mestrado Integrado em Engenharia Informática

## Resumo

O presente trabalho baseou-se no desenvolvimento de funções capazes de segmentar imagens, independentemente da existência de iluminação não uniforme, oclusão ou outros tipos de problemas, bem como a deteção de moedas através da transformada de Hough. Para tal, recorreu-se à linguagem de programação *Matlab*, a qual possui uma vasta biblioteca de ferramentas para processamento de imagem.

A elaboração deste projeto permitiu aumentar a capacidade de desenvolvimento e compreensão dos algoritmos referentes a processamento de imagem e, conseqüentemente, retirar conclusões sobre o impacto causado pelos diversos passos inerentes ao processo de segmentação de uma imagem.

Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Análise e Especificação</b>	<b>2</b>
2.1	Descrição do Projeto . . . . .	2
2.2	Especificação de Requisitos . . . . .	2
<b>3</b>	<b>Concepção da Resolução</b>	<b>3</b>
3.1	Introdução de Ruído . . . . .	3
3.2	Pré-Processamento . . . . .	4
3.3	Segmentação . . . . .	5
3.3.1	Imagem 1 . . . . .	5
3.3.2	Imagem 2 . . . . .	7
3.3.3	Imagem 3 . . . . .	8
3.4	Deteção de Moedas . . . . .	9
3.4.1	Moedas por Tipo . . . . .	11
3.5	Influência do Ruído no Resultado Final . . . . .	11
3.6	Erros e Limitações . . . . .	12
<b>4</b>	<b>Conclusões e Trabalho Futuro</b>	<b>13</b>

Lista de Figuras

1	Imagem sem ruído . . . . .	3
2	Comparação de ruído <i>salt and pepper</i> (esquerda) com gaussiano (direita). . . . .	3
3	Ruído gaussiano com variância 0.1 . . . . .	4
4	Imagem suavizada com filtro mediana . . . . .	4
5	Tratamento de iluminação não uniforme . . . . .	5
6	Tratamento de iluminação não uniforme com vários operadores morfológicos . . . . .	5
7	Aplicação do detetor de <i>edges</i> Canny à primeira imagem . . . . .	6
8	<i>Closing</i> com elemento estruturante circular . . . . .	6
9	Segmentação final da primeira imagem . . . . .	6
10	Aplicação do detetor de <i>edges</i> Canny à segunda imagem . . . . .	7
11	Aplicação de operadores morfológicos para conectar objetos . . . . .	7
12	Segmentação final da segunda imagem . . . . .	8
13	Resultado da aplicação de um <i>threshold</i> adaptativo . . . . .	8
14	Remoção de objetos com menos de 60 píxeis . . . . .	9
15	Segmentação final da terceira imagem . . . . .	9
16	Deteção de moedas na primeira imagem . . . . .	10
17	Deteção de moedas na segunda imagem . . . . .	10
18	Deteção de moedas na terceira imagem . . . . .	10
19	Histogramas de dimensões e contagem de tipos . . . . .	11
20	Segmentações da segunda e terceira imagens . . . . .	11
21	Segmentações primeira imagem sem ruído e com ruído, respetivamente . . . . .	12

# 1 Introdução

O presente relatório é o resultado da resolução do segundo trabalho prático da unidade curricular de Visão por Computador.

O foco deste trabalho passou por implementar um método capaz de detetar moedas em variadas imagens, cada uma com um problema diferente. Para tal, teve-se por base a utilização das ferramentas de processamento de imagem fornecidas pela linguagem *Matlab*, bem como métodos de segmentação e de deteção de características mencionados nas aulas da unidade curricular.

Este projeto baseou-se na implementação de duas grandes fases que deveriam aplicar um determinado algoritmo consoante os problemas associados à imagem de entrada no programa. É de destacar que a função que faz todo o tratamento necessário recebe uma lista de parâmetros específicos, retornando, também, um conjunto de valores.

Dito isto, o grande objetivo deste projeto foi o aumento da experiência dos alunos no âmbito do processamento de imagem. Quer isto dizer que se pretendia conseguir o desenvolvimento de ferramentas que implementassem, passo a passo, algoritmos capazes de segmentar, com grande exatidão, as imagens, de modo a facilitar a deteção de moedas, independentemente da disposição das mesmas, das características de fundo, ou outros fatores que possam influenciar esta deteção.

Nos capítulos seguintes serão demonstrados os problemas, formas de resolução e testes efetuados de modo a obter os resultados ideais para o problema proposto.

## 2 Análise e Especificação

### 2.1 Descrição do Projeto

O projeto em questão consistiu na realização de duas grandes tarefas. A primeira consistia na segmentação de uma imagem, ou seja, diferenciar o fundo dos objetos que se pretendem analisar. Para isso, a norma é colocar o fundo a preto e os objetos a branco, permitindo a fácil análise da imagem. Já a segunda consistia na deteção de moedas, ou seja, circunferências, na imagem segmentada a partir do método à escolha, sendo que neste caso se optou pela transformada de Hough.

### 2.2 Especificação de Requisitos

O principal objetivo deste tutorial foi contar o número de moedas numa determinada imagem, sendo que, para tal, são necessários realizar certos passos. Assim, serão apresentados os requisitos associados à resolução do problema, de forma detalhada.

1. A função deve receber uma imagem em tons de cinzento, um tipo de ruído e os parâmetros a este associado.
2. Ruído gaussiano pode ser definido pela sua média e variância, já ruído *salt and pepper* é definido pela sua densidade.
3. Deve ser introduzido ruído à imagem de entrada, tendo em conta o tipo e os parâmetros passados como argumento.
4. Deve ser efetuado um pré-processamento da imagem, sendo este dependente de cada imagem, já que cada uma possui as suas características.
5. Em seguida, deve ser realizada a segmentação da imagem, de modo a facilitar a deteção das moedas.
6. A partir da imagem segmentada deve ser possível aplicar um método que calcule o número de moedas na imagem.
7. Deve ser apresentado um histograma dos tamanhos dos objetos detetados, ou seja, os seus raios, ou áreas.
8. O programa deve devolver a imagem segmentada, com e sem ruído para efeitos de comparação, o valor de *Signal to Noise Ratio* para confirmar o ruído introduzido na imagem, a imagem pré-processada e o número e tipos das moedas presentes na imagem.

### 3 Conceção da Resolução

De modo a executar e descrever de forma simples e direta todo o processo de resolução do enunciado, foi definido que se explicaria todo este, passo a passo, demonstrando todos os resultados obtidos.

#### 3.1 Introdução de Ruído

De modo a aplicar ruído a uma imagem, foi utilizada uma função fornecida pela linguagem, em que esta recebe o tipo de ruído, bem como os parâmetros a este associados. Assim, através da utilização da função *imnoise*, é facilmente obtida uma imagem ruidosa a partir de uma imagem em tons de cinzento.

Na Figura 1 é possível observar a figura original, antes de ser inserido qualquer ruído.



Figura 1 - Imagem sem ruído.

Em seguida, tenha-se em consideração a Figura 2 onde, na imagem da esquerda, foi aplicado ruído do tipo *salt and pepper* com densidade de 0.02 e, na da direita, ruído gaussiano, onde a média e a variância tomam valores de 0 e 0.01, respetivamente. Como se pode ver, o primeiro ruído coloca píxeis pretos e brancos em posições aleatórias, ao invés do ruído gaussiano, que segue uma função, aplicando ruído uniformemente a toda a imagem.



Figura 2 - Comparação de ruído *salt and pepper* (esquerda) com gaussiano (direita).

Aumentando, por exemplo, a variância do ruído gaussiano, a imagem perde mais qualidade, sendo possível observar uma maior quantidade de píxeis ruidosos, como se vê na Figura 3.

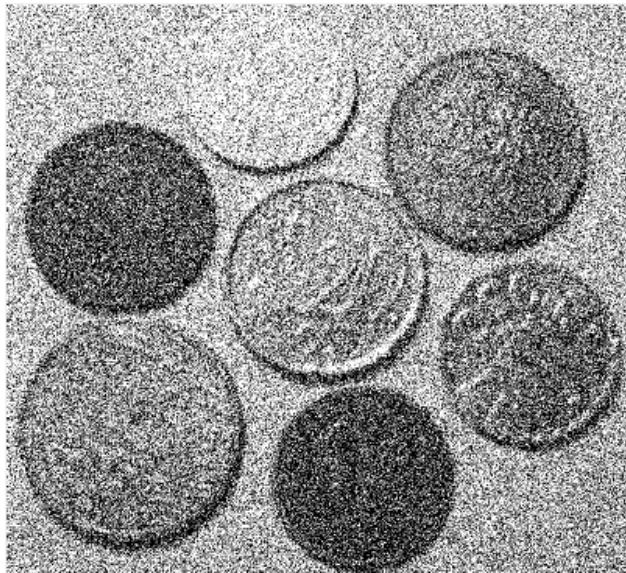


Figura 3 - Ruído gaussiano com variância 0.1.

### 3.2 Pré-Processamento

Devido às grandes variações presentes nos histogramas das imagens, o primeiro passo do processamento foi o ajuste das intensidades dos pixels, permitindo obter uma imagem com melhor contraste, ou seja, componentes de alta frequência mais acentuados, através da função *imadjust*. Em seguida, devido ao ruído introduzido, a imagem foi suavizada, tendo-se utilizado um filtro mediana, devido às suas capacidade na remoção de intensidades arbitrárias, como se pode ver pela Figura 4.



Figura 4 - Imagem suavizada com filtro mediana.

No caso das imagens 2 e 3, ou seja, as imagens cujas variações na iluminação eram bastante elevadas, foi efetuado um tratamento da iluminação não uniforme, resultando em imagens com menores variações de iluminação, como se vê na Figura 5. Para a realização deste tratamento, a imagem foi filtrada com um elemento estruturante circular de tamanho 15, sendo que se fez a subtração deste resultado à imagem suavizada.





Figura 5 - Tratamento de iluminação não uniforme.

Já no caso da imagem 1, foi efetuado um novo aumento do contraste na imagem, de modo a que as diferenças nas intensidades ficassem o mais realçadas possível. Em seguida, foi efetuado um processo semelhante ao anterior para lidar com a iluminação nas outras imagens, onde se testou utilizar vários elementos estruturantes, estando o resultado apresentado na Figura 6.



Figura 6 - Tratamento de iluminação não uniforme com vários operadores morfológicos.

### 3.3 Segmentação

Considere-se a segmentação de uma imagem como a divisão da mesma em várias regiões, sendo, neste caso, fundo e objetos, onde o primeiro é representado por pixels pretos e os segundos por pixels brancos.

Em seguida serão apresentados todos os passos para a segmentação de cada imagem, visto que cada uma necessita de um processo de segmentação apropriado às suas características. Serão, ainda, apresentados os resultados finais obtidos, prontos para a deteção de circunferências.

#### 3.3.1 Imagem 1

De todas as imagens a analisar, esta foi a que mais desafios trouxe ao grupo, uma vez que as diferentes tonalidades na imagem tornaram a diferenciação dos objetos com o fundo uma tarefa complicada.

A partir da imagem resultante do pré-processamento, normalizada, foi aplicado um *threshold* global de valor 0.42, seguido da aplicação do método Canny, onde se obteve o resultado na Figura 7. Isto permitiu ignorar alguns componentes que seriam detetados pela simples aplicação do detetor de *edges*, ajudando na segmentação da imagem.

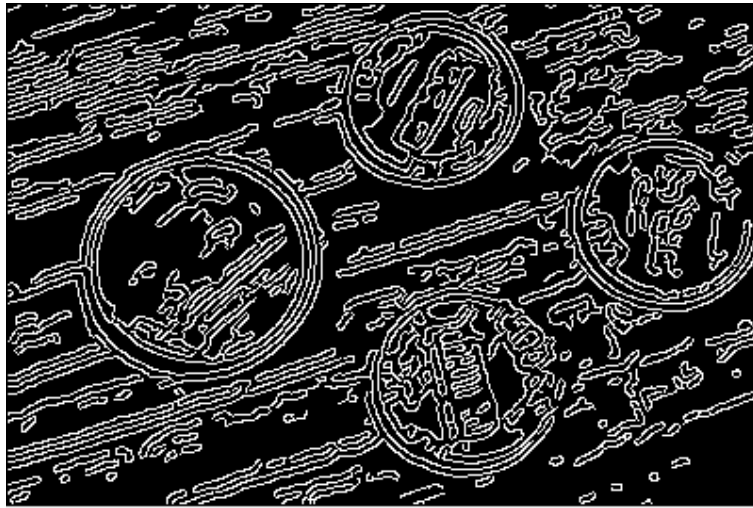


Figura 7 - Aplicação do detetor de *edges* Canny à primeira imagem.

Como a imagem ainda não estava suficientemente segmentada, removeram-se os objetos compostos por menos de 100 píxeis. Seguidamente, aplicou-se um *closing* com um elemento estruturante circular de tamanho 4, como se pode ver pela Figura 8.

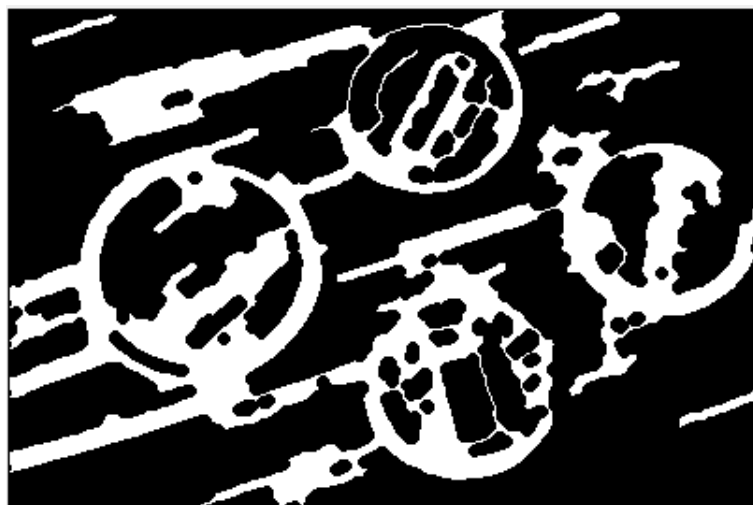


Figura 8 - *Closing* com elemento estruturante circular.

Por fim, foi efetuado um preenchimento dos buracos existentes nos objetos com auxílio da função *imfill*, seguido da remoção dos objetos com menos de 200 píxeis. Este processo permitiu obter a Figura 9, onde se pode ver a segmentação final da Imagem 1, pronta a efetuar a deteção de circunferências. Note-se que esta segmentação não é perfeita, mas permite a fácil identificação dos locais onde se encontram as moedas. No entanto, seria necessário mais processamento de modo a obter uma segmentação melhor.



Figura 9 - Segmentação final da primeira imagem.

### 3.3.2 Imagem 2

Partindo da imagem suavizada, o primeiro passo consistiu em aplicar um detetor de *edges*, o método Canny, com um fator de Fudge de 1.5, obtendo-se a Figura 10.

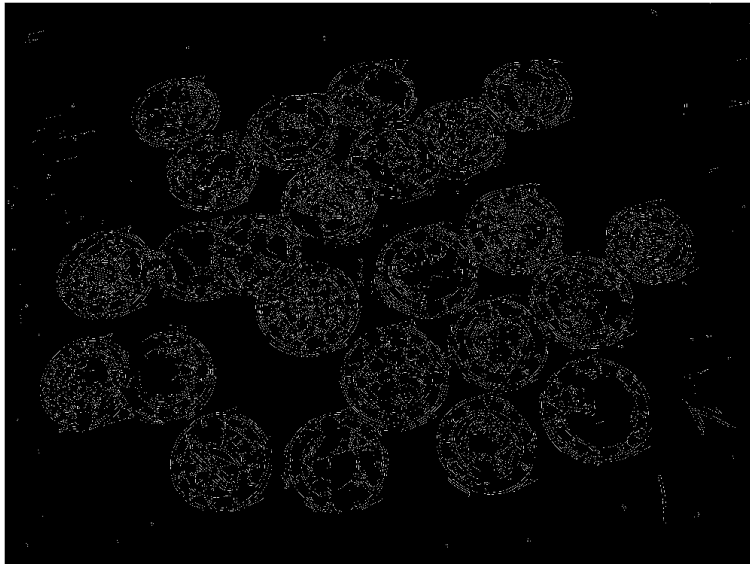


Figura 10 - Aplicação do detetor de *edges* Canny à segunda imagem.

Como o método Canny retorna uma imagem com buracos nas moedas, é necessário a aplicação de operadores morfológicos de modo a preencher os mesmos. Assim, aplicando uma dilatação quer por discos, quer por linhas, é possível preencher buracos nos objetos, obtendo-se a Figura 11.

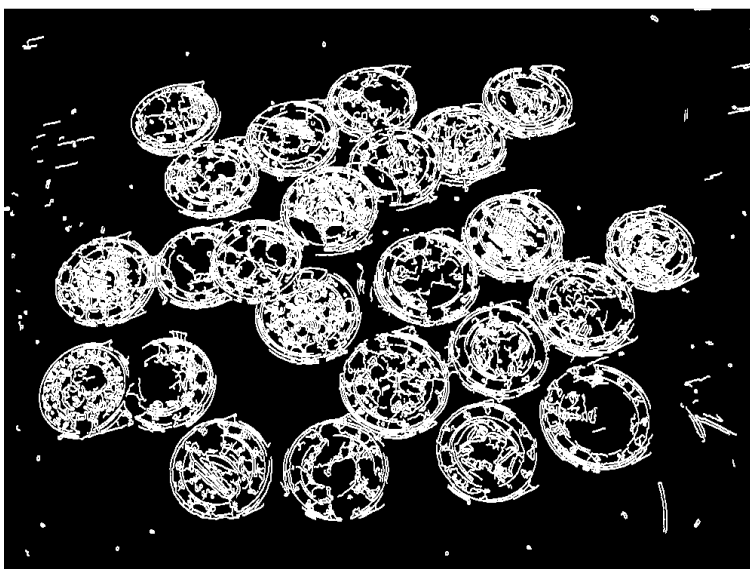


Figura 11 - Aplicação de operadores morfológicos para conectar objetos.

Por fim, é necessário melhorar os cortes entre os objetos, pelo que se efetuaram erosões com um elemento estruturante com forma de diamante, seguidos da remoção de objetos com menos de 200 píxeis, normalmente objetos associados a ruído. Dito isto, o resultado final da segmentação desta imagem pode ser observado na Figura 12, onde se vê com clareza as localizações das moedas, devido à utilização da função *imfill* para preencher os objetos em estudo.

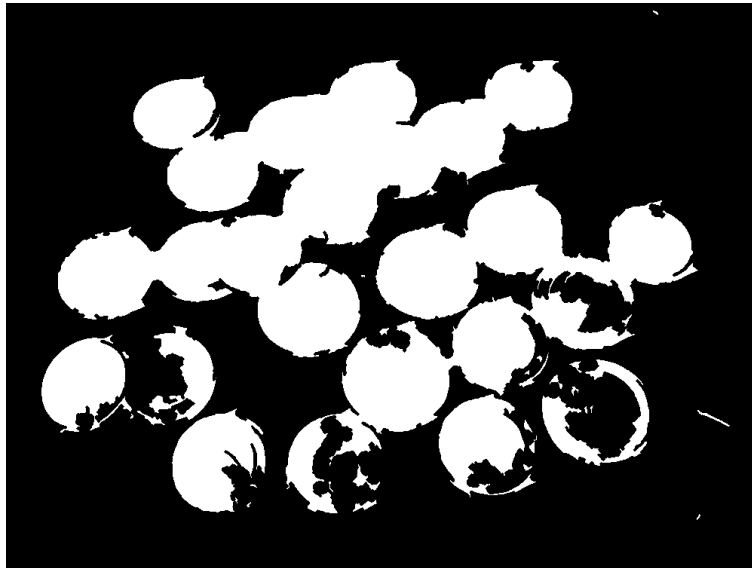


Figura 12 - Segmentação final da segunda imagem.

### 3.3.3 Imagem 3

Tendo em conta as grandes variações de iluminação presentes nesta imagem, mesmo após o tratamento da iluminação não uniforme, optou-se por utilizar um *threshold* adaptativo (*adpatthresh*) com sensibilidade de 55%, sendo que esta influencia o número de pixels associados a objetos em comparação com os pixels associados a fundo. Assim, o resultado da sua aplicação é apresentado na Figura 13, onde se podem ver com clareza as moedas, apesar de estas não estarem preenchidas e existir bastante ruído na imagem.



Figura 13 - Resultado da aplicação de um *threshold* adaptativo.

Para tentar resolver estes problemas, a ideia passou por ajustar os contrastes, sendo aplicado um novo *threshold*, desta vez com a ajuda da função *imbinarize*, culminando na remoção de objetos com menos de 60 pixels, como se pode ver pela Figura 14. Note-se que o valor de pixels mínimo por objeto tinha que ser baixo devido à existência da moeda no topo da imagem.



Figura 14 - Remoção de objetos com menos de 60 píxeis.

Por fim, de modo a obter uma imagem pronta a analisar pelo método da transformada de Hough, apenas foi necessário preencher os buracos presentes nos objetos, obtendo-se a Figura 15. É de realçar que ainda existem píxeis indesejados na segmentação. No entanto, as moedas são detetáveis, pelo que se trata de uma segmentação apropriada.



Figura 15 - Segmentação final da terceira imagem.

### 3.4 Detecção de Moedas

De modo a efetuar a deteção de moedas em cada imagem, a ideia passou por aplicar o método da transformada de Hough, método este que já se encontra implementado na linguagem com a função *imfindcircles* e é invariante à oclusão. Dito isto, apenas foi necessário otimizar os parâmetros da função, uma vez que esta recebe um intervalo para os raios das circunferências, a polaridade dos objetos a encontrar e, ainda, um valor de sensibilidade que ajudará a detetar mais ou menos objetos.

A primeira imagem possui 4 moedas, sendo que, como se pode ver pela Figura 16, o método desenvolvido é capaz de as detetar e localizar, apesar das falhas existentes na segmentação.

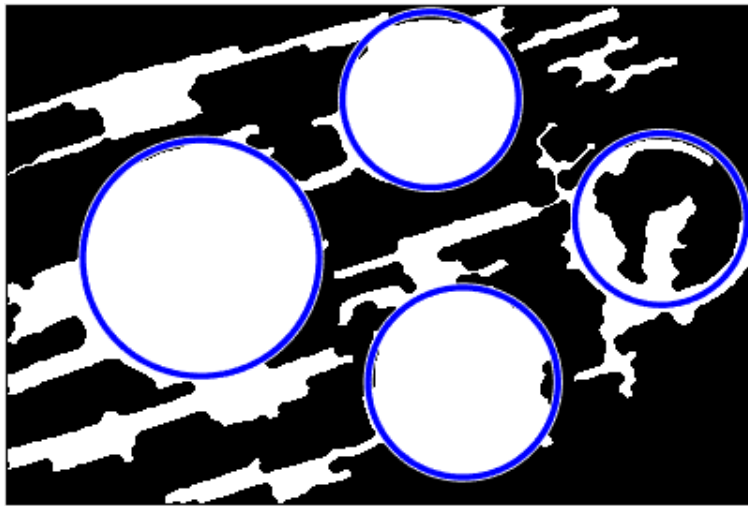


Figura 16 - Detecção de moedas na primeira imagem.

Já a segunda imagem contém 24 moedas, sendo que algumas estão cobertas por outras moedas ou por sombras, ocorrendo oclusão. No entanto, a segmentação permitiu tratar destes problemas, sendo que a deteção e localização ocorrem como expectável, como se pode observar pela Figura 17.

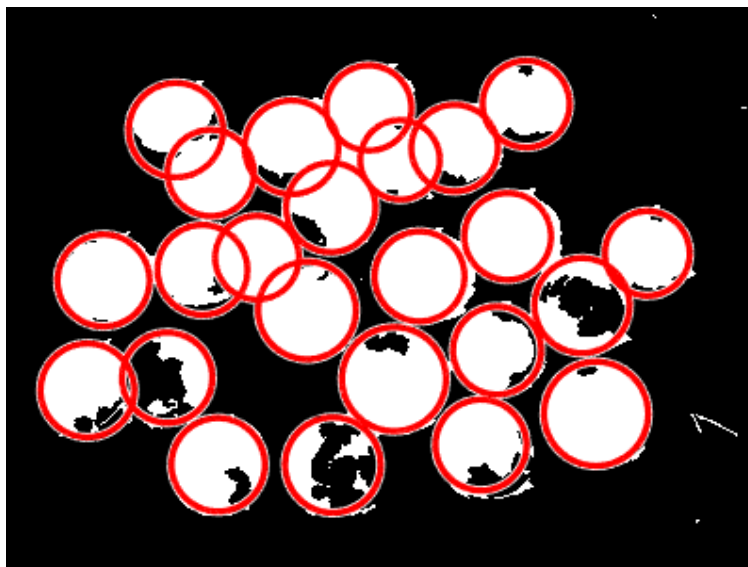


Figura 17 - Detecção de moedas na segunda imagem.

No que toca à terceira imagem, independentemente do ruído ainda visível, as moedas são detetadas e localizadas corretamente pelo algoritmo, como representado na Figura 18.

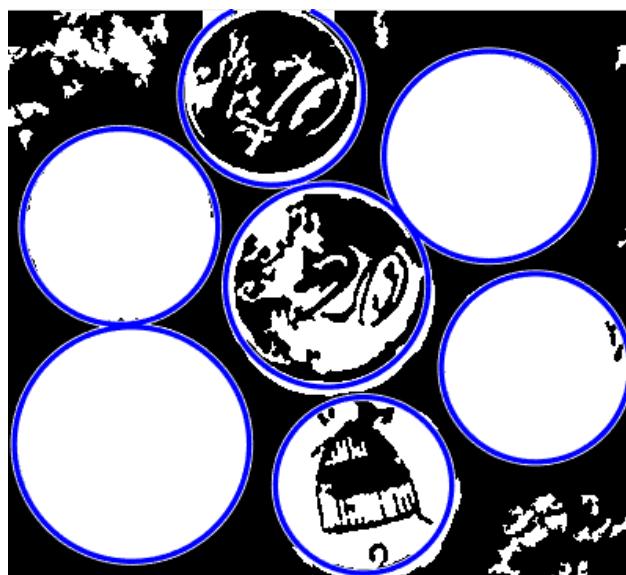


Figura 18 - Detecção de moedas na terceira imagem.

3.4.1 Moedas por Tipo

No que toca à associação das moedas aos seus tipos, o primeiro pensamento passou pela implementação de um método *SIFT*, porém este não trataria bem de moedas do mesmo tipo com diferentes faces. O segundo pensamento passou pela implementação de uma rede neuronal que fizesse a associação dos tipos a cada moeda. No entanto, este processo poderia tornar-se moroso e implicaria a implementação de, mais uma vez, um método *SIFT*, sendo que os resultados nunca seriam 100% plausíveis como mencionado anteriormente. Por fim, o grupo assumiu que o principal desafio deste projeto era a deteção das moedas na imagem e não a sua associação a cada tipo, pelo que esta foi efetuada com base nas dimensões de cada moeda. Note-se que, devido a este processo, o algoritmo está dependente de cada imagem, visto que as moedas se encontram a distâncias diferentes da câmara.

Assim sendo, como se pode ver pela Figura 19, a partir dos histogramas das dimensões das moedas, bem como da análise das imagens originais, é possível estimar valores próximos dos adquiridos pelas moedas. Note-se que, devido às variações na profundidade da segunda imagem e a pequena variação dos tamanhos entre os tipos das moedas, a contagem das moedas por tipo não é tão fiável quantos os outros exemplos, visto que existem 15 moedas de 2 euros e não 16 como obtido.

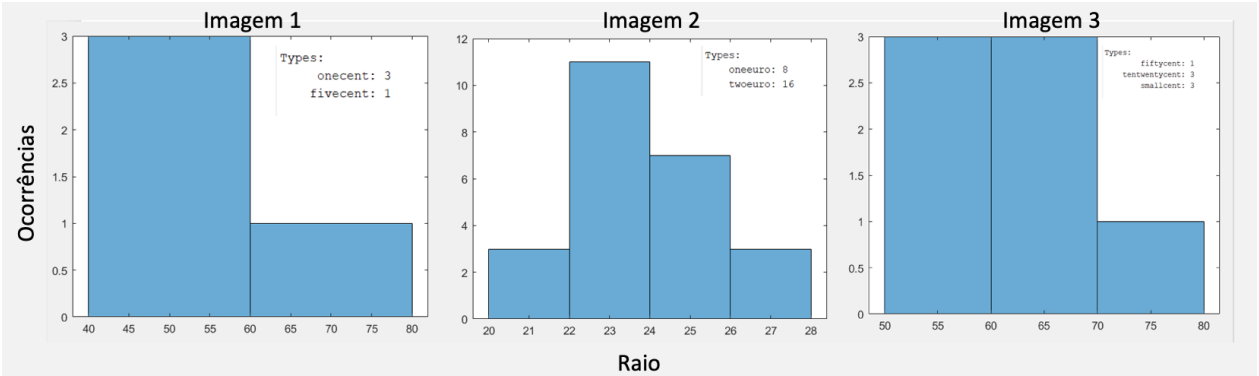


Figura 19 - Histogramas de dimensões e contagem de tipos.

3.5 Influência do Ruído no Resultado Final

No que toca à influência do ruído nas imagens, se este tiver um valor reduzido, a sua influência após a segmentação é bastante reduzida, como se pode ver pela Figura 20, onde se apresentam as segmentações das imagens 2 e 3, sem ruído. Note-se, no entanto, que, se o ruído for bastante elevado, os objetos terão mais buracos quer no seu corpo, quer nos seus contornos, tornando a deteção bastante mais complicada.

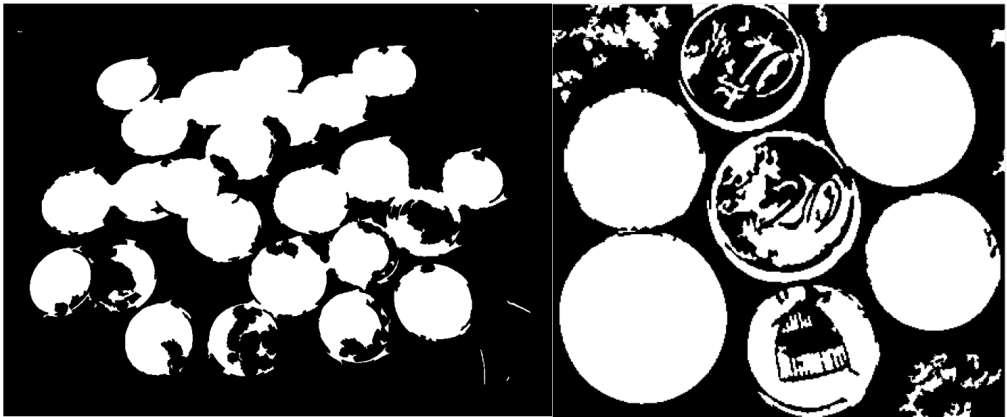


Figura 20 - Segmentações da segunda e terceira imagens.

No caso da primeira imagem, como a sua segmentação não foi a ideal, o ruído afeta mais o resultado da segmentação, como se vê na Figura 21, onde existe uma área que foi detetada como *foreground*, quando deveria ser fundo na imagem do lado direito. Este tipo de situações tornam a deteção mais complicada, apesar de existirem vestígios da existência das circunferências no local em questão.

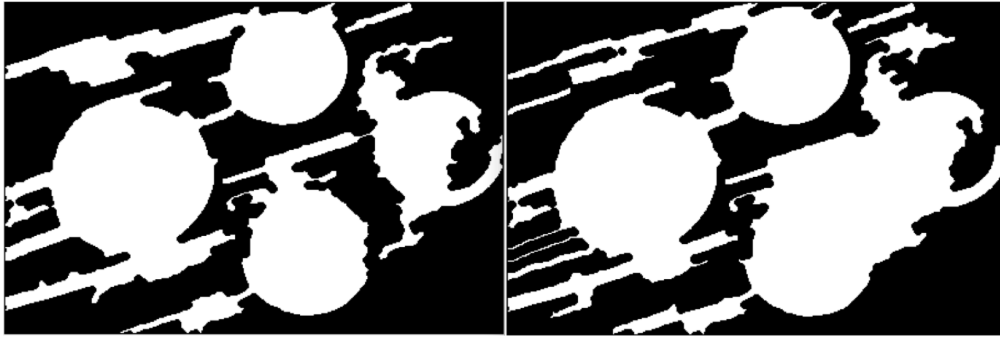


Figura 21 - Segmentações primeira imagem sem ruído e com ruído, respetivamente.

### 3.6 Erros e Limitações

Como mencionado ao longo da explicação do método desenvolvido, existem componentes que poderiam ser melhorados com mais tempo de análise. Este aspeto é notório, especialmente, no que toca à segmentação da primeira imagem, onde a diferenciação entre as partes escuras do fundo e as moedas em si não foi ideal. Apesar disto, com a aplicação de mais algumas operações morfológicas nos locais certos e com parâmetros adequados, este problema seria resolvido facilmente.

Outro problema do projeto foi o facto de não existir um método capaz de segmentar qualquer uma das imagens, independentemente de qual se trata, uma vez que cada caso é um caso. Assim, de forma a otimizar cada uma das segmentações, o grupo optou por ter casos específicos, permitindo uma maior diferenciação entre fundo e objetos, não se perdendo dados.

Assim, pode-se dizer que a qualidade das segmentações depende das condições de iluminação, da existência de oclusão e das características do fundo tendo em conta os algoritmos aplicados. Depende também, como seria de esperar, do ruído na imagem, sendo que com o aumento deste, a probabilidade de todas as moedas serem detetadas diminui.

Por fim, é de realçar o facto de que, como a função associada à transformada de Hough já se encontra definida na linguagem, a existência de falsos positivos e/ou falsos negativos está, também, associada à sensibilidade desta função. Assim, o grupo tentou utilizar valores que maximizam a deteção real, sendo que, por raras vezes, a deteção falha.



## 4 Conclusões e Trabalho Futuro

Ao longo do presente projeto encontra-se representado o segundo tutorial da unidade curricular de Visão por Computador. Neste contexto foram abordados métodos distintos de modo a segmentar imagens, realçando as moedas contidas em cada uma das mesmas. Foi, ainda, temática deste projeto o método de deteção de circunferências com a transformada de Hough, o qual se revelou de extrema importância, uma vez que permite localizar e contar as moedas numa determinada imagem, independentemente de existir oclusão das mesmas.

Tendo em conta os problemas associados à segmentação de cada imagem, o grupo construiu métodos capazes de fornecer resultados que permitem a fácil deteção de objetos, apesar de algum ruído e fundo não terem sido completamente removidos da análise, como foi o caso da primeira e última figuras.

Apesar disso, a deteção e consequente localização das circunferências foi efetuada sem qualquer problema, tirando proveito de um método de extração de características de alto nível, a transformada de Hough. Este permitiu, ainda, saber quais as dimensões de cada objeto encontrado, característica útil para a associação de um determinado tipo a cada objeto.

Em suma, a realização deste trabalho exigiu a aplicação de todos os conhecimentos lecionados em contexto de aula, permitindo que o grupo cumprisse todos os objetivos propostos no enunciado, conciliando assim a teoria e a prática.