



**Agostinho Pires**

**Projeto i-RoCS - Sistema de Detecção de Sujidade  
baseada em Visão por Computador**

**Project i-RoCS - Dirt Detection System based on  
Computer Vision**





**Agostinho Pires**

**Projeto i-RoCS - Sistema de Detecção de Sujidade  
baseada em Visão por Computador**

**Project i-RoCS - Dirt Detection System based on  
Computer Vision**

*“Being portuguese, is being an explorer and adventurer”*

— José Mourinho





**Agostinho Pires**

**Projeto i-RoCS - Sistema de Detecção de Sujidade  
baseada em Visão por Computador**

**Project i-RoCS - Dirt Detection System based on  
Computer Vision**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor António José Ribeiro Neves, Professor auxiliar do da Universidade de Aveiro, e do Professor Doutor Pedro Nicolau Faria da Fonseca, Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.



**o júri / the jury**

presidente / president

Prof. Doutor José Luis Costa Pinto de Azevedo  
professor Auxiliar da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor António José Ribeiro Neves  
professor Auxiliar da Universidade de Aveiro

Prof. Doutor Hugo Pedro Martins Carriço Proença  
professor Catedrático da Universidade da Beira Interior - Departamento de Informática



**agradecimentos /  
acknowledgements**

Aos meus orientadores, Professor António Neves e Professor Pedro Nicolau Faria da Fonseca por todo o suporte que me deram ao longo do desenvolvimento do projeto, pelos conselhos e pela transferência de conhecimento.

Um especial agradecimento ao Daniel Canedo, pela disponibilidade que teve comigo, pelo tempo gasto nas inúmeras sessões que fizemos para esclarecimento de dúvidas em determinados pontos do projeto.

Aos meus colegas, Joana Simões e João Nunes, porque sem vocês o meu percurso académico não teria sido o mesmo.

À minha amiga, Marta Marques, pela disponibilidade que teve para dar de um jeito diferente o seu contributo para a escrita desta dissertação.

Por último, mas não menos importante, à minha Família, por estarem presentes, pelo suporte que me deram ao longo destes anos académicos e que nunca me deixaram de apoiar em qualquer momento.

A todos vocês, que me acompanharam nesta etapa estou muito agradecido!



## Palavras Chave

Visão por Computador, Aprendizagem Auto-máquina, Aprendizagem Profunda, KNN, GMM, Redes Neurais Artificiais, Processamento de Imagem Digital, Perceptron, Detecção de Objetos

## Resumo

A finalidade principal desta dissertação é avaliação de algoritmos de visão por computador com potencial para efetuar a detecção de sujidade no chão, através da captura de imagens digitais. A abordagem traçada vai de encontro com a aquisição de conhecimentos pelo executante, por algoritmos e métodos de aprendizagem auto-máquina.

A aquisição destes conhecimentos é uma das principais motivações do executante em elaborar esta dissertação, pois esta abordagem tem presentes uma nova maneira de pensar com o uso de tecnologias mais contemporâneas como a visão por computador, onde métodos de aprendizagem auto-máquina surgem e apresentam bons resultados.

O projeto IROCS pode beneficiar com esta dissertação de uma implementação que permite descartar e enquadrar formas diferentes de detecção de sujidade através de captura de imagens.

Existem outras maneiras de efetuar análise de imagens visando detecção, mas no caso da sujidade e conforme a tecnologia existente na atualidade, uma câmara digital é a melhor hipótese.

Através da captura de vídeo onde as imagens digitais são o principal recurso, estudou-se o processamento de imagem, aplicando vários métodos tanto de extração de características, como classificação das mesmas, como a finalidade de detecção da sujidade, aos quais foram aplicados critérios de forma diferente.

Esta dissertação apresenta métodos de aprendizagem máquina, métodos com ou sem supervisão, com classificação por rótulo ou probabilidade, utilizando um pré-processamento recorrendo a gradientes e com extração de características em blocos, retirando dos blocos a média e o desvio padrão como recurso para a classificação. Os resultados obtidos foram motivadores, ofertando a capacidade de validar algoritmos como soluções com valência e descartar outros com menos potencial, em simultâneo, ter perceção das limitações que podem existir, verificando e analisando estes possíveis problemas e se são possíveis de contornar.



**Keywords**

Computer Vision, Machine Learning, Deep Learning, KNN, GMM, Artificial Neural Network, Digital Image Processing; Perceptron, Object Detection

**Abstract**

The main purpose of this dissertation is to evaluate computer vision algorithms with the potential to detect dirt on the floor, through the capture of digital images. The approach is the acquisition of knowledge by the performer, using algorithms and machine learning methods.

This acquisition of this knowledge is one of the main motivations of the practitioner in preparing this dissertation, as this approach presents a new way of thinking with the use of more contemporary technologies such as computer vision, where machine learning methods emerging and giving good results.

The IROCS project can benefit from this dissertation, with an implementation that allows discarding and framing different forms of dirt detection through image capture.

There are other ways to perform image analysis for detection, but in the case of dirt and according to current technology, a digital camera is the best option.

Through video capture, where digital images are the main resource, image processing and applying several methods, both of extraction of features, as classification of the same, as the purpose of detecting dirt, to which criteria were applied different way.

This dissertation presents machine learning methods, methods with or without supervision, with classification by label or probability, using a pre-processing using gradients and with block feature extraction, removing the mean and standard deviation from the blocks as a resource for the classification.

The results obtained were motivating, offering the ability to validate algorithms as solutions with valence and discard others with less potential, at the same time, being aware of the limitations that may exist, verifying and analyzing these possible problems and whether they are possible to overcome.



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>v</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>Acronyms List</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivo . . . . .	2
1.2 Motivação . . . . .	3
1.3 Contribuições . . . . .	3
1.4 Estrutura do documento . . . . .	4
<b>2 Fundamentos</b>	<b>5</b>
2.1 Noções e Conceitos . . . . .	5
2.1.1 Imagem Digital . . . . .	5
2.1.2 Vídeo Digital . . . . .	7
2.1.3 Visão Humana . . . . .	8
2.2 Visão por Computador . . . . .	11
2.3 Processamento de imagem digital . . . . .	12
2.4 Inteligência artificial . . . . .	13
2.4.1 Aprendizagem máquina . . . . .	14
2.4.2 Redes neurais artificiais . . . . .	15
2.5 Revisão da literatura científica . . . . .	19
2.6 Conclusão . . . . .	20

<b>3</b>	<b>Pré-processamento</b>	<b>21</b>
3.1	conjunto de dados . . . . .	21
3.2	Espaço de cor . . . . .	26
3.3	Gradiente de Cor . . . . .	28
3.3.1	Gradientes . . . . .	28
3.3.2	Magnitudes . . . . .	31
3.4	Extração das caraterísticas . . . . .	31
3.5	Conclusão . . . . .	33
<b>4</b>	<b>Classificação</b>	<b>35</b>
4.1	K-means . . . . .	35
4.2	Modelo de Misturas Gaussianas . . . . .	37
4.2.1	Distribuição Gaussiana . . . . .	37
4.2.2	Algoritmo de maximização de expectativa . . . . .	38
4.3	K vizinhos mais próximos . . . . .	39
4.4	Rede Neural Perceptron . . . . .	41
4.5	Conclusões . . . . .	42
<b>5</b>	<b>Implementação</b>	<b>45</b>
5.1	Algoritmos Implementados . . . . .	45
5.1.1	Pré-processamentos da imagem digital . . . . .	46
5.1.2	Classificações da imagem digital . . . . .	47
5.2	Critérios de classificação . . . . .	51
5.2.1	Interseção pelo Produto . . . . .	51
5.2.2	Interseção Maioritária . . . . .	52
5.3	Conclusão . . . . .	52
<b>6</b>	<b>Resultados Experimentais</b>	<b>53</b>
6.1	Análise . . . . .	53
6.1.1	Análise do Espaço de cor . . . . .	53
6.1.2	Análise dos gradientes . . . . .	54
6.1.3	Análise da velocidade de extração de caraterísticas . . . . .	58
6.1.4	Análise do tempo de treino de cada método . . . . .	59
6.1.5	Análise do tempo de teste de cada método . . . . .	60

6.1.6	Análise da classificação de cada método . . . . .	62
6.2	Comparação . . . . .	69
6.2.1	Comparação entre os espaços de cor . . . . .	69
6.2.2	Comparação entre os gradientes . . . . .	69
6.2.3	Comparação dos tempos de treino consoante o método . . . . .	70
6.2.4	Comparação dos tempos de teste consoante o método . . . . .	70
6.2.5	Comparação da qualidade das classificações obtidas . . . . .	70
6.3	Conclusões . . . . .	71
<b>7</b>	<b>Conclusão</b>	<b>73</b>
7.1	Algoritmo Proposto . . . . .	73
7.1.1	Pré-processamento . . . . .	74
7.1.2	Classificação . . . . .	76
7.2	Abordagens Futuras . . . . .	79
7.3	Conclusões finais . . . . .	79
<b>8</b>	<b>Referências</b>	<b>81</b>
<b>A</b>	<b>Anexos</b>	<b>85</b>
A.1	Pseudocódigo dos Algoritmos . . . . .	85
A.1.1	Perceptron: . . . . .	87
A.2	Exemplos de deteções . . . . .	89
A.2.1	Exemplos das Gaussian Mixture Model (GMM) . . . . .	89
A.2.2	Exemplos das Artificial Neural Network (ANN) . . . . .	96
A.2.3	Exemplos das K-Nearest Neighbor (KNN) . . . . .	100



# Lista de Figuras

1.1	Fluxograma do algoritmo proposto. . . . .	3
2.1	Representação das 3 dimensões de uma imagem Red Green Blue (RGB). . . . .	6
2.2	Sobreposição dos 3 canais. . . . .	7
2.3	Decomposição do espaço de cor YUV. . . . .	8
2.4	Imagem de um corte vertical do olho humano. . . . .	9
2.5	Imagem que compara uma máquina ao olho humano. . . . .	10
2.6	Imagem do corte vertical do olho e retina. . . . .	10
2.7	Representação do espectro das cores visíveis. . . . .	11
2.8	Esquemático dos métodos de aprendizagem máquina com supervisão. . . . .	14
2.9	Esquemático dos métodos de aprendizagem máquina sem supervisão. . . . .	15
2.10	Exemplo matemático do dimensionamento de um neurão. . . . .	16
2.11	Exemplo gráfico do degrau simples. . . . .	17
2.12	Exemplo gráfico do degrau bipolar. . . . .	17
2.13	Exemplo gráfico da função de ativação logística. . . . .	18
2.14	Exemplo gráfico da função de ativação hiperbólica. . . . .	18
3.1	Exemplo das imagens presentes no ACIN data. . . . .	22
3.2	Exemplo das imagens presentes no IPA data. . . . .	22
3.3	Exemplo das imagens retiradas na fábrica da Renault Cacia, S.A. . . . .	23
3.4	Exemplo das imagens retiradas na fábrica da Atena Automação Industrial, Lda .	24
3.5	A câmara digital que será integrada no protótipo do robô. . . . .	25
3.6	Imagens reproduzidas com sujidade artificial. . . . .	26
3.7	Representação visual da equação 3.10, apresentado as variações de intensidade diagonal. . . . .	29

3.8	Representação visual das equações 3.11 e 3.12, apresentado as variações de intensidade nos eixos x e y, respectivamente. . . . .	29
3.9	Representação animada de uma janela deslizante. . . . .	32
4.1	Exemplo de um espaço de características do método K-means. . . . .	36
4.2	Exemplo da representação das distâncias entre pontos. . . . .	41
6.1	Exemplo artificial de um chão laranja com existência de sombra, convertido em HSV e Lab, segunda e terceira imagem respetivamente. . . . .	54
6.2	Exemplo artificial de um chão madeira cinza, convertido em HSV e Lab, segunda e terceira imagem respetivamente. . . . .	54
6.3	Exemplo artificial de um chão azulejo cinza com existência de sombra, convertido em HSV e Lab, segunda e terceira imagem respetivamente. . . . .	54
6.4	Aplicação de um gradiente do tipo normal à Figura 6.1. . . . .	55
6.5	Aplicação de um gradiente do tipo sobel à Figura 6.1. . . . .	55
6.6	Aplicação de um gradiente do tipo laplaciano à Figura 6.1. . . . .	56
6.7	Aplicação de um gradiente do tipo normal à Figura 6.2. . . . .	56
6.8	Aplicação de um gradiente do tipo sobel à Figura 6.2. . . . .	56
6.9	Aplicação de um gradiente do tipo laplaciano à Figura 6.2. . . . .	57
6.10	Aplicação de um gradiente do tipo normal à Figura 6.3. . . . .	57
6.11	Aplicação de um gradiente do tipo sobel à Figura 6.3. . . . .	57
6.12	Aplicação de um gradiente do tipo laplaciano à Figura 6.3. . . . .	58
6.13	Representação do cálculo da IoU. . . . .	62
6.14	Exemplo de uma boa deteção com GMM, num chão do cimento. . . . .	64
6.15	Exemplo de uma boa deteção com GMM previsão, num chão de madeira. . . . .	65
6.16	Exemplo de uma boa deteção com GMM, num chão do laranja. . . . .	65
6.17	Exemplo de uma boa deteção com KNN, num chão laranja. . . . .	66
6.18	Exemplo de uma boa deteção com ANN, num chão laranja. . . . .	67
6.19	Exemplo de uma boa deteção com ANN, num chão madeira. . . . .	68
6.20	Exemplo de uma boa deteção com ANN, num chão cimento. . . . .	68
6.21	Exemplo de uma boa deteção com ANN, num chão multi-padrão. . . . .	68
7.1	Exemplo da conversão de uma imagem BGR para Lab. . . . .	74
7.2	Exemplo da divisão do espaço L, a e b, com os respetivos gradientes de cor. . . . .	75
7.3	Gráficos distribuição das características da imagem exemplo. . . . .	76

7.4	Mapa de probabilidades, interceção e aplicação da máscara ao exemplo. . . . .	77
7.5	Máscara usando GMM com recurso a previsão. . . . .	78
A.1	Deteção fraca em chão laranja com bloco 16 x 16 e critério maioritário. . . . .	89
A.2	Deteção boa em chão laranja com bloco 16 x 16 e critério maioritário. . . . .	90
A.3	Deteção fraca em chão laranja com bloco 32 x 32 e critério maioritário. . . . .	90
A.4	Deteção boa em chão laranja com bloco 32 x 32 e critério maioritário. . . . .	90
A.5	Deteção fraca em chão laranja com bloco 64 x 64 e critério maioritário. . . . .	90
A.6	Deteção fraca em chão laranja com bloco 64 x 64 e critério produto. . . . .	91
A.7	Deteção fraca em chão laranja com bloco 64 x 64 e critério produto. . . . .	91
A.8	Deteção fraca em chão de madeira com bloco 16 x 16 e critério maioritário. . . .	91
A.9	Deteção fraca em chão de madeira com bloco 16 x 16 e critério maioritário. . . .	91
A.10	Deteção boa em chão de madeira com bloco 16 x 16 e critério produto. . . . .	92
A.11	Deteção fraca em chão de madeira com bloco 32 x 32 e critério maioritário. . . .	92
A.12	Deteção boa em chão de madeira com bloco 32 x 32 e critério maioritário. . . . .	92
A.13	Deteção boa em chão de madeira com bloco 32 x 32 e critério produto. . . . .	93
A.14	Deteção boa em chão de madeira com bloco 32 x 32 e critério produto. . . . .	93
A.15	Deteção boa em chão de madeira com bloco 64 x 64 e critério produto. . . . .	93
A.16	Deteção boa em chão de madeira com bloco 64 x 64 e critério maioritário. . . . .	94
A.17	Deteção fraca em chão de madeira com bloco 64 x 64 e critério maioritário. . . .	94
A.18	Deteção fraca em chão de madeira com bloco 64 x 64 e critério produto. . . . .	94
A.19	Deteção fraca em chão de cimento com bloco 16 x 16 e critério produto. . . . .	95
A.20	Deteção fraca em chão de cimento com bloco 32 x 32 e critério maioritário. . . .	95
A.21	Deteção boa em chão de cimento com bloco 64 x 64 e critério maioritário. . . . .	95
A.22	Deteção boa em chão de laranja com bloco 16 x 16 e critério maioritário. . . . .	96
A.23	Deteção boa em chão de laranja com bloco 16 x 16 e critério produto. . . . .	96
A.24	Deteção fraca em chão de laranja com bloco 32 x 32 e critério produto. . . . .	96
A.25	Deteção fraca em chão de madeira com bloco 16 x 16 e critério produto. . . . .	97
A.26	Deteção boa em chão de madeira com bloco 16 x 16 e critério produto. . . . .	97
A.27	Deteção fraca em chão de madeira com bloco 64 x 64 e critério produto. . . . .	98
A.28	Deteção boa em chão de cimento com bloco 16 x 16 e critério maioritário. . . . .	98
A.29	Deteção fraca em chão de cimento com bloco 32 x 32 e critério maioritário. . . .	99
A.30	Deteção boa em chão de cimento com bloco 64 x 64 e critério maioritário. . . . .	99
A.31	Deteção boa em chão de laranja com bloco 16 x 16 e critério maioritário. . . . .	100

A.32 Detecção fraca em chão de laranja com bloco 32 x 32 e critério maioritário. . . . 100

# Lista de Tabelas

2.1	Espectro Visível . . . . .	11
6.1	Tempo de Processamento dos Gradientes . . . . .	55
6.2	Tempo de extração das propriedades . . . . .	58
6.3	Tempo de execução do treino da GMM. . . . .	59
6.4	Tempo de execução do treino da ANN. . . . .	60
6.5	Tempo de execução do teste da GMM, com classificação em probabilidade. . . . .	60
6.6	Tempo de execução do teste da GMM, com classificação por previsão. . . . .	61
6.7	Tempo de execução do teste da KNN. . . . .	61
6.8	Tempo de execução do teste da ANN. . . . .	62
6.9	Intersection over Union (IoU) da classificação do GMM em previsão para 10 imagens. . . . .	63
6.10	IoU da classificação do GMM em previsão para 20 imagens. . . . .	63
6.11	IoU da classificação do GMM em probabilidade para 10 imagens. . . . .	64
6.12	IoU da classificação do GMM em probabilidade para 20 imagens. . . . .	64
6.13	IoU da classificação do KNN para 10 imagens e k igual 3. . . . .	65
6.14	IoU da classificação do KNN para 10 imagens e k igual a 5. . . . .	66
6.15	IoU da classificação do ANN para 10 imagens. . . . .	66
6.16	IoU da classificação do ANN para 20 imagens. . . . .	67



# Acronyms List

<b>ACIN</b>	Automation and Control Institute	<b>IPA</b>	Institut für Produktionstechnik und Automatisierung
<b>BGR</b>	Blue Green Red	<b>IoU</b>	Intersection over Union
<b>CDD</b>	Charge Couple Device	<b>IC</b>	Integrated Circuit
<b>COBOL</b>	Common Business-Oriented Language	<b>IBM</b>	International Business Machines
<b>CAT</b>	Computerized Axial Tomography	<b>KNN</b>	K-Nearest Neighbor
<b>CNN</b>	Convolutional Neural Network	<b>ML</b>	Machine Learning
<b>DL</b>	Deep Learning	<b>MOS</b>	Metal Oxide Semiconductor
<b>DS</b>	Data Set	<b>MIT</b>	Massachusetts Institute of Technology
<b>EM</b>	Expectation Maximization algorithm	<b>ANN</b>	Artificial Neural Network
<b>FORTTRAN</b>	Translator Formula	<b>OpenCV</b>	Open Source Computer Vision Library
<b>GMM</b>	Gaussian Mixture Model	<b>PDF</b>	Probability density function
<b>HSV</b>	Hue(Tonality) Saturation Value(Brightness)	<b>RGB</b>	Red Green Blue
<b>i-RoCS</b>	Intelligent Robotic Cleaning System		



# Introdução

Com a evolução da humanidade existe uma constante procura por facilidades, maneiras mais ágeis de executar determinados processos, com maior eficiência e melhor capacidade, dado que as novas gerações pretendem ultrapassar as gerações passadas. A visão por computador é uma tecnologia que tem sido muito utilizada nos últimos anos em contexto de indústria. Com a nova revolução industrial, as tecnologias como Internet of Things (IoT) tem uma enorme procura. Neste sentido, a aplicação de sistemas inteligentes em tecnologias, já existente, justifica a importância da renovação tecnológica e industrial.

A visão por computador é uma área da ciência informática e da electrotecnia, que tem como base a aquisição de imagens digitais, que permite, através do processamento da imagem a extração de características, de modo a identificar o que é pretendido, com a aplicação de métodos de processamento, análise e segmentação de dados digitais, de modo a efetuar o reconhecimento de objetos. Tem maior aplicação em vídeo, pois um vídeo é apenas um conjunto de imagens amostradas a uma determinada frequência que transmitem sensação de movimento.

O Intelligent Robotic Cleaning System (i-RoCS) pretende proceder à criação de uma tecnologia que vá de encontro às necessidades atuais. O projeto consiste num robô que executa limpeza autónoma, com a menor interação humana possível. Pretende-se que este robô tenha incorporado um sistema de visão, dado que mais automático for, mais valências e capacidades poderá ter. Para tal, o desenvolvimento deste trabalho visa construir um algoritmo capaz de “ver” é uma necessidade fulcral deste projeto, sendo este o principal propósito desta dissertação.

Computer Vision Library (OpenCV), recorrendo à linguagem python, por forma a interligar os vários processos da construção algorítmica pretendida. Atualmente existem já algumas máquinas que usam sistemas semelhantes tais como, Liberty SC50

UVGI da Nilfisk <sup>1</sup> ou T30AMR, T7AMR e T16AMR da Tennant. <sup>2</sup>.

## 1.1 OBJETIVO

A finalidade deste projeto é a detecção de sujidade no pavimento, ou seja, através da análise de uma frame de vídeo, o equivalente a uma imagem, é possível extrair características e recursos que permitam diferenciar chão limpo de chão sujo.

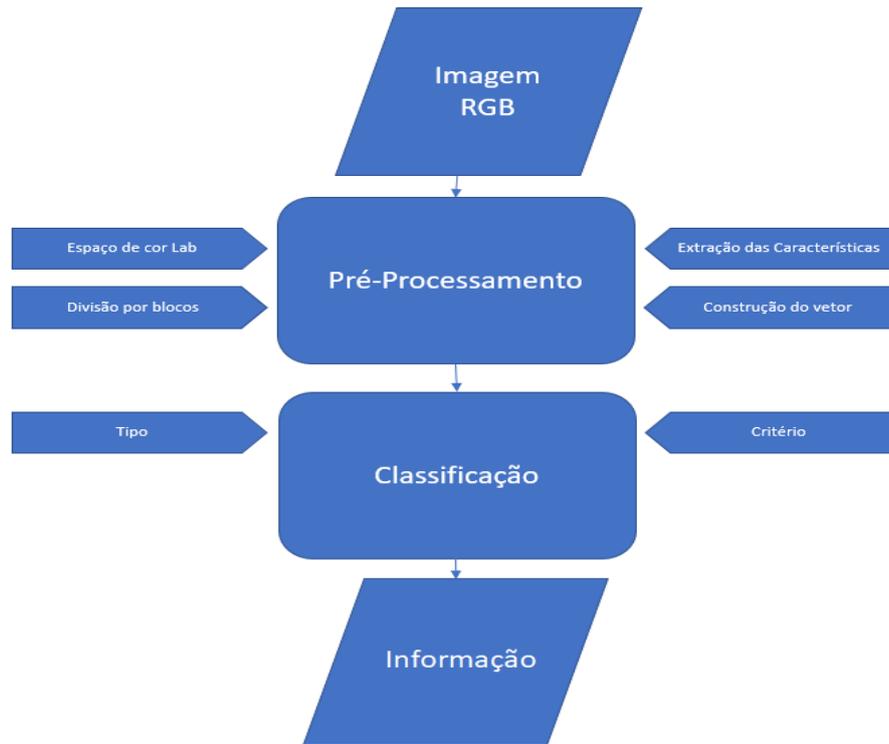
Para isso, é necessário efetuar um pré-processamento da imagem, isto é, por um método matemático transformar a informação, tornando-a mais simples o seu tratamento, para posteriormente proceder à sua classificação. Ou seja, o pretendido é a utilização de um método, também ele matemático, que agrupe a informação em diferentes tipos, visando aumentar a sua capacidade com o tempo ou com o treino.

Por fim, através da classificação previamente elaborada, devolver informação destacada do frame analisado, de modo que o resultado seja apenas uma imagem onde só esteja visível o local que se encontra sujo. Na Figura 1.1 está esquematizado de forma sintetizada o diagrama de fluxo do algoritmo implementado. Entrada do diagrama é um frame, ou seja, uma imagem RGB, em seguida é pré-processada, isto é, mudança de espaço de cor dividida em blocos, removidas as características desses blocos e construído um vetor de características enviado para classificação. A classificação é feita segundo um método *Tipo* e um *Critério* escolhido pelo utilizador, obtendo no final a informação necessária para etuar a destacação da sujidade dos restantes.

---

<sup>1</sup><https://e.pcloud.link/publink/show?code=S7jotalk>

<sup>2</sup>[https://www.tennantco.com/en\\_us/solutions/robotic-cleaning-machines.html](https://www.tennantco.com/en_us/solutions/robotic-cleaning-machines.html)



**Figura 1.1:** Fluxograma do algoritmo proposto.

## 1.2 MOTIVAÇÃO

O gosto pelo estudo de engenharias ligadas a evolução tecnológica dos dias de hoje, o interesse por novos processos eletrônicos e informáticos, quando adicionando a oportunidade de adquirir novos conhecimentos sobre Machine Learning (ML), Deep Learning (DL) e ANN, com a hipótese de poder aprofundar fundamentos em processamento de imagem e visão por computador, são as principais razões que cativam para o estudo deste tema.

Contudo, a possibilidade de comparar com outros temas tais como a detecção de objetos, devido a ser um assunto pouco estudado, com ainda poucas soluções atualmente, aplicar novas ideias, torna este projeto e o tema, um bom desafio.

## 1.3 CONTRIBUIÇÕES

O estudo de novas abordagens será sempre uma mais-valia para a evolução tecnológica, pois o que não é possível ser executado hoje com sucesso poderá ser num futuro próximo, devido ao fato da humanidade estar sempre em constante mudança e superação.

O desenvolvimento desta dissertação tem a finalidade de melhorar os sistemas já existentes de limpeza e higienização dos locais, sendo possível a sua aplicação em vários meios com características diferentes, obtendo o mesmo resultado com a ação do mesmo sistema e para tal é imprescindível o estudo para várias superfícies. Neste

seguimento, também se pretende apresentar, comparar e qualificar diferentes algoritmos. Usando métodos de ML e ANN, com e sem supervisão, de modo a obter uma análise de viabilidade e perceber quais os óbices associados a cada um deles, quando aplicados ao problema em mãos.

Concluindo, o contributo vital deste documento é científico e prático, onde são agrupados vários conhecimentos de recursos informáticos com fundamentos matemáticos, com a meta na aplicação do mesmo em contexto de indústria.

#### 1.4 ESTRUTURA DO DOCUMENTO

No Capítulo 1 é apresentado o tema da dissertação, o seu objetivo principal com uma introdução da sua implementação, as motivações mais relevantes para o estudo do mesmo e as contribuições que o algoritmo poderá ter com a sua aplicação e devida implementação.

O Capítulo 2, são apresentadas noções, conceitos e algumas curiosidades relevantes para melhor perceção do leitor no que diz respeito ao problema descrito. Retrata métodos e processos, populares no desenvolvimento de algoritmos de ML e ANN. Tem presente uma explicação do que é visão computacional, como surge, exemplos e aplicações. Também é sintetizado, o que é o processamento de imagem e a sua utilidade. Por fim, é exibido de maneira detalhada o algoritmo proposto.

Capítulo 3 é mais teórico, expõe detalhadamente os métodos matemáticos utilizados para executar transformações e conversões de dados, de modo a preceder a extração das propriedades e características da imagem, mas também apresenta os Data Set (DS) utilizados para tal, ou seja, a coleção de imagens usada como modelo para a implementação do algoritmo.

No que lhe concerne, o Capítulo 4 faz uma explicação detalhada dos vários métodos de ML utilizados para classificação, a maneira como operam, as suas características.

O Capítulo 5, apresenta de que forma foram implementados e construídos os diferentes tipos de métodos utilizados.

No Capítulo 6 é indicado qual o critério utilizado para efetuar a qualificação, apresentando uma visão global dos dados obtidos, efetuada uma análise dos procedimentos descritos no Capítulo anterior, fazendo uma comparação e discussão entre eles no que vai de encontro as valências de cada um.

Por fim, o Capítulo 7 mostra uma visão global do projeto, as suas virtudes e as suas falhas, apresentando conclusões globais do que foi dimensionando e possíveis alterações ou melhoras que poderão ser acrescentadas no futuro.

# Fundamentos

Este capítulo tem como finalidade de dar contexto ao leitor através de noções, conceitos e princípios, dando assim percepção dos métodos atualmente utilizados e a tecnologia disponível. Facultando a assimilação de como é executado o processamento imagem através de uma introdução teórica, também apresneta os algoritmos de aprendizagem automática que desempenha um papel relevante no que diz respeito a estes processos.

No final, é apresentado um breve resumo, embora detalhado, dos documentos analisados.

## 2.1 NOÇÕES E CONCEITOS

Esta secção tem presente vários princípios, que ajudam a ter melhor percepção de alguns conceitos que serão utilizados ao longo da dissertação e também algumas curiosidades notáveis.

### 2.1.1 Imagem Digital

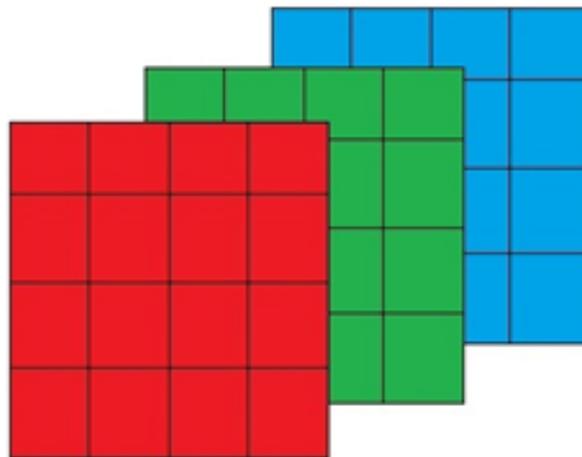
Na transição para o novo mundo digital, surgiu a necessidade de mudar, reinventar e inovar aquilo que inicialmente era conhecido. As fotografias, por sua vez, também sofreram esta transição, através da tecnologia digital, que veio facilitar esses processos.

A produção da primeira imagem digital foi no ano de 1920, pelo sistema de transmissão de imagens e cabos Bartlane, sendo os seus criadores Harry G. Bartholomew e Maynard D. McFarlane. O procedimento consistiu em “uma série de negativos em placas de zinco expostos por períodos variados, produzindo densidades variadas” [1].

Em 1957, Russell A. Kirsch inventou um dispositivo que gerava dados em formato digital, que podiam ser armazenados em um computador. Este processo era realizado através de um scanner de tambor e um tubo foto-multiplicador [1].

A obtenção de uma imagem digital é a manifestação visual de uma representação digitalmente codificada das características visuais de um objeto [2]. Sendo esta uma composição de píxeis, com um número finito e organizados para melhor percepção como a forma de uma matriz dimensional (matriz de M colunas, N linhas e D dimensões). Na Figura 2.1, consegue ter fácil percepção, que quanto maior for o número de elementos imagem maior qualidade terá a imagem e, em simultâneo, irá ocupar mais espaço de memória. Atualmente uma medida usualmente utilizada e do conhecimento geral é o megapixel, sendo cerca de 1 milhão de elementos de imagem.

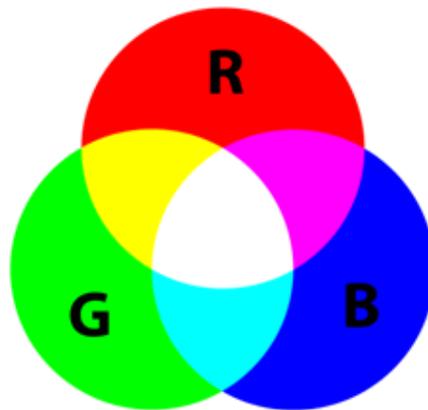
No processamento e tratamento de imagens digitais existem dois espaços de cor que tem uma importância bastante relevante: o Grey, que representa imagens a preto e branco e, o RGB, que representa imagens coloridas. O espaço denominado Grey é unidimensional, os elementos de imagem variam a sua intensidade do preto para o branco, ou seja, tons de cinza com intensidades diferentes, podendo se associar este espaço de cor às antigas imagens a preto e branco. O espaço de cor RGB é tridimensional, cada dimensão representa a cor vermelha, verde e azul, tal como no Grey são tons do mais escuro ao mais claro de cada cor. Estes valores de cada elemento de cor foram implementados para terem apenas 8 bits o que permite obter 256 valores diferentes no intervalo de 0 a 255.



**Figura 2.1:** Representação das 3 dimensões de uma imagem RGB.

A formação de imagens coloridas é a combinação dos diferentes tons de cada cor, pois através da sobreposição dos diferentes tons de cada canal da imagem obtém-se a imagem colorida pretendida como se pode observar na Figura 2.2 <sup>1</sup>.

<sup>1</sup><https://www.hisour.com/pt/rgb-color-model-24867/>



**Figura 2.2:** Sobreposição dos 3 canais.

Quando se fala numa imagem digital deve-se ter em conta o facto que todos os modelos de representação atuais, tem por base o estudo do olho humano, de modo a adquirir conhecimentos para dimensionar o espaço de cor, que permitem resolver problemas de eficientemente.

Para reproduzir uma imagem digital num determinado espaço e, ao mesmo tempo, conseguir obter da melhor maneira a sua aquisição, estudou-se essencialmente a estrutura física do olho, mas também outras variáveis, tais como, a gama do espectro visível e percepção de cor.

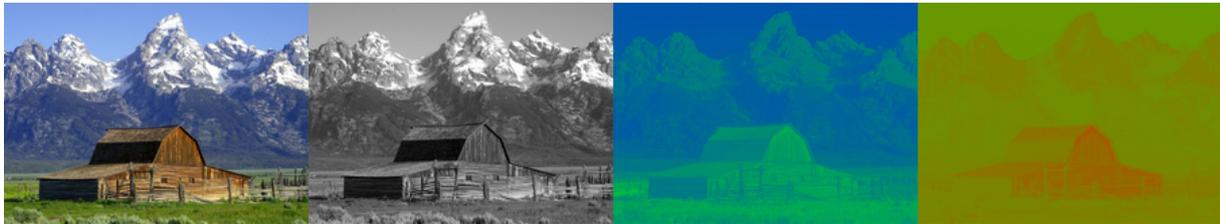
### **2.1.2 Vídeo Digital**

O vídeo digital é nada mais do que a amostragem, a uma elevada frequência, de várias imagens digitais. As imagens são denominadas frames, o ritmo a que os frames são amostrados foi refenciado por frame rate e cada frame é composto por um número de píxeis. A evolução do mesmo acontece com a evolução tecnologia, com a procura de melhor resolução e definição.

Antes do aparecimento das tecnologias digitais já existia transmissão e salvaguarda através do vídeo analógico, sendo este a representação de imagens em forma de sinal analógico. As câmaras digitais foram criadas inicialmente por sensores, Metal Oxide Semiconductor (MOS) [3]. A primeira imagem foi adquirida por um Charge Couple Device (CDD), em 1969 [4] pelo senhor Willard S. Boyle, que ganhou o prémio nobel da física [5] fundamentado pela tecnologia MOS [3].

Deste modo, é motivador analisar o espaço de cor YUV, dado que quando o sinal analógico televisivo foi alterado do antigo sinal preto e branco, para o novo sinal a cor (ambos analógicos), as antigas televisões continuaram em modo totalmente funcional, ao contrário do que aconteceu com a transferência do sinal analógico para digital, onde foi necessário comprar aparelhos para descodificar o sinal.

Na transição analógia televisiva, ao contrário do que aconteceu com a transferência do sinal analógico para digital, onde foi necessário comprar aparelhos para decodificar o sinal. A implementação executada para que as televisões não deixassem de operar foi continuar a transmitir na mesma frequência o sinal cinza que no espaço de cor YUV representa o canal Y, transmitindo em outras frequências mais duas componentes U e V onde a sobreposição dos 3 cria o sinal colorido, as televisões antigas apenas filtravam o sinal preto e branco, pois foi para tal programada, enquanto as televisões a cor efetuavam a filtragem dos 3 canais. No intuito desta dissertação é motivador fazer esta análise, porque é mais simples efectuar a identificação a olho nu do conteúdo da imagem, através do canal em tons de cinza do que pelos canais que fornecem a componente de cor, tal como se pode ver na Figura 2.3 <sup>2</sup>, por isso é que imagens em tons de cinza são melhores e mais utilizadas, para executar o processamento de imagem com mais eficácia.



**Figura 2.3:** Decomposição do espaço de cor YUV.

O espaço de cor YUV é utilizado tanto para imagem como para vídeo devido às suas propriedades serem bastante motivadoras. O mais relevante a reter é o facto que Y é a componente denominada luma (luminância) e o UV são denominados croma (crominância), U a projeção do azul e o V a projeção do vermelho.

### 2.1.3 Visão Humana

A percepção visual é a capacidade do ser humano que lhe permite interpretar o meio ao seu redor, através da interação da luz refletida pelos objetos. Esta aptidão não é única do ser humano, mas também de outros seres, o modo como é interpretada pelo olho é que varia, sendo que o olho humano assimila cores dentro de uma certa gama de comprimentos de onda, denominado espectro do visível.

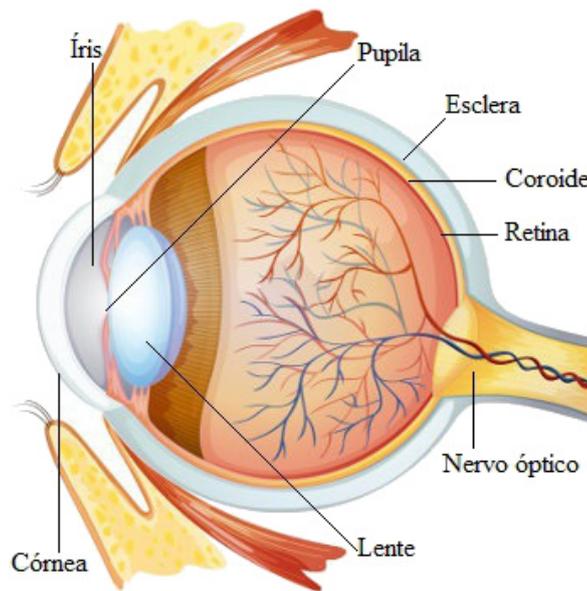
O olho humano pode-se dizer ser um “sensor” ótico bastante complexo, porque hoje em dia ainda existem bastantes processos que não foram totalmente esclarecidos, pois, o funcionamento “mecânico” é conhecido, mas o “processamento de imagem” ainda levanta muitas questões.

---

<sup>2</sup><https://en.wikipedia.org/wiki/YUV/media/File:Barn-yuv.png>

### *Estrutura*

Através do estudo do processo “mecânico” pode-se identificar, nomear e relacionar vários componentes deste “sensor”, os mais relevantes: a córnea, que envolve todo olho de modo a proteger do ambiente exterior; a lente cristalina, que está diretamente interligada a músculos e fibras, que a fazem mudar de forma conforme a zona que o ser humano pretende focar e desfocar; a retina, onde estão localizadas as células óticas que descodificam a informação a ser enviada para o cérebro e o nervo ótico, este nervo tem a função de enviar a informação para o cérebro, como se pode verificar na Figura 2.4 <sup>3</sup>.

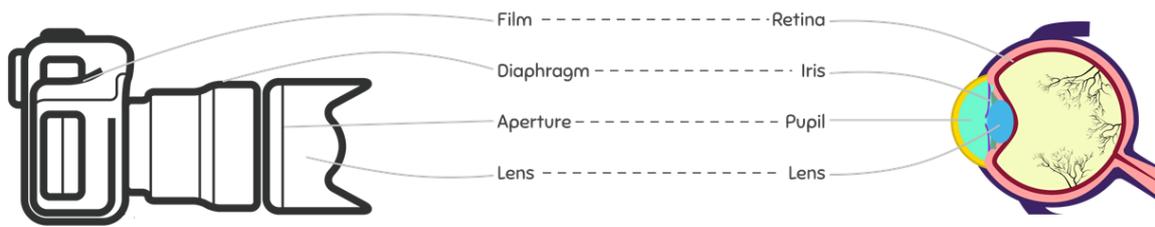


**Figura 2.4:** Imagem de um corte vertical do olho humano.

Da análise do olho humano foram retiradas várias informações úteis para a aquisição de uma imagem por uma máquina eletrônica, consegue-se fazer certas comparações entre a estrutura do olho humano e a estrutura de uma máquina de vídeo ou de fotografia. A córnea é relacionada com o invólucro da máquina, o que protege o interior do exterior, a lente cristalina é facilmente comparável a lente da máquina, os músculos e fibras estão diretamente relacionados com o diafragma da máquina, que executa o foco da imagem, mas também faz zoom na imagem, a retina a zona onde se encontram os sensores óticos. Pode-se ter percepção destas comparações com a análise da Figura 2.5.

Um dado curioso quando se compara uma máquina ao olho humano, é que o olho humano tem mais células óticas junto ao seu eixo numa zona designada de fovea, do que no resto da retina, tem um ponto cego exatamente onde se encontra a ligação ao nervo ótico, enquanto numa máquina por norma tem os sensores óticos uniformemente distribuídos e não tem ponto cego.

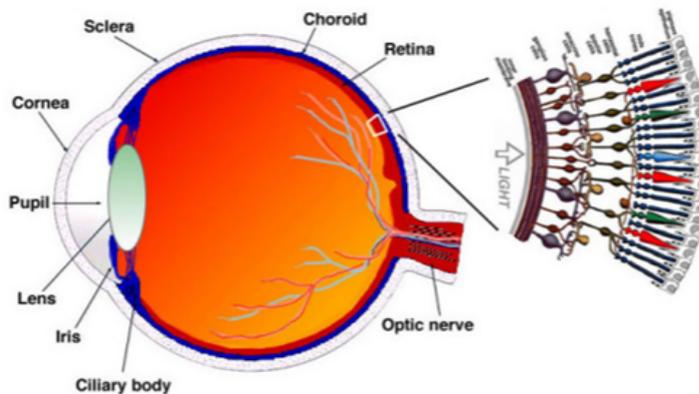
<sup>3</sup><https://mundoeducacao.uol.com.br/biologia/estrutura-interna-dos-olhos.htm>



**Figura 2.5:** Imagem que compara uma máquina ao olho humano.

### *Composição da Retina*

Relativamente à composição da retina, é essencialmente composta por duas categorias de células óticas distintas, sendo estes designados de bastonetes e cones. Os bastonetes são responsáveis pela visão escotópica, isto é, a compreensão dos tons de cinza, que permite ver com escassez de luz, os cones são responsáveis pela visão fotópica, ou seja, a percepção de cor que é mais sensível a comprimentos de onda na zona do vermelho, verde e azul como está ilustrado na Figura 2.6 <sup>4</sup>. Através da aquisição destes conhecimentos, consegue-se ter percepção que o implementado do RGB, não foi estimado arbitrariamente, mas sim com foco nas características do olho humano.



**Figura 2.6:** Imagem do corte vertical do olho e retina.

### *Espetro Visível*

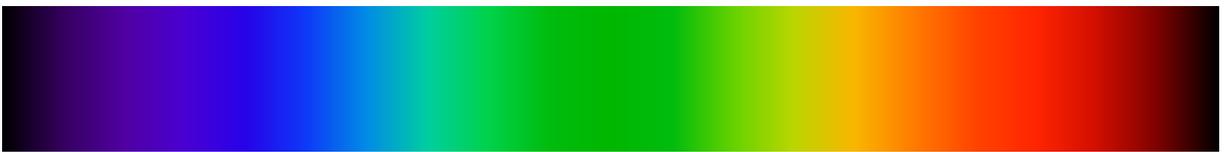
O ser humano tem por norma sensibilidade na gama do espectro onde a radiação é composta por fótons com comprimentos de onda entre os 380 – 750 nm [6], o mais sensível no intervalo entre os 500 – 565 nm, correspondendo este intervalo à zona dos verdes. No caso geral o comportamento é o apresentado, mas cada ser humano difere e estas características variam de pessoa para pessoa. Os intervalos das cores que se

<sup>4</sup><http://luztecnologiaarte.weebly.com/luz-e-fisiologia-da-visatildeo.html>

encontram dentro deste espectro estão descritos na Tabela 2.1 e na Figura 2.7 <sup>5</sup> pode-se visualizar o espectro.

**Tabela 2.1:** Comprimentos de onda máximos e mínimos, em  $\approx$  nm, decada cor visível pelo olho humano.

Cor		Violeta	Azul	Ciano	Verde	Amarelo	Laranja	Vermelho
Comprimento de onda	Min	380	440	485	500	565	590	625
	Max	440	485	500	565	590	625	740



**Figura 2.7:** Representação do espectro das cores visíveis.

### *Percepção de Cor*

A compreensão das cores é mediada pelo complexo processo de estimulação dos neurónios, este estímulo ocorre quando a luz incide em diferentes foto-receptores no olho. As características que permitem que a visão humana tenha diferentes percepções da cor são fulcralmente quatro:

**1. Luminosidade:**

A quantidade de luz presente no local.

**2. Brilho**

A quantidade de energia que irradia o objeto.

**3. Cor dominante**

Sensação de cor que o objeto transmite.

**4. Saturação ou Grau de pureza**

Porção de cor existente no elemento relacionada com o tom de cinza-médio.

## 2.2 VISÃO POR COMPUTADOR

A visão por computador ocorre da junção de tecnologias que tinham até à altura propósitos e finalidades diferentes, mas com a junção da mesma deram a origem a uma nova tecnologia inovadora com múltiplos usos. Esta nova forma de pensar, este novo meio de resolver problemas, veio apresentar várias soluções com a aplicação de uma

<sup>5</sup>[https://en.wikipedia.org/wiki/Visible\\_spectrum/media/File:Spectrum.svg](https://en.wikipedia.org/wiki/Visible_spectrum/media/File:Spectrum.svg)

câmara a um sistema, usando a mesma para obter informações. Um princípio básico de um sistema visual é suprimir a resposta a características que ocorrem com frequência, ao mesmo tempo que se mantém sensível às características que se desviam da norma [7].

Larry Roberts é denominado o pai desta nova área científica [8], tendo na sua tese de doutoramento (1960), discutido a possibilidade de fazer extração de informação 3D, por perspectiva 2D [9]. Em 1966, Seymour [10], acreditou que seria possível simular um sistema de visão humana, atando uma câmara a um computador. Após esta ideia inicial, mais tarde em 1978, investigadores deram conta que havia necessidade de efetuar uma análise mais "low-level", tais como *edge detection* e *segmentation*. Um dos investigadores com um trabalho consideravelmente relevante neste contexto foi David Marr. no Massachusetts Institute of Technology (MIT), fazendo uma abordagem *bottom-up* para compreender o cenário do seu estudo [11].

Esta tecnologia tem como finalidade poder criar modelos comportamentais de características do ser humano em sistemas, sendo estas características de conteúdo visual. Do ponto de vista científico, a visão por computador é o processamento de imagem e implementação de processos matemáticos e físicos, que estudam os dados da imagem. Do ponto de vista de engenharia, esta tecnologia ajuda na criação de sistemas autónomos. Contudo, de um ponto de vista mais objetivo e analítico a visão por computador é relacionar um ambiente 3D através da aquisição de informação em 2D, principalmente usando o conceito de padrão.

Algumas aplicações:

- Detecção facial;
- Detecção de Objetos;
- Navegação;
- Controlo de processos.

### 2.3 PROCESSAMENTO DE IMAGEM DIGITAL

O processamento de imagem digital é o nome dado ao processo de tratamento de imagens digitais através de um computador composto por 3 sub-processos, "*low-level*", "*mid-level*" e "*high-level*" [12]:

- **low-level:** é caracterizado pelo facto de as entradas e as saídas serem ambas imagens digitais.
- **mid-level:** é caracterizado pela elaboração de tarefas como segmentação, reconhecimento e particionamento.
- **high-level:** é caracterizado por processos de classificação, ou seja, perceções se os resultados fazem sentido.

Os avanços que permitiram que a imagem fosse digitalizada e computadorizada foram os seguintes:

- a criação dos transístores nos Laboratórios Bell em 1948 [12];
- o nascimento das linguagens de programação de alto nível Common Business-Oriented Language (COBOL) e Translator Formula (FORTRAN), nas décadas de 1950 e 1960;
- a invenção do Integrated Circuit (IC) pela Texas Instruments em 1958;
- o desenvolvimento de sistemas operativos no início do ano 1960;
- o avanço no estudo microprocessador pela Intel em 1970;
- introdução do conceito de computador pessoal por parte da International Business Machines (IBM) em 1981;
- a miniaturização progressiva de componentes iniciada na década de 1980.

No que diz respeito a este processo, como se pode verificar pelo que foi descrito acima, a criação do conceito de processamento de imagem ocorreu entre a décadas de 60 e 80 e, progressivamente com o surgimento de tecnologias que permitiram o seu desenvolvimento.

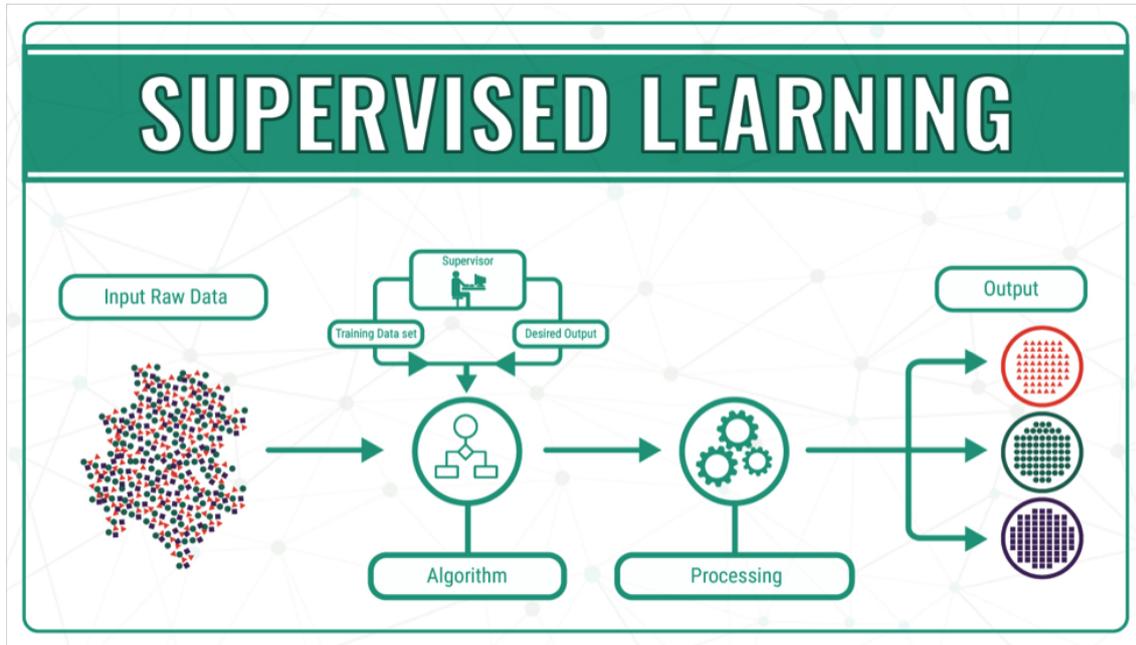
Um dado motivante do impacto que este conceito teve na área da medicina, foi o desenvolvimento do Computerized Axial Tomography (CAT), processo em que um anel de detetores circunda um objeto ou um paciente e uma fonte de Raios-X concêntrica com anel detetor que girando em torno do objeto, passando os Raios-X pelos objetos, coletados na extremidade oposta pelos detetores. A tomografia consiste em algoritmos que usam os dados detectados para construir uma imagem que representa a parte exposta do objeto, o que hoje em dia chamamos de fazer um Raio-X. O estudo da tomografia foi desenvolvido por Godfrey N. Hounsfield e Professor Allan M. Cormack, que compartilharam o Prémio Nobel da Medicina em 1979 [12].

## 2.4 INTELIGÊNCIA ARTIFICIAL

A inteligência artificial é investigado há muitos anos e, atualmente, é visto como um dos principais objetivos da evolução tecnológica, pois permite criar sistemas e processos, que conseguem ter ações próprias sem necessidade de manipulação humana, o que veio facilitar muitas coisas atualmente. Um exemplo simples de inteligência artificial, que todos nós utilizamos no dia-a dia, é quando tiramos um auto-retrato com o nosso “smartphone”. Quando vamos tirar o auto-retrato, existe um algoritmo inteligente que consegue detetar onde se encontra a face do ser humano, delineando-a com um quadrado e, ao mesmo tempo, focando esse local como prioridade.

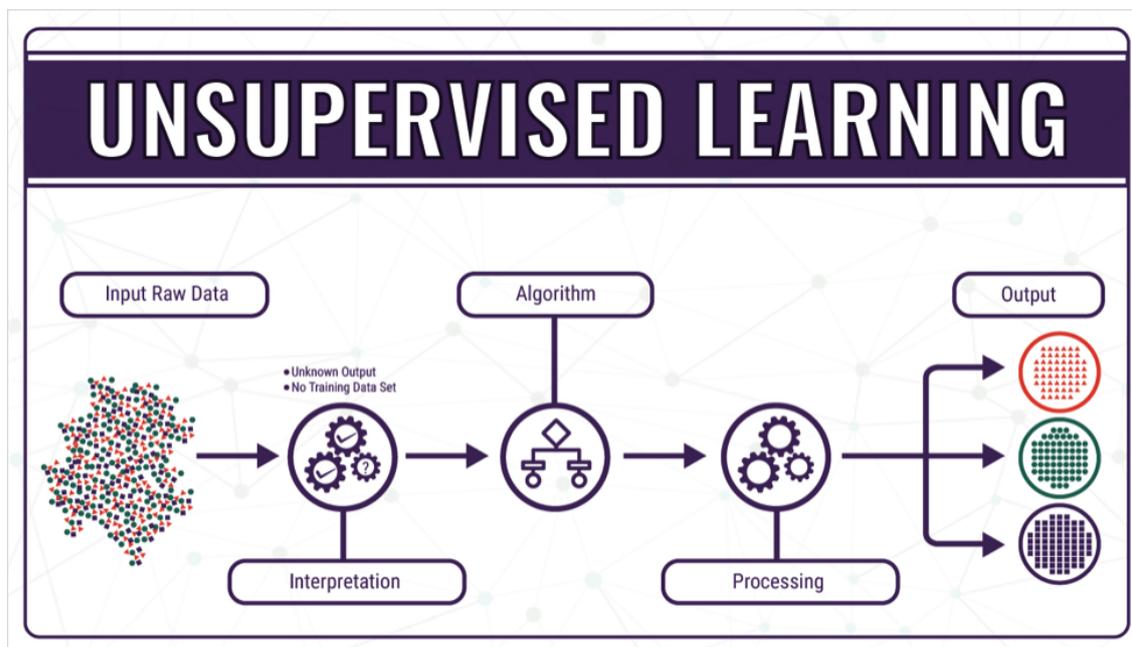
### 2.4.1 Aprendizagem máquina

Nos métodos de ML existem diversas maneiras de abordar o problema, ou seja, pode ser com ou sem recurso a anotações. No caso dos algoritmos com acesso a dados prévios, denominados algoritmos com supervisão, constroem um modelo baseado em dados que irão efetuar o treino do mesmo, de modo a fazer previsões ou terem decisões [13].



**Figura 2.8:** Esquemático dos métodos de aprendizagem máquina com supervisão [14].

Os algoritmos sem recurso a anotações, são denominados por algoritmos sem supervisão e tem como base estabelecer padrões através de um certo critério [15], isto é, através das características recebidas como entradas tentam estabelecer padrões entre os dados fazendo agrupamentos consoante o número de classes pretendidos pelo utilizador. São métodos mais complexos e com uma capacidade de falhanço maior, contudo tem outras valências, principalmente o fato de não haver necessidade de dados prévios para operarem.



**Figura 2.9:** Esquemático dos métodos de aprendizagem máquina sem supervisão [14].

#### 2.4.2 Redes neurais artificiais

As ANN são modelos inspirados no sistema nervoso central, produzidas computacionalmente e capazes de fazer aprendizagem máquina. Tem o objetivo fulcral no reconhecimento de padrões, de modo a poder fazer classificação, com recurso de uma relação matemática, em vez de efetuar o agrupamento como nos métodos ML. Quando comparados com as ANN, os modelos de aprendizagem reforçada, isto é, um sistema com supervisão que obtém melhorias, tanto com os resultados positivos como com resultados negativos (falhas de classificação), o que força a rede a aprender e adaptar-se temporalmente.

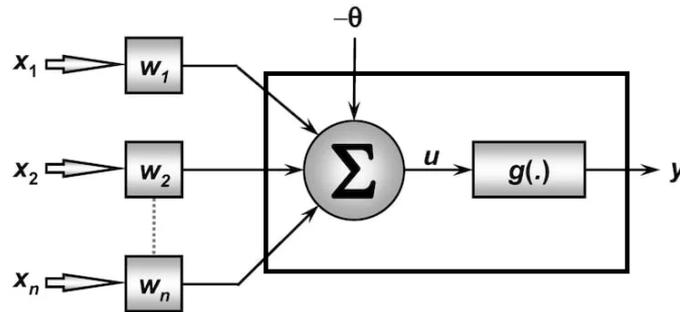
As ANN e os métodos ML, fazem parte do que é denominada inteligência artificial, ambas as áreas dedicadas ao dimensionamento de algoritmos e métodos permitem uma máquina, sistema aprender algum comportamento ou fazer o aperfeiçoamento do desempenho de alguma tarefa.

Dois exemplos simples bem presentes no dia a dia de todos, no nosso smart phone são:

- Reconhecimento de voz;
- Corretor de escrita manual.

Uma ANN simples é essencialmente composta por neurões, sendo cada neurão composto por uma função de agrupamento ou sumadora e uma função de ativação,

como se pode ver na Figura 2.10 <sup>6</sup> no retângulo.



**Figura 2.10:** Exemplo matemático do dimensionamento de um neurônio.

A função de soma, como o próprio nome indica tem como principal finalidade efectuar a soma de todos os sinais de entrada tendo em conta  $\theta$ , sendo este o limiar de activação. Esta produz à sua saída o potencial de activação  $u$ , sendo este dado como entrada na função de activação, que toma a decisão de deixa passar o sinal para a saída ou modificar o valor dado como entrada,  $u$ .

Sendo forma como o cálculo do potencial de activação é efectuada é a seguinte:

$$u = \sum_{i=1}^n w_i x_i - \theta \quad (2.1)$$

### *Funções de ativação*

A escolha da função de ativação depende muito do problema que se pretende resolver, existem algumas funções utilizadas nas mais variadas áreas da inteligência artificial, também se pode optar pela criação da mesma, se for mais de encontro com o problema em mãos, mas as existentes são facilmente implementáveis e eficazes. Algumas delas descritas a seguir.

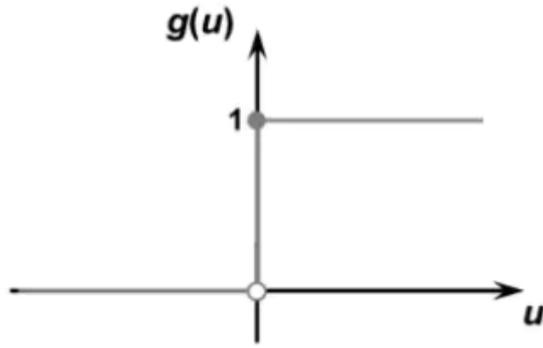
### **Degrau**

#### *Degrau simples*

A mais simples e frequentemente usada na gíria, cujo ativação é efetuada quando a entrada é igual a um e desativada quando igual a zero, como representado na expressão seguinte:

$$g(u) = \begin{cases} 1 & \text{se } (u \geq 0) \\ 0 & \text{se } (u < 0) \end{cases} \quad (2.2)$$

<sup>6</sup><https://www.oficinadanet.com.br/tecnologia/25007-o-que-sao-as-redes-neurais-artificiais>

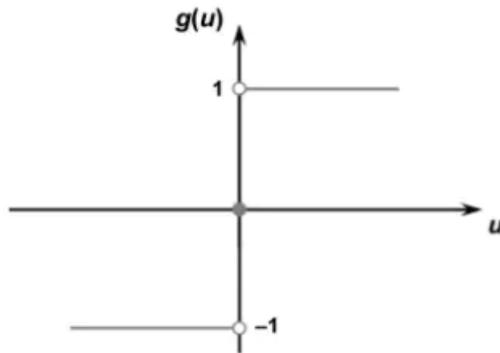


**Figura 2.11:** Exemplo gráfico do degrau simples.

### *Degrau bipolar*

Também denominada por função de ativação de sinal muito semelhante à anterior, onde apenas altera desativação feita por um sinal a menos um, como está representado na expressão seguinte:

$$g(u) = \begin{cases} 1 & \text{se } (u \geq 0) \\ -1 & \text{se } (u < 0) \end{cases} \quad (2.3)$$



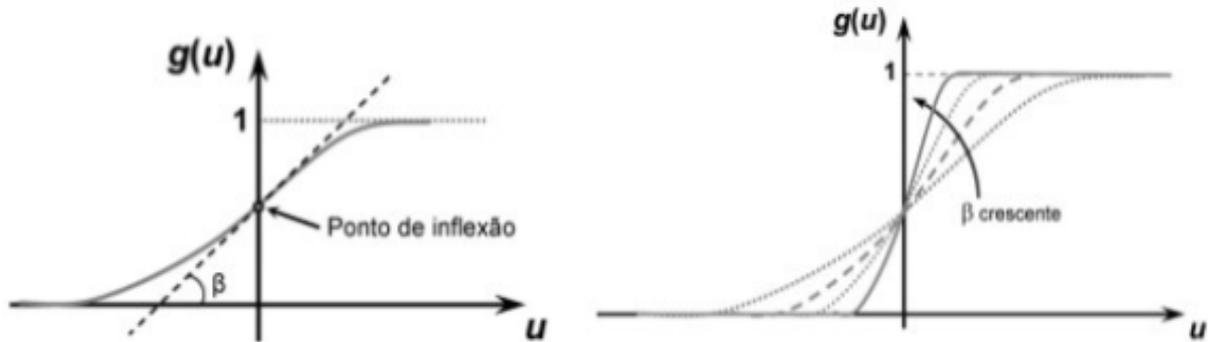
**Figura 2.12:** Exemplo gráfico do degrau bipolar.

### **Logística**

Nesta função de ativação é necessário ter em conta que, a ativação não é feita de modo repentino, ou seja, tem apenas dois estados, apenas existindo uma transição. Esta é ativa a 1 e desativa a 0, sendo que, parâmetro  $\beta$  é dimensionado através de testes, o que interfere diretamente na velocidade de transição de estado, como se pode verificar

pela expressão e Figura seguinte:

$$g(u) = \frac{1}{1 + e^{-\beta u}} \quad (2.4)$$

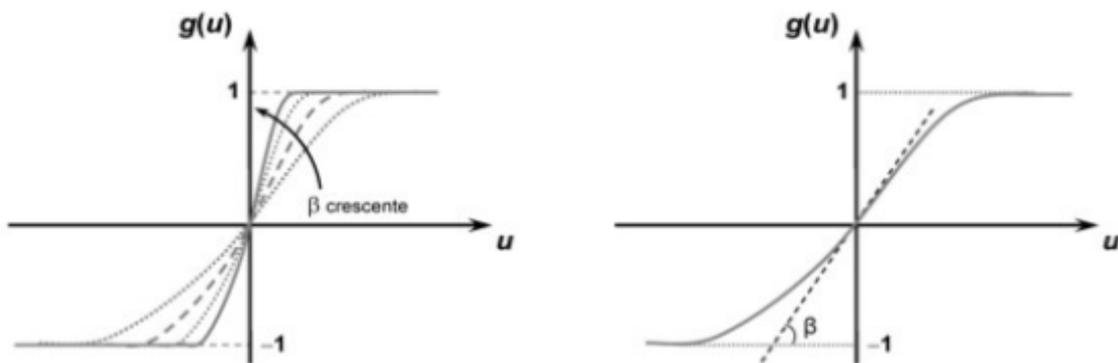


**Figura 2.13:** Exemplo gráfico da função de ativação logística.

### Tangente hiperbólica

Está é muito semelhante à anterior, apenas altera o limite de desativação, que passa a ser menos um, sendo o seu dimensionamento feito através da seguinte expressão:

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}} \quad (2.5)$$



**Figura 2.14:** Exemplo gráfico da função de ativação hiperbólica.

## 2.5 REVISÃO DA LITERATURA CIENTÍFICA

Os documentos lidos, foi efetuada inicialmente a leitura da tese do colega Rui Carapinha [16], que fez uma abordagem motivante ao tema, testando ou idealizando algumas das possíveis opções para a resolução do problema, recorrendo a vários tipos de abordagens.

O documento "*The Power of GMMs: Unsupervised Dirt Spot Detection for Industrial Floor Cleaning Robots*" [17], apresenta uma abordagem ao tema com o uso de GMM's, utilizando um método de ML, onde recorrendo a um certo número de propriedades retiradas à imagem é possível fazer a classificação de que é solo e chão, por uma classificação em probabilidade.

O documento "*Cascaded Machine-Learning Technique for Debris Classification in Floor-Cleaning Robot Application*" [18], expõe uma abordagem ao tema, através de Convolutional Neural Network (CNN) com extração de características da imagem e ainda com um vetor de ML de support, onde é possível efetuar uma classificação se a sujidade encontrada é líquido ou sólido, recorrendo a rótulo de previsão, sendo este tipo de abordagem com supervisão, isto é, tem recurso para treino de um conjunto de características de sólidos e líquidos.

O documento "*Vision-Based Dirt Detection and Adaptive Tiling Scheme for Selective Area Coverage*" [19], exhibe um procedimento de 3 fases, é inicialmente aplicado a imagem um reconhecimento de padrões, onde existe um algoritmo que se preocupa com a descoberta automática de regularidades nos dados por meio do uso de algoritmos de computador e com o uso dessas regularidades para realizar ações como classificar os dados em diferentes categorias [20]. A deteção de arestas com aprimoramento, ou seja, a identificação de diferenças de intensidades com certo padrão, sendo esta ferramenta fundamental no processamento de imagens, visão de máquinas e visão computacional, particularmente nas áreas de deteção de recursos e extração de recursos [21]; por fim, uma redução de ruído, que tende a alterar os sinais em maior ou menor grau. O algoritmo de ortogonalização de sinal e ruído local pode ser usado para evitar alterações nos sinais [22], em que neste caso os sinais são os valores dos pixels da imagem.

No documento "*A Visual Dirt Detection System for Mobile Service Robots*" [23], a proposta indicada é fundamentalmente um filtro de ruído espectral, que consiste em analisar o espectro de cada imagem e obter o residual espectral [24] e um *Dirt Thresholding*, cujo método mais simples de segmentar imagens. A partir de uma imagem em tons de cinza, a limiarização é usada para criar imagens binárias [25].

O documento "*Autonomous Dirt Detection for Cleaning in Office Environments*" [26], é outra proposta bastante semelhante à descrita no documento anteriormente referenciado, a diferença está no facto de terem acrescentaram ao algoritmo uma visão

de perspectiva normalizada, ou seja, cada imagem em cena é, portanto, uma versão plana da mesma [27], aplicando este método a um robô.

O documento "*Robot Perception of Static and Dynamic Objects with an Autonomous Floor Scrubber*" [28], apresenta todo o plano de um robô que efetua limpeza automática em locais de interesse, onde a parte mais relevante é o algoritmo de detecção de sujidade baseado em GMM, muito semelhante ao "*The Power of GMMs: Unsupervised Dirt Spot Detection for Industrial Floor Cleaning Robots*" [17].

O curso "*Machine Learning e Data Science com Python*" [29], instrui concisamente o que diz respeito aos algoritmos de ML, fazendo também uma breve introdução a Programação em python, programação orientada a objetos, bibliotecas mais úteis do python 3, focando-se em 3 algoritmos diferentes de ML, o KNN, as ANN e Kmenas, explicando bem as possíveis utilidades e as limitações de cada um deles.

## 2.6 CONCLUSÃO

Numa visão global, todos os documentos lidos tiveram o seu contributo para a escolha do caminho a prosseguir, alguns métodos não foram avaliados, para a primeira análise verificou-se que não era o caminho a seguir, tendo como base principal de trabalho os artigos: "*The Power of GMMs: Unsupervised Dirt Spot Detection for Industrial Floor Cleaning Robots*" [17] e "*Robot Perception of Static and Dynamic Objects with an Autonomous Floor Scrubber*" [28], na medida em que apresentam uma estratégia motivante na abordagem ao problema.

Os conhecimentos adquiridos ao longo do curso foram também de elevado interesse, pois permitiram de uma maneira mais criativa e mais autónoma explorar outros métodos que de igual modo foram analisados como potenciais aplicações, ao problema da detecção de sujidade por um sistema de visão computacional.

## Pré-processamento

Nesta dissertação optou-se por fazer um tratamento prévio da imagem, denominado pré-processamento, isto é, antes de desenvolver a parte de classificação cujo objetivo é localizar zonas de destaque na imagem, priorizou-se a simplificação da informação. Desta forma, melhorar algumas valências no desempenho do algoritmo, pois ao simplificar a informação cria-se uma espécie de filtragem que permite que a seguinte parte possa ser realizada de uma maneira mais rápida e menos complexa, filtrando assim o que inicialmente pode ser um elevado aglomerado de dados num conjunto menor da mesma informação.

### 3.1 CONJUNTO DE DADOS

No seguimento de uma revisão do conhecimento científico acerca do problema em questão, verificou-se que existem alguns DS de acesso aberto, bastante utilizados para o dimensionamento e teste de algoritmos de deteção de sujidade. No artigo “The Power of GMMs: Unsupervised Dirt Spot Detection for Industrial Floor Cleaning Robots” [17], é fornecido um repositório online <sup>1</sup>, onde é possível descarregar os DS usados, fornecidos por Automation and Control Institute (ACIN) e Institut für Produktionstechnik und Automatisierung (IPA), denominado se assim ACIN data e IPA data.

Estes conjuntos, fornecem uma quantidade bastante alargada de imagens em diferentes ambientes, o que proporciona a possibilidade de verificar o comportamento em superfícies variadas. Além disso, também tem presente o groundtruth das mesmas, ou seja, é possível fazer testes tanto com supervisão como sem supervisão. Estes DS, ainda disponibilizam os resultados obtidos pelos autores, o que é excelente para obter comparação com os resultados alcançados.

---

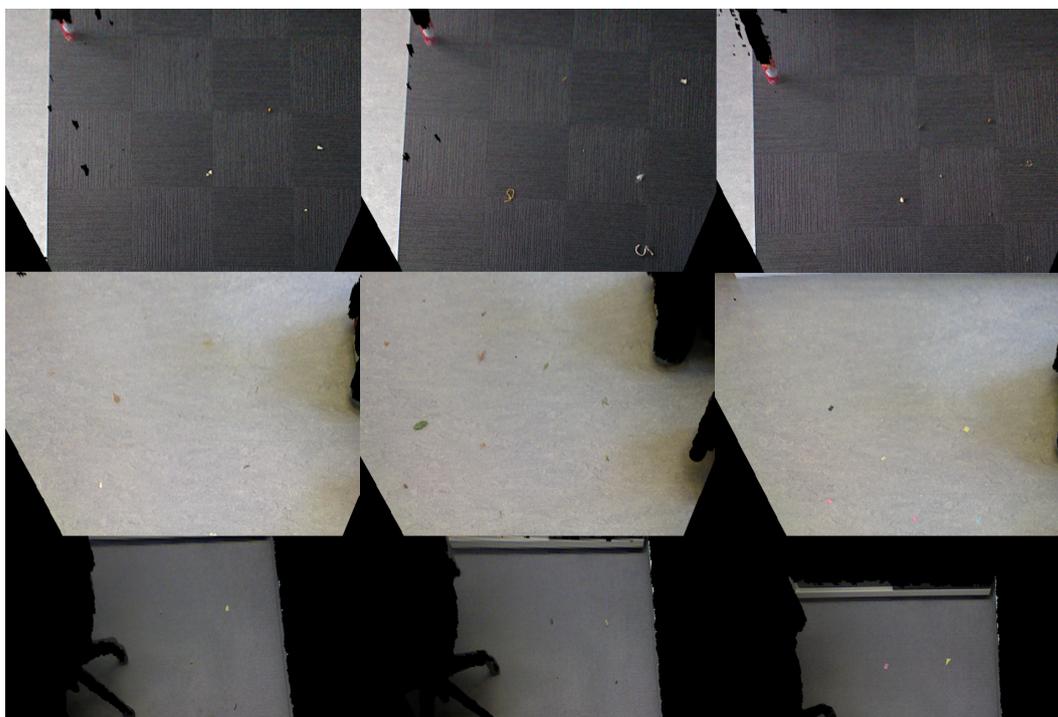
<sup>1</sup><https://goo.gl/6UCBpR>

Nas Figuras 3.1 e 3.2, estão presentes imagens existentes em ACIN data e IPA data, respetivamente.



**Figura 3.1:** Exemplo das imagens presentes no ACIN data.

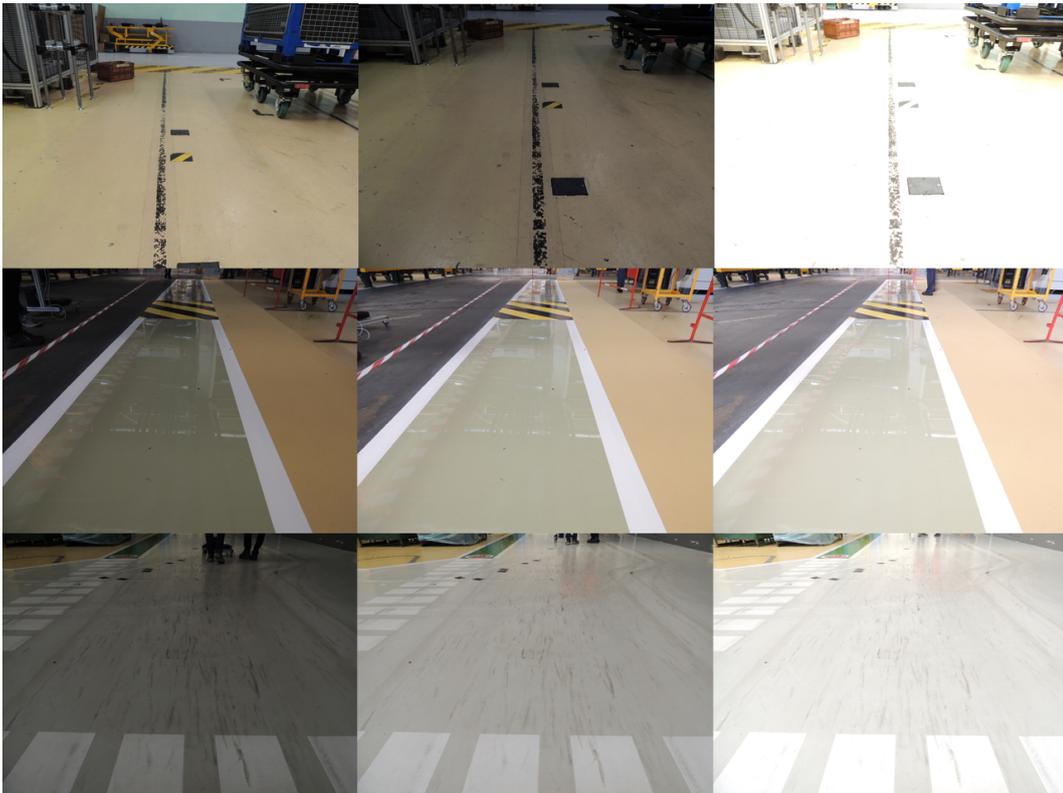
Através da Figura 3.1, consegue-se ter noção do tipo de imagens presente nestes conjuntos de dados, com imagens que desafiam a criar algo robusto, visto que as superfícies são irregulares e as imagens não têm a melhor qualidade, no que diz respeito à sua definição, bem como, também estão presente diversas categorias de sujidade com várias dimensões.



**Figura 3.2:** Exemplo das imagens presentes no IPA data.

Na Figura 3.2, pode-se visualizar que as imagens presentes neste DS em questão, têm fundos mais uniformes, embora a categoria de sujeidade seja de dimensões reduzidas. Com estes dois DS, consegue-se testar o algoritmo em cenários particularmente contrastantes, o que fornece e desafia a criação de um método de ML.

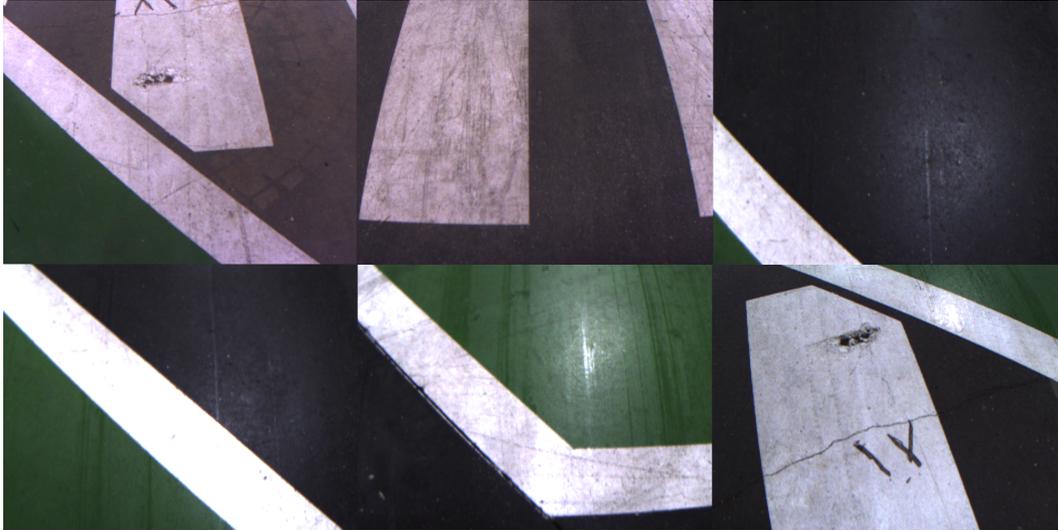
Foram fornecidos também conjuntos de imagens do pavimento da fábrica da Atena Automação Industrial, Lda e da Renault Cacia, S.A, empresas com interesse no projeto. Os DS apresentados tem falta de imagens com sujeidade, retratando na sua maioria o pavimento, porém com estas imagens pode-se pôr à prova o algoritmo no contexto em que terá utilidade.



**Figura 3.3:** Exemplo das imagens retiradas na fábrica da Renault Cacia, S.A.

Este conjunto de imagens da Figura 3.3, foi capturado através de uma câmara de fotografar de definição elevada, possui um vasto número de cenários e as capturas, cuidadosamente realizadas, apresentavam diferentes tipos de luminosidades em vários tipos de pavimento presentes, o que facilitou o trabalho, no sentido em que não existe necessidade de introdução de luminosidades artificiais para testar os métodos de classificação.

Esta coleção da Figura 3.4, tem um elevado número de imagens de pavimento, onde não se verificam grandes alterações no padrão, a sujeidade é escassa e onde o desgaste do mesmo está bastante presente. Estas capturas foram obtidas com uma Allied Vision Mako G-125C, a câmara digital que será integrada no protótipo do robô (presente na



**Figura 3.4:** Exemplo das imagens retiradas na fábrica da Atena Automação Industrial, Lda

Figura 3.5), tendo como principal finalidade o seu uso em visão industrial, devido ao facto de ser uma câmara de visão artificial super compactada (29x29 mm) e apesar das dimensões reduzidas mantendo elevados padrões de qualidade. Uma vez que é detentora da tecnologia Power-over-Ethernet, pode ser alimentada pela ligação Ethernet ou por um conector Hirose de 8 pinos da Mean Well até 24VDC.

Caraterísticas da Mako G-125C:

1. Resolução: 1280×960 pixeis
2. Sensor de captura de imagem: 1/3" CCD Sony ICX445
3. Velocidade: 30 imagens/segundo
4. Interface: GigE
5. Tamanho pixel: 3.75µm
6. Regiões de interesse (ROI)
7. Controlo de ganho e exposição
8. White balance
9. Binning (somente em versões CCD)
10. Modo gravação
11. StreamBytesPerSecond (controlo de largura de banda)

A Figura 3.5, faculta a perceção de como é feita a aquisição das imagens pela máquina, podendo-se visualizar a distância e ângulo ao pavimento a que é colocada. Esta posição permitira que retratos sejam especialmente ricos em imagens de pavimento, onde não existe uma vasta presença de objetos que não sejam o pavimento, evitando assim a presença de pontos cegos.



**Figura 3.5:** a câmara digital que será integrada no protótipo do robô.

Elaborou-se uma ferramenta que permite de uma forma bastante simplificada criar imagens com sujidade aleatória, isto é, consegue-se colocar no pavimento de uma imagem sujidade dos mais variados tipos e dimensões, possibilitando também a adição de luz artificial, apenas com o recurso a pavimentos diversos. Em alguns casos existe a necessidade de segmentação de paredes e objetos presentes, para posteriormente serem adicionadas imagens segmentadas de sujidade e modelos de luz artificial. Através deste trabalho foi possível efetuar a criação de DS, bastantes mais ricos e com melhor qualidade para os testes que validam os algoritmos. Na Figura 3.6, estão alguns exemplos destas imagens criadas artificialmente.

Estas imagens criadas permitem de uma forma direta testar os algoritmos de modo a efetuar uma validação dos modelos em ambiente favoráveis e controlados, tentando perceber o comportamento em outros ambientes não controlados.



**Figura 3.6:** Imagens reproduzidas com sujeidade artificial.

### 3.2 ESPAÇO DE COR

Neste desenvolvimento de modo a retirar da imagem digital as características foi necessário identificar qual o melhor espaço de cor a utilizar para o efeito pretendido, ou seja, estabelecer o objetivo a alcançar e tentar interpretar qual os espaços disponíveis e qual se adequa ao problema em questão de modo a obter o resultado final projetado.

Atualmente, a evolução tecnológica e a necessidade constante de melhorar funcionalidades, existe um vasto leque de espaços de cor para representação de imagens, apesar daqueles que já foram retratados anteriormente neste documento.

Quando se trabalha em visão existem vários espaços muito utilizados, tais como o Hue(Tonality) Saturation Value(Brightness) (HSV), onde são relevantes características como a tonalidade, a saturação e o brilho, sendo que cada letra da sigla significa, respetivamente, cada característica do espaço.

Este espaço tem como benefício a separação da saturação e do brilho que permite, por exemplo, fazer de uma forma eficiente a deteção de cores. No projeto em questão, a abordagem ao problema não é feita no sentido da deteção de cor, mas sim a identificação de formas de variadas dimensões, tendo como base o conceito de que estas formas serão de dimensões menores e a maioria dos elementos de imagem serão pavimento.

Na sequência do problema apresentado o espaço *Lab* foi o escolhido, este espaço destina-se ao tratamento e processamento das imagens, fazendo uma separação bastante motivadora, separando os valores que dizem respeito à luminosidade dos valores respetivos a cor. O  $L$  é uma matriz onde estão presentes os valores de luminosidade, enquanto que  $a$  é a representação da transição de azul para verde e o  $b$  é a representação da

transição de amarelo para vermelho, isolando assim os valores de luminosidade sem perder a informação relevante da cor. Este processo é efetuado com recurso ao Open Source Computer Vision Library (OpenCV), partindo inicialmente de uma imagem Blue Green Red (BGR), o mesmo espaço de cor que RGB, mas com a organização matricial inversa. Para tal, é necessário aplicar a seguinte transformação:

$$\begin{bmatrix} Z \\ Y \\ X \end{bmatrix} = \begin{bmatrix} 0.019334 & 0.119193 & 0.950227 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.412453 & 0.357580 & 0.180423 \end{bmatrix} \cdot \begin{bmatrix} B \\ G \\ R \end{bmatrix} \quad (3.1)$$

Pode-se verificar através da equação 3.1, que a transformação não é direta, sendo inicialmente convertido num novo espaço de cor XYZ a partir da imagem digital proveniente do BGR.

$$X = \frac{X}{X'} \quad (3.2)$$

$$Z = \frac{Z}{Z'} \quad (3.3)$$

sendo  $X' = 0.950456$  e  $Z' = 1.088754$

A obtenção do canal  $\mathbf{L}$  é feita através das seguintes equações:

$$L = \begin{cases} 116Y^{\frac{1}{3}} - 16 & \text{se } Y > 0.008856 \\ 903.3Y & \text{se } Y \leq 0.008856 \end{cases} \quad (3.4)$$

Os canais  $\mathbf{a}$  e  $\mathbf{b}$ , são obtidos através das seguintes equações:

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{se } t > 0.008856 \\ 7.787t + \frac{16}{116} & \text{se } t \leq 0.008856 \end{cases} \quad (3.5)$$

$$a = 500(f(X) - f(Y)) + \Delta \quad (3.6)$$

$$b = 200(f(Y) - f(Z)) + \Delta \quad (3.7)$$

onde,

$$\Delta = \begin{cases} 128 & \text{se imagens de 8 - bits} \\ 0 & \text{se imagens em vírgula flutuante} \end{cases} \quad (3.8)$$

A gama de valores obtida depois da conversão:

$$\begin{cases} 0 \leq L \leq 100 \\ -127 \leq a \leq 127 \\ -127 \leq b \leq 127 \end{cases} \quad (3.9)$$

Consoante os tipos de dados finais pretendidos, os valores são normalizados da seguinte forma:

- imagens de 8 bits :  $L = 2.55L$ ,  $a = a + 128$ ,  $b = b + 128$  ;
- imagens de 16 bits : não suportado pelo OpenCV;
- imagens de 32 bits : não necessita.

### 3.3 GRADIENTE DE COR

Os gradientes têm uma grande utilização no processamento de imagem, com a principal vantagem de retirar características. A imagem digital é uma matriz, com o cálculo do gradiente, pode-se verificar facilmente onde existe mudança de intensidade repentina, sendo essas zonas, locais onde se identificam transições. Essas alterações podem ser possíveis objetos, mudanças da tonalidade do pavimento ou sujidade, estes acontecimentos serão associados a possibilidade de ser sujidade.

#### 3.3.1 Gradientes

A definição de gradiente de imagem digital é a alteração de intensidade numa imagem, descrita por  $F(x,y)$  em ordem ao eixo das colunas e linhas. Ou seja, a representação matemática :

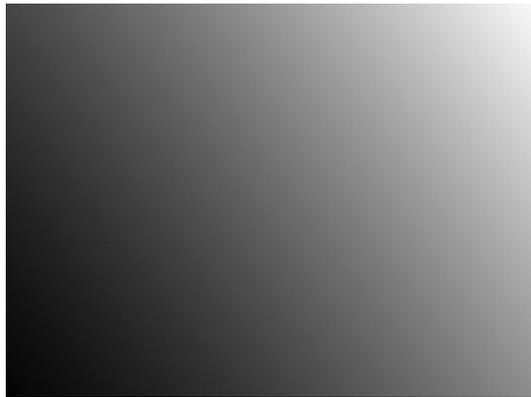
$$\nabla F = \left[ \frac{\delta F}{\delta x}, \frac{\delta F}{\delta y} \right] \quad (3.10)$$

$$\nabla F = \left[ 0, \frac{\delta F}{\delta y} \right] \quad (3.11)$$

$$\nabla F = \left[ \frac{\delta F}{\delta x}, 0 \right] \quad (3.12)$$

O gradiente é a solução numérica de equações diferenciais onde se utiliza a aproximação à derivada, ou seja, consoante o gradiente utilizado, no sentido do x, y ou diagonal, irá sempre obter-se características, que identificam mudanças de intensidade nessas direções. Depois através de operações com estas novas matrizes de valores de intensidade obtidos, consegue-se identificar melhor, visualmente ou numericamente, onde existem variações significativas e relevantes. As Figuras 3.7 e 3.8, são exemplos que melhoram a perceção de como seria o resultado do gradiente consoante diferentes direções. As mesmas têm representadas variações de intensidade a favor da direção do gradiente, onde a intensidade de elemento para elemento de imagem varia pouco inicialmente e

começa a ser cada vez mais significativa quando avaliados os elementos seguintes até ao final. Assume-se que o início da imagem seja onde a intensidade dos elementos se encontra a preto e o fim seja onde os elementos se encontram a branco. Outra observação importante a ter em consideração, é o facto das diferenças de intensidades baixas serem referentes a tons pretos e, por outro lado, as diferenças de intensidades altas serem tons cinza-claro e branco.



**Figura 3.7:** Representação visual da equação 3.10, apresentado as variações de intensidade diagonal.



**Figura 3.8:** Representação visual das equações 3.11 e 3.12, apresentado as variações de intensidade nos eixos x e y, respetivamente.

### *Normal*

Uma solução computacional, que contempla o mesmo resultado da solução analítica, para o cálculo na direção de ambos eixos, é a seguinte:

$$\frac{\delta F(x, y)}{\delta x} = F(x + 1, y) - F(x, y) \quad (3.13)$$

$$\frac{\delta F(x, y)}{\delta y} = F(x, y + 1) - F(x, y) \quad (3.14)$$

De modo a poder efetuar o cálculo das derivadas parciais, o processo computacional que é utilizado, como se pode visualizar nas equações 3.13 e 3.14, é a solução da operação de diferença entre o elemento atual e o elemento seguinte, segundo a direção do gradiente que se pretende calcular.

### *Sobel*

Outra solução é usar o conceito de convolução e núcleo, que calcula computacionalmente de forma rápida as derivadas de uma imagem. Um exemplo deste tipo de solução seria o seguinte:

$$\frac{\delta F(x, y)}{\delta x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} I \quad (3.15)$$

$$\frac{\delta F(x, y)}{\delta y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} I \quad (3.16)$$

Sendo  $I$  a representação matricial de uma imagem digital nesta operação.

Este método consiste em efetuar o cálculo de produto entre matrizes, sendo estas as matrizes presentes nas equações 3.15 e 3.16 e a imagem a qual se pretende aplicar o gradiente. O resultado obtido fornece informação de um conjunto de elementos em relação a outros. Esta lógica é feita de modo a calcular o gradiente nas direções  $x$  e  $y$ .

### *Laplacian*

Este método expõem uma maneira diferente de abordar o problema, apresentado um bom resultado também no cálculo das derivadas de segunda ordem, que também podem ser usadas para obtenção do gradiente.

$$\nabla F = \frac{\delta^2 F}{\delta x^2} + \frac{\delta^2 F}{\delta y^2} \quad (3.17)$$

A solução computacional baseia-se também no conceito de convulsão e núcleo, como se pode observar no exemplo a seguir apresentado:

$$\nabla F(x, y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} I \quad (3.18)$$

Sendo  $I$  a representação matricial de uma imagem digital nesta operação.

Esta solução representada na equação 3.18, ao nível de desempenho, é muito semelhante ao método anterior, divergindo no resultado, pois o produto entre matrizes retorna os valores das derivadas parciais de segunda ordem, nesse sentido os resultados de intensidade são ligeiramente diferentes.

### 3.3.2 Magnitudes

O cálculo da magnitude do gradiente tem uma elevada importância, porque uniformiza a informação obtida pelo gradiente, de modo que seja mais simples a interpretação do seu resultado e, em simultâneo, é visualmente mais simples de analisar.

O resultado dos gradientes por si só não transmite informação relevante, apesar de a sua ligação com outros gradientes, pelo cálculo numérico permite obter um resultado claro e conciso.

Neste trabalho o caminho escolhido foi a computação da magnitude dos gradientes  $x$  e  $y$ , com mostra na equação 3.19.

As expressões para efetuar a operação da magnitude do gradiente computacionalmente, são as seguintes:

$$\|\nabla F\| = \sqrt{\left(\frac{\delta F}{\delta x}\right)^2 + \left(\frac{\delta F}{\delta y}\right)^2} \quad (3.19)$$

Outra equação mais simples que também é usada:

$$\|\nabla F\| = \left\| \frac{\delta F}{\delta x} \right\| + \left\| \frac{\delta F}{\delta y} \right\| \quad (3.20)$$

O ângulo do gradiente, ou seja, a direção da intensidade por vezes também tem relevância em ser efetuado o seu cálculo, sendo a expressão a seguinte:

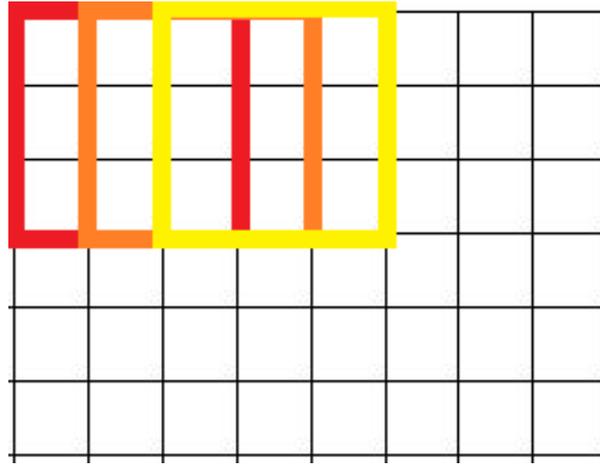
$$\theta = \tan^{-1} \left[ \frac{\delta F}{\delta x} / \frac{\delta F}{\delta y} \right] \quad (3.21)$$

## 3.4 EXTRAÇÃO DAS CARATERÍSTICAS

Com propósito de conseguir obter uma boa quantidade de características como recursos da imagem captada para análise, efetuou-se um método muito recorrente em visão por computador, a divisão da *frame* em pequenos blocos.

Esta divisão foi feita por parâmetros configuráveis e através de uma janela deslizante, iniciando o processo pela criação de um pequeno bloco logo no início da imagem, com  $n \times m$ , dimensões e deslizando na horizontal com um passo  $w$ , até percorrer toda a imagem, retirando as características de cada bloco. O exemplo da Figura 3.9 seria um bloco  $3 \times 3$  com um passo de 1, representado a vermelho a primeira posição e as restantes as seguintes posições assumidas, ou seja, as primeiras 3 iterações.

As características extraídas são a média e o desvio padrão dos elementos de imagem de cada bloco, sendo que com a aplicação de um gradiente e respectivo cálculo da sua magnitude, obtém-se uma imagem de intensidades, quando as variações são pouco bruscas pode-se analisar que representa uma superfície grande e uniforme da imagem chão ou objeto, por outro lado, quando as variações são consideravelmente bruscas é necessário identificar se nesse bloco existe a probabilidade de haver algum tipo de sujidade ou irregularidade.



**Figura 3.9:** Representação animada de uma janela deslizante.

### *Média Aritmética*

A média fornece o valor médio que representa a concentração de um determinado número de dados, podendo ser calculado através de um vetor ou de uma matriz. O bloco extraído da imagem é tratado computacionalmente como uma matriz e o cálculo da média matricial é feito através da expressão:

$$\mu(t) = \frac{1}{(n \times m)} \sum_{i=0}^n \sum_{j=0}^m x(i, j) \quad (3.22)$$

### *Desvio padrão*

O desvio padrão é definido na estatística como sendo uma medida da quantidade de dispersão ou variação de um conjunto de valores. Um desvio padrão baixo representa que um grupo dos valores tende a estar próximos da média, ou seja, para o tema abordado um valor elevado do desvio padrão transmite que aquele bloco tem um conjunto de valores muito diferentes do geral, o que pode indicar sujidade. O cálculo é feito através da seguinte expressão:

$$\sigma(t) = \sqrt{\frac{1}{n \times m} \sum_{i=0}^n \sum_{j=0}^m (x(i, j) - \mu)^2} \quad (3.23)$$

### 3.5 CONCLUSÃO

O mais relevante abordado neste capítulo é a manipulação dos dados digitais, neste caso a imagem digital. Todo este processo é efetuado para os vários canais de cada imagem segundo uma determinada ordem, ou seja, depois da alteração do espaço de cor, a imagem é dividida em 3 matrizes independentes deixando de haver sobreposição entre os canais. A seguida, são calculados para cada um dos canais os respetivos gradientes, magnitudes e sendo por fim feita a extração das características. As características finais são guardadas num vetor 2D de  $N$  comprimento, estando este comprimento diretamente associado à dimensão da janela deslizante, onde cada posição deste vetor possui o conteúdo da média e desvio padrão, na primeira e segunda posição respetivamente.

Contudo, é motivante ter o entendimento, que existem variadas maneiras de chegar ao mesmo resultado, por isso é importante mostrar os vários tipos de gradientes que foram analisados, o espaço de cor utilizado e a razão do seu estudo em relação a outros, as diferentes formas de cálculo da magnitude dos gradientes, a relação entre os valores iniciais e as características obtidas no final do pré-processamento da imagem digital. Feita a análise destes fatores, o ideal é optar por escolher aqueles que melhor encaixam no objetivo do projeto.



# Classificação

Com o intuito de aprender e como aplicar novos conhecimentos foram explorados alguns métodos simples e complexos, mas que permitem ter uma boa percepção de como os processos funcionam, sem a percepção de métodos simples é mais complicado perceber métodos mais complexos.

Os métodos de ML servem esse mesmo propósito, tendo como finalidade principal a busca por congruências que permitam caracterizar o que é pretendido, de modo a criar inteligência artificial pelo reconhecimento de padrões.

Este capítulo irá retratar essencialmente, como é possível aplicar vários algoritmos ao processamento de imagem e quais as características de cada um, explicando toda a sua construção matemática.

## 4.1 K-MEANS

O objetivo deste método consiste em efetuar uma separação de características por grupos, colocando no mesmo grupo aqueles que são similares, criando assim um agrupamento de K classes. Estas classes são um aglomerado de informação que partilham similaridade, isto é, tem qualidades equivalentes ou próximas segundo uma determinada métrica.

Este método está presente no OpenCV e nas bibliotecas do Scikit Learn, onde a métrica utilizada para separar amostras é a variância, colocando no mesmo grupo amostras com variância igual. O algoritmo K-means divide um conjunto de N amostras em K aglomerados disjuntos, estabelecendo os vários centros, estes denominados de “centróides”.

O algoritmo, o "Standard algorithm", foi dimensionado para modelação de código por impulso, em 1957 por Stuard Lloyd, apesar de a sua publicação apenas ter sido efetuada 1982 [30]. É referido com Lloyd-Forgy, porque em 1965 E.W.Forgy publicou

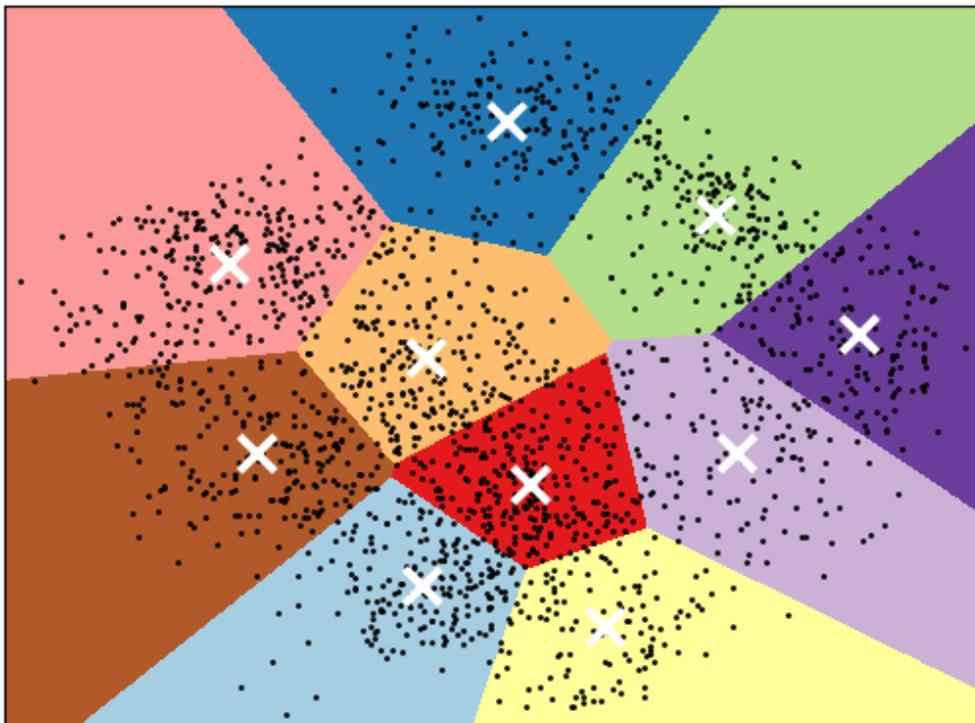
essencialmente o mesmo método [31]. Atualmente, a versão mais eficiente foi sugerida por Hartigan e Wong, em 1975 [32].

O K-means é um algoritmo de maximização de expectativa com uma matriz de covariância totalmente igual e pequena.

O método possui 3 fases:

- Inicialmente, consiste em pegar em  $k$  amostras e atribuir a cada amostra um centróide diferente, isto é, o mais próximo para cada uma;
- De seguida, cria novos "centróides", tendo como referência o valor médio das amostras anteriores, sendo que o cálculo da média de todos os valores serão o novo centro;
- Por fim, calcula a diferença entre este e os novos "centróides" e os antigos.

Estes três processos repetem-se até que os "centróides" não se movam significativamente, mas também pode terminar chegando a um valor limite por defeito ou estabelecido pelo utilizador. É importante também ter em atenção que é um método não hierárquico, ou seja, é particionado e não herda nenhum tipo de informação. É não supervisionado, definindo os limites de cada partição, grupo ou conjunto de dados, a cada iteração. Um método hierárquico tem sempre como principal recurso o conteúdo de informação herdada, tendo sido adquirida anteriormente, por exemplo, os métodos com supervisão.



**Figura 4.1:** Exemplo de um espaço de características do método K-means.

A Figura 4.1 <sup>1</sup>, é um exemplo onde se pode visualizar um conjunto de valores que foi particionado pelos diferentes *clusters* e o centro calculado para cada um dos conjuntos de características diferentes, segundo a parametrização pretendida pelo utilizador.

## 4.2 MODELO DE MISTURAS GAUSSIANAS

O GMM, tal como o K-means, é um algoritmo de *clustering*. Uma das diferenças entre ambos é o estabelecimento de fronteiras, pois o K-means é mais rigoroso, uma vez que estabelece fronteiras "inquebráveis" em relação ao GMM, isto é, a mesma amostra de uma determinada característica, pode pertencer a vários "cluster" com uma probabilidade diferente. Este problema é abordado na literatura, inicialmente em 1846 [33], embora que a sua referência é feita ao trabalho Karl Pearson, em 1894 [34].

### 4.2.1 Distribuição Gaussiana

Cada conjunto da GMM corresponde a uma distribuição gaussiana, com diferente probabilidade. A expressão de uma distribuição gaussiana de apenas uma dimensão é a seguinte:

$$N(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{(x - \mu)^2}{2\sigma^2} \quad (4.1)$$

A expressão da equação 4.1, também é conhecida por densidade de probabilidade ou por distribuição Normal, daí o uso da letra  $N$  para a representar. O parâmetro  $x$  representa o conjunto sobre o qual a distribuição é colocada, o  $\mu$  refere-se à média, que define o centro da distribuição e o  $\sigma$  controla a forma da distribuição. A distribuição gaussiana pode ser utilizada também em múltiplas dimensões.

Para espaços dimensionais, a distribuição é caracterizada por um vetor da média [Dx1] e uma matriz de variâncias [DxD], sendo D o número de dimensões da distribuição. A expressão matemática para tal é a seguinte:

$$N(x|\mu, \Sigma) = \frac{1}{2\pi^{\frac{d}{2}} \sqrt{|\Sigma|}} \exp -\left(\frac{1}{2}(X - \mu)^T \frac{1}{\Sigma}(X - \mu)\right) \quad (4.2)$$

Para duas dimensões os parâmetros das gaussianas são:

$$\left[ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1 & \sigma_{12} \\ \sigma_{21} & \sigma_2 \end{pmatrix} \right] \quad (4.3)$$

---

<sup>1</sup><https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

### 4.2.2 Algoritmo de maximização de expectativa

O algoritmo Expectation Maximization algorithm (EM), tal como K-means é um método de agrupamento, utilizado na GMM, caso exista um conjunto de dados provenientes de diferentes distribuições gaussianas, mas não existe informação de quais distribuições os dados são provenientes. Para efetuar uma resolução eficaz do problema utiliza-se o EM, ou seja, inicialmente o algoritmo começa por criar distribuições com médias e variâncias aleatórias, em seguida itera sobre dois passos:

- **E:** Através das médias e variâncias, calcula-se a probabilidade de cada dado proveniente de cada distribuição gaussiana;
- **M:** Após a obtenção das probabilidades, estas são usadas para reestimar a média e a variância das distribuições, de modo a efetuar os devidos ajustes nas mesmas.

#### *Passo E*

O GMM tem como finalidade principal alocar cada dado ao respetivo grupo, consoante a sua probabilidade de pertencer ao mesmo. A fórmula para computacionalmente obter este resultado é a seguinte:

$$r_{ik} = \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^k \pi_j N(x_i | \mu_j, \Sigma_j)} \quad (4.4)$$

Dividindo em partes as atribuições suaves, quantificando-as por um vetor de probabilidade  $r$ , ou seja, uma atribuição de um dado a um determinado grupo é feito analisando a probabilidade de pertencer ao mesmo, denominando-se por atribuição suave uma atribuição que tem uma probabilidade muito elevada de estar presente num grupo quando comparada com as restantes probabilidades. Para cada cálculo de probabilidade  $i$  é acrescentado ao vetor de probabilidades um novo valor no respetivo índice, este vetor vai ter o número de probabilidades daquele, dado correspondentes a  $k-1$ , número de grupos menos um. A soma das probabilidades do vetor tem como resultado 1.

Para formar uma distribuição de dados complexa, em vez de usar uma distribuição apenas, usa-se uma combinação linear de múltiplas distribuições gaussianas, para obter melhor resultado. Então, de modo a calcular uma mistura de gaussianas, é utilizado um conjunto de pesos para cada grupo,  $\pi_k$ . Onde  $\sum_k = K\pi_k$  e  $0 \leq \pi_k \leq 1$ , sendo este parâmetro aquele que identifica a probabilidade de um determinado dado estar no conjunto de dados  $x$ .

Através do valor do dado para aquele grupo em avaliação, este representado por  $x_i$ , calcula-se a probabilidade de esse dado estar presente na distribuição variada. Para isso, com recurso à Probability density function (PDF) de  $x_i$ , subtraindo-a com o somatório

das PDF de todos os restantes grupos, multiplicando pelo respetivo peso, obtendo a  $r_i$ , que é a probabilidade do dado, numa múltipla distribuição gaussiana.

#### *Passo M*

Este passo é efetuado em seguida ao  $E$ , após a obtenção do vetor de probabilidades faz se uma atualização dos parâmetros da GMM, através das expressões:

$$N_k^{soft} = \sum_{i=1}^N r_{ik} \quad (4.5)$$

$$\pi_k = \frac{N_k^{soft}}{N_k} \quad (4.6)$$

$$\mu_k = \frac{1}{N_k^{soft}} \sum_{i=1}^N r_{ik} x_i \quad (4.7)$$

$$\Sigma_k = \frac{1}{N_k^{soft}} \sum_{i=1}^N r_{ik} (x_i - \mu_k)(x_i - \mu_k)^T \quad (4.8)$$

Estes parâmetros inicialmente, são inicializados com um valor defeito, sendo que, apenas após algumas iterações é que os valores têm uma representação próxima daquilo que é o espaço de dados.

O parâmetro  $\pi_k$  reflete o peso daquele determinado grupo em relação ao conjunto de todos os dados, o que pode ser também avaliado como respetivo tamanho de cada grupo. Este peso é obtido através da equação 4.6, calculado para cada grupo, com o resultado da equação 4.5, em que  $N$  é o número total de todos os dados,  $N_k$  é o número de dados por cada grupo e  $N_k^{soft}$  é o peso dado pela contagem suave de cada grupo.

Após atualizar as estimativas dos pesos tem-se em conta o valor associando-o a cada  $r_{ik}$ , ou seja, todas a vezes que é calculado um novo peso de dados é multiplicado por  $r_{ik}$ , como indica a expressão presente na equação 4.7.

Por fim, é relevante realçar que aquando da atualização dos parâmetros  $\mu_k$  e  $\Sigma_k$ , utiliza-se o número efetivo de observações nesse grupo como denominador, o  $N_k^{soft}$ , em vez de dividir pela soma total dos dados daquele grupo  $N_k$ .

### 4.3 K VIZINHOS MAIS PRÓXIMOS

Dos vários métodos de ML este faz parte dos denominados por aprendizagem com supervisão, ou seja, para obter o resultado é necessário fazer um treino (no caso de KNN quando é referido ao longo do documento a treino, é a construção do banco de dados, que irá servir como referência), onde são fornecidas tanto as entradas como as saídas

para o modelo aprender quais as saídas das respectivas entradas, de modo a calcular a resposta correta quando testado e fazer estimativas a partir daí, pois o sistema aprende diretamente das informações corretas. Este foi desenvolvido por Evelyn Fix e Joseph Hodges em 1951 [35], tendo sido mais tarde otimizado por Thomas Cover [36].

O KNN é um algoritmo que permite fazer a classificação de valores numéricos com base em exemplos de treino, que estejam mais próximos no espaço de características. Para tal, é necessário ter um conjunto de valores exemplo para treino, definir a métrica que será usada para efetuar o cálculo da distância dos exemplos de treino e por fim o número de  $K$  vizinhos que serão considerados pelo algoritmo.

No que diz respeito à métrica a usar, o KNN usa o cálculo de distância. Pode ser realizada por vários métodos, por exemplo, a distância de Minkowski, Manhattan ou Euclidiana.

#### *Distância Euclidiana*

Esta distância considerada por Euclides, professor, matemático platónico e escritor da Grécia antiga, tem por definição que o menor caminho entre pontos é a menor reta possível. O cálculo é efetuado através de:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (4.9)$$

#### *Distância Manhattan*

Apresentada por Hermann Minkowski no século XIX, consiste no somatório das projeções, vertical ou horizontal, de dois pontos, ou seja, o somatório do comprimento das projeções destes dois pontos até à sua interseção, calculado da seguinte forma:

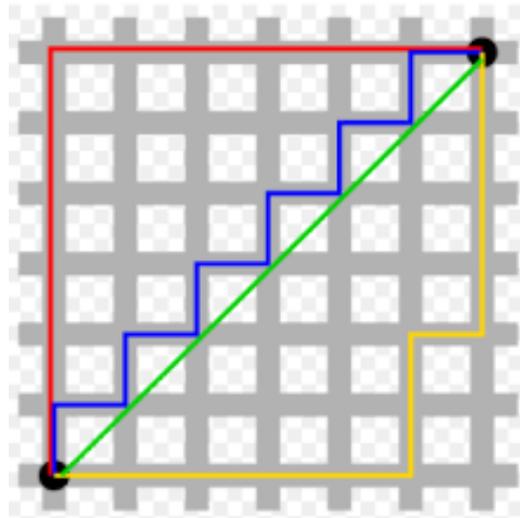
$$d(p, q) = \sum_{i=1}^n |p_i - q_i| \quad (4.10)$$

#### *Distância Minkowski*

Esta também é dimensionada por Hermann Minkowski, podendo-se considerar como uma generalização das duas anteriores, sendo uma métrica num espaço vetorial normalizado, onde o cálculo é realizado através da seguinte expressão:

$$d(p, q) = \left[ \sum_{i=1}^n |p_i - q_i|^n \right]^{\frac{1}{n}} \quad (4.11)$$

A classificação é o rótulo final que será obtido para o conjunto de valores testado, no caso do KNN vai corresponder ao valor maioritário dos K valores que tenham a menor distância, permitindo excluir um vasto valor de possibilidades, concentrando o respetivo interesse nos K valores pedidos.



**Figura 4.2:** Exemplo da representação das distâncias entre pontos.

Na Figura 4.2 <sup>2</sup>, encontra-se representado com a cor verde a distância Euclidiana e com cores diferentes possíveis distâncias de Manhattan.

#### 4.4 REDE NEURAL PERCEPTRON

Este tipo de ANN é um modelo que permite entender com facilidade o funcionamento de uma rede neural, esta foi dimensionada em 1958, por Frank Rosenblatt no Cornell Aeronautical Laboratory [37], inventada para ser uma máquina que efetuava o reconhecimento de imagens [20]. Neste trabalho foi elaborada uma execução da mesma com apenas um neurônio, com a forma que está representado na Figura 2.10, com apenas duas entradas, usando a função de somadora da equação 4.12 e a função de ativação da equação 4.13. Esta rede vai essencialmente focar-se no ajustamento de dois pesos,  $w_1$  e  $w_2$ , sendo estes os pesos associados a cada entrada. A Perceptron requer que os dados sejam o mais lineares possíveis, dado que a rede, teoricamente, faz uma separação entre os dois tipos de dados consoante as características fornecidas. Este método vai fazendo iterações até que os pesos consoante as entradas fornecidas calculem o resultado da classificação e este seja o mais correto possível. Geralmente os pesos têm valores no intervalo e 0 e 1 e o seu valor inicial é totalmente aleatório.

<sup>2</sup>[https://pt.wikipedia.org/wiki/Geometria\\_dota\\_xi](https://pt.wikipedia.org/wiki/Geometria_dota_xi)

A expressão da correção do valor dos pesos é a seguinte:

$$w^{novo} = w^{anterior} + \eta(d^{(k)} - y)x^{(k)} \quad (4.12)$$

Onde:

- constante da taxa de aprendizagem;
- $y$  valor produzido pela saída da rede;
- $d^{(k)}$  valor desejado;
- $x^{(k)}$  amostra de treino
- $w$  pesos da última iteração

Com esta rede obtém-se dois tipos de saída, 0 e 1, as quais são associadas inicialmente a um grupo de dados diferente, de modo a fazer a classificação de dois tipos de recursos distintos.

#### 4.5 CONCLUSÕES

Após a leitura e percepção dos métodos expostos, é necessário ter noção das limitações para que a utilização dos mesmos seja efetuada de forma correta, ou seja, os parâmetros controláveis dos diferentes métodos têm as suas deficiências, uma parametrização errada leva a que os resultados não sejam eficientes.

##### *Kmeans*

Este algoritmo tem como necessidade fundamental que o número de grupos seja específico, adaptando-se bem a um grande número de amostras, o que possibilita o seu uso numa grande variedade de aplicações nas mais diversas áreas da ciência.

Com a finalidade fulcral de minimizar a inércia ou o critério da soma dos quadrados no grupo, o K-menas escolhe de forma coerente os centróides.

Desvantagens:

- Pressupõe que os conjuntos têm sempre características convexas e isotópicas, o que não se verifica;
- Resposta fraca quando o conjunto é alongado, isto é, muito disperso no espaço de características;
- Resposta ineficiente quando existe muita variedade de irregularidades nas características.

Vantagens:

- Facilidade na implementação;
- Uma boa velocidade de execução, demorando pouco tempo;

- Não necessita de treino com supervisão;
- Permite a classificação para um elevado número de grupos.

### *GMM*

Este algoritmo é de uma complexidade superior em relação aos restantes, sendo que utiliza metodologias matemáticas mais avançadas. É importante que o número de grupos seja específico, para que os resultados sejam os pretendidos. Este é muito semelhante ao K-means, embora apresente melhor performance no que diz respeito à classificação.

Vantagens:

- Boa classificação dos dados;
- Boa velocidade de execução;
- Não necessita de treino com supervisão;
- Permite a classificação para um elevado número de grupos.
- Fronteiras entre grupos são mais subjetivas, o mesmo dado pode pertencer a vários grupos consoante a probabilidade de fazer parte do mesmo.

Desvantagens:

- Complexidade na implementação;
- A classificação é bastante afetada quando a qualidade dos dados não é boa;
- A classificação é bastante afetada quando existem características irrelevantes.

### *KNN*

O mais importante a ter em consideração quando usado o KNN é que consoante o número de classes presentes no modelo, o K escolhido deve ser sempre um valor com paridade oposta, para que não exista empate de classificação, bem como também deve ter-se em conta o número de amostras com que o sistema foi treinado, visto que, ao utilizar um K grande num número reduzido de amostras, não irão ser obtidas boas conclusões. Tendo em conta estas dependências, o valor de K não deve ser muito pequeno, pois fica sujeito a pontos de ruído, caso contrário se for muito elevado a vizinhança pode conter elementos de várias classes. A classificação deste método depende fortemente do modelo dos dados, por isso a normalização dos mesmos é algo a ter em consideração quando utilizado o KNN.

Numa visão geral, o que é conhecido à partida é que este método tem as seguintes vantagens:

- Simplicidade na implementação;
- Facilmente moldável, bastante flexibilidade;
- Para sistemas com poucos parâmetros apresenta bons resultados.

As desvantagens de igual relevância são:

- Necessita supervisão;
- O tempo de execução aumenta com a complexidade;
- A precisão degrada com a existência de ruído;
- A classificação é bastante afetada quando existem características irrelevantes.

### *ANN Perceptron*

Este tipo de rede necessita de treino, no sentido de que quanto maior o número de amostras melhor será a qualidade dos pesos obtidos. Sendo a linearidade dos dados o fator mais importante a ter em conta no dimensionamento deste tipo de rede simples.

As vantagens desta rede são:

- Simples de implementar;
- Uma boa velocidade de classificação;
- Quanto maior o número de amostras melhor qualidade tem a rede;
- Simples de aumentar a complexidade, utilizando mais neurões com as mesmas funcionalidades.

Apresenta as desvantagens:

- Necessita supervisão;
- A precisão é fraca em dados não lineares;
- A classificação é bastante afetada quando existem características irrelevantes.

# Implementação

O trabalho desenvolvido no âmbito desta dissertação passou por várias fases de execução, nomeadamente pesquisa, testes e validações, onde foram abordadas várias opções para alcançar, de modo mais eficiente, o melhor resultado. Na fase de pesquisa tentou-se explorar os métodos existentes, ou seja, procurar soluções eficientes, analisar estudos de colegas da área, visualizar os resultados de algumas aplicações distintas. O principal foco da pesquisa foi todo ele na área da visão por computador, mais concretamente em metodologias de inteligência artificial, também procurando, ao mesmo tempo, adquirir conhecimentos através de percepção de métodos simples de ML e ANN.

Em seguida passou-se à fase de implementação e teste onde através dos conhecimentos adquiridos anteriormente, se pôs em prática o implementado, criando os algoritmos para resolução do problema, aqueles que se equacionou como possíveis metodologias, que pudessem apresentar bons resultados. Nesta etapa efetuou-se a implementação do pré-processamento de imagens, onde se analisou o espaço de cor a trabalhar, qual o gradiente que apresenta melhor qualidade, melhor velocidade de execução, tudo isto para otimizar a extração das características e os recursos das imagens em análise. Por sua vez, também se elaborou a implementação da classificação, sendo esta parte mais extensa, devido à implementação de vários algoritmos de classificação e teste dos mesmos, de modo a poder retirar as devidas conclusões sobre cada um, face aquilo que era conhecido anteriormente.

## 5.1 ALGORITMOS IMPLEMENTADOS

O algoritmo final deste trabalho, como tem sido refletido ao longo deste documento, é composto por duas partes muito relevantes da sua estrutura, que se dividem em pequenas partes, estas são o pré-processamento e a classificação das características da

imagem digital. Ambos os processos foram sujeitos ao estudo, implementação e teste. Nesta secção, esta presente a forma como foram implementados e testados estes pontos, até chegar ao resultado.

### 5.1.1 Pré-processamentos da imagem digital

Tendo em conta o capítulo 3, esta subsecção tem com objetivo explicar como foram utilizadas as ferramentas descritas nesse capítulo recorrendo ao uso de pseudocódigo.

#### **Pseudocódigo:**

Ler imagem BGR

Converter imagem BGR para Lab

Retirar da imagem o valor de linhas, colunas e dimensões

Dividir a imagem 3D em 3 canais L, a e b

Ciclo for das 3 dimensões:

    Inicializar class Gradiente fornecendo um canal como entrada

    Calcular Gradiente (Algoritmo 1)

    Calcular Magnitude (Algoritmo 2)

    Retirar características da imagem (Algoritmo 3)

    Guardar características num vetor tridimensional

Fim do Ciclo

Nota: Pseudocódigo dos algoritmos em apêndice.

Existem alguns tipos de formas de calcular o gradiente para o pretendido. Para melhor perceção deste método foi implementado um gradiente denominado normal, bem como também foram analisados outros tipos disponíveis na biblioteca do OpenCV, como se pode verificar no Algoritmo 1.

De forma a compreender aquele que seria o que apresenta melhor eficiência, ou seja, quando efectuado o cálculo da magnitude, qual seria o mais rápido quando comparado com a qualidade do resultado e se as transições de tonalidade ficam bem descritas, uma vez que, através desta análise é possível ter a perceção de qual gradiente que se pode assimilar de forma adequada ao pretendido.

No caso das características das imagens, foram retiradas as médias e os desvios padrões de pequenos blocos da imagem, sendo a dimensões do bloco um parâmetro bastante relevante, pois é crucial na eficiência do algoritmo. O tamanho do bloco e do passo foi implementado para ter resto da divisão por 16, igual a 0 e o passo entre blocos para ser metade do valor do bloco. Estes valores variam muito dependendo do

tamanho da imagem, por isso, quando é testado o sistema é preciso ter em atenção a imagem dada como entrada, ou seja, estabelecer uma relação entre o tamanho da imagem e o bloco, pois blocos muito pequenos em imagens com muitos píxeis e blocos muito grandes em imagens pequenas fazem com que se obtenha péssimos resultados.

### 5.1.2 Classificações da imagem digital

Através do conhecimento descrito no capítulo 4, efetuou-se a implementação dos respetivos algoritmos descritos no mesmo, recorrendo ao uso de pseudocódigo para expor de forma simplificada como é a estrutura dos diferentes métodos.

#### **Pseudocódigo Kmeans:**

Ciclo for para as 3 dimensões:

    Usar o método Kmeans do OpenCv para obter o vetor de classificação

    Dividir o vetor em dois vetores distintos:

        Classe 0 para os que tem label 0

        Classe 1 para os que tem label 1

    Normalizar valores

    Assumir que a classe que tem maior número de valores é chão:

    Se classe 0 > classe 1 então classe 0 é sujidade, guardar o centro do cluster

    Se não então classe 1 é sujidade, guardar o centro do cluster

    Ciclo for para o comprimento do vetor:

        Se Distância Euclidiana entre centro e ponto for > limite então é sujidade

        Guardar o valor a verdadeiro num vetor 3D no canal respetivo

        Se não é chão

        Guardar o valor a falso num vetor 3D no canal respetivo

    Fim do ciclo

Fim de ciclo

Sujidade = (canal 0 and canal 1) or (canal 0 and canal 2) or (canal 1 and canal 2)

Ciclo for para correr o booleano "Sujidade"

    Se Sujidade igual a verdadeiro:

        Colocar na máscara o respetivo bloco como sujidade

Fim de ciclo

Aplicar a máscara à imagem

Nota: Limite estabelecido 0.7, com valores normalizados de 0 a 1 e Distância Euclidiana em apêndice, algoritmo 4.

Com o uso do Kmeans pode-se agrupar em diferentes grupos informação com características diferentes, neste caso foi parametrizado para reconhecer dois tipos diferentes de "clusters", os critérios usados foram "(cv.TERM\_CRITERIA\_EPS + cv.TERM\_CRITERIA\_MAX\_ITER, 10, 1.0) e cv.KMEANS\_RANDOM\_CENTERS"[38], após esta identificação é feito o reconhecimento de qual é o tipo de grupo correspondente a cada label, sujidade ou chão, sendo estes classificados com um label 0 ou 1. De seguida, para refinar um pouco mais é estabelecido um limite, sendo que, o grupo do chão tem por norma um elevado número de pontos, estabelecendo-se um limite cujo um ponto com uma distância euclidiana normalizada superior a 0.7, da relação entre o centro do conjunto com o ponto mais longínquo, tem probabilidade de ser sujidade. Como tal, foram identificados esses pontos para os diferentes canais, no final para aprimorar o método, faz-se uma interseção entre os vários canais, cujo objetivo passa por analisar quais foram as identificações semelhantes nos mesmos e obter uma classificação mais otimizada. Por fim, a máscara é aplicada à imagem de modo identificando as zonas de sujidade.

### **Pseudocódigo GMM:**

Ciclo for para as 3 dimensões:

Usar o método GMM.fit do scikit-learn para o treino

Probabilidades = Obter probabilidades (GMM.predict\_proba)

Predições = Obter predições (GMM.predict)

Verificar qual cluster é chão e sujidade:

Se Contagem de Predições igual 1 > Contagem de Predições igual 0:

Para garantir que o label 0 é sempre chão:

Predição not Predição

Cluster igual 0

Fim do ciclo

Produtório Probabilidades = canal 1 and canal 2 and canal 3

Produtório Predições = canal 1 and canal 2 and canal 3

Ciclo for para as N

Se Produtório Probabilidade > critério:

O bloco tem é de sujidade

Se Produtório Predição igual 1:

O bloco tem é de sujidade

Fim de ciclo

Aplicar máscara à imagem

Nota:  $N$  são o número de características retiradas à imagem

Através do GMM, tal como no Kmeans, efetua um estudo sem supervisão onde se obtém o agrupamento em conjuntos diferentes da informação. O método proposto foi implementado de maneira muito semelhante como o descrito em "The power of gmms"[17] e com o uso da biblioteca do "scikit-learn"[39]. Este foi parametrizado para fazer identificação de dois conjuntos de valores distintos, para ter uma variância do tipo esférica, sendo que, o pretendido é detetar chão e sujidade, duas distribuições de probabilidades diferentes e do tipo esférico, porque os conjuntos, como se pode verificar com o Kmeans, agrupam-se com uma forma bastante circular nas três dimensões.

Os resultados fornecidos são de dois tipos: um é a probabilidades do dado pertencer aquele conjunto, e o outro é a previsão do conjunto a que pertence, no fundo foi estabelecido um critério em que valores com probabilidade superior a um determinado valor pertencem a um determinado conjunto e as restantes não, após efetuar este processo para os vários canais fez-se a interseção entre eles e construiu-se a máscara consoantes os valores obtidos por esta interseção, de modo que quando aplicada a imagem apenas fique visível a sujidade presente na mesma.

### **Pseudocódigo KNN:**

Vetor saídas = Cálculo das Saídas para treino (Algoritmo 4)

Ciclo for para as 3 dimensões:

    Ciclo for para  $N$ :

        Vetor resultado para cada canal = classificação do KNN (Algoritmo 8)

    Fim do Ciclo

Fim do ciclo

Ciclo for para  $N$ :

    Classificação = vetor resultado (canal 1 + canal 2 + canal 3)/3

    Se Classificação  $> 1.5$  então bloco tem sujidade

Fim do ciclo

Aplicar máscara à imagem

Este algoritmo proposto é lhe executada uma parametrização simples, implementado apenas para identificar 2 tipos de classes diferentes, como os restantes. Consoante as limitações, o algoritmo devolve o número dos  $K$  vizinhos mais próximos não tendo um valor limite a dar como retorno cabe ao utilizador efetuar o estudo de quantos vizinhos quer que lhe sejam devolvidos. Os recursos que o algoritmo usa são características da imagem obtida em bloco e, como os restantes métodos, é um método de supervisão pois

as saídas são obtidas também dos blocos a classificar, ou seja, um bloco com mais de metade dos valores dos pixels a um, na imagem binária, é considerado um bloco com sujidade sendo estes dados fornecidos ao algoritmo como base para efetuar a classificação. Aqueles que apresentam distâncias menores ao dado avaliado, são considerados vizinhos próximos que serão usados para classificação.

O processo é sempre efetuado para cada canal da imagem de forma independente, porque as imagens da forma em que foram repartidas as suas características, com forme o espaço de cor escolhido têm valências diferentes, apresentando resultados de classificação diferente, por isso efetua-se a intercessão dos vários canais para refinar o resultado. Neste algoritmo foi equacionado que caso dois canais da imagem tenham a mesma classificação, esse é o rótulo atribuído ao bloco em questão.

### **Pseudocódigo ANN Perceptron:**

Ciclo for para as 3 dimensões:

    Inicializar Perceptron class

    Treinar a rede

    Ciclo for para N:

        Testar a rede guardando os valores num vetor

    Fim de ciclo

Fim de ciclo

Guardar Pesos

Ciclo for para N:

    Resultado = Treino(Canal 1 + canal 2 + canal 3)

    Se Resultado > 1.5:

        Este bloco é sujidade

    Se não

        Este bloco é chão

Fim de ciclo

Aplicar máscara à imagem

Esta rede neural segue o mesmo perfil dos restantes métodos, que se pode identificar como o padrão neste estudo, uma vez que, faz uma classificação para cada canal de forma independente, treinando a rede para cada um deles. Neste sentido, a Perceptron implementada tem apenas um neurão como está representado na figura 2.10 e o número de pesos é igual ao número de características extraídas ao bloco, ou seja, duas características. O treino consiste em adaptar os valores dos pesos tendo em conta as

entradas e as saídas, de modo a convergir para um valor, que seja o mais adequado possível para efetuar a classificação no momento do teste.

A operação de produzir a testagem é efetuada para cada característica de forma individual, através da função de teste da Perceptron implementada, este método da classe com o auxílio dos pesos, executa a função somadora e passa o potencial de ativação a função de ativação, reproduzir a saída consoante o modo como foi implementada, neste caso a função de ativação usada é o step, pois como só existem apenas duas classes, identifica-se a classe zero como sendo chão e a classe 1 como sendo sujidade.

Por fim, tal como efetuado no KNN é feita a interseção dos resultados obtidos em cada canal de forma a encontrar congruência, ou seja, se em dois canais distintos obtiverem a mesma classificação, admite-se que esse é o valor correspondente aquele bloco em questão.

## 5.2 CRITÉRIOS DE CLASSIFICAÇÃO

Após a separação dos vários canais que formam a imagem e análise do seu conteúdo, pode-se ter a perceção, como esperado, que os canais dadores de mais informação relevante são o **a** e **b**, enquanto existe abstração da luz e só se identificam por norma dois padrões distintos, sendo estes o chão e possível sujidade. Por conseguinte, programou-se dois tipos de critérios diferentes, ambos de interseção, mas com cálculos diferentes.

Devido a estes fatores, quando se obtém resultados de tipos diferentes, sendo estes resultados por valores de probabilidade, previsão ou rótulo, é necessário utilizar diferentes tipos de critérios e métodos para executar a interseção dos mesmos, de modo a obter resultados minimamente coerentes, em que o seu significado tenha valor relevante para deteção das diferentes classes na imagem e para construção da máscara a aplicar à imagem.

### 5.2.1 Interseção pelo Produto

A abordagem por produto, é boa quando se trata de efetuar o produto de probabilidades, sendo que não elimina possibilidades apenas as refina. A expressão que representa como foi implementado é a seguinte:

$$p(X|\sigma) = \prod_{c \in (L,a,b)} p(X^c|\sigma^c) \quad (5.1)$$

Através da equação do ponto a cima, é possível efetuar a interseção das probabilidades respetivas aos diferentes canais, para os dois tipos de conjuntos de dados diferentes consoante os dois tipos diferentes de características fornecidas.

Os valores que podem ser obtidos através do cálculo acima representado, serão sempre entre 0 e 1. Neste sentido, o critério estabelecido para diferenciar o que é pretendido, daquilo que é irrelevante, é efetuar filtragem por valores do seguinte intervalo, 0.6 a 0.9, valores abaixo serão sempre identificados como chão.

Efetuar este tipo de interseção em classificação por rótulo, faz com que os resultados obtidos não sejam tão bons.

### 5.2.2 Interseção Maioritária

Para efetuar interseção de classificação por rótulo de forma eficiente, implementou-se a seguinte equação, que representa a regra do valor maioritário, ou seja:

$$F(X, \sigma) = \sum_{c \in (L, a, b)} F(X^c | \sigma^c) \quad (5.2)$$

Como se pode verificar no ponto acima, é feito um somatório dos valores da classificação, os valores podem ser 0 ou 1, consoante os valores obtidos, os valores que se podem obter deste cálculo serão sempre entre 0 e 3. Assim, para fazer uma identificação maioritária através dos valores presentes em F, um valor que se encontre acima de 1.5 será sempre admitido como um bloco de sujidade e um valor abaixo será sempre admitido como um bloco de solo. Por sua vez, também é possível classificar como valores apenas com resultado 3 sujidade, mas os resultados obtidos tem falha de classificação em quantidade, mas são mais fiáveis com este tipo de notação.

### 5.3 CONCLUSÃO

Estas implementações visam complementar-se e ao mesmo tempo serem testadas de forma paralela, de modo a poder obter resultados diferenciados e entender se os resultados obtidos são equivalentes ao esperado. Cada algoritmo tem um método de classificação distinto, por isso apresentam diferente ponto a favor e contra.

## Resultados Experimentais

Este capítulo apresenta a validação de resultados, onde através de um determinado critério, diferente para cada método avaliados os resultados obtidos, conforme o que era esperado, se estariam corretos e retirando as devidas conclusões de cada classificação.

Utilizando um método de avaliação de métrica, ou seja, um método que permite qualificar os algoritmos testados quanto a sua qualidade, comparando com os resultados corretos os resultados obtidos.

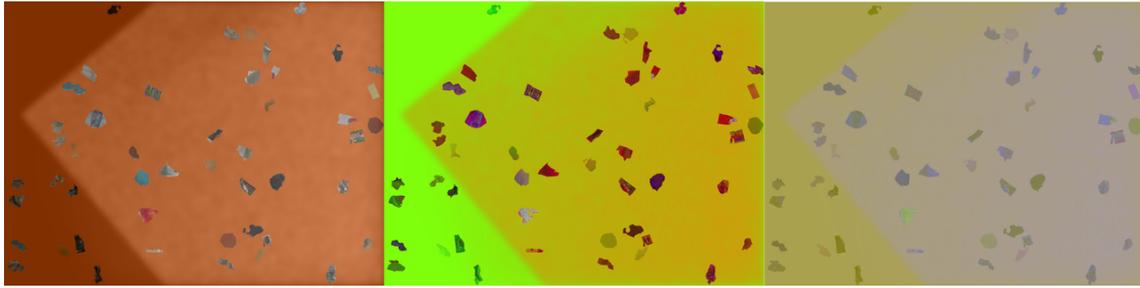
### 6.1 ANÁLISE

Esta secção apresenta a análise das várias partes dos algoritmos testados. Tem como objetivos esclarecer as várias valências e défices de cada parte.

#### 6.1.1 Análise do Espaço de cor

Os dois espaços de cor analisados foram o HSV e Lab, devido ao facto das suas características de conversão são aquelas que mais se podem adequar ao contexto pretendido, pois dividem a imagem em três canais de diferentes características. Exemplos de resultados dos dois tipos de espaço de cor que foram analisados, encontram-se nas Figuras 6.1, 6.2 e 6.3.

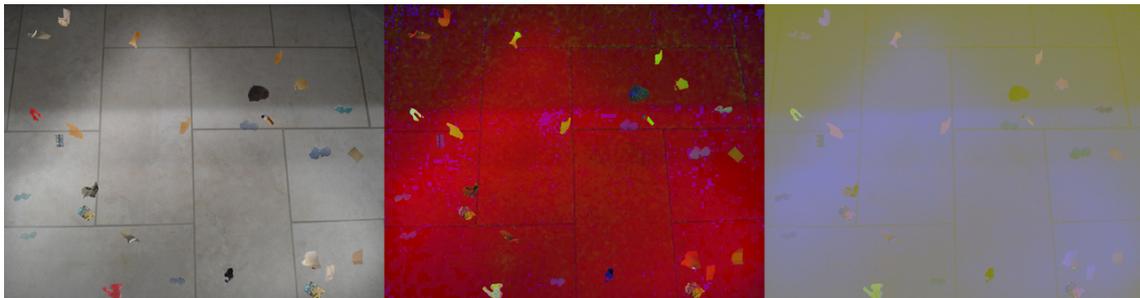
Foram escolhidos estes 3 exemplos, porque neles temos presentes 3 contrastes motivadores de analisar e não no que diz respeito a cor, mas sim à luminosidade, ao ângulo da imagem e às suas diferenças nas várias irregularidades do chão.



**Figura 6.1:** Exemplo artificial de um chão laranja com existência de sombra, convertido em HSV e Lab, segunda e terceira imagem respetivamente.



**Figura 6.2:** Exemplo artificial de um chão madeira cinza, convertido em HSV e Lab, segunda e terceira imagem respetivamente.



**Figura 6.3:** Exemplo artificial de um chão azulejo cinza com existência de sombra, convertido em HSV e Lab, segunda e terceira imagem respetivamente.

### 6.1.2 Análise dos gradientes

O principal diagnóstico feito aos gradientes, fundou-se essencialmente em dois pontos de igual importância, a qualidade do resultado e o tempo de processamento.

Sendo o pré-processamento a fase principal do algoritmo, as preocupações do tempo de execução foram consideravelmente ponderadas, tendo em conta que tempo de execução é importante no que diz respeito ação do sistema, foi tido o cuidado de não começar a criar um algoritmo lento logo no primeiro passo. Para tal, testou-se os gradientes presentes na Secção 3.3.1, tendo em atenção os dois aspetos referidos.

### *Tempo de processamento do gradiente*

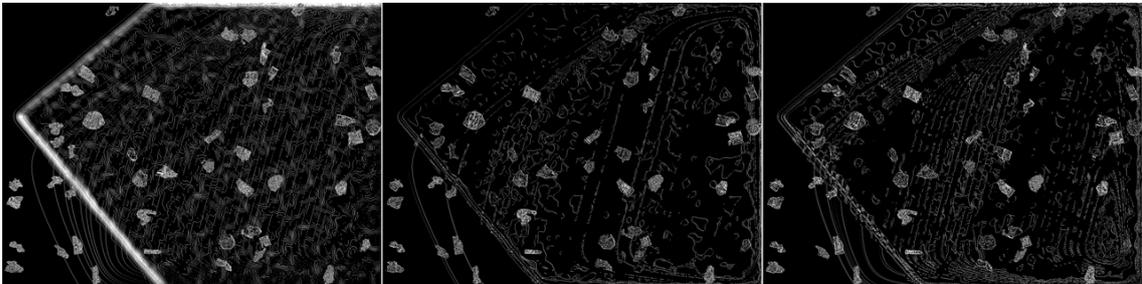
A Tabela 6.1 apresentada, constam os valores do tempo em segundos, tempo este que demora a executar cada tipo de gradiente, sendo de salientar o facto de que, os métodos Sobel e Laplaciano foram usados *kernels* iguais a três como indicado na Secção 3.3.1.

**Tabela 6.1:** Tempo de execução de cada tipo de gradiente consoante as diferentes imagens de exemplo.

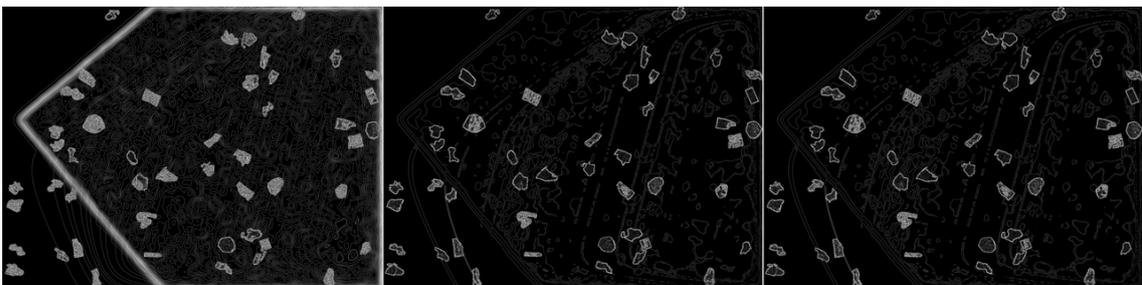
Imagem		Laranja	Madeira cinza	Azulejo cinza
Gradientes Tempo (s)	Normal	12.46	4.99	3.72
	Sobel	4.90	1.67	1.33
	Laplacian	14.43	5.24	4.09

### *Qualidade do gradiente*

As imagens seguintes representam o resultado do cálculo dos diferentes tipos de gradientes apresentados na Secção 3.3.1, a magnitude dos canais L, a e b. Podendo-se também verificar quais as diferenças apresentadas por cada canal ao nível da intensidade dos pixels e também obter uma percepção mais fácil da qualidade dos mesmos, fazendo a comparação visual entre eles.



**Figura 6.4:** Aplicação de um gradiente do tipo normal à Figura 6.1.



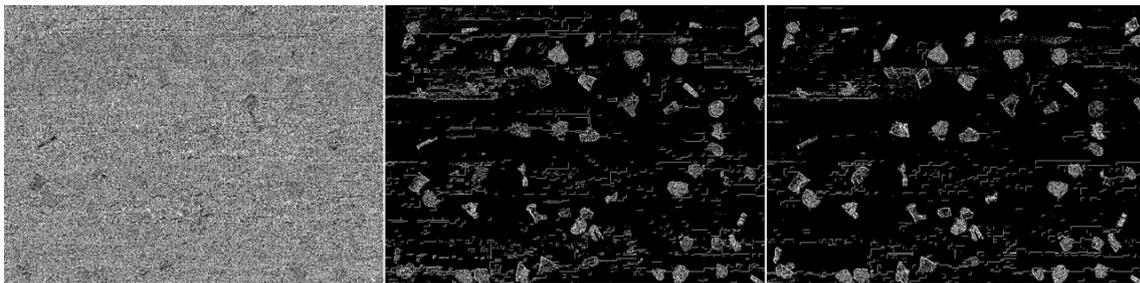
**Figura 6.5:** Aplicação de um gradiente do tipo sobel à Figura 6.1.



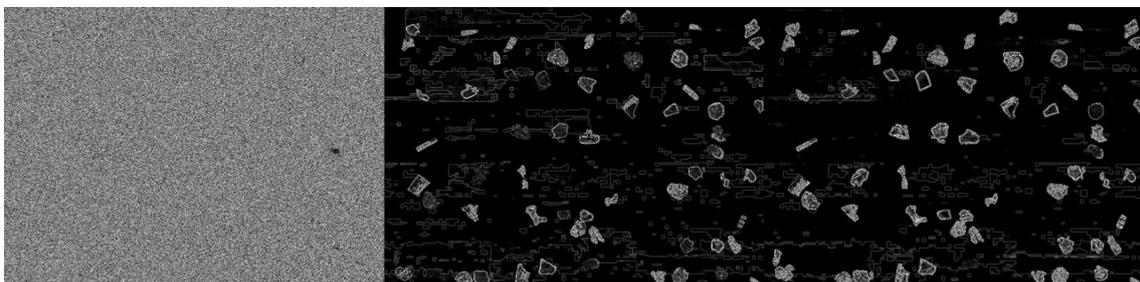
**Figura 6.6:** Aplicação de um gradiente do tipo laplacian à Figura 6.1.

Numa análise inicial através das representações visuais do primeiro exemplo é possível retirar algumas conclusões relevantes. A primeira é que os gradientes normal e sobel apresentam, no que diz respeito à qualidade de resultado, um resultado muito semelhante e a segunda é que os resultados obtidos para o método laplaciano para os canais a e b, não são bons, porque não identificam com a mesma qualidade que os outros dois.

Quando feita uma análise do valor real do pixel verifica-se que a interpretação visual gráfica corresponde com a análise do valor numérico.



**Figura 6.7:** Aplicação de um gradiente do tipo normal à Figura 6.2.



**Figura 6.8:** Aplicação de um gradiente do tipo sobel à Figura 6.2.

Este segundo exemplo, apesar de não ter presente diferentes tipos de sombra, obteve-se um resultado motivador e relevante, para com a sua análise para obter conclusões.

Dessa análise visual pode-se acrescentar, que um chão com texturas ou algumas irregularidades com tons muito próximos, no que diz respeito ao canal L, apresenta



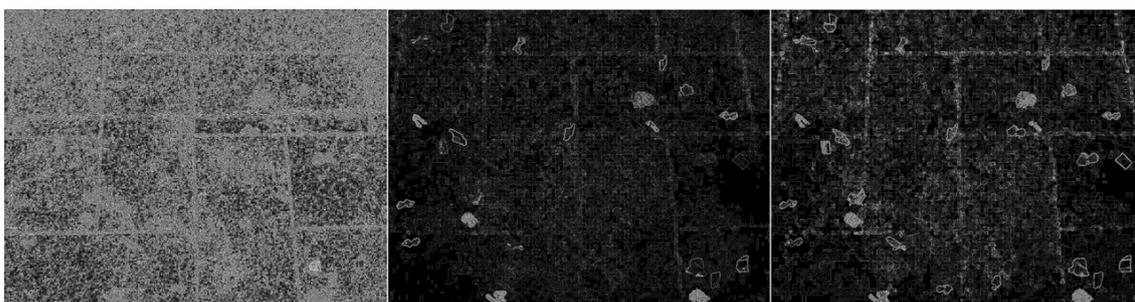
**Figura 6.9:** Aplicação de um gradiente do tipo normal à Figura 6.2.

um cálculo de magnitude de gradiente que não transmite informação relevante para a futura classificação, enquanto o método laplaciano no que diz respeito ao canal L apresenta um resultado com melhor qualidade, mas mesmo assim abaixo do expectável.

As conclusões visíveis no primeiro exemplo continuam a ser válidas e são ainda mais reforçadas, pelo facto de o método laplaciano não ter boa performance nos canais a e b, mas com a mesma qualidade para o canal L que diz respeito aos valores de luminosidade.



**Figura 6.10:** Aplicação de um gradiente do tipo normal à Figura 6.3.



**Figura 6.11:** Aplicação de um gradiente do tipo sobel á Figura 6.3.

Este último exemplo em análise é uma imagem mais próxima do contexto real, é um chão de azulejo cinza com sujidade artificial, onde a imagem foi retirada com um certo ângulo em relação ao chão, não se verificando os normais noventa graus, tendo também presente no chão sombra, texturas e irregularidades. Sendo este do ponto de vista de análise destes métodos bastante ricos em características, pode-se reforçar as validações



**Figura 6.12:** Aplicação de um gradiente do tipo laplacian à Figura 6.3.

feitas até agora, salientando que zonas com mais luminosidade e com mais nitidez na imagem, irão apresentar no cálculo da sua magnitude de gradiente, sempre melhores resultados no que diz respeito às mudanças de tom.

### 6.1.3 Análise da velocidade de extração de caraterísticas

A extração de propriedades ou caraterísticas da imagem, tal como está retratado na Secção 3.4, o principal objetivo de análise deste ponto, vai ao encontro do tempo de execução, mas também a qualidade das caraterísticas, porque blocos demasiado grandes podem ter uma grande percentagem de chão e uma pequena de sujidade e são rotulados como chão erradamente, mas também blocos demasiado pequenos podem levar a que o tempo de processamento seja elevado e ao mesmo tempo que a qualidade dos resultados seja fraca.

Neste ponto o foco foi para a análise do tempo de processamento, sendo feito esse diagnóstico aquando da classificação dos blocos. Foi necessário também ter em atenção que os tamanhos dos blocos tem sempre de ter uma relação de forma linear com a dimensão da imagem, isto é, ser linear consoante o número de pixels que a imagem digital tem presente.

Utilizando os mesmos três exemplos das Figuras 6.1, 6.2 e 6.3, ricos para demonstrações efetuou-se o teste de velocidade de extração como está demonstrado na Tabela 6.2.

**Tabela 6.2:** Tempo em segundos de extração das caraterísticas das diferentes imagens exemplo, consoante a dimensão do bloco e o passo.

Bloco	128 x 128	64 x 64	32 x 32	16 x 16
Passo	64	32	16	8
Laranja 6.1	0.124	0.359	1.00	3.432
Madeira cinza 6.2	0.036	0.091	0.32	1.169
Azulejo cinza 6.3	0.134	0.192	0.966	2.204

#### 6.1.4 Análise do tempo de treino de cada método

O tempo que o método demora a efetuar o treino é algo importante ser analisado, embora que quando testado, no caso dos algoritmos com supervisão este tempo é irrelevante, apenas atuando sobre aquilo ao qual já foram previamente treinados. Contudo, esta análise é feita de modo a obter perceção de como o dimensionamento do treino tem impacto no tempo de processamento e para o caso dos algoritmos sem supervisão para ter noção do impacto deste tempo no total da execução o algoritmo final.

Não foi feita uma análise para blocos de dimensão 128 x 128 com um passo de 64, devido ao facto destes blocos serem demasiado grandes para as imagens em análise.

Os conjuntos de imagem de seguida em análise foram obtidos através da ferramenta de criação de DS com lixo artificial, usando os fundos pavimentos presentes nas Figuras 6.1, 6.2 e 6.3.

##### *Sem Supervisão*

**Tabela 6.3:** Tempo em segundos da execução do treino da GMM, para diferentes conjuntos de imagem, com características e tamanhos diferentes, mas também para vetores de propriedades de diferentes dimensões consoante o tamanho do bloco e passo.

Bloco	64 x 64			32 x 32			16 x 16		
Passo	32			16			8		
Nº Imagens	10	20	30	10	20	30	10	20	30
Laranja	0.311	0.527	0.704	0.294	0.535	0.762	0.935	0.902	1.25
Madeira cinza	0.244	0.321	0.748	0.489	0.955	1.34	1.47	3.14	4.51
Azulejo cinza	0.588	0.983	1.36	1.32	2.557	5.17	4.35	8.32	16.26

É de reparar que o algoritmo das GMM fornece duas saídas diferentes, embora que o treino para ambas é efectuada pelos mesmos valores não existindo assim 2 tipos de treinos distintos para estas duas saídas diferentes.

##### *Com Supervisão*

A análise do tempo de treino é mais relevante quando se trata de algoritmos com supervisão, porque apesar de não estar refletida na ação do algoritmo é motivador efetuar uma análise, para ter perceção de que o tempo de treino é mais elevado quando comparado com algoritmos sem supervisão.

O KNN é um algoritmo que recebe um determinado conjunto de valores e respetivos rótulos, ou seja, tanto entradas como saídas e depois tenta encontrar os K vizinhos mais próximos, por isso o treino prévio deste é a remoção de características e atribuição de rótulo, não efetuando um treino como os outros algoritmos.

**Tabela 6.4:** Tempo em segundos da execução do treino da ANN, para diferentes conjuntos de imagem, com características e tamanhos diferentes, mas também para vetores de propriedades de diferentes dimensões consoante o tamanho do bloco e passo.

Bloco	64 x 64			32 x 32			16 x 16		
Passo	32			16			8		
Nº Imagens	10	20	30	10	20	30	10	20	30
Laranja	52.1	10.2	45.7	55.2	49.9	69.8	71.7	205.9	292.9
Madeira cinza	38.9	49.9	45.2	85.1	61.1	255.4	93.2	733.1	1052.9
Azulejo cinza	49.4	51.9	195.5	50.5	552.4	797.3	1024.1	2214.2	3208.2

Fazendo uma análise da Tabela 6.4, pode-se verificar que quanto maior o número de imagens para treino e quanto menor o bloco escolhido para deteção, o tempo de treino aumenta. No caso do treino quanto mais amostras e mais resolução melhor a qualidade do treino.

### 6.1.5 Análise do tempo de teste de cada método

Este tempo realmente irá ter impacto, dado que está diretamente conectado ao tempo de processo do algoritmo num todo, sendo a análise do mesmo é deveras fundamental independentemente do algoritmo que esteja a ser tratado, com ou sem supervisão.

Os conjuntos de imagem a seguir em análise foram obtidos através da ferramenta de criação de DS com lixo artificial, usando os fundos pavimentos presentes nas Figuras 6.1, 6.2 e 6.3.

#### *Sem Supervisão*

**Tabela 6.5:** Tempo em segundos da execução do teste da GMM com classificação em probabilidade, para diferentes conjuntos de imagem, com características e tamanhos diferentes, mas também para vetores de propriedades de diferentes dimensões consoante o tamanho do bloco e passo.

Bloco	64 x 64			32 x 32			16 x 16		
Passo	32			16			8		
Nº Imagens	10	20	30	10	20	30	10	20	30
Laranja	0.005	0.012	0.018	0.007	0.017	0.025	0.015	0.033	0.049
Madeira cinza	0.007	0.016	0.435	0.013	0.030	0.450	0.043	0.099	0.173
Azulejo cinza	0.012	0.026	0.038	0.034	0.084	0.138	0.119	0.265	0.435

Como se pode verificar nas Tabelas 6.5 e 6.6, consoante o que pretendemos analisar, os tempos, independentemente do número de imagens, é bastante reduzido e pouco significativo quando se faz uma comparação à unidade.

**Tabela 6.6:** Tempo em segundos da execução do teste da GMM com classificação por previsão, para diferentes conjuntos de imagem, com características e tamanhos diferentes, mas também para vetores de propriedades de diferentes dimensões consoante o tamanho do bloco e passo.

Bloco	64 x 64			32 x 32			16 x 16		
Passo	32			16			8		
Nº Imagens	10	20	30	10	20	30	10	20	30
Laranja	0.005	0.007	0.011	0.16	0.009	0.014	0.046	0.016	0.024
Madeira cinza	0.004	0.009	0.013	0.006	0.015	0.022	0.019	0.041	0.075
Azulejo cinza	0.006	0.013	0.020	0.017	0.035	0.061	0.048	0.107	0.180

*Com Supervisão*

**Tabela 6.7:** Tempo em segundos da execução do teste da KNN, para k igual a três, para diferentes conjuntos de imagem, com características e tamanhos diferentes, mas também para vetores de propriedades de diferentes dimensões consoante o tamanho do bloco e passo.

Nº Imagens	10								
Bloco	64 x 64			32 x 32			16 x 16		
Passo	32			16			8		
K	1	3	5	1	3	5	1	3	5
Laranja	0.229	0.228	0.236	5.33	5.48	2.10	103.1	103.1	63.5
Madeira cinza	2.91	3.01	2.09	56.6	55.4	36.5	995.3	974.4	600.2
Azulejo cinza	34.9	35.5	35.5	635.2	614.1	371.1	12176	11834	10237

O KNN apresenta tempos de testagem bastante elevados, verificando o aumento do tempo com o aumento do número de imagens que foram objeto de teste. O estudo deste método para 10 imagens é suficiente para ter noção dos tempos de execução, pois analisando a Tabela 6.7 consegue-se ter perceção que o tempo é demasiado elevado, não sendo necessário um análise mais longa, isto é, com maior número de imagens.

O ANN apresenta o tempo de teste relativamente bom, com uma velocidade de resposta boa consoante a entrada, que vai ao encontro do que é pretendido e o que é esperado deste algoritmo.

**Tabela 6.8:** Tempo em segundos da execução do teste da ANN, para diferentes conjuntos de imagem, com características e tamanhos diferentes, mas também para vetores de propriedades de diferentes dimensões consoante o tamanho do bloco e passo.

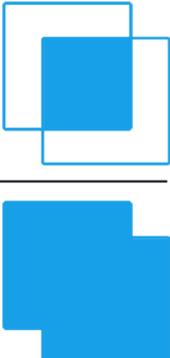
Bloco	64 x 64			32 x 32			16 x 16		
Passo	32			16			8		
Nº Imagens	10	20	30	10	20	30	10	20	30
Laranja	0.001	0.001	0.001	0.003	0.003	0.003	0.014	0.013	0.014
Madeira cinza	0.003	0.003	0.0037	0.011	0.011	0.011	0.046	0.049	0.047
Azulejo cinza	0.008	0.009	0.009	0.036	0.035	0.036	0.154	0.142	0.143

### 6.1.6 Análise da classificação de cada método

Os algoritmos apresentados no Capítulo 4 usam métodos de classificação e também critérios de classificação. Deste modo esta análise foi feita tanto de forma a comparar métodos que usem critério e classificação similar, mas também para comparar resultados entre os métodos diferentes, de modo a poder tirar conclusões da sua eficácia no contexto global.

Ao fazer uma análise qualitativa, foi utilizado com recurso de classificação de qualidade o IoU [40], cujo propósito de estabelecer uma relação ou rácio entre a deteção feita e a esperada. O cálculo é feito como mostra a Figura 6.13, ou seja, a área dos blocos detetados em relação a área dos blocos esperados.

As imagens utilizadas para os seguintes testes foram criadas aleatoriamente em grupos de 10 e 20 imagens. Nesta criação aleatória, existem apenas três fatores também aleatórios considerados, quantidade de sujidade, tipo de sujidade e a presenta de luminosidade em cada imagem, criando assim um conjunto de imagens com características totalmente aleatória e todas elas de maneira distinta, de modo a por em prova os algoritmos em contextos semelhantes, mas com características diferentes.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


**Figura 6.13:** Representação animada do cálculo da IoU.

**Tabela 6.9:** IoU da classificação do GMM em previsão para 10 imagens, consoante o bloco, passo e critério utilizado.

Imagens	10					
	64 x 64		32 x 32		16 x 16	
Passo	32		16		8	
Critério	Maioritário	Produto	Maioritário	Produto	Maioritário	Produto
Laranja	0.017	0.017	0.167	0.164	0.6034	0.603
Madeira	0.465	0.465	0.442	0.442	0.568	0.567
Cimento	0.193	0.192	0.399	0.399	0.407	0.407

**Tabela 6.10:** IoU da classificação do GMM em previsão para 20 imagens, consoante o bloco, passo e critério utilizado.

Imagens	20					
	64 x 64		32 x 32		16 x 16	
Passo	32		16		8	
Critério	Maioritário	Produto	Maioritário	Produto	Maioritário	Produto
Laranja	0.009	0.008	0.084	0.0837	0.411	0.412
Madeira	0.476	0.4762	0.454	0.454	0.508	0.577
Cimento	0.204	0.204	0.335	0.335	0.422	0.422

Através da observação das Tabelas 6.9 e 6.10, pode-se visualizar que dependendo das características e dos tamanhos dos blocos dos diferentes conjuntos de imagens, a classificação tem um resultado no que diz respeito à qualidade de deteção diferente. Neste caso os resultados da GMM previsão transmitem que o algoritmo melhora consoante a diminuição do tamanho do bloco, tendo um resultado do cálculo do IoU bastante semelhante para ambos os critérios de refinação dos resultados utilizados. Verifica-se também que para imagens com menos resolução como o caso do conjunto do chão laranja blocos menores tem melhor performance, enquanto para imagens com uma resoluções maiores as dimensões do bloco não melhoram significativamente a deteção, como no caso do chão de madeira.

No caso da GMM probabilidade, onde os resultados do cálculo do IoU estão presentes nas Tabelas 6.11 e 6.12, a análise é muito como a GMM previsão, pois se verifica o que foi analisado acima, embora que no que concerne ao critério o resultado é bastante distinto, pois os resultados do critério de interceção via produto, explicado na Secção 5.2.1, obtém resultados significativamente melhores no que diz respeito à média de resultados do cálculo do IoU.

Os exemplos de imagens 6.14, 6.15 e 6.16, são casos de deteções bem sucedidas para os 3 tipos de chão exemplo. Pode-se visualizar que com blocos de dimensão 32 e 16

**Tabela 6.11:** IoU da classificação do GMM em probabilidade para 10 imagens, consoante o bloco, passo e critério utilizado.

Imagens	10					
Bloco	64 x 64		32 x 32		16 x 16	
Passo	32		16		8	
Critério	Maioritário	Produto	Maioritário	Produto	Maioritário	Produto
Laranja	0.044	0.119	0.182	0.227	0.505	0.737
Madeira	0.438	0.758	0.499	0.729	0.609	0.800
Cimento	0.209	0.361	0.471	0.727	0.538	0.749

**Tabela 6.12:** IoU da classificação do GMM em probabilidade para 20 imagens, consoante o bloco, passo e critério utilizado.

Imagens	20					
Bloco	64 x 64		32 x 32		16 x 16	
Passo	32		16		8	
Critério	Maioritário	Produto	Maioritário	Produto	Maioritário	Produto
Laranja	0.022	0.055	0.091	0.113	0.361	0.506
Madeira	0.435	0.779	0.487	0.759	0.607	0.829
Cimento	0.239	0.417	0.408	0.642	0.550	0.797

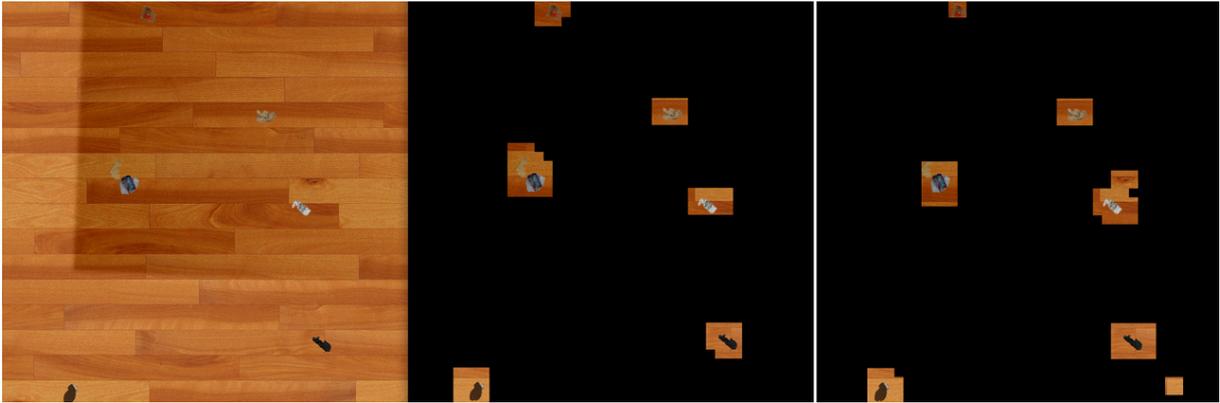


**Figura 6.14:** Exemplo de uma boa deteção com GMM, usando um bloco de 16 x 16 com critério maioritário, num chão do cimento. Imagem original, classificação usando GMM probabilidade e GMM previsão, respetivamente.

para este tipo de imagens, com luz artificial, as deteções são feitas com uma taxa de sucesso bastante elevada.

Encontra-se em anexo mais alguns exemplos de deteções, com boa e fraca qualidade de classificação.

Com a análise das Tabelas 6.13, 6.14, pode-se verificar que o IoU é bastante fraco, pois a classificação obtida não tem o sucesso esperado independentemente dos critérios



**Figura 6.15:** Exemplo de uma boa detecção com GMM, usando um bloco de 32 x 32, num chão do madeira. Imagem original, classificação usando GMM probabilidade e GMM previsão, respetivamente.



**Figura 6.16:** Exemplo de uma boa detecção com GMM, usando um bloco de 16 x 16 com critério produto, num chão do laranja. Imagem original, classificação usando GMM probabilidade e GMM previsão, respetivamente.

**Tabela 6.13:** IoU da classificação do KNN para 10 imagens e k igual 3, consoante o bloco, passo e critério utilizado.

Imagens	10					
	64 x 64		32 x 32		16 x 16	
Bloco	32		16		8	
Passo	Maioritário	Produto	Maioritário	Produto	Maioritário	Produto
Laranja	0.023	0.017	0.274	0.197	0.110	0.003
Madeira	0.125	0.067	0.341	0.183	0.190	0.081
Cimento	0.014	0.005	0.109	0.092	0.001	0

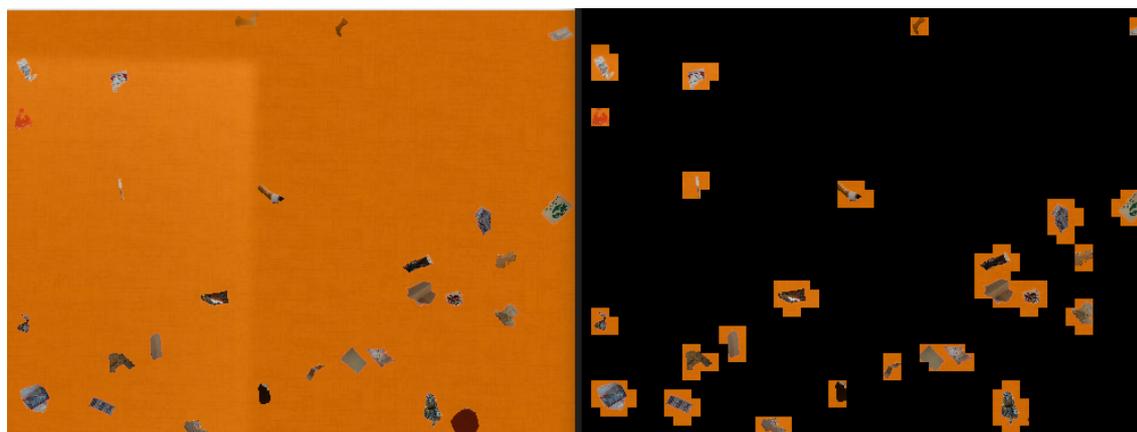
ou parâmetros utilizados.

A Figura 6.17 é um exemplo de uma boa detecção do algoritmo KNN, este tem uma taxa de sucesso muito fraca como se pode ver pelas Tabelas 6.13 e 6.14, em anexo seguem outros exemplos maioritariamente de insucesso.

O algoritmo ANN apresenta um IoU bastante bom, como se pode ver nas Tabelas 6.15 e 6.16, com uma taxa de sucesso mais significativa com o uso do critério maioritário,

**Tabela 6.14:** IoU da classificação do KNN para 10 imagens e k igual a 5, consoante o bloco, passo e critério utilizado.

Imagens	10					
Bloco	64 x 64		32 x 32		16 x 16	
Passo	32		16		8	
Critério	Maioritário	Produto	Maioritário	Produto	Maioritário	Produto
Laranja	0.009	0	0.012	0	0.006	0
Madeira	0.083	0	0.111	0	0.003	0
Cimento	0.001	0	0.013	0.001	0	0



**Figura 6.17:** Exemplo de uma boa deteção com KNN, usando um bloco de 32 x 32 com critério maioritário, num chão do laranja. Imagem original, classificação usando KNN, respetivamente.

**Tabela 6.15:** IoU da classificação do ANN para 10 imagens, consoante o bloco, passo e critério utilizado.

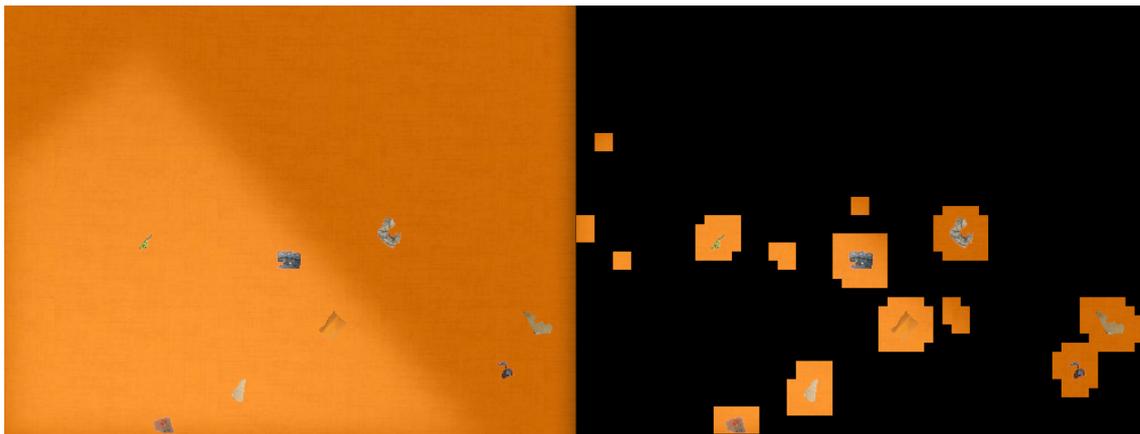
Imagens		10					
Bloco		64 x 64		32 x 32		16 x 16	
Passo		32		16		8	
Critério		M	P	M	P	M	P
Laranja	IoU	0.938	0.893	0.917	0.837	0.888	0.806
	Falhas	3	0	9	3	37	9
Madeira	IoU	0.849	0.526	0.796	0.628	0.694	0.601
	Falhas	4	0	4	0	23	3
Cimento	IoU	0.831	0.462	0.746	0.213	0.703	0
	Falhas	2	0	1	0	11	0

com a diminuição do tamanho do bloco diminui a percentagem de sucesso em maior parte dos exemplos analisados até mesmo em alguns casos não e feita qualquer tipo de deteção. Nestas Tabelas acrescentaram-se as falhas, sendo estes valores representantes da média de quantos blocos a mais foram detetados sem sujidade. O critério de interceção por

**Tabela 6.16:** IoU da classificação do ANN para 20 imagens, consoante o bloco, passo e critério utilizado.

Imagens		20					
Bloco		64 x 64		32 x 32		16 x 16	
Passo		32		16		8	
Critério		M	P	M	P	M	P
Laranja	IoU	0.962	0.861	0.9227	0.812	0.847	0.860
	Falhas	2	4	12	23	18	20
Madeira	IoU	0.823	0.512	0.763	0.625	0.701	0.552
	Falhas	0	0	5	0	31	8
Cimento	IoU	0.832	0.123	0.771	0	0.656	0
	Falhas	3	4	10	0	14	0

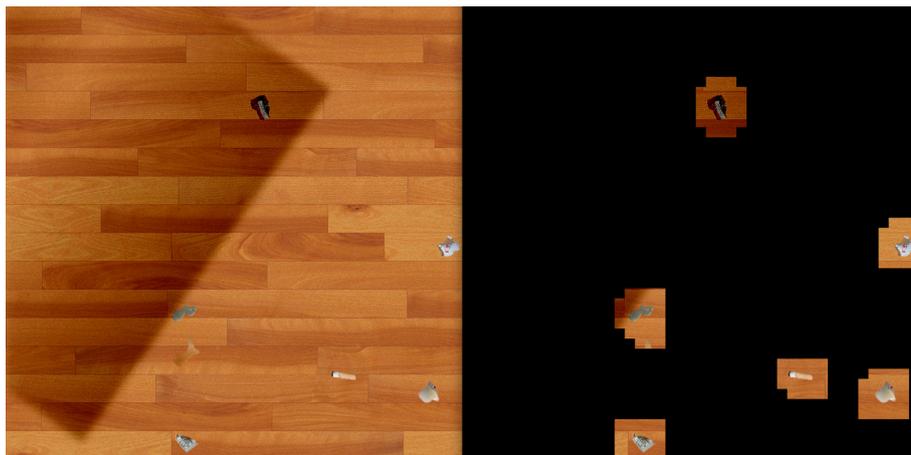
produto, no que diz respeito a falhas, apresenta maior sucesso, não apresentando uma taxa de deteção tão boa, ou seja, apresenta um IoU mais baixo.



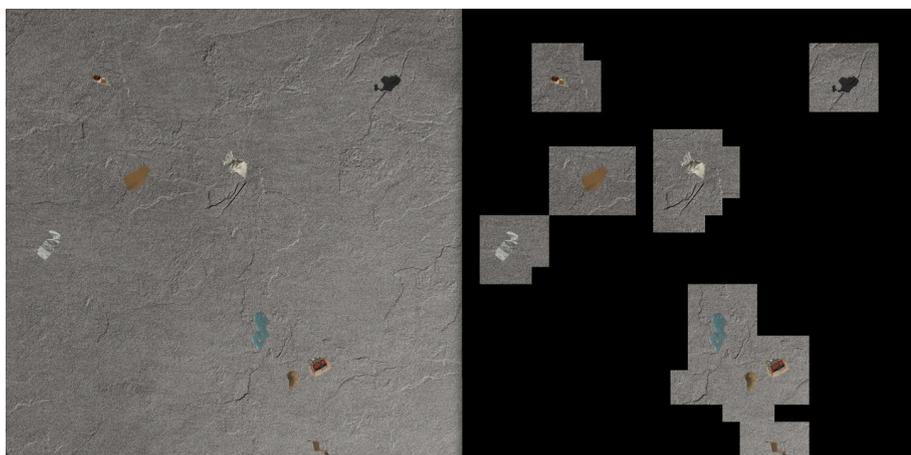
**Figura 6.18:** Exemplo de uma boa deteção com ANN, usando um bloco de 16 x 16 com critério maioritário, num chão do laranja. Imagem original, classificação usando ANN, respetivamente.

As Figuras 6.18, 6.19 e 6.20, são exemplos de deteções com uma taxa de sucesso bastante elevado, encontrando se mais exemplos de deteções do ANN em anexo.

A deteção presente na Figura 6.21, é um tipo de imagem com um chão bastante motivador cheio de caraterísticas e padrões, o que torna a deteção neste tipo e chão algo bastante complexo. O ANN implementado segundo a parametrização descrita e critério utilizado é o único método que consegue fazer deteção neste tipo de chão de todos os que estiveram em estudo.



**Figura 6.19:** Exemplo de uma boa detecção com ANN, usando um bloco de 32 x 32 com critério produto, num chão do madeira. Imagem original, classificação usando ANN, respetivamente.



**Figura 6.20:** Exemplo de uma boa detecção com ANN, usando um bloco de 64 x 64 com critério maioritário, num chão do cimento. Imagem original, classificação usando ANN, respetivamente.



**Figura 6.21:** Exemplo de uma boa detecção com ANN, usando um bloco de 32 x 32 com critério maioritário, num chão do multi-padrão. Imagem original, classificação usando ANN, respetivamente.

## 6.2 COMPARAÇÃO

Esta secção visa fazer uma comparação teórica dos resultados analisados, ou seja, comparar do ponto de vista de performance os vários métodos utilizados.

### 6.2.1 Comparação entre os espaços de cor

Como se pode verificar pelos exemplos em análise na Secção 6.1.1, a transformação Lab é mais suave, sendo que, as características da imagem obtida depois da transformação são muito semelhante, independente da imagem em análise, pois como se pode visualizar no que diz respeito à luminosidade, a zona com menos luminosidade tem um tom mais amarelado e a zona com mais luminosidade um tom mais claro e azulado. A sujidade assume características bastantes semelhantes com tons próximos e não existe qualquer tipo de degradação da imagem ou introdução de qualquer tipo de ruído.

A transformação HSV tem várias qualidades, tais como uma boa separação visual entre sujidade e chão, as características da imagem estão mais nítidas e mais vivas dependendo da sua cor. Noutra perspetiva é necessário realçar o facto da transformação varia um pouco consoante o tipo de chão em análise, o que foi motivo de ponderação pois pretendia-se criar algo que efetuasse uma transformação semelhante para vários tipos de chão.

Na Figura 6.3 a segunda imagem, que diz respeito a transformação HSV, de salientar que a imagem sofreu uma ligeira degradação com a transformação, obtendo-se locais da imagem que retratam um efeito de sombra artificial da sujidade, ou seja, existe uma introdução artificial de ruído o que deteriora os dados das características da imagem.

Contudo, é de realçar que uma avaliação de características no espaço HSV, tem uma melhor utilização quando é relevante efetuar uma análise ou deteção de cor, pois como se pode ver nas Figuras 6.3, 6.2 e 6.1, nas imagens centrais que representam o espaço de cor HSV, é facilmente visível por análise visual a relevância das cores da sujidade em relação ao resto.

### 6.2.2 Comparação entre os gradientes

Começando por efetuar a comparação de um dos aspetos mais importantes o tempo de execução ou de processamento, pode-se verificar, uma vez simplificado e objetivo na Tabela 6.1, que o gradiente que apresenta menor tempo de execução é o Sobel e para o problema apresentado menor é melhor.

No que diz respeito à qualidade do resultado do cálculo da magnitude dos gradientes, como descrito na Secção 6.1.2, os gradientes Sobel e normal, quando comparados verifica-se que são muito semelhantes tanto a nível visual como valor numérico do elemento de imagem, embora seja possível observar maior capacidade de deteção de pormenores no gradiente normal em relação ao Sobel, mas não é de todo significativa.

Por fim, comparando o método laplaciano com os restantes, verifica-se que ao nível de qualidade de resultados fica um pouco a quem do expectável, pois não produz resultados com a mesma qualidade de definição e deteção que os outros dois produzem, exceto em imagens mais complexas com luminosidades mais ponderantes, mas não é de todo significativo.

### **6.2.3 Comparação dos tempos de treino consoante o método**

No caso dos algoritmos sem supervisão onde o efeito do tempo de treino é diretamente relevante no tempo de execução do algoritmo no seu todo, o que se pode verificar efetuando comparações entre os valores da Tabela 6.3 que, para um grande número de imagens ele é bastante reduzido, então para apenas uma unidade é bastante insignificante quando comparado com o tempo de extração presente na Tabela 6.2 ou com o tempo de execução do gradiente presente na Tabela 6.1.

O tempo de treino é bastante maior, nos algoritmos com supervisão, mas em contrapartida este tempo em nada afeta o algoritmo de deteção final no seu todo, pois o treino deve ser feito à parte, guardando no caso do KNN o conjunto de amostras que ajuda a deteção dos vizinhos e no caso do ANN os pesos que fazem a identificação consoante a entrada fornecida.

### **6.2.4 Comparação dos tempos de teste consoante o método**

Quando comparados, os tempos de teste dos diversos algoritmos, é perceptível que os valores do KNN são consideravelmente elevados para apenas 10 imagens, quando comparados com os restantes. Ambos os métodos GMM e ANN apresentam uma velocidade de teste satisfatória, em que os resultados de ambos são notoriamente semelhantes, embora a velocidade de execução do ANN, quando comparados a unidade é melhor. Contudo, o GMM previsão e GMM probabilidade apresentam também uma velocidade de resposta parcialmente equivalente, embora que a velocidade de execução do modo de previsão é mais rápida.

### **6.2.5 Comparação da qualidade das classificações obtidas**

Os valores Tabelados são a média dos resultados obtidos do IoU para os mesmos conjuntos, mas com número de imagens distintos, ou seja, quando se faz uma verificação do resultado com mais imagens é de notar que a qualidade da classificação das GMM não varia muito, mantendo sempre mais ou menos com o mesmo rácio. Embora que quando comparados os dois tipos de classificação da GMM, a classificação por probabilidade obtém melhor resultado independentemente do tamanho do bloco ou do critério a usar. No caso dos critérios a interseção via produto apresenta melhores resultados para a GMM probabilidade. Como está presente em [17] a interseção aparenta ser um método bastante válido, sendo que, no caso das GMM's, quando se obtém as probabilidades

do produto das mesma faz com que o resultado seja mais refinado, não eliminando possibilidades. No entanto, quando se obtém a mesma classificação através de rótulo o produto das classificações obtidas para diferentes canais já tem resultados diferentes.

O método KNN para uma classificação de dois tipos de classes com um grande aglomerados de valores de características tem um comportamento fraco, quando comparado com os outros dois diagnósticos estudados. A ANN quando comparada com os restantes apresenta uma boa performance de classificação, mas apresenta para algumas parametrizações algumas falhas, não deixando de ter, do ponto de vista do estudo apresentado, resultados motivadores.

### 6.3 CONCLUSÕES

Seguindo o sentido até agora escolhido de apresentação, análise do estudo feito e resultados obtidos, através das observações presentes na Secção 6.2.1, pode-se concluir que o espaço Lab tem melhores características para o problema em abordagem, no sentido que fornece melhores resultados no que diz respeito a homogeneidade do resultado consoante a entrada que lhe é fornecida. Por sua vez, também efetua uma distribuição de características por canais, que vai mais de encontro com o pretendido para análise, no sentido em que colocam de parte em um só canal a luminosidade, permitindo que o algoritmo tenha uma maior valência.

Deste modo, indo ao encontro de que foi diagnosticado para os gradientes estudados, é possível retirar ilações que o gradiente que demonstra melhor performance nos dois pontos dimensionados como mais relevantes, tempo de execução e qualidade de resultado, é o Sobel. Apresenta maior velocidade de processamento que os restantes e apresenta resultados de boa qualidade quando comparado com os restantes estudados.

No que diz respeito ao tempo de treino, apesar do que já foi referido, este tempo no caso dos algoritmos com supervisão, se fizessem parte do algoritmo final de classificação seria totalmente não viável, pois tornaria a deteção muito lenta. Contudo, como esperado o treino dos algoritmos sem supervisão é muito mais rápido que os algoritmos com, pois consoante o número de imagens o tempo de execução aumenta em todos eles, nuns mais significativamente que em outros, como se pode verificar pelas Tabelas 6.3 e 6.4.

O tempo de teste é aquele de maior importância, pois é o tempo de encontrar o resultado pretendido, um algoritmo que demore muito tempo a encontrar o resultado no problema em questão não é viável, sendo que neste caso maior velocidade de execução é o mais valorizado. Por isso, através das análises e comparações feitas pode se concluir que o algoritmo com melhor performance no que corresponde ao tempo de execução do teste de modo a obter o rótulo ou probabilidade, é o ANN, embora os dois algoritmos

das GMM também tenham uma boa performance. O KNN dimensionado apresenta uma velocidade de teste bastante elevada em relação aos restantes o que o torna a pior hipótese neste diagnóstico.

As classificações obtidas quando comparadas qualitativamente com cálculo o IoU, observa-se que tanto as GMM, como as ANN estudadas apresentam bom resultados com qualidades distintas, mas que tornam ambas duas soluções motivadores, enquanto o KNN fica muito à quem da expectativa e por isso o torna de certo modo a alternativa mais fraca, mas também inviável para o problema que se pretende resolver.

Concluindo, da análise feita nesta Secção através dos resultados apresentados, as duas opções mais motivadores para proposta de algoritmo nas condições em que foram feitos os testes, são a ANN e GMM, contudo com pontos positivos semelhantes mas em alguns pormenores apresentam qualidades de classificação divergentes, como por exemplo a questão de classificações erradas é mais recorrente na ANN e a questão de uma qualificação com menor qualidade nas GMM.

## Conclusão

Neste capítulo serão apresentadas as conclusões finais do estudo, que possibilitou a aquisição de novos conhecimentos acerca da inteligência artificial, até a melhoria das técnicas de programação utilizadas.

Inicialmente, é feito o esclarecimento detalhado do pré-processamento, a razão dos meios utilizados para a obtenção do objetivo, correlacionando os dados obtidos através do estudo com a opinião do executante, efetuando o mesmo para a parte da classificação efetuando uma análise mais detalhada, mais rica em pormenores explicando o que é o algoritmo final proposto. Com isto, pretende-se que o leitor fique mais informado e que tenha melhor percepção dos métodos aconselhados pelo executante do estudo.

Por fim, são apresentadas possíveis melhorias, possíveis abordagens que poderiam ter sido testadas e possíveis acréscimos ao que foi feito, efetuando uma breve conclusão final de tudo o descrito neste documento.

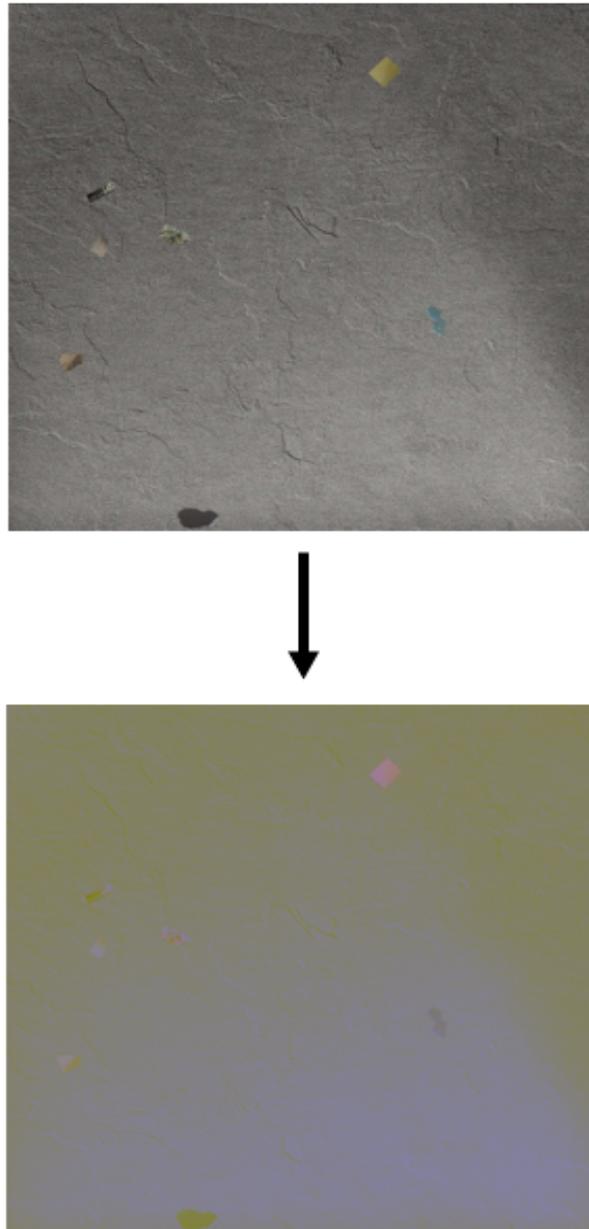
### 7.1 ALGORITMO PROPOSTO

A proposta de algoritmo vai ao encontro do documento que teve como base este estudo [17], com os métodos implementados pelo executante, ou seja, o espaço utilizado foi o Lab descrito na secção 6.1.1, o gradiente usado foi o Sobel presente na subsecção 3.3.1, com o cálculo da magnitude como apresentados na equação 3.19 da subsecção 3.3.2, tendo sido a extração de propriedades das imagens feita como está representado na 3.4, características estas que estão descritas nas sub-secções, no que diz respeito ao pré-processamento.

A proposta de classificação, é o uso de GMM em modo de probabilidades tal como está descrito na secção 4.2, como está explicado na subsecção 5.1.2 e indo ao encontro do pseudocódigo explicado das GMM com o critério de interseção por produto.

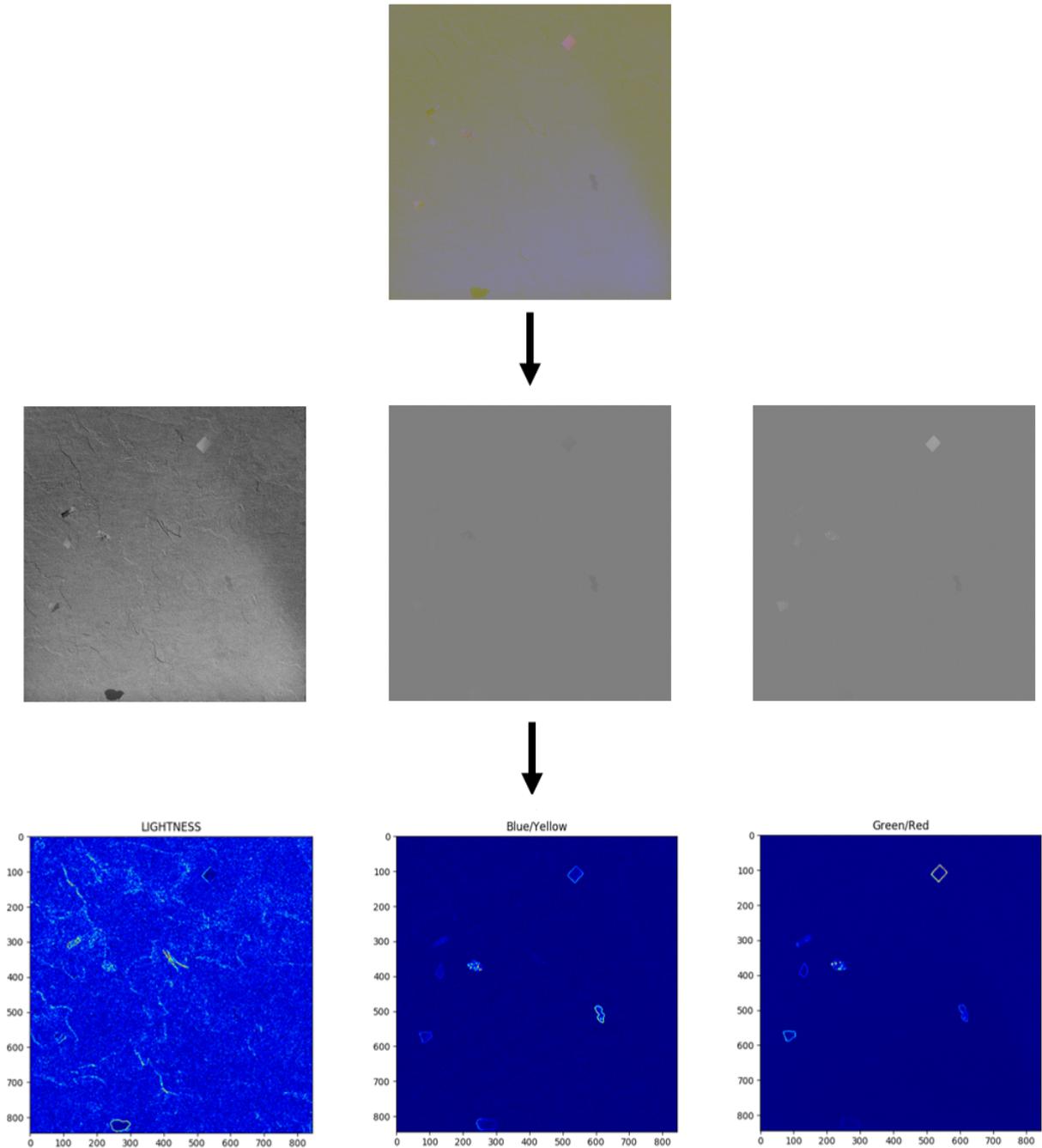
### 7.1.1 Pré-processamento

Este processamento inicial, é executado diretamente na imagem recebida, ou seja, é a primeira interação entre a entrada e o algoritmo, recorreu-se a exemplos gráficos.



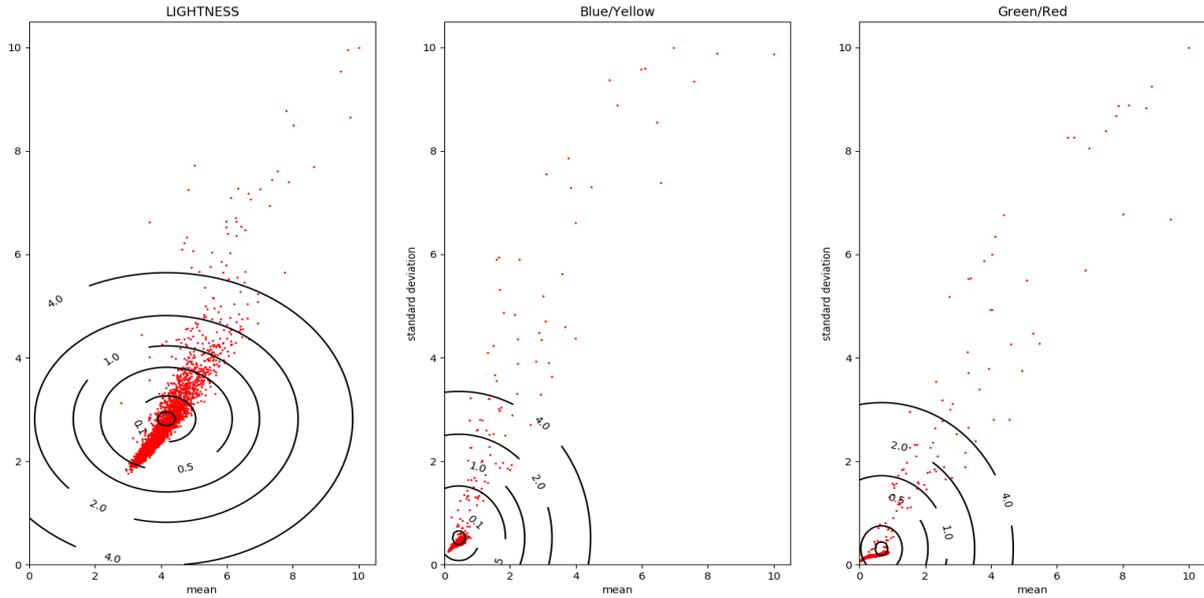
**Figura 7.1:** Exemplo da conversão de uma imagem BGR para Lab. Imagem têm presença de irregularidades e sombras.

Ao efetuar a divisão dos três canais, possível observar visualmente e confirmar com os valores dos pixels, que ambos registam diferentes tipos de informação, tendo o canal L o maior volume de informação e os restantes uma quantidade mais pequena, sendo mais relevante para a identificação do pretendido, como já foi descrito anteriormente.



**Figura 7.2:** Exemplo da divisão do espaço L, a e b, com os respectivos gradientes de cor.

A magnitude do gradiente é mais interessante visualizar num mapa de cor, como nas Figuras inferiores da Figura 7.2, conseguindo ter a percepção das transições de intensidade. Os mapas a e b, correspondentes a transições do azul para amarelo e verde para vermelho, tem uma quantidade de detalhe mais simplificada quando comparados com o L, o mapa do da luminosidade, tem uma quantidade de detalhe muito maior, o que permite concluir que os canais a e b irão servir como filtro de informação irrelevante quando intercetados com o L.



**Figura 7.3:** Distribuição das características da imagem exemplo.

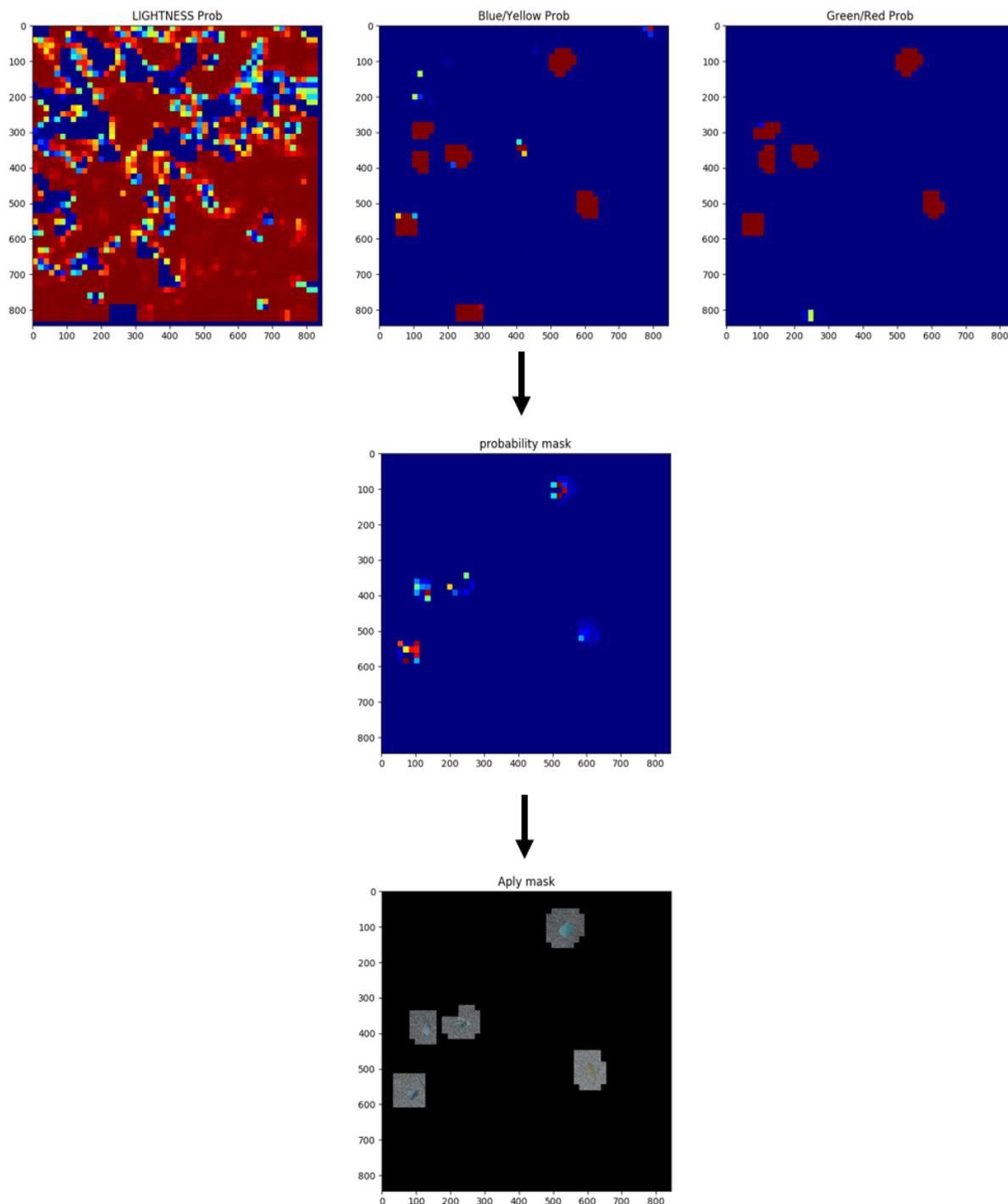
Nos gráficos da Figura 7.3, é possível visualizar os valores das características da imagem desvio padrão em função da média, assimilando que os valores nos círculos mais exteriores irão possivelmente representar blocos onde existe alguma percentagem da probabilidade, uma vez que quanto mais afastados do centro do conjunto a probabilidade aumenta.

A extração destas características da imagem foram feitas no caso deste exemplo com blocos de dimensão 32 x 32 e passo de 16, sendo que, que do ponto de vista do executante, para este tipo de imagem, conforme as dimensões da mesma é o melhor atribuir estes parâmetros. Os gráficos encontram-se normalizados de 0 a 10 por escolha do executante apenas.

### 7.1.2 Classificação

A escolha de uma classificação GMM recorrendo a classificações através de probabilidade, é a melhor opção do ponto de vista do executante, na medida em que, quando generalizando todos os resultados obtidos, apresenta a melhor combinação de tempo e qualidade de resultado para os distintos tipos de imagem analisados. Em relação à GMM que utiliza recurso de previsão é uma excelente opção também, o ganho é em velocidade e perda em qualidade, mas quando comparado a unidade com a velocidade não tem um ganho assim tão significativo enquanto a qualidade do resultado é sempre um aspeto de vital interesse neste assunto.

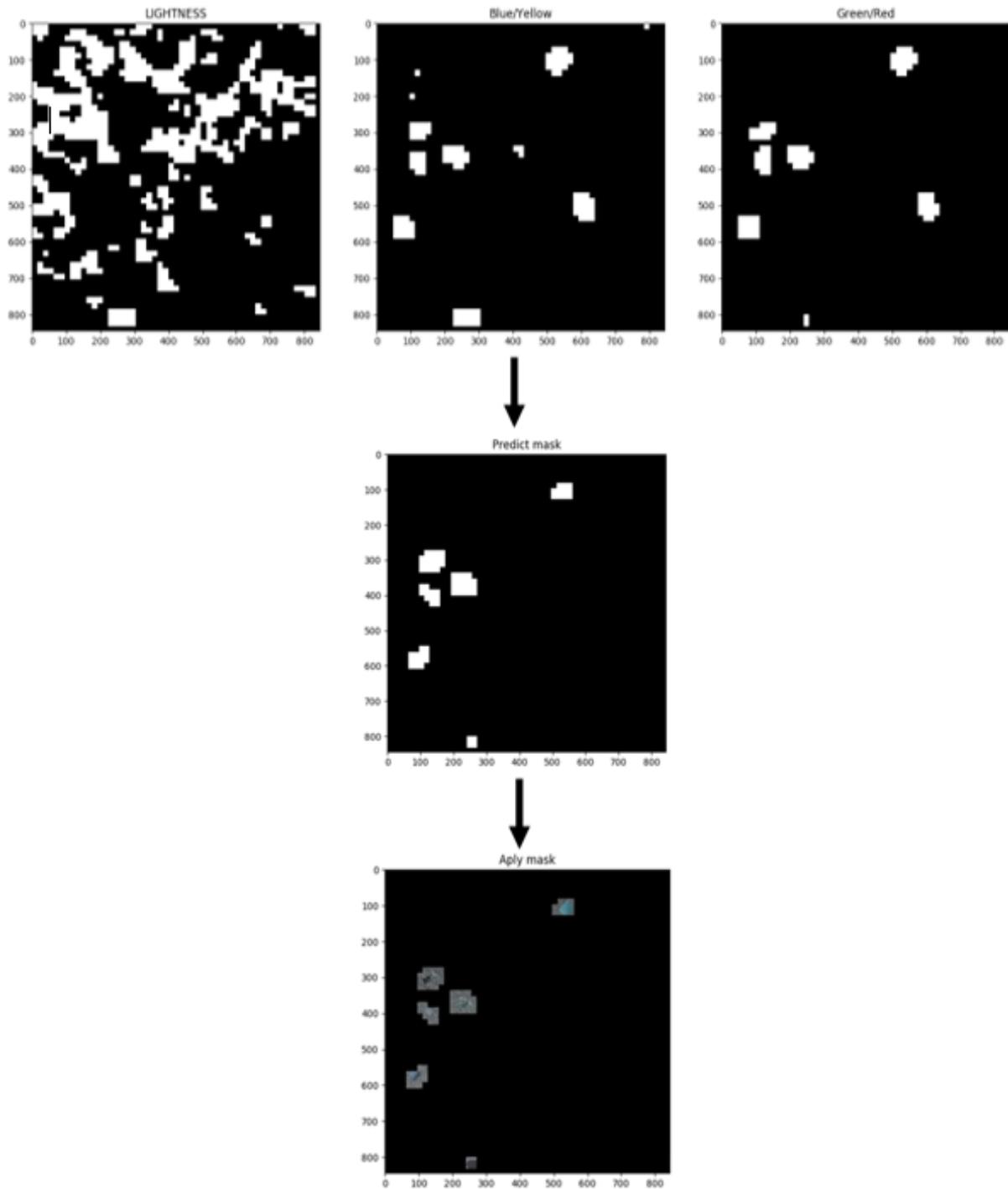
No exemplo da Figura 7.4, as imagens iniciais dizem respeito ao mapa de probabilidades, onde uma probabilidade alta de ter sujidade é representada por bordô e uma probabilidade baixa por azul, no mapa da luz pode-se verificar que como a quantidade



**Figura 7.4:** Mapa de probabilidades, interceção e aplicação da máscara ao exemplo.

de informação é tão elevada o algoritmo não consegue identificar de forma correta quais são as zonas da sujidade, enquanto os outros dois conseguem de maneira mais eficiente, quando intercetados o resultado não é perfeito, mas é de boa qualidade pois apenas não deteta uma mancha de sujidade, obtendo uma máscara com um IoU bastante elevado.

Para o exemplo da GMM apresentado na Figura 7.5, é usando as saídas através



**Figura 7.5:** Máscara usando GMM com recurso a previsão.

de previsão, apresenta bom resultado, mas verificando-se em grande escala melhor qualidade o GMM usando as saídas de probabilidade, como se pode ver através das tabelas 6.9, 6.10, 6.11 e 6.12, onde se encontram os valores obtidos para IoU.

## 7.2 ABORDAGENS FUTURAS

Outro tipo de soluções pensadas antes e durante a realização do estudo e dimensionamento do algoritmo foi a utilização de métodos de classificação de DL, sendo também sugerido por Rui Carapinha na sua tese de mestrado [16]. No entanto, identificou-se efetivamente que o problema em mãos, era um problema de ML, focando-se nestes tipos de métodos deixando de parte as metodologias de DL. Contudo, continua a ser uma possível abordagem apesar de não ter sido utilizada no estudo apresentado e até mesmo ser uma opção que leve a bons resultados.

A ANN Perceptron que foi dimensionada, foi utilizada pelo executante com o intuito de perceber como opera uma rede neural simples, tendo sido estudada com a maior simplicidade, mas uma possível melhoria seria tornar a rede mais complexa usando um maior número de neurões e aumentando o número de entradas, de modo a obter uma melhoria no resultado.

Concluindo, o fato de poder extrair mais recurso, características ou propriedades da imagem de uma forma diferente das utilizadas, com o intuito de criar mais uma entrada nos algoritmos é uma boa filosofia para melhoria dos mesmos, pois com mais entradas é possível obter as mesmas saídas, mas possivelmente estas saídas seriam de melhor qualidade, dado que é fornecida mais informação útil há classificação.

## 7.3 CONCLUSÕES FINAIS

O algoritmo proposto tem uma execução consideravelmente satisfatória, tendo proporcionado ao executante a possibilidade de aprender e melhorar nos vários temas abordados ao longo do estudo, que teve como base de elaboração o documento "The power of gmms"[17], embora que com algumas diferenças: o uso de classificação para 2 classes, o tipo de gradiente usado é o Sobel e o critério de interseção é o mesmo.

Ao longo do trabalho foram surgindo impasses, um deles foi naturalmente a percepção e estudo das possíveis abordagens que poderiam ser úteis para o problema, o uso dos dataset's disponíveis de contexto real tornaram-se um obstáculo para a validação dos métodos apresentados, apresentados devido à complexidade da imagem ser grande, noutros casos existe pouca informação quando se compara a informação do que pode ser sujidade e o que pode ser solo. Neste sentido, os dataset criados, validam com melhor resultado o funcionamento dos algoritmos, pois fornecem a informação necessária para o melhor funcionamento dos algoritmos apresentados, como era expectável. Uma vez que a segmentação de alguma sujidade não está cem por cento correta, faz com que os algoritmos de classificação com supervisão faz com algumas deteções não sejam de todo coerentes introduzindo classificações erradas, dependendo muito do tamanho do solo e do pavimento, que está em análise.

O dimensionamento onde foi utilizado o Kmeans não teve o resultado esperado, ficando muito à quem das expetativas, logo o algoritmo apesar de ter sido implementado, como não apresenta resultados de todo relevantes, por isso os resultados não foram incluídos na seção de análise. A forma como o dimensionamento foi pensado traz bastante falhas e introduções de ruído, porque o que o algoritmo faz é identificar, entre as caraterísticas recebidas, dois tipos de classes diferentes, assumindo sempre que a que apresenta maior número de caraterísticas é solo, aquelas caraterísticas que estiverem a uma distância normalizada entre o centro deste conjunto e o ponto mais afastado maior que 0.6, por exemplo, são caraterizadas como sujidade, sendo que, efetivamente vai identificar sujidade, ao mesmo tempo vai identificar muitos mais blocos que não apresentam sujidade, por isso não se torna uma solução viável, logo apresenta bons resultados.

Contudo, o estudo apresenta algumas soluções para a classificação de sujidade no solo, o algoritmo proposto não foi capaz de validar aquilo que era pretendido, como apresentado em "The power of Gmms"[17], no que diz respeito a imagem 6.21, não possível obter um resultado semelhante ao que se consegue obter com a ANN devido à complexidade da imagem e ser difícil para a GMM conseguir identificar 2 classes distintas nesse tipo de pavimento, ao contrário dos restantes contextos que apresentam resultados consideravelmente satisfatórios.

## Referências

- [1] T. H. V. M, «"Introduction". Fundamental of Digital Imaging,» pp. 1–6, 2008.
- [2] « Federal Agencies Digital Guidelines Initiative Glossary,» URL: [http : / / www . digitizationguidelines . gov/term.php?term=digitalimage](http://www.digitizationguidelines.gov/term.php?term=digitalimage).
- [3] W. J.B, «The Electronics Revolution: Inventing the Future,» pp. 245–248, 2017.
- [4] J. R. Janesick, «Scientific charge-coupled devices,» pp. 3–4, 2001.
- [5] «"2009 Nobel Prize in Physics awarded to Kao, Boyle, and Smith ,» 2009.
- [6] C. Starr, *Biology: Concepts and Applications*. Thomson Brooks/Cole, 2005, p. 94.
- [7] C. K. T. Poggio., «"Predicting the Visual World: Silenceis Golden. Nature Neuroscience ,» pp. 9–10, 1999.
- [8] T. S. Huang, «"Computer Vision : Evolution And Promise ,» pp. 21–25, 1996. DOI: 10.5170/CERN-1996-008.21.
- [9] Y. Aloimonos, «"Special Issue on Purposive and Qualitive Active Vision ,» 1992.
- [10] S. Papert, «"The Summer Vision Project ,» 1966.
- [11] D. Marr, «"Vision: A Computational Investigation into the Human Representation and Processing of Visual Information ,» 1982.
- [12] R. E. W. Rafael C. Gonzalez, *Digital Image Processing* . Pearson Prentice Hall, 2008, pp. 1–7, ISBN: 0-13-168728-x978-0-13-168728-8.
- [13] «"Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. ,» pp. 151–170, 1996. DOI: 10.1007/978-94-009-0279-4\_9.
- [14] 2018. URL: [https : / / bigdata - makesimple . com / machine - learning - explained - understanding-supervised-unsupervised-and-reinforcement-learning/](https://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/).
- [15] «"An analysis of single-layer networks in unsupervised feature learning ,» p. 3, 2011.

- [16] R. Carapinha, «Project i-RoCS - Dirt Detection System based on Computer Vision,» University of Aveiro, jul. de 2020, pp. 91–102.
- [17] G. A. H.-F. G. P. J. V. M, «The power of gmms: Unsupervised dirt spot detection for industrial floor cleaning robots. In: Towards Autonomous Robotic Systems: 18th Annual Conference 2017 (TAROS), Guldford UK,» pp. 436–449, 2007.
- [18] «"Cascaded Machine-Learning Technique for Debris Classification in Floor-Cleaning Robot Application  
,» 2018. URL: <https://www.mdpi.com/>.
- [19] «"Vision-Based Dirt Detection and Adaptive Tiling Scheme for Selective Area Coverage  
,» 2018. URL: <https://www.hindawi.com/>.
- [20] C. M. Bishop, «"Pattern Recognition and Machine Learning  
,» 2006.
- [21] S. E. Umbaugh, «"Digital image processing and analysis : human and computer vision applications with CVIPtools  
,» 2010.
- [22] S. Chen Yangkang; Fomel, «"Random noise attenuation using local signal-and-noise orthogonalization  
,» 2015. DOI: 10.1190/GEO2014-0227.1.
- [23] «"A Visual Dirt Detection System for Mobile Service Robots  
,» 2016.
- [24] X. H. L. Zhang, «"Saliency Detection: A Spectral Residual Approach  
,» p. 2, 2007. DOI: 10.1109/CVPR.2007.383267.
- [25] G. C. Shapiro Linda G. Stockman, «Computer Vision,» pp. 9–10, 2002.
- [26] «"Autonomous Dirt Detection for Cleaning in Office Environments  
,» 2013.
- [27] J. D'Amelio, *Perspective Drawing Handbook*. Dover Publication, 2003, p. 19.
- [28] «"Robot Perception of Static and Dynamic Objects with an Autonomous Floor Scrubber  
,» 2002.
- [29] *"Machine Learning e Data Science com Python*  
, 2017. URL: <https://www.udemy.com/course/machine-learning-e-data-science-com-python/>.
- [30] S. P. Lloyd, «Least square quantization in PCM,» pp. 129–137, 1982. DOI: 10.1109/TIT.1982.1056489.
- [31] E. Forgy, «Cluster analysis of multivariate data: efficiency versus interpretability of classifications,» pp. 768–769, 1965.
- [32] J. Hartigan, «Clustering algorithms. [S.l.]: John Wiley Sons, Inc.,» 1975.
- [33] G. McLachlan, «Finite Mixture Models, Wiley,» 2000.
- [34] C. Améndola, «"Moment varieties of Gaussian mixtures  
,» 2015. DOI: 10.18409/jas.v7i1.42.
- [35] J. L. Fix Evelyn; Hodges, «Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. USAF School of Aviation Medicine, Randolph Field, Texas.,» 1951. URL: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf>.

- [36] N. S. Altman, « "An introduction to kernel and nearest-neighbor nonparametric regression"(PDF). The American Statistician.,» pp. 175–185, 1992. DOI: 10.1080/00031305.1992.10475879. URL: <https://ecommons.cornell.edu/bitstream/handle/1813/31637/BU-1065-MA.pdf;jsessionid=055851E9C5B279A3A6EC00FCBA78BE9A?sequence=1>.
- [37] F. Rosenblatt, « "The Perceptron—a perceiving and recognizing automaton ,» 1957.
- [38] URL: [https://docs.opencv.org/4.5.2/d1/d5c/tutorial\\_py\\_kmeans\\_opencv.html](https://docs.opencv.org/4.5.2/d1/d5c/tutorial_py_kmeans_opencv.html).
- [39] URL: <https://scikit-learn.org/stable/modules/mixture.html>.
- [40] URL: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>.



## Anexos

### A.1 PSEUDOCÓDIGO DOS ALGORITMOS

---

**Algorithm 1** Calculo do gradiente

---

```

1: if gradiente = "normal" then
2:   for  $i = 1, 2, \dots, linhas$  do
3:     for  $j = 1, 2, \dots, colunas$  do
4:       Gradiente x = Ponto 3.13
5:       Gradiente y = Ponto 3.14
6:       Magnitude[i,j] = Calculo da magnitude
7:     end for
8:   end for
9: end if
10: if grad = "sobel" then
11:   Gradiente x = Ponto 3.15
12:   Gradiente y = Ponto 3.16
13:   for  $i = 1, 2, \dots, linhas$  do
14:     for  $j = 1, 2, \dots, colunas$  do
15:       Magnitude[i,j] = Calculo da magnitude
16:     end for
17:   end for
18: end if
19: if gradiente = "laplacian" then
20:   Gradiente = Ponto 3.18
21:   for  $i = 1, 2, \dots, linhas$  do
22:     for  $j = 1, 2, \dots, colunas$  do
23:       Magnitude[i,j] = Calculo da magnitude
24:     end for
25:   end for
26: end if
27: return Magnitude

```

---

---

**Algorithm 2** Calculo da magnitude

---

```
1: Magnitude = Ponto 3.19
2: return Magnitude
```

---

---

**Algorithm 3** Extração de características

---

```
1: for  $i = 1, 2, \dots, linhas$  do
2:   for  $j = 1, 2, \dots, colunas$  do
3:     média = Ponto 3.22
4:     desvio padrão = Ponto 3.23
5:     VetorCaraterísticas.append(média, desvio padrão)
6:   end for
7: end for
8: return VetorCaraterísticas, média, desvio padrão
```

---

---

**Algorithm 4** Calculo das Saídas

---

```
1: for  $i = 0, passo, linhas - dimensão + 1$  do
2:   for  $j = 0, passo, colunas - dimensão + 1$  do
3:     bloco retirado a imagem
4:     média do bloco
5:     if média > 120 then
6:       Saídas.append(1) sujidade
7:       break;
8:     end if
9:     Saídas.append(0) chão
10:   end for
11: end for
12: return Saídas
```

---

O número de linhas e colunas, presente nos algoritmos, as linhas e colunas é sempre relativo à imagem que está em análise.

---

**Algorithm 5** Distância Euclidiana

---

```
1: for  $i = 1, \dots, N$  do
2:   distância += Ponto 4.9
3: end for
4: return distância
```

---

---

**Algorithm 6** Distância Manhattan

---

```
1: for  $i = 1, \dots, N$  do
2:   distância += Ponto 4.10
3: end for
4: return distância
```

---

---

**Algorithm 7** Distância Minkowski

---

```
1: for  $i = 1, \dots, N$  do
2:   distância += Ponto 4.11
3: end for
4: return distância
```

---

---

**Algorithm 8** KNN

---

```
1: for  $i = 1, \dots, N$  do
2:   distâncias[i] = Distância Euclidiana entre dados conhecidos e novo bloco
3: end for
4: Vizinhaça = K vizinhos mais próximos(sorted)
5: for index em Vizinhaça do
6:   if Saídas[index] == 0 then
7:     classe0 +=1
8:   end if
9:   if Saídas[index] == 1 then
10:    classe1 +=1
11:  end if
12: end for
13: Verificar qual das classes é maioritária:
14: if classe0 > classe1 then
15:   return 0
16: end if
17: return 1
```

---

Relativamente aos algoritmos,  $N$  é o número de características, que no caso de estudo são apenas 2, média e desvio padrão.

### A.1.1 Perceptron:

*Perceptron Class*

---

**Algorithm 9** Init

---

```
1: Amostras para treino = Guardar as características
2: Saídas para treino = Guardar o numero de rotulos
3: Taxa = Guardar taxa de aprendizagem
4: Épocas = Guardar numero de épocas a usar
5: Limite = Guardar o limite
6: N = numero de amostras
7: Atributos = numero de características
8: Pesos = iniciar pesos
```

---

---

**Algorithm 10** Treino

---

```
1: for amostra em Amostras de treino do
2:   amostra.insert(0,-1)
3: end for
4: for  $i = 0, passo, Atributos$  do
5:   Presos.append(random)
6: end for
7:
8: Pesos.insert(0,Limite)
9: Népcas = 0
10: Erro = False
11:
12: while True do
13:   for index em N do
14:     for j em Atributos+1 do
15:       u += Calculo do potencial de activação
16:     end for
17:     y = Step(u) Obter a saída através da função de activação
18:     if y diferente de Saída de treino then
19:       Erro auxiliar = Saída de treino - y
20:       for k em Atributos+1 do
21:         Pesos[k] = Recalculado através da somatório do valor anterior com a
22:         multiplicação Taxa, Erro auxiliar e amostra
23:       end for
24:       Erro = True
25:     end if
26:   end for
27:   Népcas += 1
28:   if not Erro ou Népcas > Épocas then
29:     break
30:   end if
31: end while
32: return Népcas - 1
```

---

---

**Algorithm 11** Teste

---

```
1: amostra.insert(0,-1)
2: for index em Atributos+1 do
3:   u += Calculo do potencial de activação
4: end for
5: y = Step(u) Obter a saída através da função de activação
6: return y
```

---

---

**Algorithm 12** Step

---

```
1: if  $u \geq 0$  then  
2:   return 1  
3: end if  
4: return 0
```

---

---

**Algorithm 13** Guardar Pesos

---

```
1: Abrir ficheiro em modo de escrita  
2: Escrever no ficheiro o  $n^\circ$  de pesos no cabeçalho  
3: for index em  $n^\circ$  de Pesos do  
4:   Escrever respetivo peso  
5: end for  
6: Fechar ficheiro
```

---

*Load Perceptron Charges*

---

**Algorithm 14** Guardar Pesos

---

```
1: Abrir ficheiro em modo de leitura  
2:  $N =$  Ler a 1 $^\circ$  linha para saber quantos são os pesos a ler  
3: for index em  $N$  do  
4:    $\text{Pesos}[\text{index}] =$  Valor lido do ficheiro  
5: end for  
6: Fechar ficheiro  
7: return Pesos
```

---

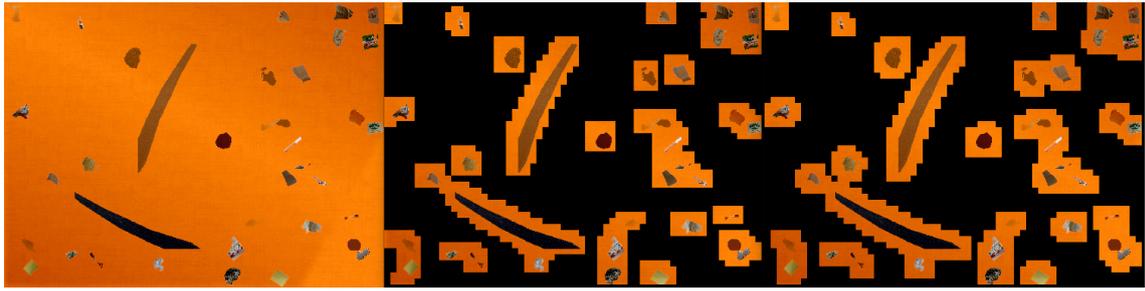
## A.2 EXEMPLOS DE DETEÇÕES

### A.2.1 Exemplos das GMM

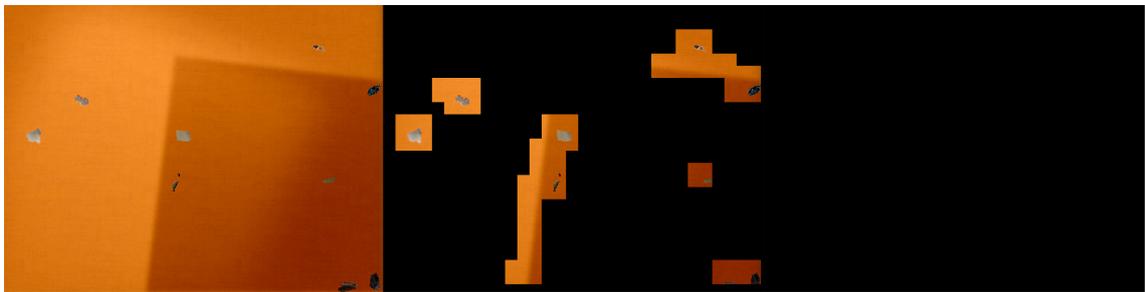
Os exemplos seguem sempre a mesma ordem imagem recurso para classificação, classificação em probabilidade e classificação por previsão.



**Figura A.1:** Detecção fraca em chão laranja com bloco 16 x 16 e critério maioritário,



**Figura A.2:** Detecção boa em chão laranja com bloco 16 x 16 e critério maioritário,



**Figura A.3:** Detecção fraca em chão laranja com bloco 16 x 16 e critério maioritário,



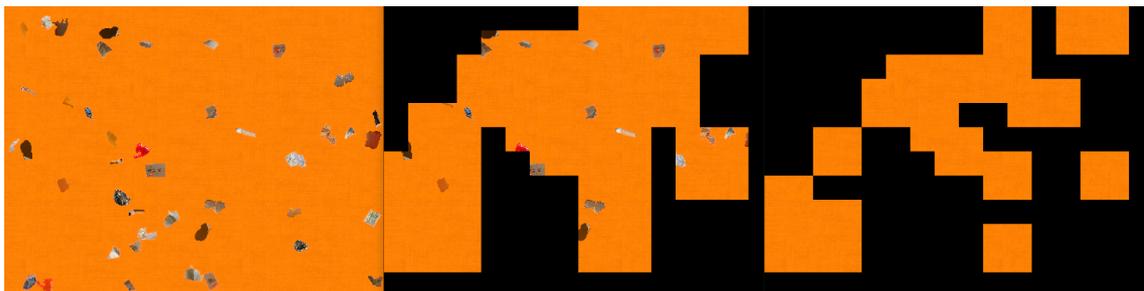
**Figura A.4:** Detecção boa em chão laranja com bloco 16 x 16 e critério maioritário,



**Figura A.5:** Detecção fraca em chão laranja com bloco 16 x 16 e critério maioritário,



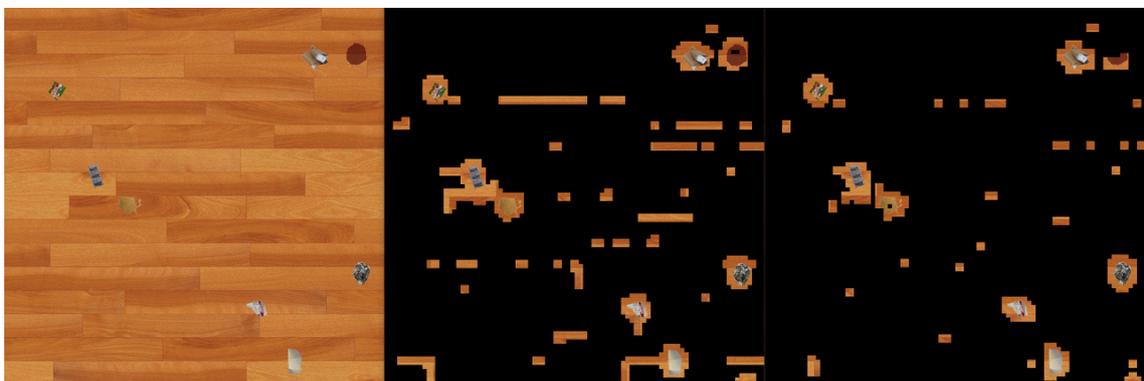
**Figura A.6:** Detecção fraca em chão laranja com bloco 16 x 16 e critério produto,



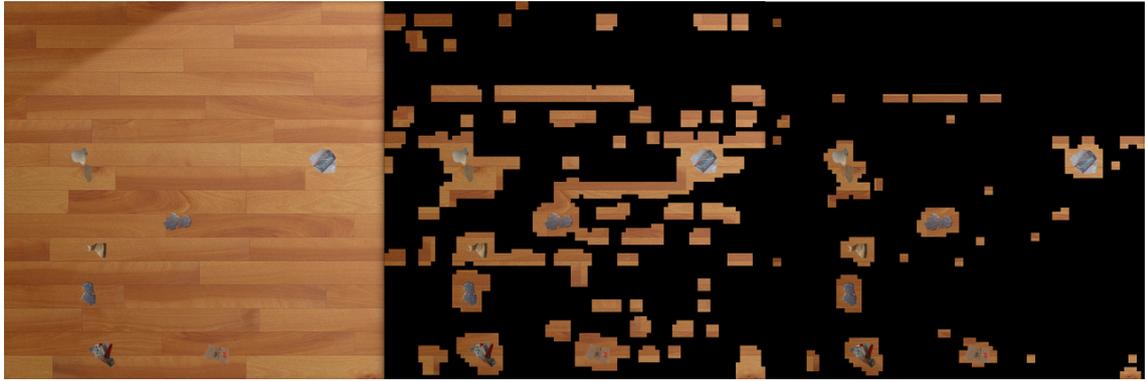
**Figura A.7:** Detecção fraca em chão laranja com bloco 16 x 16 e critério produto,



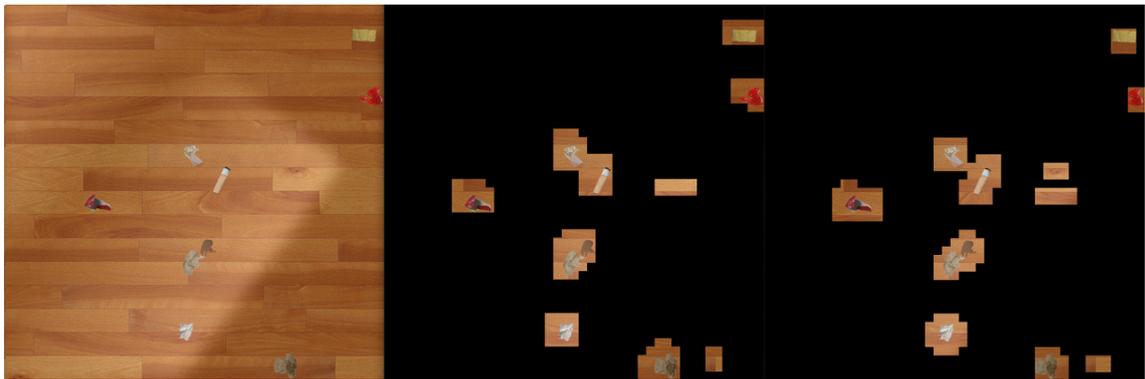
**Figura A.8:** Detecção fraca em chão de madeira com bloco 16 x 16 e critério maioritário,



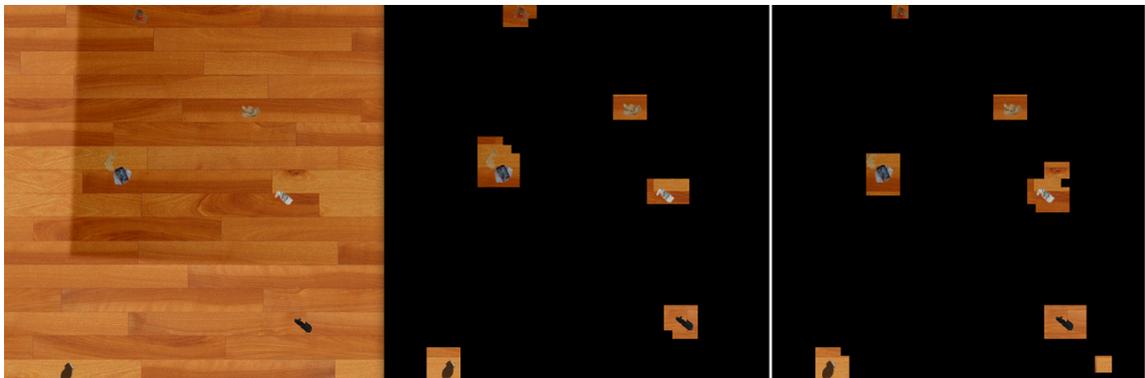
**Figura A.9:** Detecção fraca em chão de madeira com bloco 16 x 16 e critério maioritário,



**Figura A.10:** Detecção boa em chão de madeira com bloco 16 x 16 e critério produto,



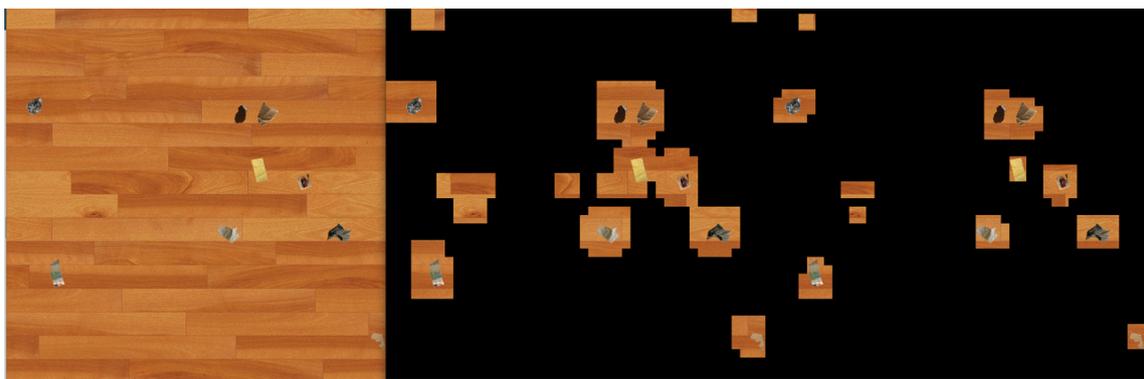
**Figura A.11:** Detecção fraca em chão de madeira com bloco 32 x 32 e critério maioritário,



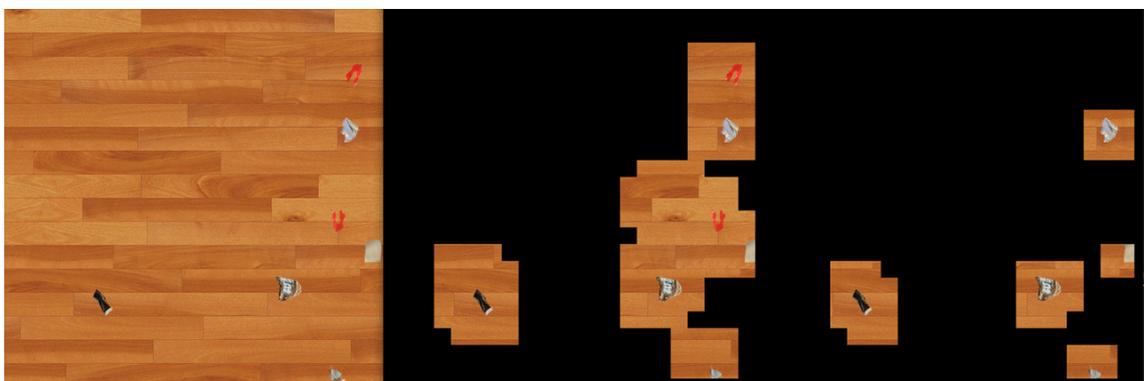
**Figura A.12:** Detecção boa em chão de madeira com bloco 32 x 32 e critério maioritário,



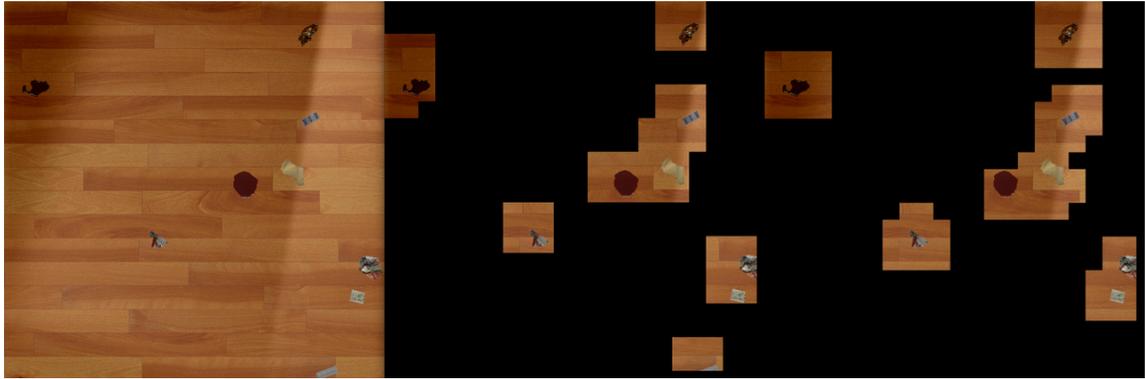
**Figura A.13:** Detecção boa em chão de madeira com bloco 32 x 32 e critério produto,



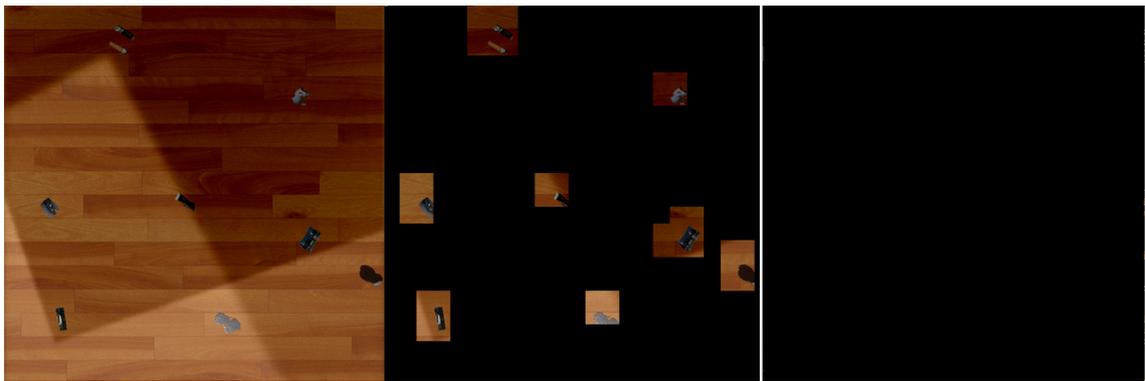
**Figura A.14:** Detecção boa em chão de madeira com bloco 32 x 32 e critério produto,



**Figura A.15:** Detecção boa em chão de madeira com bloco 64 x 64 e critério produto,



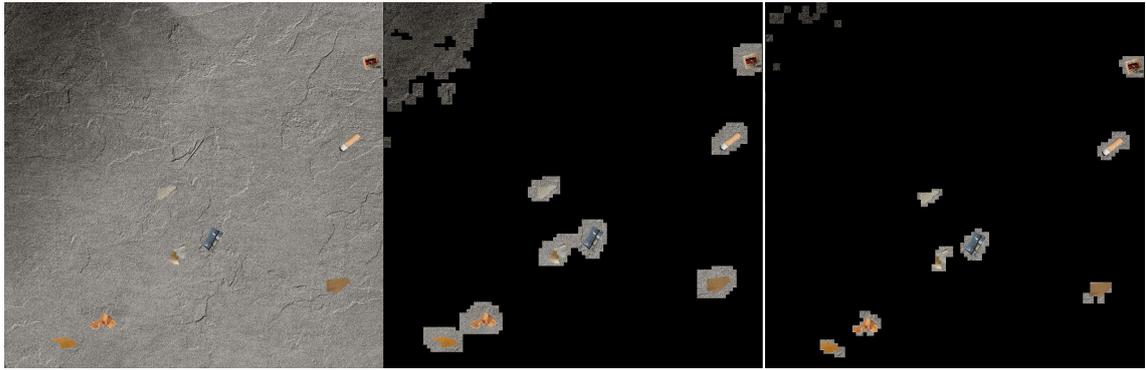
**Figura A.16:** Detecção boa em chão de madeira com bloco 64 x 64 e critério maioritário,



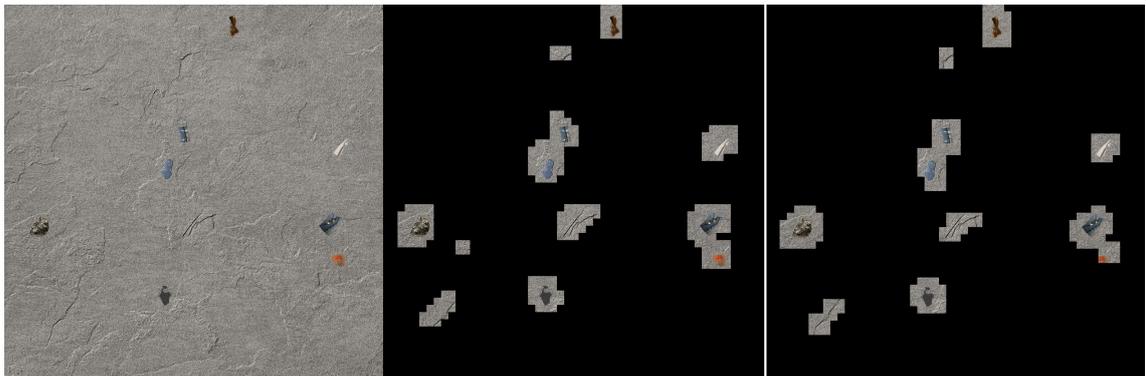
**Figura A.17:** Detecção fraca em chão de madeira com bloco 64 x 64 e critério maioritário,



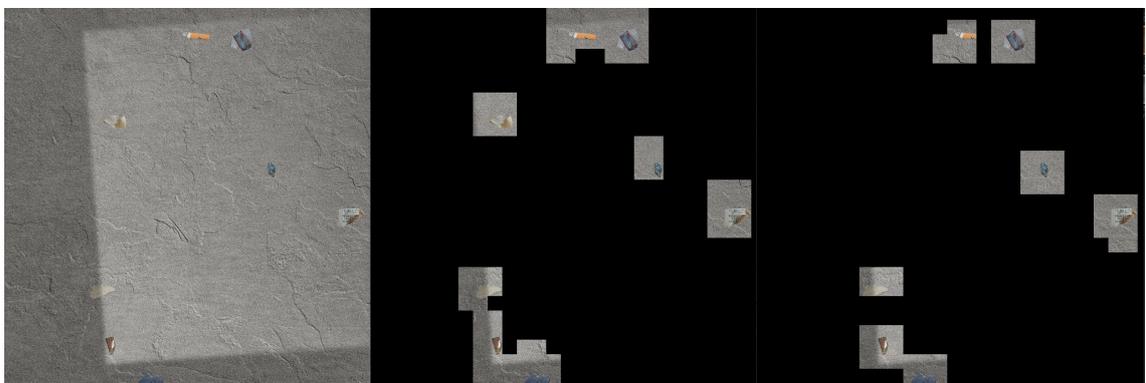
**Figura A.18:** Detecção fraca em chão de madeira com bloco 64 x 64 e critério produto.



**Figura A.19:** Detecção fraca em chão de cimento com bloco 16 x 16 e critério produto.



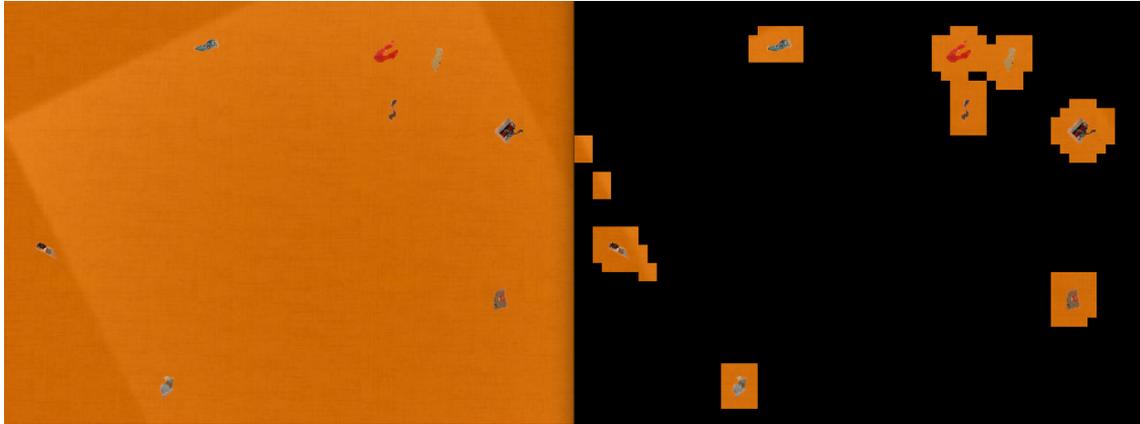
**Figura A.20:** Detecção fraca em chão de cimento com bloco 32 x 32 e critério maioritário.



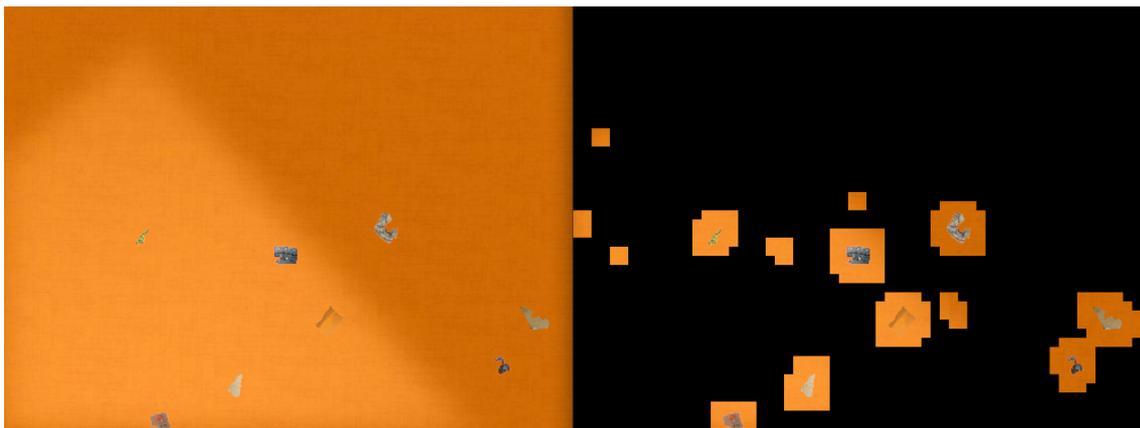
**Figura A.21:** Detecção boa em chão de cimento com bloco 64 x 64 e critério maioritário.

### A.2.2 Exemplos das ANN

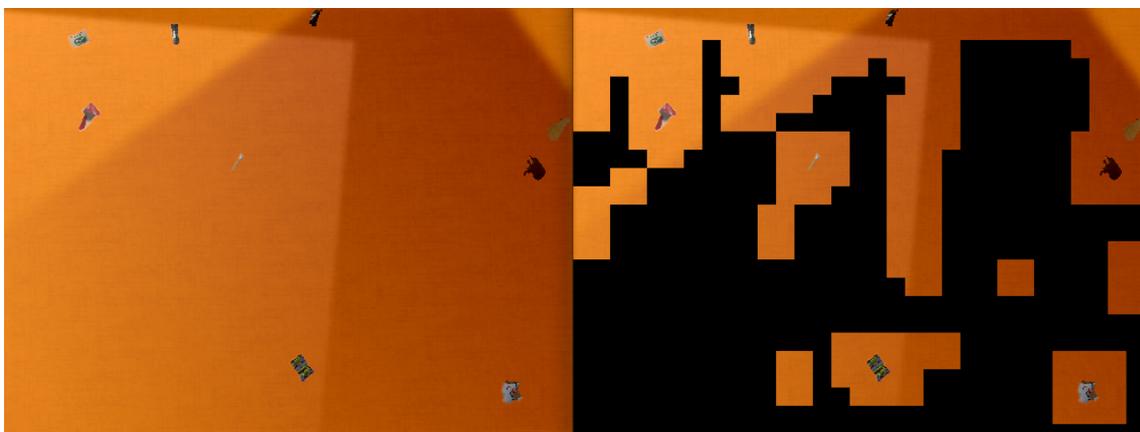
Os exemplos seguem sempre a mesma ordem imagem recurso para classificação e classificação.



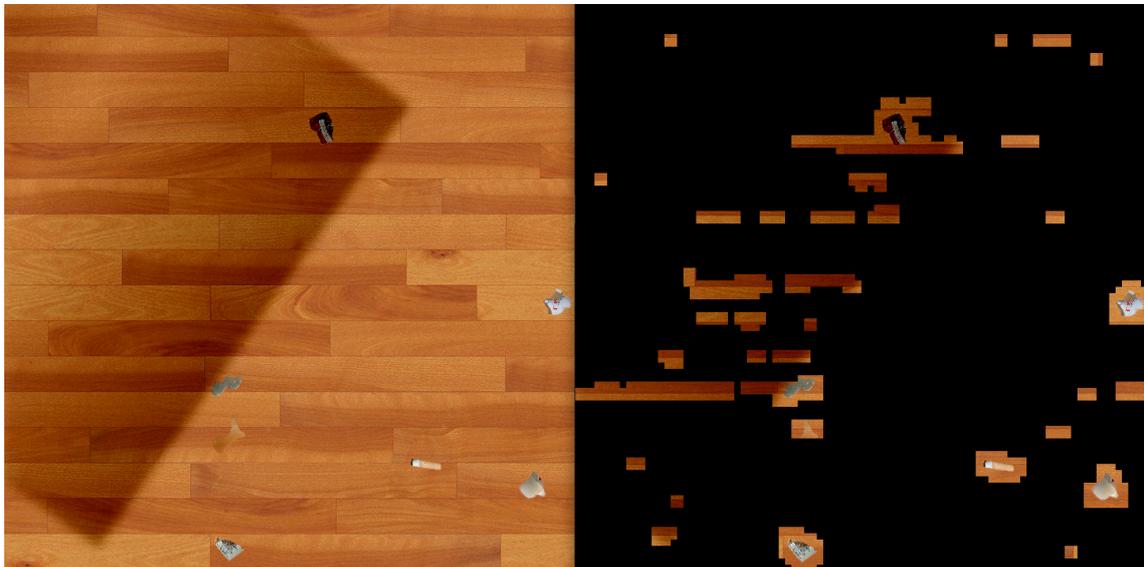
**Figura A.22:** Detecção boa em chão de laranja com bloco 16 x 16 e critério maioritário.



**Figura A.23:** Detecção boa em chão de laranja com bloco 16 x 16 e critério produto.



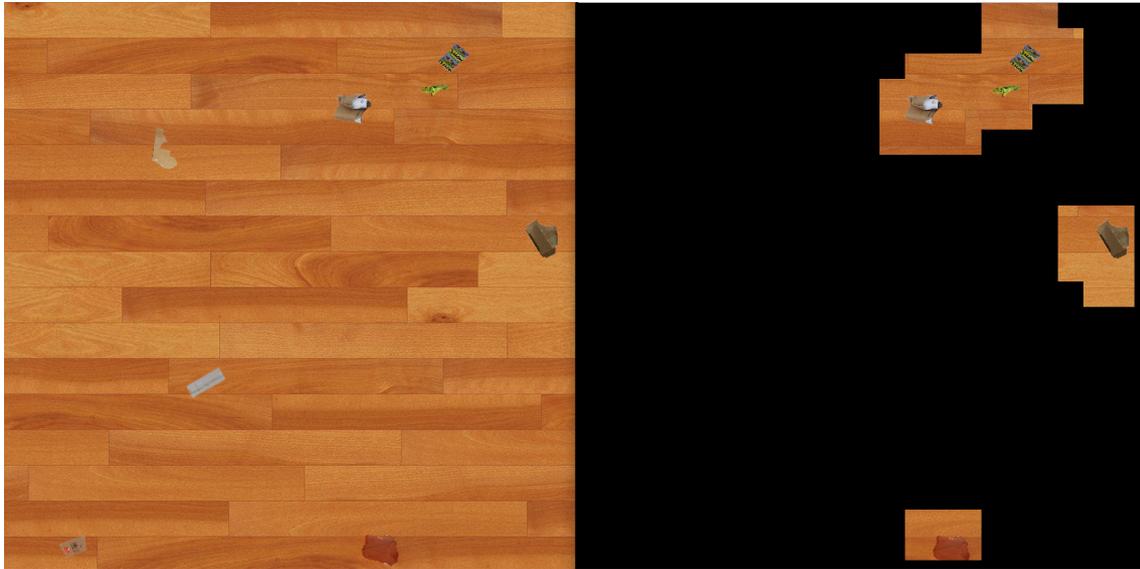
**Figura A.24:** Detecção fraca em chão de laranja com bloco 32 x 32 e critério produto.



**Figura A.25:** Detecção fraca em chão de madeira com bloco 16 x 16 e critério produto.



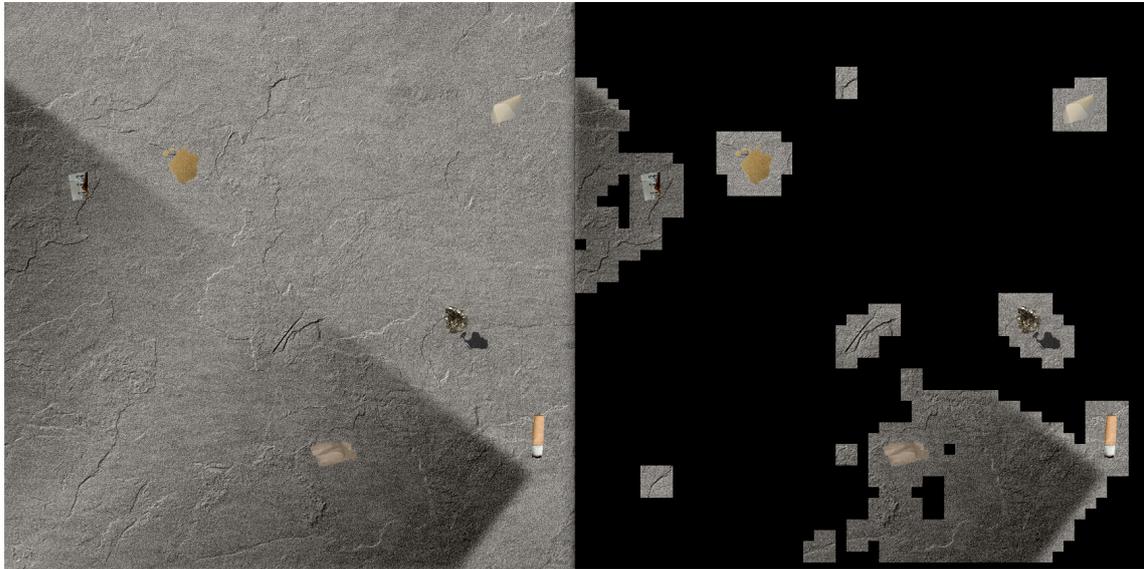
**Figura A.26:** Detecção boa em chão de madeira com bloco 16 x 16 e critério produto.



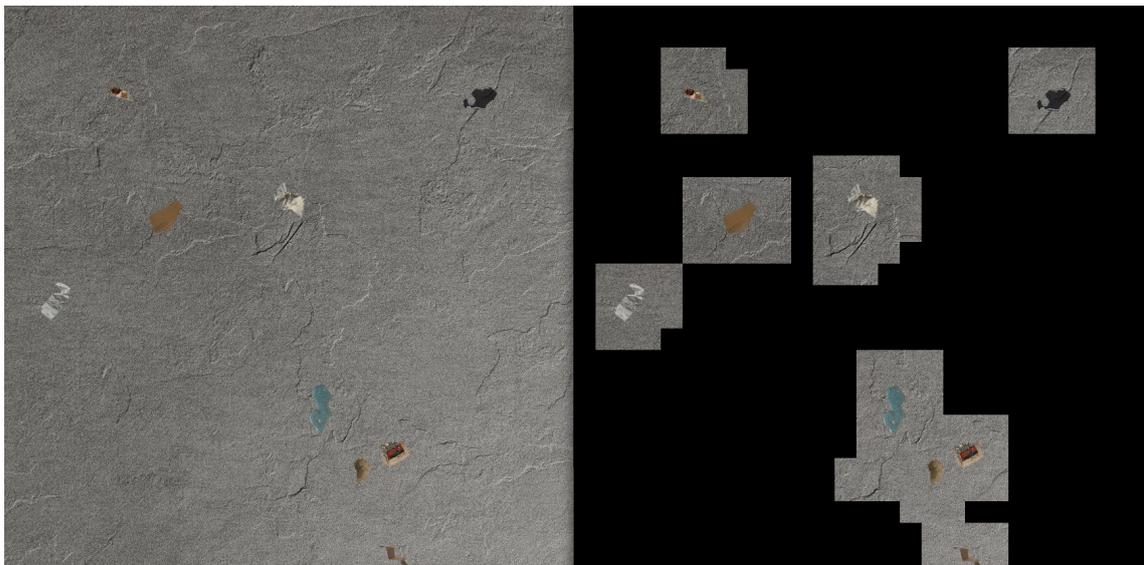
**Figura A.27:** Detecção fraca em chão de madeira com bloco 64 x 64 e critério produto.



**Figura A.28:** Detecção boa em chão de cimento com bloco 16 x 16 e critério maioritário.



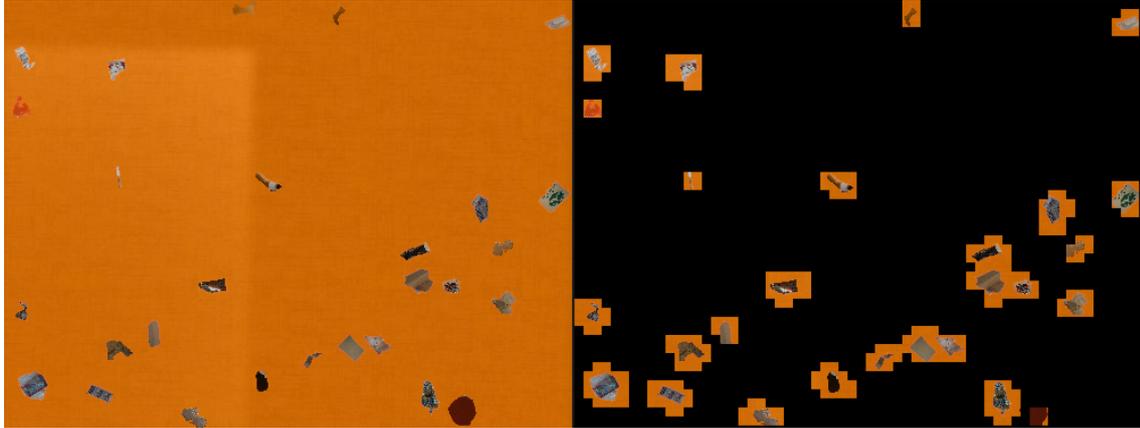
**Figura A.29:** Detecção fraca em chão de cimento com bloco 32 x 32 e critério maioritário.



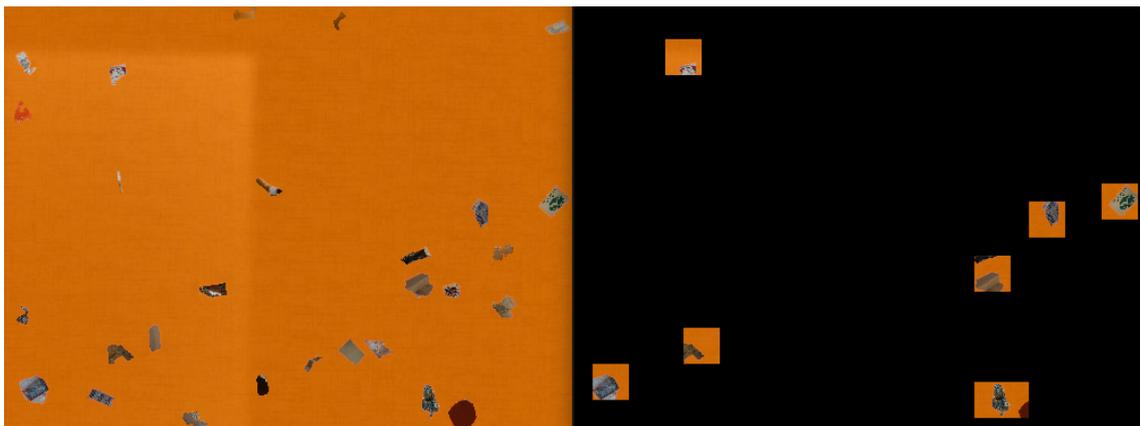
**Figura A.30:** Detecção boa em chão de cimento com bloco 64 x 64 e critério maioritário.

### A.2.3 Exemplos das KNN

Os exemplos seguem sempre a mesma ordem imagem recurso para classificação e classificação.



**Figura A.31:** Detecção boa em chão de laranja com bloco 16 x 16 e critério maioritário.



**Figura A.32:** Detecção fraca em chão de laranja com bloco 32 x 32 e critério maioritário.