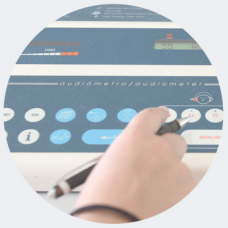
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA BIOMÉDICA






Segmentação Automática da Região Plantar Para Apoio ao Diagnóstico da Neuropatia Periférica

RUI PEDRO ROCHA MESQUITA
Outubro de 2022

Segmentação Automática da Região Plantar Para Apoio ao Diagnóstico da Neuropatia Periférica

Rui Pedro Rocha Mesquita
1171050

Mestrado em Engenharia Biomédica
Departamento de Física
Instituto Superior de Engenharia do Porto
2021/2022

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Biomédica

Candidato: Rui Pedro Rocha Mesquita, Nº 1171050, 1171050@isep.ipp.pt
Orientação científica: Luís Filipe Martins Coelho, lfc@isep.ipp.pt
Coorientação: Manuel Fernando dos Santos Silva, mss@isep.ipp.pt

Mestrado em Engenharia Biomédica
Departamento de Física
Instituto Superior de Engenharia do Porto
21 de outubro de 2022

Agradecimentos

"It is during our darkest moments that we must focus to see the light. - Aristotle"

Os últimos anos foram particularmente stressantes, com os desenvolvimentos da pandemia, início de conflitos armados na Europa e a necessidade de conciliar múltiplos objetivos pessoais. Contudo, foi possível contar com um conjunto de pessoas que surgiram como uma luz nestes momentos mais sombrios.

Em primeiro lugar, gostaria de agradecer à minha família, aos meus pais pelo apoio, suporte e carinho ao longo, quer dos anos de mestrado, quer dos de licenciatura, e às minhas irmãs por fazerem aquilo que os irmãos melhor fazem, nunca nos deixar desistir, mesmo quando a motivação é baixa. Seguidamente, gostaria de agradecer a um grupo pelo qual guardo uma relação especial, os meus amigos, tanto os novos, como os mais antigos. Com eles criei memórias que para sempre guardo, e que até ao dia de hoje definem a pessoa que me tornei.

O último agradecimento vai para os meus orientadores de projeto, o professor Luis Coelho, e o professor Manuel Silva, pela sua orientação e paciência no que toca à elaboração do seguinte documento e por sempre acreditarem nas minhas capacidades.

Resumo

De acordo com a organização mundial de saúde, o número de pessoas portadoras de diabetes tem crescido. A diabetes é a grande causadora de complicações como cegueira, deficiência renal, ataques cardíacos, tromboses e amputações dos membros inferiores [1].

As neuropatias estão entre as mais comuns das complicações a longo termo da diabetes. No mínimo, metade de todas as úlceras no pé em fase terminal podem ser evitadas através de uma correta gestão de cuidados, tratamentos e da educação dos pacientes. Contudo, a falta de tempo e informação adequada pode incutir um regime de cuidados desadequados à doença em questão [2].

Este tipo de condições pode ser de maior impacto ainda, uma vez que o sistema de saúde encontra-se cada vez mais sobrecarregado e com falta de recursos humanos. Logo, exames como estes podem ser negligenciados a favor de outro tipo de procedimentos mais sérios, causando então uma complicação grave no paciente em questão. Com o recurso a técnicas modernas de *software* e robótica seria possível ultrapassar a barreira humana do exame procedendo desta forma à automação do mesmo.

Nota-se que, uma grande porção dos trabalhos efetuados pelos médicos não podem ser automatizados, visto que se relacionam diretamente com a saúde do paciente. Todavia, existem pequenas tarefas, como o caso do exame à neuropatia periférica, que poderiam ser automatizados e efetuados por máquinas providas de algoritmos inteligentes.

Assim, o objetivo do trabalho centra-se na automação do exame à neuropatia periférica. Através de técnicas de *machine leaning* e inteligência artificial irá ser replicado o processo de visão humana, ou seja, a identificação de quais os pontos na porção plantar do pé do paciente que devem ser avaliados, e quais as suas coordenadas no espaço. Seguidamente, esta informação pode ser processada e enviada para um braço robótico, o qual irá efetuar o exame em si.

O primeiro grande marco do projeto consiste em introduzir inteligência arti-

ficial para uma tarefa de segmentação. Apesar da iteração anterior do trabalho, utilizando técnicas de processamento de imagem e segmentação, ter sido sucedida, a revolução tecnológica e os avanços ao nível da computação continuam cada vez mais robustos. Desta forma, partiu-se para uma solução mais sofisticada do ponto de vista tecnológico, a aplicação de processos de segmentação com recurso a algoritmos de *machine learning* e aprendizagem profunda.

A aplicação de tais conceitos permite ao utilizador uma abstração dos processos inerentes à segmentação de imagem, uma vez que os modelos desta natureza conseguem interpretar o ambiente em que estão inseridos e efetuar os devidos ajustes para chegar ao resultado esperado. O modelo em estudo foi o das redes de segmentação U-Net, famosas pela sua capacidade de produzir excelentes resultados face a uma dimensão reduzida de dados de treino. O modelo desenvolvido foi treinado com o recurso a um subconjunto de dados de treino, e outro subconjunto de dados de avaliação. Nota-se que, em ambas as partições, existia um par imagem - máscara em que a partição de treino foi expandida utilizando processos de aumento de dados, como distorções elásticas, morfológicas e mudanças de saturação.

Para os treinos da rede, foram definidos hiperparâmetros específicos, para os quais se fez variar maioritariamente o valor das *epochs* entre iterações. Primeiramente foi desenhado um modelo para fazer a segmentação unicamente dos pontos inerentes aos dedos do pé (segmentar 3 pontos por pé), o qual foi treinado ao longo de 10, 20 e 50 *epochs*. Uma vez verificada a eficácia do modelo implementado para uma segmentação simples, procedeu-se ao treino da rede preditiva final (segmentação de 9 pontos por pé).

Para validar os resultados obtidos, desenhou-se um *script* de teste que utiliza o modelo gerado como entrada para fazer a segmentação das imagens do subconjunto de teste. Assim é possível avaliar o resultado esperado (obtido manualmente) face aquele que o modelo produziu. Durante esta execução ainda são calculadas métricas típicas para este tipo de algoritmos de segmentação e métricas personalizadas, desenvolvidas especialmente para este trabalho, as quais contam o número de pontos detetados em diferentes regiões do pé.

Após o treino e validação das seis redes, verificou-se que o modelo que apresentou melhor comportamento a nível de qualidade de saída e desempenho foi o de 20 *epochs*. Mais tarde, foram ainda gerados mais dois modelos com vista a tentar atingir resultados ainda melhores, procedendo a um aumento do número de dados de treino, e a um aumento do número de *workers*. Contudo, nenhum destes novos modelos exibiu um comportamento melhor que os anteriores.

Palavras-Chave: Diabetes, neuropatia periférica, inteligência artificial, rede U-Net, segmentação.

Abstract

According to the World Health Organization, the number of people with diabetes has been growing. Diabetes is the major cause of complications such as blindness, kidney impairment, heart attacks, thrombosis and lower limb amputations [1].

Neuropathies are among the most common of the long-term complications of diabetes. At least half of all end-stage foot ulcers can be prevented through proper care management, treatment and patient education. However, lack of time and adequate information can instil a care regime that is inappropriate for the disease in question [2].

These types of conditions can be even more impactful as the healthcare system is increasingly overstretched and understaffed. Therefore, tests such as these can be neglected in favour of other more serious procedures, causing a serious complication for the patient in question. With the use of modern software and robotic techniques, it would be possible to overcome the human barrier of the examination, thus proceeding to its automation.

It is noted that a large portion of the work done by doctors cannot be automated, since it is directly related to the patient's health. However, there are small tasks, such as the peripheral neuropathy examination, which could be automated and carried out by machines provided with intelligent algorithms.

Thus, the objective of the work is focused on the automation of the peripheral neuropathy examination. Through machine leaning techniques and artificial intelligence, the human vision process will be replicated, that is, the identification of which points on the plantar portion of the patient's foot should be assessed, and their coordinates in space. Then, this information can be processed and sent to a robotic arm, which will perform the examination itself.

The first major milestone of the project is to introduce artificial intelligence for a segmentation task. Although the previous iteration of the work, using image processing and segmentation techniques, was successful, the technological

revolution and advances in computing continue to be more and more robust. In this way, a more sophisticated solution from the technological point of view was started, the application of segmentation processes using machine learning and deep learning algorithms.

The application of such concepts allows the user an abstraction of the inherent processes to the image segmentation, once the models of this nature are able to interpret the environment in which they are inserted and make the necessary adjustments to reach the expected result. The model under study was the U-Net segmentation networks, famous for their ability to produce excellent results with a small amount of training data. The model developed was trained using a subset of training data, and another subset of evaluation data. Note that in both partitions there was an image-mask pair, where the training partition was expanded using data augmentation processes, such as elastic and morphological distortions and saturation changes.

For the network trainings, specific hyperparameters were defined, for which the value of epochs was mostly varied between iterations. Firstly, a model was designed to perform the segmentation only of the points inherent to the toes (segment 3 points per foot), which was trained throughout 10, 20 and 50 epochs. Once verified the effectiveness of the model implemented for a simple segmentation, we proceeded to the training of the final predictive network (segmentation of 9 points per foot).

To validate the results obtained, a test script was designed that uses the generated model as input to perform the segmentation of the images of the test subset. Thus, it is possible to evaluate the expected result (obtained manually) against the one produced by the model. During this execution, typical metrics for this type of segmentation algorithms and custom metrics are also calculated, which were developed especially for this work, counting the number of points detected in different regions of the foot.

After training and validation of the six networks, it was verified that the model that presented the best behaviour in terms of output quality and performance was the one with 20 epochs. Later, two more models were generated in order to try to achieve even better results, increasing the number of training data and the number of "workers". However, none of the new generated models had a better behaviour than the previous ones.

Keywords: Diabetes, peripheral neuropathy, artificial intelligence, U-Net, segmentation.

Índice

Agradecimentos **v**

Índice **i**

Índice de Figuras **v**

Acrónimos **ix**

1 Introdução **1**
1.1 Contextualização 2
1.2 Motivação 3
1.3 Objetivos 4
1.4 Calendarização 4
1.5 Organização do relatório 4

2 Inteligência Artificial **7**
2.1 O que é inteligência artificial 7
2.2 A história da inteligência artificial 9
2.3 Inteligência artificial forte e fraca 10
2.4 Métodos de aprendizagem das redes neuronais 11
2.4.1 Aprendizagem supervisionada 11
2.4.2 Aprendizagem não supervisionada 12
2.4.3 Aprendizagem por reforço 13
2.5 Aplicações da inteligência artificial 13
2.5.1 Aplicações na medicina 14
2.5.2 Aplicações no mundo digital 15
2.6 Conclusão do capítulo 2 17

3 Visão Artificial com Aprendizagem Automática **19**
3.1 Como funciona a visão artificial 19

3.2 Redes para segmentação de imagem (U-Net) 21
3.3 Tarefas com o recurso a visão artificial 22
3.3.1 Reconhecimento 23
3.3.2 Análise de movimento 23
3.4 Aplicações 24
3.4.1 Análise de imagem médica 24
3.4.1.1 Visão artificial para a deteção de nódulos pulmonares 25
3.4.1.2 Visão artificial e marcas moleculares para o diagnóstico de fibrose pulmonar 25
3.4.1.3 Modelos preditivos na hipertensão pulmonar . . . 26
3.4.1.4 IA na pandemia de SARS-COV-2 26
3.4.2 Realidade virtual e realidade aumentada na medicina . . . 26
3.5 Conclusão do capítulo 3 27

4 Materiais e métodos **29**
4.1 *Pipeline* geral de execução 29
4.2 Conjunto de dados 30
4.2.1 Descrição do conjunto de dados 30
4.2.2 Criação das máscaras binárias 31
4.2.3 Criação das máscaras RGB 32
4.2.4 Aumento de dados 32
4.3 Modelo de dados 33
4.3.1 Framework computacional 33
4.3.2 Arquitetura 35
4.4 Treino da rede 36
4.5 Localização dos pontos segmentados pelo algoritmo 38
4.6 Conclusão do capítulo 4 38

5 Análise de Resultados **41**
5.1 Avaliação de desempenho 41
5.2 Modelo de análise de 3 pontos 43
5.2.1 10 epochs 44
5.2.2 20 epochs 46
5.2.3 50 epochs 47
5.3 Modelo de análise de 9 pontos 49
5.3.1 10 epochs 50
5.3.2 20 epochs 52
5.3.3 50 epochs 55
5.4 Ajustes ao modelo com melhor desempenho 56
5.4.1 Expansão do conjunto de dados 56
5.4.2 Aumento do tamanho de lote 59

5.5 Filtro para a deteção de pontos 60
5.6 Obtenção de coordenadas reais 61
5.7 Conclusão do capítulo 5 64

6 Conclusões **65**
6.1 Conclusão 65
6.2 Trabalhos futuros 67

Referências Bibliográficas **69**

Índice de Figuras

1.1 Diagrama de Gantt com o plano de trabalhos 5

2.1 Rede neuronal artificial com múltiplas camadas 8
2.2 Classificação de imagens com uma rede convolucional 12
2.3 Evolução das número de publicações de artigos relacionados com IA . 15

3.1 Blocos de construção de uma CNN 21
3.2 Arquitetura da U-Net 22

4.1 *Pipeline* geral da arquitetura do modelo de inteligência artificial para a segmentação dos pés. 30
4.2 Exemplo de imagem do conjunto de dados. 31
4.3 Exemplo de máscara para uma imagem do conjunto de dados. 32
4.4 Exemplo de máscara RGB para uma imagem do conjunto de dados. . 32
4.5 Manipulações das imagens de entrada. a) imagem original; b) inversão horizontal; c) inversão vertical; d) rotação; e) saturação Hue; f) distorção ótica e de rede 34
4.6 Arquitetura U-Net com as etiquetas correspondentes às classes no código. A zona vermelha corresponde às operações efetuadas pela classe *encoder_block*, a verde às operações da *decoder_block*, e a laranja às operações da *conv_block*. 36

5.1 Esquema da distribuição de diferentes áreas de interesse do pé, de acordo com as dimensões propostas por Price, Carina *et al* [3]. 43
5.2 Evolução dos valores de perda de validação e perda de treino ao longo de 10 *epochs* com um lote de tamanho dois. 44
5.3 Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 10 *epochs*. 45
5.4 Sobreposição da máscara manual e da máscara gerada pelo modelo de 10 iterações, sobre a imagem de origem. 45

5.5 Evolução dos valores de perda de validação e perda de treino ao longo de 20 *epochs* com um lote de tamanho dois. 46
5.6 Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 20 *epochs*. 47
5.7 Sobreposição da máscara manual e da máscara gerada pelo modelo de 20 iterações, sobre a imagem de origem. 47
5.8 Evolução dos valores de perda de validação e perda de treino ao longo de 50 *epochs* com um lote de tamanho dois. 48
5.9 Imagem original, máscara de segmentação obtida manualmente, e máscara de segmentação gerada pelo modelo de 50 *epochs*. 49
5.10 Sobreposição da máscara manual e da máscara gerada pelo modelo de 50 iterações, sobre a imagem de origem. 49
5.11 Valores de métricas extraídas em cada modelo. 50
5.12 Evolução dos valores de perda de validação e perda de treino ao longo de 10 *epochs*, com um lote de tamanho dois, para o modelo de 9 pontos. 51
5.13 Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 10 *epochs* e 9 pontos. . . . 51
5.14 Sobreposição da máscara manual, e da máscara gerada pelo modelo de 10 iterações e 9 pontos, sobre a imagem de origem. 52
5.15 Número de pontos detetados para diferentes áreas dos pés ao longo das 20 imagens de avaliação do modelo de 10 iterações. 52
5.16 Evolução dos valores de perda de validação e perda de treino ao longo de 20 *epochs*, com um lote de tamanho dois, para o modelo de 9 pontos. 53
5.17 Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 20 *epochs* e 9 pontos. . . . 53
5.18 Sobreposição da máscara manual e da máscara gerada pelo modelo de 20 iterações e 9 pontos, sobre a imagem de origem. 54
5.19 Número de pontos detetados para diferentes áreas dos pés ao longo das 20 imagens de avaliação do modelo de 20 iterações. 54
5.20 Evolução dos valores de perda de validação e perda de treino ao longo de 50 *epochs*, com um lote de tamanho dois, para o modelo de 9 pontos. 55
5.21 Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 50 *epochs* e 9 pontos. . . . 56
5.22 Sobreposição da máscara manual e da máscara gerada pelo modelo de 50 iterações e 9 pontos, sobre a imagem de origem. 56
5.23 Número de pontos detetados para diferentes áreas dos pés ao longo das 20 imagens de avaliação do modelo de 50 iterações. 57
5.24 Valores de métricas extraídas em cada modelo de 9 pontos. 57
5.25 Evolução dos valores de perda de validação e perda de treino ao longo de 50 epochs, com um lote de tamanho dois, para o modelo de 9 pontos com um conjunto de dados expandido. 58

5.26 Métricas extraídas para o modelo de 50 *epochs* e 9 pontos e para o modelo de 50 *epochs* e 9 pontos com o conjunto de dados expandido. . 58
5.27 Evolução dos valores de perda de validação e perda de treino ao longo de 30 *epochs* com um lote de tamanho três para o modelo de 9 pontos. 59
5.28 Métricas extraídas para o modelo de 50 *epochs* e 9 pontos, e para o modelo de 30 *epochs* e 9 pontos com tamanho de lote 3. 60
5.29 Representação visual de quais os pontos encontrados pelo algoritmo de recolha de pontos para uma das máscaras geradas pelo modelo preditivo. 61
5.30 Processo de segmentação dos quadrados amarelos. a) Filtro HSV, b) Segmentação após dilatação e erosão, c) Resultado da segmentação. . 62
5.31 Imagem original com a máscara manual e imagem original com as coordenadas reais dos pontos segmentados pelo modelo preditivo. . . . 62
5.32 Dispositivo OAK-D e as suas diferentes câmaras. 63
5.33 Imagens obtidas pelas câmaras *stereo*. a) captura da câmara esquerda, b) captura da câmara direita, c) sobreposição das duas imagens, dando origem à imagem de disparidade. 63
5.34 a) disparidade resultante da sobreposição das camaras *stereo*, b) introdução do mapa de cores sobre os valores de disparidade. 64
5.35 Resultado de uma previsão em tempo real com a imagem de profundidade para a extração do valor de z. 64

Acrónimos

Acrónimo	Descrição	Página
NPD	Neuropatia Periférica Diabética	2
IA	Inteligência Artificial	7
MIT	Instituto de Tecnologia de Massachusetts	10
EUA	Estados Unidos da América	10
IoT	Internet das Coisas	15
OMS	Organização Mundial de Saúde	19
CNN	Rede Neuronal Convolucional	20
RV	Realidade Virtual	24
TAC	Tomografia Axial Computorizada	25
NLST	*National Lung Screening Trial*	25
ROC	Característica de Funcionamento do Recetor	25
TACAR	Tomografia Axial Computorizada de Alta Resolução	25
HAP	Hipertensão Arterial Pulmonar	26
RA	Realidade Aumentada	26
JPEG	*Joint Photographic Experts Group*	31
GIMP	GNU *Image Manipulation Program*	31
GIF	*Graphics Interchange Format*	31
GPU	Unidade de Processamento Gráfico	34
CUDA	*Compute Unified Device Architecture*	37
BCE	*Binary Cross-Entropy*	37
DCS	Dice-Sørensen *coefficient*	37
HSV	*Hue Saturation Value*	61
ROI	Região de Interesse	63

Capítulo 1

Introdução

Segundo a organização mundial de saúde, o número de pessoas portadoras de diabetes tem crescido, em 1980 registavam-se 108 milhões de afetados, contrastando com os 422 milhões de 2014. A prevalência sofre um aumento mais significativo em países de baixos e médios rendimentos, face aos de elevados rendimentos [1]. A diabetes é a grande causadora de complicações como cegueira, deficiência renal, ataques cardíacos, tromboses e amputações dos membros inferiores [1]. A diabetes é uma doença crónica que ocorre quando o pâncreas não consegue produzir a quantidade necessária de insulina, ou o sistema não é capaz de a usar eficientemente. No que lhe concerne, a insulina é a hormona que regula a quantidade de glucose no sangue, permitindo que as células a absorvam a partir da corrente sanguínea, para a converter em energia ou armazenar para o futuro. Hiperglicemia, ou elevado glicose no sangue, são efeitos comuns de uma diabetes não controlada e acartam consequências graves para todo o sistema, principalmente no sistema nervoso e vascular [4]. Atualmente a diabetes pode ser dividida em 3 categorias distintas.

A diabetes tipo 1 resulta de uma destruição das células beta do sistema autoimune no pâncreas, resultando numa falha total de produção de insulina [5]. Esta classe de diabetes representa cerca de 5% a 10% de todos os casos da doença. Os seus fatores de risco incluem fatores autoimunes, genéticos e ambientais. Sendo que até aos dias de hoje, não existe uma forma conhecida de se prevenir esta categoria de diabetes [5].

Apesar de ainda não existir uma "cura" para esta condição, os dados mais recentes e os resultados de estudos preliminares sugerem as células T (células do sistema imunitário, as quais são originadas a partir de células estaminais da medula óssea, e ajudam a defender o corpo de infeções) como a escolha para estratégias preventivas [6].

A diabetes tipo 2 é o tipo de diabetes mais comum, que conta com cerca de 90% a 95% de todos os pacientes diabéticos e espera-se ainda que este número aumente para 439 milhões até 2030. Esta patologia resulta de interações complexas entre ambientes hereditários, juntamente com outros fatores de risco, como obesidade e um estilo de vida sedentário. A diabetes tipo 2 e as complicações a si inerentes constituem um problema para a saúde pública a nível mundial, afetando quase todas as populações, tanto em países desenvolvidos, como em desenvolvimento [5].

Outro tipo de diabetes, menos conhecida, é a diabetes gestacional, que se resume numa intolerância à glucose que afeta algumas mulheres durante a gravidez. O controlo e manutenção dos níveis de glicemia em mulheres com diabetes gestacional é crucial para prevenir complicações durante o desenvolvimento da criança. As mulheres que já tenham contraído este tipo de doença durante uma gravidez anterior, têm uma maior probabilidade de a contrair novamente, e o risco de contrair diabetes do tipo 2, durante a sua vida, aumenta para cerca de 20% a 50% [5, 1].

Um dos problemas associados à diabetes é a neuropatia periférica. Esta doença é uma desordem normalmente encontrada na prática clínica. Tendo em conta o envelhecimento da população, a pandemia da diabetes e a obesidade, a prevalência da neuropatia periférica está a aumentar, a qual constitui uma preocupação significativa em termos de saúde pública [7].

1.1 Contextualização

As neuropatias estão entre as mais comuns das complicações a longo termo da diabetes, afetando cerca de 50% dos pacientes. No passado, a falta de consciência e a gestão inadequada da neuropatia periférica diabética (NPD) levou à morbilidade e a custos de saúde elevados, os quais poderiam ser facilmente evitados. No mínimo, metade de todas as úlceras no pé em fase terminal podem ser evitadas através de uma correta gestão de cuidados, tratamentos e da educação dos pacientes. Contudo, a falta de tempo e falta de informação adequada pode incutir um regime de cuidados desadequados à doença em questão [2].

Embora alguns pacientes possam sofrer com sintomas extremamente violentos, outros com marcas de neuropatia periférica mais acentuadas podem ser assintomáticos. O diagnóstico requer a examinação extensa e cuidada dos membros inferiores. O controlo da doença passa também por identificar se esta condição é causada pela diabetes ou por outra causa diferente [2].

A inteligência artificial está a revolucionar e fortalecer a medicina moderna através da introdução de tecnologias que podem prever, aprender e atuar, quer seja em identificar relações existentes entre códigos genéticos ou controlar robôs

de assistência em cirurgia. Consegue identificar pequenos padrões, os quais poderiam passar despercebidos ao ser humano. Atualmente a inteligência artificial tem um papel importante no que toca aos descobrimentos de novas drogas e medicamentos, testes clínicos e cuidados do paciente [8].

Similarmente com o que acontece nas áreas acima mencionadas, é possível concluir que se poderia desenhar um processo de automação para o diagnóstico e exame da NPD. Para este efeito é necessário cobrir duas áreas em particular, o processo de identificação, localização e mapeamento das zonas do membro inferior que se pretendem avaliar, e seguidamente o processo de atuação, ou seja, de que forma realizar o procedimento mecânico do exame.

Não obstante, todos estes temas podem ser resolvidos recorrendo à inteligência artificial, algoritmos de segmentação de imagem e à robótica, respetivamente.

1.2 Motivação

Durante os últimos anos tem-se assistido diariamente a notícias sobre o estado do sistema de saúde. As instituições de saúde estão constantemente sobrelotadas, quer seja por falta de profissionais da área, quer seja pela busca desnecessária por parte da população. O agravar desta situação foi particularmente notável nos dois últimos anos de pandemia (2020/2021). O aparecimento do SARS-COV-2 (COVID-19) só veio expor ao de cima o que muitos já sabiam, o sistema de saúde não consegue lidar eficazmente com um grande volume de pedidos, muito menos eficientemente.

Desta forma, tudo o que foram pedidos não relacionados com COVID-19 foram colocados em fila de espera. Contudo, interprete-se isto como um alerta para aquilo que se vive atualmente. Na indústria moderna seria impensável ter profissionais para realizar todo o tipo de tarefas que uma linha de produção necessita. Isto porque muitas destas tarefas são demasiado repetitivas e excessivamente dispendiosas a nível de tempo humano. Como se ultrapassa este problema? Automatizando as tarefas mais simples para que os funcionários de carne e osso se possam focar nas tarefas que realmente importam e que não podem ser de todo automatizadas. Se atualmente os avanços tecnológicos, a inteligência artificial e a robótica parecem ter um futuro tão próspero, então, porque se continua a deixar a indústria da saúde fora da era moderna?

Nota-se que, uma grande porção dos trabalhos efetuados pelos médicos não podem ser automatizados, visto que se relacionam diretamente com a saúde do paciente, e nenhum utente se sentiria seguro ao receber uma avaliação clínica por parte de uma máquina e não de um médico real. Todavia, existem pequenas tarefas, como o caso do exame à neuropatia periférica, realizado em pacientes portadores de diabetes, que poderiam ser automatizados e efetuados por máquinas

providas de algoritmos inteligentes. Neste caso em concreto, todo o exame poderia ser efetuado pela máquina, cabendo apenas ao médico observar o relatório final e tomar as medidas necessárias em função do mesmo.

1.3 Objetivos

O objetivo primário de todo trabalho passa pela automação do exame à neuropatia periférica, o qual é tipicamente realizado por um profissional de saúde. Através de técnicas de *machine leaning* e inteligência artificial irá ser replicado o processo de visão humana, ou seja, a identificação de quais os pontos na porção plantar do pé do paciente que devem ser avaliados, e quais as suas coordenadas no espaço. Seguidamente, esta informação será processada e enviada para um braço robótico, o qual irá efetuar o exame em si. Na extremidade do robô estará colocado um monofilamento, o qual é tipicamente utilizado no exame à neuropatia diabética. De acordo com os dados fornecidos pelo algoritmo de visão artificial, o braço irá deslocar-se para as posições que serão avaliadas, e fará o filamento entrar em contacto com o pé do paciente. O paciente, de seguida, terá a missão de indicar se sentiu o toque, ou se não identificou nenhum tipo de contacto. O robô irá tentar executar o procedimento para o mesmo ponto até um máximo de três vezes, caso o paciente continue sem responder, então, o braço será deslocado até ao próximo ponto.

1.4 Calendarização

O Diagrama de Gantt ilustrado na Figura 1.1 sumariza o plano de trabalhos para este projeto.

1.5 Organização do relatório

Ao longo do Capítulo 1 é apresentado o tema selecionado para a elaboração da dissertação. No mesmo é realizada uma descrição do que se pretende obter com a realização do seguinte trabalho, é também fornecida uma contextualização, motivação e agenda de trabalhos.

Durante o Capítulo 2 expõem-se os tópicos de interesse para o projeto, a inteligência artificial e *machine learning*. Nesta secção apresenta-se a evolução deste tema, desde a sua origem até aos dias modernos, os algoritmos mais conhecidos para este tipo de tecnologia, e por fim, as aplicações existentes, com ênfase nas aplicações médicas.

O Capítulo 3 retoma os conteúdos do capítulo posterior, contudo agora com uma direção mais voltada para a visão artificial, explicando em que consiste este conceito, quais os algoritmos de inteligência artificial que já existem nesta área,

Figura 1.1: Diagrama de Gantt com o plano de trabalhos

como funcionam os mecanismos de visão artificial, e quais as suas principais funções na vida quotidiana.

No Capítulo 4 entra-se na parte prática do trabalho, onde é possível perceber quais as abordagens utilizadas no que toca à arquitetura do *software*, desenvolvimento de algoritmos, processamento de dados, treino e teste de modelos de *machine learning*, e deteção dos pontos segmentados.

Os principais resultados obtidos são descritos no Capítulo 5. Esta secção conta com a lógica e raciocínio aplicados ao longo do trabalho para se conseguir atingir o objetivo pretendido.

No último capítulo, o 6, são reunidas as principais conclusões e perspetivados os futuros desenvolvimentos.

Capítulo 2

Inteligência Artificial

A inteligência artificial (IA) é um termo geral que implica o uso de um modelo computacional com um comportamento inteligente, e com a mínima intervenção humana. É descrita como a ciência, e a engenharia, de criar máquinas inteligentes, oficialmente nascida em 1956 [9]. É um ramo da ciência que se foca em criar máquinas inteligentes e 'software' que consiga executar múltiplas tarefas que requerem inteligência humana, é um sistema que replica várias funções que o Homem consegue executar fazendo o uso de grandes quantidades de dados para atingir o desempenho desejado [10]. O que era outrora apenas um sonho dos filmes de ficção científica é agora uma realidade derivada dos desenvolvimentos do mundo tecnológico. Na medicina, estes métodos já estão presentes tanto na forma física, como na virtual. A forma virtual conta com aplicações informáticas derivadas da gestão de informação e recursos com base em algoritmos de aprendizagem profunda. Relativamente à forma física, aqui refere-se essencialmente a robôs utilizados para a assistência cirúrgica, ou com a população mais envelhecida. Nota-se que, a nano-robótica também já faz uso destas técnicas para a distribuição controlada de fármacos [9].

2.1 O que é inteligência artificial

Desde a revolução industrial, houve um vasto desenvolvimento no campo tecnológico. Muito trabalho manual foi substituído por tecnologia, a qual ajudou de uma forma drástica a humanidade. A inteligência artificial foi uma destas inovações [10]. A inteligência artificial é definida como o ramo da ciência e engenharia responsável pela compreensão computacional daquilo que é normalmente designado como comportamento inteligente, e a criação de produtos que exibam esse mesmo comportamento. Os programas que permitem aos computadores ope-

rar de forma aparentemente inteligente para o observador externo são designados como sistemas inteligentes [11].

Uma das técnicas mais populares de IA são as redes neuronais artificiais. Esta técnica é fortemente inspirada no sistema nervoso biológico, que consiste em redes de processos computacionais (neurónios) altamente interligados que conseguem efetuar computação de forma paralela para processamento de dados e representação lógica dos mesmos. Esta categoria de redes possui propriedades que as tornam extremamente apelativas, como a sua habilidade para aprender a partir de exemplos históricos, analisar dados não lineares, gerir informação imprecisa e permitir uma aplicação generalizada do modelo a um conjunto de dados independentes [11]. Em 1983, McCulloch e Pitts definiram um neurónio como sendo uma unidade de estado mutável que recebe informação de outros neurónios, como se pode ver na Figura 2.1, sendo que em função do total dessa mesma informação se define se o neurónio em causa permanece ativo ou permuta o seu estado para inativo. [12].

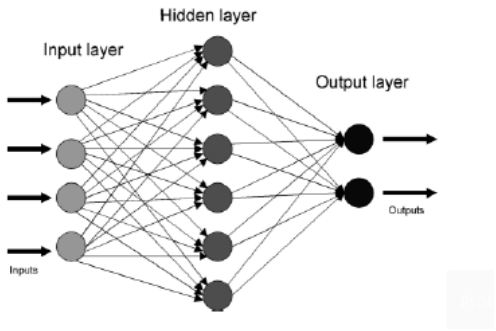


Figura 2.1: Rede neuronal artificial com múltiplas camadas [11]

Ao aplicar algoritmos que replicam os processos de verdadeiros neurónios é possível ensinar a rede a resolver um vasto conjunto de problemas. Cada um destes neurónios é classificado como um nó unitário. Recebe como parâmetro de entrada um valor numérico proveniente de outras unidades, ou de fontes externas e cada entrada possui um peso que irá ter influência na soma final de todas as entradas. Se a soma for superior a um determinado limite, então a saída desse nó será 1, caso contrário a mesma será 0 [12].

Para se atingir a solução face a um problema, é necessário atribuir os pesos e os limites necessários para que a rede consiga produzir o resultado expectável. Este processo pode ser conseguido através de um processo iterativo, no qual são apresentados exemplos cuja solução é conhecida, um após o outro. A este processo chama-se aprendizagem, ou treino, uma vez que se assemelha ao processo que o ser humana emprega quando tenta aprender algo. No caso do computador a aprendizagem reduz-se ao ajuste dos parâmetros da rede em cada exemplo novo

que lhe é apresentado e estas alterações conduzem a um aumento da qualidade da solução final [12, 13].

Desta perspetiva, o treino de uma rede neuronal é um problema não linear de minimização. O algoritmo mais conhecido para o treino é o de propagação retroalimentada (*backpropagation*). Os modelos de *backpropagation* distinguem-se dos de *feed-forward* uma vez que os primeiros vão continuamente melhorando o resultado ao utilizar as saídas da iteração anterior como entradas da iteração seguinte. Os modelos de *feed-forward* apenas estabelecem os valores de entrada uma única vez e executam a previsão do resultado [12, 14].

2.2 A história da inteligência artificial

Embora seja difícil de localizar temporalmente, as primeiras aparições de conceitos relacionados com IA podem ser provavelmente traçadas até 1940, particularmente em 1942, quando Isaac Asimov, um escritor americano de ficção científica, publicou uma pequena história, o *Runaround*. O desenvolver da narrativa gira em torno de um robô desenvolvido pelos engenheiros Gregory Powell e Mike Donavan e das três leis da robótica. Um robô não deverá, de forma alguma, colocar a vida do ser humano em risco; o robô deverá obedecer imparcialmente às ordens humanas, exceto quando em conflito com a primeira regra; o robô deverá sempre proteger a sua existência desde que não entre em conflito com a primeira e segunda regra. Este trabalho inspirou gerações de cientistas e engenheiros na área da robótica, inteligência artificial e ciência de computadores [15].

Possivelmente, uma das máquinas mais famosas da história foi a *The Bombe*, nome atribuído pelo governo britânico ao dispositivo desenvolvido pelo matemático Alan Turing durante a Segunda Guerra Mundial para decifrar o Enigma, o código utilizado pelos alemães para trocar mensagens encriptadas entre si. Esta máquina, até hoje considerada o primeiro computador eletromecânico funcional, com dimensões de 3 por 3 por 0,6 metros e com o peso de cerca de uma tonelada, deixou Turing com a dúvida acerca da inteligência de tais máquinas. Sendo que em 1950, o mesmo publica um artigo denominado *Computing Machinery and Intelligence* [16, 15]. Contudo, a expressão inteligência artificial só ficou oficialmente conhecida seis anos depois, quando, em 1956, Marvin Minsky e John McCarthy organizaram um programa de investigação em inteligência artificial de cerca de oito semanas no Dartmouth College em New Hamsphire. Este *workshop* de verão marca oficialmente o começo da "primavera de IA", patrocinado pela Rockefeller Foundation, reuniu aqueles que mais tarde seriam conhecidos como os "Fundadores da inteligência artificial" [17].

Após esta conferência, seguiram-se duas décadas que viram um sucesso imenso no que diz respeito à IA. Um desses exemplos foi o programa ELIZA, criado entre 1964 e 1966 por Joseph Weizenbaum no Instituto de Tecnologia de Massachusetts

(MIT). A ELIZA era uma ferramenta de processamento natural de linguagem que simulava uma conversa com um humano. Desta forma, foi um dos primeiros programas capaz de tentar passar no teste de "Turing". Naturalmente que histórias de sucesso como esta fomentavam cada vez mais projetos e investigação na área [15].

No entanto, em 1973 os elevados custos na investigação para a inteligência artificial começaram a ser fortemente criticados pelos congressos e governos. Nesse mesmo ano, James Lighthill publica um artigo onde questiona o otimismo apontado para a IA, dizendo que estas máquinas apenas conseguiriam atingir níveis amadores no que toca a jogos como o xadrez, e que o senso comum estaria sempre além das suas capacidades. Em resposta, o governo britânico decide cessar o apoio financeiro a investigações relacionados com IA em quase todas as universidades. O governo dos Estados Unidos da América (EUA) rapidamente decide optar por esta opção também. Este período inicializa assim o "inverno da IA" [15, 18].

Uma razão para a falta de progresso inicial no campo da IA e pela qual a realidade recuou bruscamente em relação às expectativas reside na forma específica em que os primeiros sistemas "inteligentes" tentaram replicar a inteligência humana. Ou seja, todos eles eram um conjunto de regras que tentavam reduzir a inteligência humana a um conjunto de operações lógicas. E apesar de este tipo de formalização funcionar bem em determinados casos, como os jogos de xadrez, tem um comportamento bastante pobre no que toca a reconhecimento de imagens ou sons. Para este tipo de tarefas é necessário que o algoritmo consiga interpretar os dados de forma correta. Dado que tais algoritmos não possuíam esses atributos, nasceram então as redes neuronais artificiais em 1969. No entanto, este trabalho ficou estagnado devido à falta de capacidade de processamento dos computadores da altura. Mais tarde estas redes voltaram a surgir na forma de algoritmos de *deep learning*, sendo que em 2015 a Google lança o AlphaGo, uma IA capaz de bater o campeão mundial de Go, um jogo consideravelmente mais complexo que o xadrez. Naquela altura ninguém esperava que um computador conseguisse derrotar o humano num jogo de tamanha complexidade [15].

Presentemente, as redes neuronais artificiais de aprendizagem profunda são a base da maioria das aplicações de reconhecimento facial e reconhecimento de imagem dos gigantes tecnológicos como a Google, Microsoft, Apple, e Meta [15].

2.3 Inteligência artificial forte e fraca

O princípio por detrás da inteligência artificial forte é que as máquinas consigam pensar, ou por outras palavras, que consigam replicar a mente humana no futuro. A inteligência artificial forte acredita que um futuro próximo trará cada vez mais

estas máquinas com capacidades para realizar todas as tarefas que o ser humano consegue, sendo desta forma também elas dotadas de razão [19].

Por outro lado, a inteligência artificial fraca é uma categoria de IA cuja função/tarefa é limitada a uma área em particular. Este tipo de inteligência simula o sentido cognitivo humano. Possui o potencial de beneficiar a sociedade ao automatizar processos que, de outra forma, seriam muito desgastantes ao nível de tempo, e ao analisar os dados em questão de forma que o humano nem sempre consegue [19, 20].

Nesta abordagem existe a falta de consciência humana, apesar de ser capaz de a simular em determinadas ocasiões. Uma ilustração clássica do funcionamento de IA fraca é a experiência da sala chinesa de John Searle's. Esta experiência diz que uma pessoa que esteja fora da sala tem a ilusão que consegue ter uma conversa em chinês com a pessoa na sala, se esta estiver a receber instruções de como responder às conversas em chinês [20].

2.4 Métodos de aprendizagem das redes neuronais

O processo de ensino de uma rede neuronal pode variar bastante de acordo com o efeito pretendido. Atualmente, considera-se que existem três categorias de *machine learning*: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço. A primeira consiste numa metodologia onde o algoritmo sabe previamente quais as entradas e as saídas esperadas [21]. Na segunda, o algoritmo atua sem que exista a intervenção humana. A rede tenta encontrar padrões nos dados que lhe são fornecidos [22]. A terceira técnica decorre também sem intervenção humana, sendo que a rede interage com o meio e vice-versa até se chegar ao resultado pretendido [23].

2.4.1 Aprendizagem supervisionada

A aprendizagem supervisionada utiliza um conjunto de treino para ensinar ao modelo qual a saída desejada. Este conjunto de treino consiste numa base de dados cujos valores de entrada e os respetivos valores de saída já são conhecidos, o que permite ao algoritmo aprender ao longo do tempo. Este processo pode ser observado na Figura 2.2. O algoritmo valida a sua precisão pela função de perda (*loss function*), ajustando este valor de forma a minimizar o erro [21].

Este tipo de aprendizagem pode ser classificada em duas categorias relacionadas com mineração de dados: classificação e regressão. Nos algoritmos de classificação, o algoritmo utilizado pretende classificar com elevada precisão os dados de teste em categorias específicas e bem definidas. Reconhece entidades específicas nos dados e tenta tirar conclusões. Algoritmos comuns para esta categoria são os classificadores lineares, máquinas de suporte de vetores, árvores de

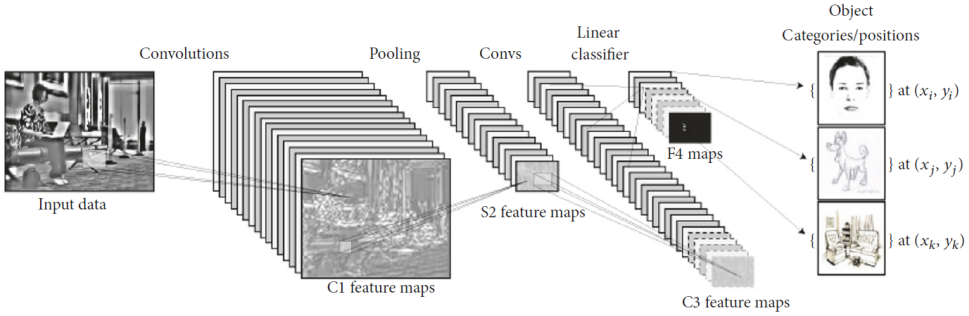


Figura 2.2: Classificação de imagens com uma rede convolucional [24]

decisão, *k-nearest neighbor*. Os algoritmos de regressão são utilizados para perceber quais as relações existentes entre variáveis dependentes e independentes. São tipicamente utilizados para fazer projeções, como retorno de vendas para um determinado negócio [21].

2.4.2 Aprendizagem não supervisionada

Aprendizagem não supervisionada utiliza algoritmos de *machine learning* para analisar e agrupar conjuntos de dados que não se encontrem identificados. Estes algoritmos descobrem padrões escondidos ou grupos de dados sem ser necessário a intervenção humana. A sua capacidade para descobrir semelhanças/diferenças em conjuntos de dados torna-os ideais para análise exploratória de dados, estratégias de venda cruzadas, segmentação de mercados, e reconhecimento de imagem [22, 25].

Algoritmos de aprendizagem não supervisionada são utilizados em três tarefas particulares: *clustering*, associação, e redução de dimensões.

O *clustering* é uma forma de agrupar que estipula que um ponto pode apenas existir num *cluster*. O algoritmo de *K-means clustering* é um exemplo comum deste método, onde cada ponto é atribuído a K grupos, onde K representa o número de grupos com base na distância entre o centroide de cada grupo. Os pontos que estiverem mais próximos de um determinado centroide serão anexados a esse grupo [25].

Uma regra de associação é um método para encontrar a relação entre variáveis de um conjunto de dados. Estes métodos são utilizados com frequência para a análise de produtos no mercado, permitindo às empresas perceber qual a relação entre os seus diferentes produtos [22].

Enquanto mais dados trazem geralmente resultados mais precisos, podem também, por outro lado, impactar o desempenho dos algoritmos de *machine learning* e também tornar a visualização dos dados difícil. A redução de dimen-

sões é uma técnica utilizada quando o número de *features*, ou dimensões, de um conjunto de informações é muito elevado. Reduz o número de entradas para um número que seja viável para gestão, garantindo simultaneamente que não existem perdas a nível da integridade dos elementos [22].

2.4.3 Aprendizagem por reforço

A aprendizagem por reforço estuda como os sistemas naturais e artificiais conseguem prever as consequências e otimizar o seu comportamento em ambientes cujas ações originam estados diferentes. Os animais, desde os mais humildes até aos menos modestos, deparam-se com múltiplas situações deste género de otimização e de uma forma extraordinária são capazes de os resolver eficazmente. Fazem-no ao aprender a selecionar as ações nas quais obterão a recompensa e evitarão a penalização [23].

Conforme os estudos, os ambientes de tomada de decisão são essencialmente caracterizados por alguns conceitos como, estado do espaço (estados são as localizações num labirinto, existência de um determinado estímulo, ou até mesmo as posições num tabuleiro se o caso em questão for um jogo), um conjunto de ações (direções nas quais se mover, premir um botão, direções no tabuleiro) e resultados que sejam importantes (encontrar queijo, obter água, ganhar o jogo). As ações podem levar o agente a diferentes estados no ambiente, e podem produzir resultados [23].

Na aprendizagem por reforço a máquina interage com o ambiente ao produzir ações, sendo que estas afetam o estado do ambiente, que em retorno resultam na máquina receber uma recompensa escalar, ou penalização. Desta forma, o objetivo da máquina é aprender a atuar para maximizar as recompensas futuras. A aprendizagem por reforço está relacionada com o campo da teoria de decisão, e teoria de controlo [25].

2.5 Aplicações da inteligência artificial

Uma vez que a inteligência artificial tenta replicar o pensamento e lógica humana, as aplicações desta tecnologia no mundo moderno são elevadas. Podem-se observar estes sistemas na área da saúde, para a gestão, tanto de utentes, como de profissionais de saúde. A IA aliada à visão artificial tem trazido, também, um grande impacto na forma como se realiza o processamento de imagens médicas e da geração de diagnósticos [9, 26].

No que se refere à era digital, é notória a presença da IA nos dias que decorrem. Todos os gigantes da indústria tecnológica, de uma forma, ou de outra, já desenvolvem equipamentos com IA integrada, que os tornam não só mais apelativos ao público, mas também mais sofisticados do ponto de vista do *software* [27].

A IA é, do mesmo modo, um tópico de interesse no que diz respeito ao *marketing* digital, especialmente na divulgação e gestão de anúncios [28].

2.5.1 Aplicações na medicina

Como já foi mencionado anteriormente, as aplicações da IA na medicina podem ser divididas em duas categorias específicas: aplicações virtuais e físicas. Atualmente, os sistemas pensados para os cuidados de saúde não estão apenas focados nas interações entre utentes e prestadores de cuidados de saúde, mas têm em conta, também, organizações de elevada escala. É também importante que não sejam estacionárias, ou seja, que aprendam com as suas próprias experiências e prosperem ao implementar processos contínuos de aperfeiçoamento. Neste sentido, algoritmos que, de certa forma, englobem IA são propícios a atingir um nível de progresso significativo [9].

Cada vez mais, a IA começa a ganhar terreno nas diferentes vertentes da medicina. Entre outras aplicações no ramo virtual, podem-se considerar os registos médicos eletrónicos, onde com os devidos algoritmos é possível para cada indivíduo navegar no histórico de doenças hereditárias da família, ou até mesmo doenças crónicas, e prever qual o risco de o indivíduo ser portador, ou desenvolver, essa mesma doença [26].

Uma das mais amplas aplicações surge na radiologia, com o progresso feito nos últimos anos no que toca ao processamento e reconhecimento de imagem, ao qual se tem visto um crescimento quer na disponibilidade de dados digitais disponíveis para tarefas desta origem, quer no aumento do poder de processamento computacional. Estes fatores, aliados com o aumento na acessibilidade a exames radiológicos, levam a que se consiga produzir resultados a uma velocidade nunca vista [26].

A inteligência artificial também tem tido notórios avanços na oncologia, onde em certos exames, como ao cancro da mama, a IA provou conseguir ter um desempenho superior nas leituras face ao Homem. O mesmo acontece também em exames ao cancro do pulmão. Sendo que neste caso, num estudo de Yu *et al.* [29], foi provada a precisão da IA, e os resultados demonstram que a IA consegue prever com elevada precisão o prognóstico, e dessa forma também decidir qual o tratamento a seguir para beneficiar a saúde do paciente [26].

A segunda forma de aplicação da IA na medicina inclui objetos físicos, dispositivos médicos e robôs cada vez mais sofisticados no que toca à distribuição de cuidados. Talvez uma das mais promissoras abordagens seja o uso de robôs como ajudantes, por exemplo, um robô de companhia para pessoas de idade mais elevada, com capacidades cognitivas em decadência ou então com mobilidade limitada. Os robôs de cuidados de saúde japoneses são provavelmente a forma mais avançada desta tecnologia que se conhece [9].

Uma das outras formas físicas da utilização de IA reside na cirurgia assistida. Com a tecnologia existente o cirurgião tem acesso a uma visão aumentada, com o recurso a câmaras 3D, e imagens infravermelho. Dispõe também de um conjunto de capacidades mecânicas sobre-humanas, como articulação intuitiva dos instrumentos, eliminação dos tremores e ainda a escalabilidade do movimento [9].

No entanto, a cirurgia assistida por IA ainda é muito recente, levando a que não exista uma produção de valor tão alta face aos outros exemplos apresentados. Nota-se, no entanto, que quanto mais cirurgias esta executa, e maior variabilidade anatómica encontra, melhor a compreensão da complexidade anatómica se torna, e devido a isso as cirurgias passam a ser executadas de forma mais rápida e segura. Espera-se que a IA na cirurgia continue a fazer progressos e consiga aumentar a segurança, reduzir a intervenção humana, permitir a reprodutibilidade e, até mesmo, a automação de alguns eventos [9].

2.5.2 Aplicações no mundo digital

Indubitavelmente, a inteligência artificial é um ramo que se encontra em rápida expansão, sendo uma das áreas de investigação mais saliente dos últimos anos, como é possível verificar na Figura 2.3. É um campo que possui vantagens imensas e tem sido aplicada a múltiplas áreas da indústria com grande sucesso. Nestas inclui-se a classificação de imagens, reconhecimento da fala, carros autónomos e visão artificial. Por outro lado, o rápido desenvolvimento da ciência de dados, incluindo sensores mais robustos e sofisticados, a "internet" das coisas (IoT), computação de última geração e a capacidade de analisar vastas quantidades de dados, fornecem uma boa base para os processos e desenvolvimentos de *machine learning* [30].

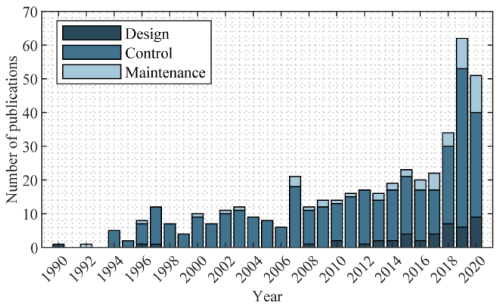


Figura 2.3: Evolução das número de publicações de artigos relacionados com IA [30].

Na era digital, algumas das aplicações mais conhecidas estão presentes nos assistentes de voz pessoais, como a SIRI da Apple, ou a ALEXA da Amazon [27]. A geração de linguagem natural é utilizada para gerar fala a partir de dados num computador com IA, especialmente aplicando redes de aprendizagem profunda.

As redes de aprendizagem profunda são das mais populares zonas de estudo do campo da IA. Ainda relacionado com o tema anterior, existe o reconhecimento de voz, que utiliza geralmente os modelos de Hidden Markov. Nos anos recentes, redes neuronais profundas têm ganho bastante destaque, uma vez resolverem problemas de reconhecimento. O objetivo do reconhecimento de voz é o oposto do exemplo apresentado anteriormente, ou seja, traduzir linguagem humana e converter para um formato cujo computador consiga interpretar [27].

Algo que também tem sofrido bastantes avanços recentemente é a realidade virtual, ou a realidade aumentada. A realidade virtual é uma simulação 3D de um ambiente gerado automaticamente, com o qual é possível interagir de forma real [27].

Por um lado, um dos temas mais controversos relaciona a inteligência artificial com as redes sociais, e o *marketing* digital. Ao contrário do que se pensa, as redes sociais há muito que se afastaram da simples ideia de serem um lugar onde as pessoas podem interagir e se conectar. As mais diversas empresas utilizam este meio como uma ferramenta de comércio *online*, serviços para o cliente, *marketing*, relações-públicas e muito mais. E como a IA atua no meio disto tudo? Tudo acontece com um processo de três fases. Existem ferramentas alimentadas por inteligência artificial que conseguem traçar um perfil de uma determinada pessoa com base nos conteúdos das suas páginas pessoais, isto permite detetar consumidores, perceber qual a audiência de um produto e fazer um estudo do mercado. Seguidamente, esta informação é utilizada por outro conjunto de ferramentas que irão tratar de criar anúncios personalizados para cada indivíduo. Num passo seguinte é possível analisar o que a concorrência faz para vender aquele produto e proceder a um ajuste para se conseguir destaque no mercado, e assim ser a selecionada pelo consumidor final [28].

Uma das outras áreas em que a inteligência artificial já não é novidade é a indústria dos videojogos. De facto, nos dias que se vivem, a IA é uma característica de um bom videojogo. Os primeiros jogos que surgiram não incluíam inteligência artificial, dado que eram bastante simples e estavam desenhados para serem jogados por dois jogadores. Só em 1970, com o lançamento de "Computer Space" é que foi pela primeira vez introduzido o conceito. Desde aí, os desenvolvedores tentaram criar IA mais complexas com o intuito de incentivar os jogadores a gastarem mais moedas nas máquinas de arcada [31]. Presentemente, o consumidor quando compra um videojogo não só espera uma qualidade gráfica que simule a realidade, como também interações entre elementos e personagens não jogáveis que se assemelhem à realidade.

2.6 Conclusão do capítulo 2

Neste capítulo foram apresentados alguns aspetos relativos à inteligência artificial, quer no que toca ao seu funcionamento, à sua evolução, algoritmos existentes e aplicações práticas que já se encontram em ação. De um ponto de vista futurista, a IA ainda tem muito por onde se desenvolver. Apesar de ser uma ferramenta extremamente poderosa e com um leque bastante amplo de aplicações, ainda não é a superinteligência com intuito de conquistar o mundo, como se descreve em alguns filmes de ficção cientifica. De facto, a IA ainda apresenta muitas limitações como a mono execução de problemas, ou seja, considerando todos os eventos que ocorrem no mundo real e o tempo e dados necessários para treinar, a IA é tipicamente limitada apenas a uma tarefa, por exemplo, determinar qual a melhor jogada num jogo de xadrez [27].

No que toca a reconhecimento de padrões, a IA é excelente a extrair um exato e determinado padrão. No entanto, no que toca a problemas que não utilizem valores numéricos, como os de associação, estes algoritmos já não conseguem funcionar eficazmente. A simbologia é também um problema complexo para a IA. Enquanto o humano consegue associar o símbolo de "cavalo + riscas = zebra", a IA não é portadora de conhecimento para fazer esta ligação de ideias [27].

Concluindo, é possível ver que ainda existem vários problemas por resolver com inteligência artificial e que assim ainda existe muito espaço para melhorar este conceito.

Capítulo 3

Visão Artificial com Aprendizagem Automática

O conceito de visão artificial tem atraído cada vez mais indivíduos, quer do mundo académico, quer do mundo industrial. No contexto de tecnologias de assistência, a literatura prova que os algoritmos de visão artificial conseguem efetivamente explorar e satisfazer diferentes necessidades do utilizador, como aponta a Organização Mundial de Saúde (OMS). Prematuramente, é fácil de prever que oportunidades para explorar conceitos mais desafiantes, e também mais recompensadores, dependem da velocidade com que os conhecimentos dos fundamentos por detrás da ferramenta envolvem, isto é o caso do que acontece normalmente quando o conhecimento é transferido do ponto de vista teórico para aplicações práticas no campo. Após uma década onde o desempenho da visão artificial se manteve praticamente inalterada, nos últimos 5 anos tem sido apressado devido ao impacto da aplicação de aprendizagem profunda. Este tema, que antes retinha apenas um grupo restrito de investigadores, tornou-se agora um dos temas de estudo mais convencionais, contribuindo para desbloquear aplicações em diversas áreas como, imagem e vídeo, reconhecimento de voz, imagem médica e veículos autónomos [32]

3.1 Como funciona a visão artificial

A visão artificial é um campo onde a inteligência artificial está presente e que permite aos computadores e sistemas retirar informações significativas a partir de imagens digitais, vídeos, ou outro tipo de entrada visual. Desta forma, é possível ao sistema em questão tomar decisões ou proceder a recomendações baseadas nessa mesma informação. Se a IA permite ao computador "raciocinar" então a

visão artificial permite-lhes observar [33].

A visão artificial funciona equivalentemente à visão humana, com a exceção de que os humanos já se encontram um passo à frente. O olho Humano tem a vantagem de uma vida de contexto para treinar a distinção entre diferentes objetos, as suas partes, a distância a que se encontram, se estão em movimento ou parados, ou até mesmo se existe algo de errado na imagem, ou não [32]. O desenvolvimento das Redes Neuronais Convolucionais (CNN) teve uma tremenda influência no ramo da visão artificial nos últimos anos, e é responsável pelo grande salto na capacidade de reconhecer objetos. Este progresso tem sido possível graças também ao avanço no poder computacional, e também no que toca ao aumento da quantidade de dados disponíveis para se treinar as redes neuronais [33, 34].

As CNN usam *kernels*, também conhecidos como filtros, para detetar características ao longo de uma imagem. Um *kernel* não é nada mais que uma matriz de valores, chamados pesos, treinados para detetar características específicas. Como o seu nome indica, a ideia principal por detrás de uma CNN é convolver espacialmente o *kernel* de uma determinada imagem de entrada, e verificar se a característica que se quer detetar se encontra presente. Para fornecer um valor representativo da confiança de que essa característica se encontra presente, é executada uma convolução ao computar o produto escalar do *kernel* e a área de entrada onde o *kernel* se encontra sobreposto (a área da imagem original para a qual o *kernel* olha) [33, 34].

Para facilitar a aprendizagem dos pesos do *kernel*, as saídas das camadas convolucionais são somadas com um termo de tendência e seguidamente fornecidas a uma função de ativação não linear. As funções de ativação são normalmente não lineares, como a "Sigmoid", "TanH" e a "ReLU". Dependendo da natureza dos dados e da tarefa de classificação, estas funções de ativação são escolhidas de acordo. O resultado destas operações resulta em melhores resultados para a tarefa em prática, menos suscetibilidade a problemas de gradiente de fuga, e produzem representações bastante mais eficientes [34, 24].

Para acelerar o processo de treino e reduzir a quantidade de memória consumida pela rede, a camada convolucional é normalmente seguida de uma camada de *pooling* para remover a redundância presente nas características de entrada. Por exemplo, o *pooling* máximo move a janela por cima da entrada e simplesmente envia para a saída o valor máximo nessa janela, efetivamente reduzindo o número de pixeis importantes na imagem. Como se pode constatar na Figura 3.1, CNN profundas podem conter vários pares de camadas convolucionais e camadas de *pooling*. Por fim, uma camada completamente conectada, comprime o volume da camada anterior em vetores de características e depois uma camada de saída que calcula os resultados (confiança e probabilidade) para as classes de saída através de uma rede densa. Este resultado é depois passado para uma função de

regressão como a *Softmax*, que mapeia tudo para um único vetor, cujos elementos podem-se somar até 1 [34, 24].

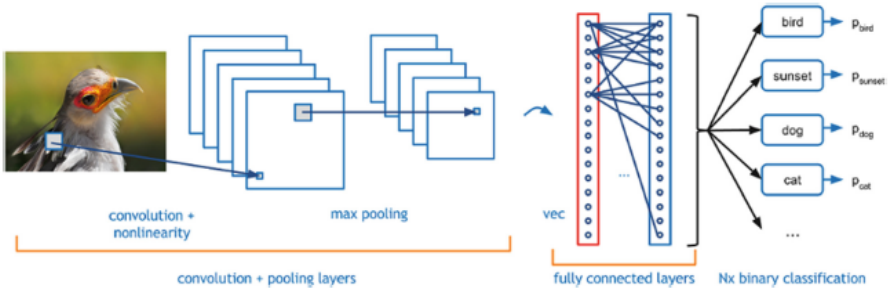


Figura 3.1: Blocos de construção de uma CNN [34].

3.2 Redes para segmentação de imagem (U-Net)

Como tem sido mencionado ao longo deste capítulo, as redes convolucionais profundas conseguem um desempenho superior às redes superficiais em muitas tarefas que envolvem o processamento de imagens. Apesar destas redes já existirem há algum tempo, o seu sucesso sempre esteve muito limitado pela pela necessidade de existência de vasto conjunto de treino disponível para proceder ao treino do algoritmo. Existem modelos de redes já desenvolvidos (pré-treinados) que permitem uma mais fácil utilização devido à capacidade de abstração do modelo matemático inerente à mesma. Um destes exemplos, é a rede de segmentação U-Net. Nota-se ainda que, este tipo de redes beneficia muito de um processo de manipulação de dados chamado *data augmentation*, que permite disponibilizar ao algoritmo variações de dados e assim utilizar as amostras existentes de forma mais eficiente [35, 36].

A ideia principal do modelo U-Net é complementar uma rede de contração comum com camadas sucessivas, onde operadores de *pooling* são substituídos por operadores de *upsampling*. Desta forma, estas camadas aumentam a resolução da saída. Um aspeto importante desta arquitetura reside no grande número de canais associados aos operadores de *upsampling* que permitem à rede propagar informação de contexto para camadas de resolução mais elevada [37].

A arquitetura da rede encontra-se ilustrada na Figura 3.2. De uma forma geral divide-se em duas operações, contração (lado esquerdo) e expansão (lado direito). O curso de contração segue a arquitetura típica de uma rede convolucional. Consiste na aplicação repetida de duas convoluções 3x3, cada uma seguida de uma ReLU e uma operação 2x2 de *pooling* máximo com dois passos para *downsampling*. A cada passo de *downsampling* o número de canais é duplicado. Todos os passos no curso de expansão consistem no *upsampling* do mapa de ca-

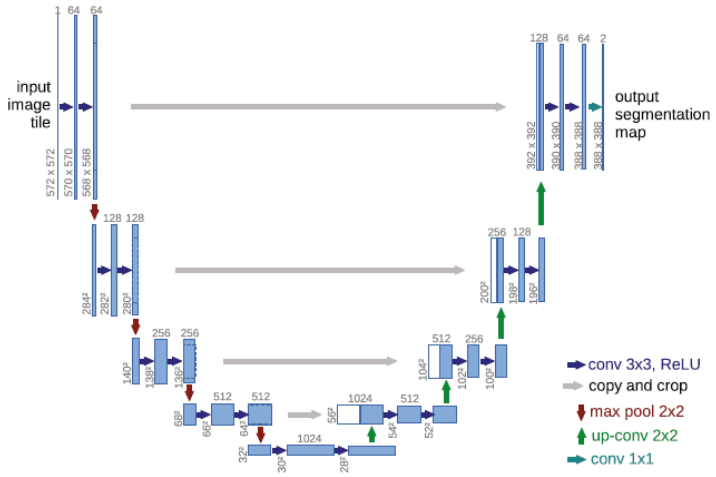


Figura 3.2: Arquitetura da U-Net. Cada retângulo azul corresponde a um multicanal de mapeamento de características. O número de canais encontra-se no topo de cada retângulo. As dimensões x e y estão presentes no canto inferior esquerdo. As setas indicam diferentes tipos de operações [35].

racterísticas cortadas, seguido de uma convolução 2x2 que reduz o número de canais para metade, uma concatenação com o mapa correspondente da rota de contração, e duas convoluções 3x3, cada uma seguida por uma ReLu. O corte é necessário devido à perda de pixeis nas bordas em cada convolução. Na camada final é utilizada uma convolução 1x1 para mapear cada vetor de 64 componentes para o número de classes desejado. No total, a rede tem 23 camadas convolutivas [35].

Para permitir uma colocação do mapa de segmentação de saída sem descontinuidades, é importante selecionar o tamanho da matriz de entrada de modo que todos os procedimentos de colocação 2x2 *max-pooling* sejam aplicados a uma camada com um tamanho uniforme de x e y [35].

3.3 Tarefas com o recurso a visão artificial

A visão artificial é um tópico recorrente em muitas tarefas comuns do dia a dia. O ser humano encontra-se constantemente exposto a este tipo de algoritmos nas mais diversas aplicações. O reconhecimento é uma das tarefas base onde a visão artificial está presente. A análise e interpretação de padrões são algumas das bases que permitem a existência dos famosos algoritmos de reconhecimento facial, ou de reconhecimento de íris [38]. A análise de movimento é outro dos temas que, geralmente, capta o interesse dos investigadores, devido ao leque de aplicações que esta tecnologia abre, desde a área da segurança, da realidade virtual, e da saúde

ao desporto [39]. Seguidamente são apresentadas algumas tarefas com recurso a visão artificial de maior destaque devido à sua utilidade.

3.3.1 Reconhecimento

O processamento de imagem digital ou outro tipo de processamento de imagem evoluíram de forma a gerar curiosidade e a atrair a atenção dos investigadores. Na maioria dos casos, os indivíduos procuram conseguir a integração destas técnicas noutros campos de estudo. Suportada pelo rápido desenvolvimento de outras disciplinas como matemática, álgebra linear, estatística, computação e neurociências, a análise de imagem começa a ganhar imenso reconhecimento por parte da comunidade científica[38].

O reconhecimento de padrões é um ramo da visão artificial que se foca na identificação de objetos através da transformação de imagens para obter melhor qualidade de imagem e melhor interpretação. Este processo extrai informações para tomar decisões baseadas nas imagens obtidas por sensores. As *frameworks* mais comuns utilizadas em visão artificial envolvem aquisição de imagem, pré-processamento, extração de características, deteção e segmentação, processamento de alto nível, e tomada de decisão. Estas *frameworks* consistem em dois grupos principais: análise 3D morfológica e otimização de pixeis. A análise 3D é a teoria base para processamento de imagem e deteção de padrões, onde a otimização de pixeis está relacionada com a caracterização morfológica dos pixeis, incluindo análise estrutural e componentes internos para uma melhor compreensão das funções de vetores [38].

3.3.2 Análise de movimento

A deteção de ações humanas pelo computador está a ganhar cada vez mais interesse. Uma parte significativa desta tarefa passa por registar movimentos, um processo denominado por "captura de movimento humano". Apesar de este termo cobrir muitos aspetos, é principalmente referenciado no que toca a captura de movimentos de grande dimensão, como os movimentos da cabeça, tronco, e membros [40, 41, 39]. A análise do movimento humano tem um enorme interesse para a investigação, dadas as suas inúmeras aplicações, como desempenho físico, avaliação, diagnóstico médico, e realidade virtual. Ou seja, de certa forma pode-se classificar estas aplicações em três grupos distintos: vigilância, controlo e análise. A área da vigilância cobre as aplicações onde os sujeitos se encontram a ser monitorizados temporalmente, e possivelmente monitorizados para ações específicas. Um exemplo típico onde isto acontece é nos parques de estacionamento, onde o sistema rastreia o movimento dos indivíduos para validar se estes poderão ou não estar a assaltar um carro. O controlo passa por utilizar os movimentos capturados para funcionalidades relacionadas com o controlo. Pode ser utilizado como uma

interface para jogos, ambientes virtuais, ou animação. O terceiro grupo, análise, encaixa-se no uso dos dados obtidos para proceder a diagnósticos, estando muitas das vezes relacionado com a área médica. É utilizado em diagnósticos ortopédicos ou em atletas de alta competição para perceber de que forma é possível melhorar o seu desempenho [41, 40].

Atualmente, os métodos utilizados para a deteção de objetos em movimento são o método de subtração por *frame*, o método de subtração de fundo, e o método de fluxo ótico. No método de subtração de *frame* a presença de objetos em movimento é detetada ao calcular a diferença entre duas imagens consecutivas. O método de subtração de fundo utiliza a diferença entre o fundo da imagem atual e o fundo da imagem em movimento para detetar o objeto que se encontra em movimento, com um simples algoritmo, contudo é bastante sensível a mudanças externas do ambiente. O método de fluxo ótico consiste em calcular o campo de fluxo ótico da imagem, e fazer um processamento de grupos conforme as distribuições de características do fluxo ótico da imagem.[40].

3.4 Aplicações

Nesta secção descrevem-se quais as aplicações que a visão artificial possui na área da saúde. Esta tecnologia encontra-se bem presente em todo o tipo de tarefas que envolvam imagem médica, ou métodos de diagnóstico cujas saídas resultem em imagens que possam ser posteriormente interpretadas por estes algoritmos. Os estudos revelaram que em muitos casos, os algoritmos preditivos de visão artificial e IA conseguiram obter desempenhos superiores aos dos próprios médicos [42].

Estas técnicas fornecem também um futuro interessante para o ensino de jovens cirurgiões com o recurso a ambientes de realidade virtual (RV). A RV irá permitir aos estudantes realizar processos complexos que outrora não seriam possíveis sem a supervisão de um entidade sénior [43].

3.4.1 Análise de imagem médica

A inteligência artificial tem transformado a entrega de cuidados e serviços no sistema de saúde. A revolução digital na medicina aponta para um crescimento exponencial de dados interconectados com muitos outros elementos de outras fontes digitais, como a genómica, imagem médica e registos de saúde eletrónicos [44].

Este tema relaciona-se com a melhoria e desenvolvimento de novas técnicas dos métodos de análise para imagens médicas. Primeiramente, a integração de informação multimodal, proveniente de diferentes diagnósticos e de diferentes técnicas de imagem, é essencial para ter uma caracterização compreensiva da

região que se encontra sob examinação. A extração de características é um dos processos-chave para a análise de imagem médica [44].

3.4.1.1 Visão artificial para a deteção de nódulos pulmonares

O rastreio do cancro do pulmão através de tomografia axial computorizada (TAC) de baixa dose demonstrou reduzir a mortalidade para 20% no National Lung Screening Trial (NLST) do Instituto Nacional do Cancro. Está atualmente incluído em diretrizes de rastreio norte-americanas. Os resultados falso-positivos conduzem a procedimentos invasivos, e verifica-se que existe um número elevado destes casos, mas a IA está a ser usada para melhorar a precisão do diagnóstico. Os estudos comparam o desempenho entre radiologistas especializados e algoritmos de aprendizagem profunda utilizando imagens do tórax. Ardila *et al* [45] propuseram uma abordagem de ponto a ponto para a deteção do cancro do pulmão utilizando apenas os dados de entrada de TAC do NLST. O modelo tridimensional da CNN foi criado a partir de análise de volumes inteiros de TAC. Os dados de treino incluíam volumes com a patologia confirmada. Consequentemente, a CNN foi treinada para identificar regiões de interesse e desenvolver um modelo preditivo de risco de cancro. Nos testes, este modelo alcançou uma área sob a curva da característica de funcionamento do recetor (ROC) de 94,4% já o grupo de radiologistas profissionais obteve um desempenho equivalente ou abaixo da do algoritmo [42].

3.4.1.2 Visão artificial e marcas moleculares para o diagnóstico de fibrose pulmonar

Apesar da declaração em 2018 da Fleisher Society em expandir as recomendações de diagnóstico para a evolução radiográfica da doença fibrótica pulmonar, o diagnóstico continua a ser um tema desafiante devido à variabilidade interobservador, mesmo entre radiologistas experientes. Os algoritmos de aprendizagem profunda propostos por alguns investigadores classificam a doença pulmonar utilizando imagens de tomografia axial computorizada de alta resolução (TACAR). Num estudo de Walsh *et al* [46], foram utilizados algoritmos de aprendizagem profunda para a automação da classificação da fibrose pulmonar, em função de imagens de TACAR. O estudo utilizou uma base de dados de 1157 imagens de TACAR de pacientes anónimos com fibrose pulmonar difusa. O desempenho do algoritmo foi comparado com a maioria dos votos de 91 radiologistas torácicos. A precisão mediana dos radiologistas foi de cerca de 70,7% face aos 73,3% do algoritmo [42].

3.4.1.3 Modelos preditivos na hipertensão pulmonar

Os modelos de *machine learning* apuram a precisão da previsão, face aos modelos de regressão convencionais. Na lesão vascular relacionada com a hipertensão pulmonar, por exemplo, os papéis da inflamação e das citosinas não são compreendidos na sua totalidade. Num estudo, algoritmos de *machine learning* não supervisionados foram utilizados para classificar pacientes com hipertensão arterial pulmonar (HAP) do grupo I da Organização Mundial de Saúde em grupos imunitários proteómicos (citocinas, quimiocinas e fatores de crescimento). As características clínicas e os resultados foram subsequentemente comparados entre os grupos. Foram identificados quatro grupos de HAP com perfis proteómicos distintos. Estes grupos imunitários tinham diferenças significativas nos parâmetros de risco clínico e prognóstico a longo prazo. Em comparação com outros aglomerados, os doentes do aglomerado 1 tinham conjuntos únicos de proteínas não preparadas (ligante indutor de apoptose relacionado com o factor de necrose tumoral, factor inibidor da migração dos microfagos, ligante de quimiocina 4, 5 e 7) e tinham as taxas de sobrevivência sem transplantação menos favorável em 5 anos (30,8%, 95% CI 17,3-54,8%) [47].

3.4.1.4 IA na pandemia de SARS-COV-2

O novo coronavírus de 2019 (COVID-19) cresceu de forma descontrolada desde a sua descoberta como um surto de gripe em Wuhan, China, em janeiro de 2020. O rápido reconhecimento do surto foi, em parte, devido a um algoritmo de IA epidemiológico chamado BlueDot, o qual recolhia informações de noticiários por todo o mundo e anúncios oficiais para disponibilizar aos clientes acerca de novidades sobre potenciais epidemias. O algoritmo BlueDot previu a dispersão do COVID-19 para fora de Wuhan, com base nos dados de viagens gerados pela Associação Internacional de Transportes Aéreos. Este algoritmo já teria sido sucedido anteriormente ao prever a difusão internacional do vírus Zika, no sul da Florida, em 2016 [42].

Para além dos modelos de rastreamento de doenças infeciosas, a IA também teve grande importância na elaboração de diagnósticos e estratégias de tratamento. Por exemplo, os algoritmos de aprendizagem profunda contribuíram imenso no que toca ao diagnóstico da doença através de análises a imagens de radiografias torácicas [42].

3.4.2 Realidade virtual e realidade aumentada na medicina

Os avanços científicos em tecnologias como a realidade virtual e na realidade aumentada (RA) têm conseguido rápidos progressos na última década. Esta tecnologia tem-se tornado acessivelmente práticas devido à dimensão de aplicações nos domínios da indústria, particularmente no setor da saúde, para aprimorar

as práticas clínicas atuais. O uso de tecnologias, RV introduz muitas vantagens na educação de cirurgiões acerca de diferentes intervenções cirúrgicas de forma a conseguir um planeamento pré-operativo, e navegação intraoperativa para a cirurgia. A RV pode ser utilizada em diversas formas de educação, como a execução de procedimentos em órgãos humanos simulados [43].

Um exemplo encontra-se presente na forma como cirurgiões e engenheiros podem colaborar no desenho e modelação 3D de ambientes que preservam a anatomia e mantenham as condições para procedimentos cirúrgicos num mundo virtual. Outro exemplo é com a simulação através de RV, onde médicos podem realizar procedimentos cirúrgicos passo a passo [43].

Para a cirurgia laparoscópica, existe um *software* de simulação com RV que oferece formação para diferentes procedimentos através de módulos, por exemplo, para costura, colecistectomia, hérnia ventral e *bypass* gástrico. Estes módulos técnicos podem ser utilizados para formar os cirurgiões acerca da utilização de conhecimentos básicos para métodos minimamente invasivos, navegação com câmara, manuseamento de instrumentos, manipulação de objetos, recorte e corte [43].

3.5 Conclusão do capítulo 3

Ao longo deste capítulo foi dado a conhecer um pouco acerca da visão artificial, do seu modo de funcionamento, da sua evolução, das tarefas nas quais esta tecnologia é aplicada, e, por fim, em aplicações reais, focadas particularmente na saúde. A área da visão artificial tem um impacto particularmente extenso devido aos desenvolvimentos no campo da inteligência artificial e dos algoritmos de aprendizagem profunda. Por um lado, os algoritmos de visão artificial funcionam na base de arquiteturas semelhantes aos de aprendizagem profunda, contudo, muitas das arquiteturas modernas apresentam resoluções com um nível de complexidade superior de forma a permitir treinar as redes de visão de forma mais eficaz, ou seja, obter resultados de alta precisão com menos dados do que normalmente são necessários para a tarefa em questão.

Grande parte das tarefas de visão artificial passam por dois princípios básicos, reconhecimento de objetos, ou análise de movimento. Sendo que, cada um destes tópicos pode ser dissecado em subtópicos mais específicos, como reconhecimento de expressões humanas, análise de movimento para estudos biomecânicos, entre outros. A aplicação da visão artificial na medicina tem contribuído de forma muito positiva para a área. Esta tecnologia não só permite recriar cenários realistas para ensinar jovens médicos (simulações de realidade virtual), como também permite analisar com mais atenção ao detalhe as imagens provenientes de determinados exames médicos.

Capítulo 4

Materiais e métodos

Os modelos modernos de machine learning, como os de aprendizagem profunda, normalmente, contam com uma vasta gama de parâmetros, que lhes permite facilmente generalizar o problema em causa quando treinados com enormes quantidades de dados etiquetados [48]. Contudo, nem sempre é o caso em que o conjunto de dados disponíveis para treino apresenta a quantidade ideal. Felizmente, existem certos modelos de redes neuronais que já têm este impasse em conta e conseguem operar eficazmente, mesmo em conjuntos de dados de pequenas dimensões, recorrendo ao aumento de dados. Neste capítulo será apresentada a arquitetura global de todos os algoritmos desenvolvidos, desde o processamento dos dados de alimentação da rede, à arquitetura do modelo, aos algoritmos de treino e, por fim, os algoritmos de teste.

4.1 Pipeline geral de execução

A *pipeline* correspondente ao funcionamento de todo o sistema pode ser visualizada na Figura 4.1. De uma forma geral, a mesma não se afasta muito daquilo em que consiste uma *pipeline* genérica para operação e execução de algoritmos de *machine learning*. Possuindo uma secção para processamento dos dados iniciais, na qual a partir do conjunto de dados iniciais se realiza um processo designado por "aumento de dados"que gera um conjunto novo de dados através de manipulações controladas dos originais. Com o conjunto de dados processado, utilizou-se 70% para treino e 30% para avaliação da rede. Através do conjunto de avaliação consegue-se extrair métricas que são depois utilizadas num processo de *feedback* contínuo para o modelo de treino. Por fim, o modelo que apresentar resultados mais promissores é *deployed* para o ambiente de execução, ao qual o utilizador final tem acesso. O código utilizado para gerar toda a *pipeline* e as suas múltiplas

funções foi validado através da criação de testes unitários, e pode ser consultado no repositório de GitHub [49]. As instruções para montar o ambiente de execução podem ser obtidas a partir do ficheiro README, também ele presente no repositório.

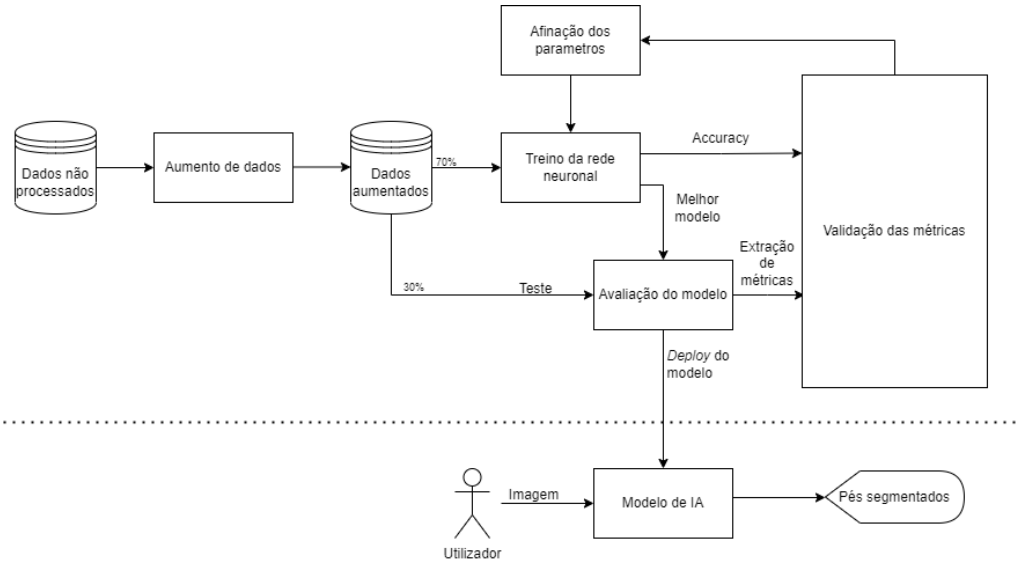


Figura 4.1: *Pipeline* geral da arquitetura do modelo de inteligência artificial para a segmentação dos pés.

4.2 Conjunto de dados

Nesta secção serão apresentados os processos utilizados para se obter o grupo de dados necessário para alimentar o modelo de treino da rede U-Net. Tendo como referência a *pipeline* apresentada anteriormente, as subsecções seguintes abordam a fase de pré-processamento dos dados originais.

4.2.1 Descrição do conjunto de dados

Como se tem mencionado ao longo deste documento, uma das características vitais para o treino e sucesso de uma rede neuronal reside na quantidade de dados disponíveis. Neste trabalho foram utilizadas 66 imagens de pés, das quais 46 foram alocadas para o processo de treino da rede, e as restantes 20 para a avaliação da mesma. As imagens adquiridas consistem em fotos tiradas num ambiente de luminosidade não controlada, onde o indivíduo em questão se encontrava numa posição de decúbito dorsal (deitado de barriga para cima), tendo as pernas inseridas em dois orifícios de uma tela preta, de forma a ter como destaque a porção plantar do pé. As condições anteriormente referidas podem ser observadas na Fi-

gura 4.2. Todas as imagens encontram-se no formato *Joint Photographic Experts Group* (JPEG).

Figura 4.2: Exemplo de imagem do conjunto de dados.

4.2.2 Criação das máscaras binárias

Uma das partes cruciais para se conseguir treinar uma rede neuronal consiste em fornecer ao algoritmo de treino, não só as entradas corretas, neste caso, serão as imagens do conjunto de dados, mas também quais as saídas esperadas, ou seja, qual o resultado esperado para a segmentação de cada entrada.

O ponto anterior traduz-se na criação de uma máscara de segmentação para cada uma das 66 imagens que se tem no conjunto de dados inicial. Estas máscaras foram criadas utilizando o GNU *Image Manipulation Program* (GIMP), um *software* de código aberto compatível com múltiplos sistemas operativos, como Linux, Windows e MacOS, que permite a edição de imagens de alta qualidade, bem como a manipulação gráfica das mesmas [50]. Com recurso a esta ferramenta foram marcados, de acordo com a literatura [51, 52], os respetivos pontos que se pretendiam segmentar sobre a imagem base dos pés. Seguidamente a imagem guia foi removida de forma a se obter apenas os pontos marcados num fundo transparente. O resultado desta manipulação pode ser constatado na Figura 4.3. Todas as máscaras foram gravadas no formato *Graphics Interchange Format* (GIF), uma vez ser o melhor formato para imagens com transparência e limitado número de cores. De acordo com estes fatores, considera-se um formato que não perde qualidade com a compressão [53].

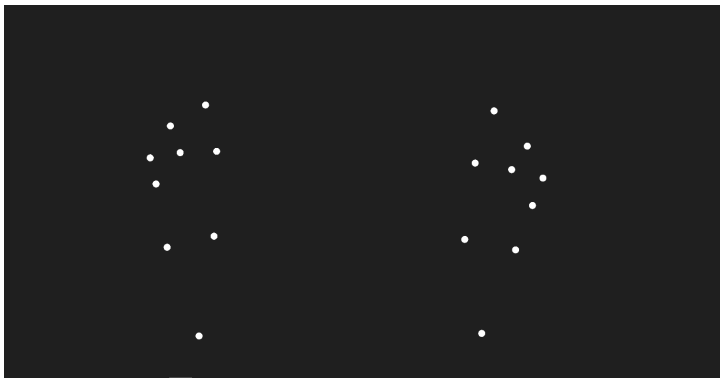


Figura 4.3: Exemplo de máscara para uma imagem do conjunto de dados.

4.2.3 Criação das máscaras RGB

Equitativamente à subsecção anterior, um dos procedimentos também realizados foi a criação de máscaras de segmentação, mas, desta vez, marcando cada ponto com uma cor diferente. Por conseguinte, cada ponto marcado no pé terá uma cor única, funcionando como uma espécie de "id" para cada ponto. Este tipo de segmentação permite analisar não só se os pontos são corretamente identificados, mas também, se a sua posição não se encontra trocada com a de outro ponto. Esta segmentação pode ser observada na Figura 4.4. Estas máscaras correspondem também a um formato, como aquele mencionado na subsecção anterior.

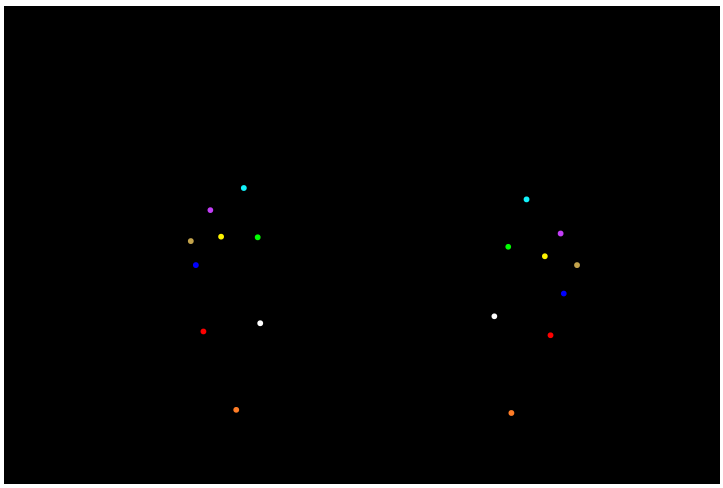


Figura 4.4: Exemplo de máscara RGB para uma imagem do conjunto de dados.

4.2.4 Aumento de dados

Apesar de 66 imagens parecer um número mais que razoável para o ser humano aprender um conceito simples, como o de segmentar pontos num pé humano, a

realidade é que os algoritmos de inteligência artificial necessitam de um número bastante mais elevado de dados para conseguirem atingir o objetivo pretendido. Para conseguir estender o conjunto de dados inicial é necessário proceder ao aumento de dados (*data augmentation*).

O aumento de dados recorre a alterações de entrada que mantêm as etiquetas de saída correspondentes e é um método amplamente utilizado para expandir o número e a diversidade de conjuntos de treino. O aumento de dados é frequentemente utilizado para aumentar o desempenho na visão por computador e surgiu como uma estratégia popular de regularização implícita para contrariar a sobreposição de modelos de aprendizagem profunda. Apesar do facto da maioria das estruturas de aprendizagem profunda apoiarem simples transformações de imagem, a lista inclui frequentemente apenas algumas iterações de inverter, rodar, escalar, e cortar [48].

Ao longo deste trabalho, para cada imagem e a sua respetiva máscara foram aplicadas cinco transformações para obter uma maior diversidade face ao conjunto de dados inicial. As operações aplicadas foram as seguintes: inversão horizontal, inversão vertical, rotação, saturação, distorção de rede e distorção ótica, que podem ser observadas na Figura 4.5. A inversão horizontal consiste numa inversão da imagem de entrada em torno do eixo y, na inversão vertical ocorre uma inversão da imagem de entrada em relação ao eixo x, a rotação, tal como o nome indica, consiste numa rotação da imagem de entrada até um ângulo de 45° (para este trabalho), na operação de saturação os valores de Hue e de saturação são alterados de forma aleatória, durante a distorção ótica promove-se um desvio da projeção retilínea normal, já a distorção de rede consiste no mapeamento de famílias de curvas diferentes, arranjadas no sistema de rede [48].

Através destas técnicas de aumento de dados, foi possível gerar 276 imagens e máscaras de treino a partir das 46 imagens iniciais.

4.3 Modelo de dados

Esta secção passa por explicar quais as escolhas que foram efetuadas aquando da implementação do modelo de processamento de dados. Na iminente necessidade de efetuar processamento de baixo nível num conjunto de dados, torna-se imperativo a escolha de uma *framework*, de um modelo e de parâmetros de treino.

4.3.1 Framework computacional

É um enorme empreendimento criar uma rede neural de aprendizagem profunda e tê-la pronta para enfrentar os desafios da era moderna. De forma a utilizar a aprendizagem profunda para alcançar os objetivos desejados, é necessário ligar e coordenar demasiados elementos de uma forma metódica. As empresas precisam

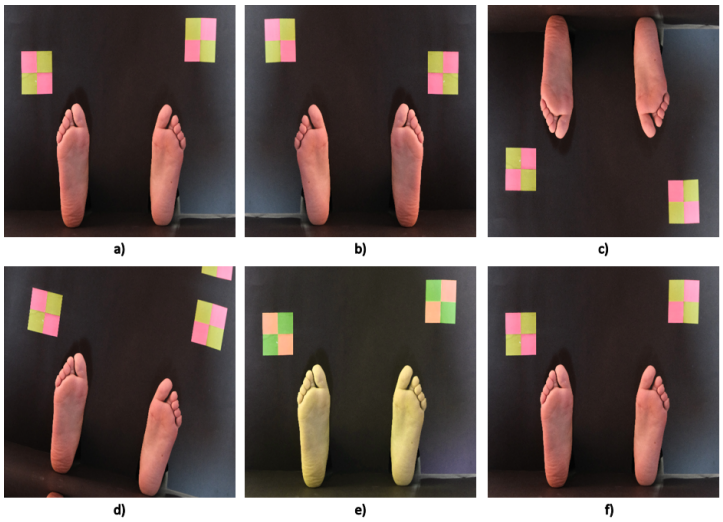


Figura 4.5: Manipulações das imagens de entrada. a) imagem original; b) inversão horizontal; c) inversão vertical; d) rotação; e) saturação Hue; f) distorção ótica e de rede

de um nível de abstração elevado para conseguirem lidar com o trabalho de campo intensivo, a fim de oferecer soluções mais simples, mais rápidas e melhores para a investigação. Isto permitiria que os investigadores e os promotores passassem mais tempo em projetos importantes e menos tempo a fazer atividades de rotina [54].

As "interfaces" e *frameworks* de aprendizagem profunda oferecem uma abstração razoável relativamente à complexidade subjacente das tarefas básicas de *machine learning*. Os investigadores e programadores podem utilizar estas ferramentas para enfrentar questões mais desafiantes. O Keras, PyTorch, TensorFlow, MXNet, Caffe, Microsoft's CNTK, entre outras opções, são alguns exemplos [55, 54].

Uma biblioteca de *machine learning* e aprendizagem profunda chamada PyTorch foi desenvolvida pela META, Inc. Tal como o nome sugere, é baseada em Python, e procura fornecer uma forma de solucionar os problemas de desempenho do NumPy, oferecendo uma biblioteca baseada em tensores que tem sido afinada para a tarefa específica de aprendizagem profunda com o recurso a Unidades de Processamento Gráfica (GPU). Esta biblioteca oferece operações matemáticas em tensores multidimensionais e fornece estruturas de dados para os mesmos. Conta ainda com um vasto leque de ferramentas de serialização eficazes e tensores arbitrários [55].

O PyTorch foi a ferramenta escolhida para o trabalho por um conjunto de simples razões. Oferece uma estrutura que é simples de usar, expandir, criar, e depurar. É simples para a comunidade se adaptar, uma vez estar escrito em

Python. Tanto os investigadores como os programadores podem concluir o trabalho com facilidade. A sua base assenta em C++, promovendo um elevado desempenho em cenários reais, sem que os programadores tenham que se preocupar com a interferência do Python. Para além destes pontos, o PyTorch é a biblioteca de aprendizagem profunda mais utilizada na área da investigação [54].

4.3.2 Arquitetura

O modelo adotado para o problema em questão foi o de uma U-Net. Para se programar o modelo em questão, seguiu-se a arquitetura de rede proposta por Ronnenberger *et al.* [35], a qual pode ser consultada na Figura 3.2.

Programaticamente, esta proposta consiste em quatro classes: *conv_block, encoder_block, decoder_block, build_unet*, sendo que três destas são invocadas pela classe principal, a *build_unet*. Todas as classes mencionadas estendem a uma classe pai, a *torch.nn* que possui os principais blocos de construção das redes neuronais da biblioteca PyTorch. Cada uma destas ainda conta com dois métodos comuns, o decorador *__init__*, que permite modificar o método de inicialização da classe pai, e o método *forward*, que efetua a implementação específica daquela classe.

Cada uma destas classes representa uma zona particular do modelo a implementar. A classe *conv_block*, tal como o nome indica, foi desenvolvida com o intuito de tratar das operações de convolução presentes no modelo. Utiliza uma convolução de duas dimensões, seguido de uma normalização de lote, e de uma ativação linear corrigida. Esta operação é repetida duas vezes por cada bloco do caminho contráctil do modelo, como se pode observar a laranja na Figura 4.6.

A classe *encoder_block* corresponde a duas operações de convolução e a uma de *pooling* máximo. Programaticamente, recorre à classe descrita anteriormente, *conv_block*, para as operações de convolução e aplica ainda mais uma operação de *pooling* máximo através da função *nn.MaxPool2d*. Este conjunto de manipulações representa um bloco de codificação e pode ser observado a vermelho na Figura 4.6.

A classe *decoder_block* executa as seguintes operações: uma convolução de duas dimensões transposta, uma concatenação dos tensores e duas convoluções de duas dimensões. Estas operações conseguem-se com a *nn.ConvTranspose2d*, para a convolução transposta, com a *torch.cat*, para a concatenação e, por fim, com a classe *conv_block* para as convoluções. Este processo corresponde a um bloco de descodificação, que pode ser observado a verde na Figura 4.6.

Finalmente, a classe *build_unet* invoca quatro vezes a classe *encoder_block* para construir o caminho contráctil da rede, utilizando as saídas da operação anterior como entradas da operação seguinte. A classe *conv_block* é invocada de modo a construir a zona de transição entre o caminho contráctil e o caminho expansível

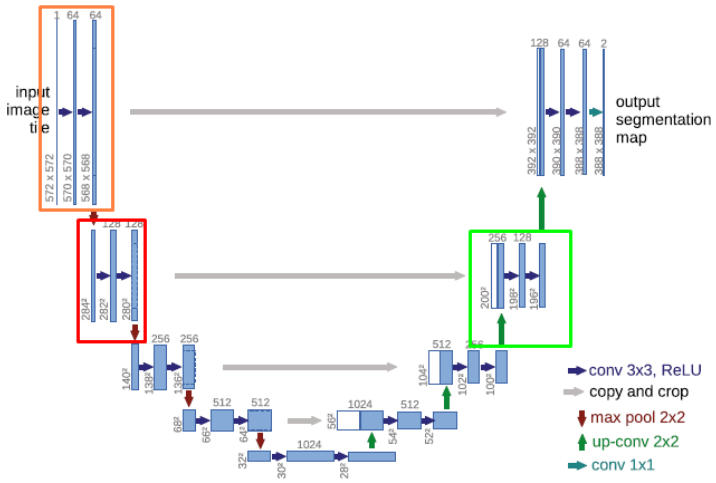


Figura 4.6: Arquitetura U-Net com as etiquetas correspondentes às classes no código. A zona vermelha corresponde às operações efetuadas pela classe *encoder_block*, a verde às operações da *decoder_block*, e a laranja às operações da *conv_block*.

da rede. Seguidamente, a classe *decoder_block* é invocada quatro vezes de modo a gerar o caminho expansível e, por fim, é efetuada uma última convolução com um filtro de um por um.

4.4 Treino da rede

Para o treino da rede foi desenvolvido um código de treino, *train.py*, que juntamente com módulos auxiliares procede ao treino do modelo com os dados previamente organizados.

Um destes módulos desenvolvido com o intuito de auxiliar o treino foi o *data.py*. Dentro deste existe uma classe *DriveDataset* que estende à classe pai *Dataset* do sub-módulo *torch.utils.data* do PyTorch. Esta classe tem como objetivo fazer a manipulação dos vetores de imagem *numpy* de forma a chegar às propriedades pretendidas e seguidamente os transformar em tensores.

Após o conjunto de dados ser transformado em tensores, é possível agora aplicar uma função iterativa de dados, como a *DataLoader* disponibilizada pelo PyTorch. Esta função, na sua raiz, representa uma iteração típica de Python sobre um conjunto de dados, mas oferecendo a possibilidade de utilizar um maior número de parâmetros, como tamanho de lote, número de *threads*, tipo de dados e muitos mais.

No passo seguinte invoca-se o modelo previamente desenhado na secção anterior e procede-se à indicação de qual o *hardware* a partir do qual se pretende

executar o processamento para o treino da rede. Para o trabalho em causa, dado que o computador onde a rede irá ser treinada possui uma placa gráfica com suporte de núcleos de *Compute Unified Device Architecture* (CUDA), foi selecionado como *hardware* de processamento principal a GPU.

Como função de otimização, mais uma vez, foi utilizada uma disponibilizada pelo pacote de *utils* do PyTorch. O Adam é um algoritmo para otimização de funções objetivas e estocásticas de primeira ordem, baseando-se em estimativas adaptativas de momentos de ordem inferior. O método é simples de implementar, a nível de processamento computacional é bastante eficiente, os requisitos de memória são baixos, e é invariável ao redimensionamento diagonal dos gradientes [56].

As funções de perda definem como os modelos de redes neuronais calculam o erro na sua globalidade a partir dos resíduos das mesmas a partir de cada lote de treino. No que lhe concerne, esta ação caracteriza a forma como os pesos internos da rede são ajustados na propagação posterior, assim a escolha da função de perda tem influência direta no desempenho global do modelo [57].

A escolha padrão para tarefas de segmentação ou outro tipo de classificação é a entropia binária cruzada (BCE). Em situações em que uma métrica particular, como o coeficiente de dados ou interseção sobre a união, são utilizadas para avaliar o desempenho do modelo, é comum fazer experiências com funções de perda que derivem destas métricas [57, 58].

A função de perda selecionada para o modelo apresentado foi a de perda de dados BCE. Esta função mistura a função de perda de dados, com a de BCE, a qual é a função padrão para o tipo de segmentações em questão. A junção destes dois métodos permite alguma diversidade na função de perda, ao mesmo tempo que beneficia da estabilidade da BCE. A equação 4.1 representa a equação para o cálculo da função Dice-Sørensen *coefficient* (DSC), a função de perda de dados [58].

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{4.1}$$

Onde X representa as entradas e Y representa as saídas. O coeficiente de dados é uma medida de sobreposição de duas máscaras, sendo que o valor 1 representa uma sobreposição perfeita e 0 a não sobreposição. Já a equação 4.2 representa a equação de BCE combinada com a de DSC.

$$J(w) = -\frac{1}{N}\sum_{n=1}^{N} H[y_n log\hat{y}_n + (1 - y_n)log(1 - \hat{y}_n)] \tag{4.2}$$

Nesta equação, $j(w)$ é a função de perda de dados BCE, o N a dimensão do lote, o $\hat{y}$ representa o valor previsto pelo algoritmo e o y representa o valor esperado.

Através das funções descritas anteriormente é possível obter qual o valor de perda de validação por cada *epoch*. Desta forma, é possível perceber se este valor melhorou ou piorou face à *epoch* anterior. No caso de o valor ser melhor que o anterior, então procede-se ao arquivo do estado do modelo atual. Caso contrário, o treino prossegue para a próxima iteração de treino sem guardar o modelo.

No final do processo de treino, de modo a ter um registo de como foi o desempenho do algoritmo ao longo das múltiplas iterações, é escrito para um ficheiro de texto todos os *logs* respetivos à evolução dos valores de perda de validação. É também efetuado um gráfico com os valores de perda de validação e os valores de perda de treino.

4.5 Localização dos pontos segmentados pelo algoritmo

Após o algoritmo treinado ser corrido sobre o conjunto de dados de teste, irá ser possível consultar a imagem original com a respetiva máscara prevista pelo modelo. Seguidamente será então necessário proceder à localização no plano destes mesmos pontos de forma a se conseguir efetuar outro tipo de operações em função da posição destes. Este objetivo pode ser facilmente alcançado através de funções como a *SimpleBlobDetector* da biblioteca "OpenCV". Este tipo de funções permite identificar conjuntos de pixeis com propriedades iguais numa imagem. Aplicando diretamente esta função, alguns pontos na imagem não serão assinalados, de forma que é necessário alterar os valores de limite, cor e área para se conseguir uma deteção completa de todos os pontos.

Uma vez sucedido, este método retorna um vetor de objetos cujos atributos permitem identificar a posição no eixo do x e y, bem com a dimensão do ponto em questão.

4.6 Conclusão do capítulo 4

Ao longo deste capítulo foram descritas quais as técnicas, algoritmos e estratégias utilizadas de forma a se conseguir montar o algoritmo para a criação do modelo de treino e, seguidamente, para o treino do mesmo. De igual importância reside o processo de aumento do conjunto de dados de forma a obter um conjunto maior e mais diversificado. Contudo, todo o processo só se encontra concluído após ser possível proceder à validação do modelo. Dessa forma, existe o conjunto de dados de teste, bem como os algoritmos de teste, que testam o modelo treinado com estes dados específicos, e tratam ainda de aplicar mais um conjunto de

operações sobre os resultados obtidos para as previsões, de forma a extrair métricas representativas do desempenho do modelo. No caso do trabalho em questão, para além das métricas padrão utilizadas em problemas típicos de segmentação, ainda foram criadas mais uma série de métricas personalizadas para a validação do problema em concreto.

Capítulo 5

Análise de Resultados

Ao longo deste capítulo serão expostos os resultados obtidos para diferentes iterações e parametrizações do modelo desenvolvido. Durante o processo de treino de uma rede neuronal, um dos passos importantes reside na otimização e configuração do modelo em si, e dos seus parâmetros. Neste capítulo serão demonstrados todos os resultados obtidos, juntamente com o processo de reflexão seguido face a esses mesmos resultados e às soluções encontradas para lidar com eles. De igual importância tem-se também a análise de problemas resultantes da conversão das coordenadas de imagem para coordenadas reais. Nota-se que, todos os modelos que serão mencionados foram treinados no mesmo computador, no qual, os componentes principais são a seguir apresentados: um processador AMD Ryzen 5 3600, 16 GB de memória RAM e uma GPU NVIDIA GeForce GTX 1660 Ti com 6 GB de memória GDDR6.

5.1 Avaliação de desempenho

Ao longo do processo de avaliação serão analisados alguns valores relacionados com métricas típicas para redes neuronais, como a Accuracy, Precision, pontuação de Jaccard, pontuação de F1 e o Recall, uma vez que são os valores mais explorados e comuns em problemas de segmentação e como tal permitem um termo de comparação com valores padrão. Serão também analisados um conjunto de métricas específicas ao projeto, como o número de pontos detetados em determinadas regiões do pé, que serão utilizados para validar quais as áreas do pé que algoritmo consegue segmentar com mais facilidade.

Com o modelo gerado e treinado, é necessário agora proceder à validação do mesmo. Os algoritmos de teste têm como objetivo avaliar o desempenho do modelo ao aplicar o mesmo ao conjunto de dados de teste e analisar quais os

resultados que o algoritmo gera face aqueles que seriam esperados.

Os *scripts* de teste executam uma iteração sobre as imagens fornecidas para esse mesmo fim e, utilizando o modelo treinado, que o utilizador indica no início da execução, é efetuada uma previsão de qual a máscara esperada para essa mesma imagem. Seguidamente o algoritmo gera uma nova imagem que consiste numa matriz de 1x3, onde o primeiro elemento consiste na imagem original, o segundo na máscara original, e o terceiro na máscara que o modelo gerou. Desta forma o utilizador é facilmente capaz de perceber quais as discrepâncias entre o resultado esperado e gerado. Num outro *script* é ainda gerada uma segunda imagem que consiste numa matriz 1x2, onde o primeiro elemento corresponde à imagem original, com a máscara original sobreposta, e o segundo elemento corresponde à imagem original com a máscara prevista pelo modelo sobreposta. Desta forma torna-se ainda mais simples de observar qual o resultado ideal.

Para obter uma validação ainda mais concreta, procedeu-se ao cálculo de diversas métricas de forma a avaliar diferentes componentes do resultado gerado pelo modelo. Este tipo de métricas foi dividido em duas categorias: métricas padrão e métricas personalizadas para o problema em questão. As métricas padrão correspondem a diversas funções provenientes da biblioteca *sklearn.metrics*, tais como a pontuação de Jaccard, a pontuação de F1, o Recall, a Precision e a Accuracy. A pontuação do coeficiente de semelhança de Jaccard é definida como a dimensão da intersecção dividida pela dimensão da união de dois conjuntos de etiquetas e é utilizado para comparar o conjunto de etiquetas previstas com o conjunto de etiquetas pré-definidos. A pontuação de F1 pode ser interpretada como a média harmónica entre a Precision e o Recall. Esta métrica atinge o seu melhor valor para 1, e o seu pior em 0. A contribuição relativa da Precision e do Recall é igual. A fórmula para a pontuação de F1 encontra-se representada na equação 5.1

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{5.1}$$

O Recall consiste no rácio entre o número de verdadeiros positivos, e falsos negativos. Consiste, intuitivamente, na capacidade de o classificador para encontrar todas as amostras positivas. Por outro lado, a Precision é o rácio entre o número de verdadeiros positivos e falsos positivos, sendo que esta métrica representa a capacidade do classificador não rotular como positiva uma amostra que é negativa. Por fim, a Accuracy representa o número de etiquetas que foram corretamente identificadas.

Para as métricas personalizadas procurou-se analisar se todos os pontos nos pés estariam a ser bem identificados. Para isso, decidiu dividir-se a imagem em vários segmentos, e dentro de cada um destes procurar quantos pontos estariam

a ser atribuídos pelo modelo. O esquema de segmentação de imagem pode ser visto na Figura 5.1.

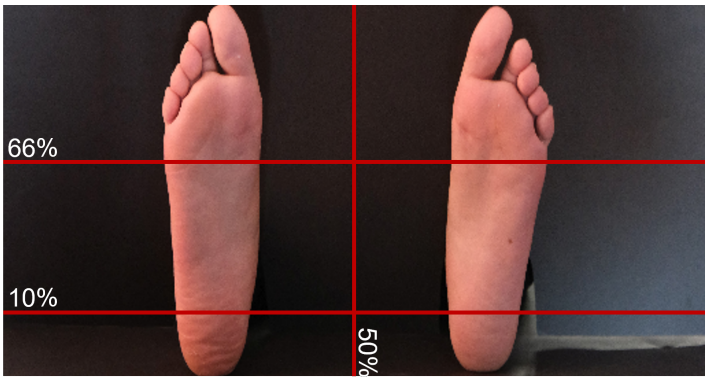


Figura 5.1: Esquema da distribuição de diferentes áreas de interesse do pé, de acordo com as dimensões propostas por Price, Carina *et al* [3].

Tal como observado na Figura 5.1, a imagem original foi dividida em seis regiões de interesse. Estas áreas permitem extrair métricas relativamente ao número total de pontos na imagem, ao número de pontos no pé direito e no pé esquerdo, número de pontos detetados no calcanhar, na zona plantar e, finalmente, na zona do metatarso e falanges.

Todo este conjunto de métricas é calculado com o recurso a uma classe *customMetrics*. Dentro desta classe existem os métodos responsáveis não só por detetar a localização dos pontos na imagem, mas também aqueles que calculam as métricas mencionadas acima.

No final da execução de cada *script* de teste é gerado um relatório. Este relatório contém as imagens com as respetivas previsões de segmentação do algoritmo, bem como os valores das métricas padrão e personalizadas. É possível, ainda, observar os gráficos gerados para algumas das métricas anteriores.

5.2 Modelo de análise de 3 pontos

Para um teste inicial do modelo implementado para a rede de segmentação U-Net aplicou-se uma tarefa de segmentação com um nível de complexidade inferior ao proposto no problema original. Deste modo, o conjunto de dados utilizado manteve-se o mesmo que o enunciado no Capítulo 4, contando igualmente com o processo de aumento de dados, mas com um conjunto de máscaras de treino diferentes. Nestas novas máscaras aponta-se a diferença de, em vez de estarem segmentados 9 pontos ao longo de toda a porção plantar, apenas se encontrarem segmentados 3 pontos pertencentes às extremidades dos dedos. Durante a

análise destes modelos o foco estará presente no numero de *epochs* utilizadas, uma vez que é um dos hiperparâmetros com mais impacto no resultado final do algoritmo. O objetivo será, a partir de um número mais pequeno de iterações de treino, perceber de que forma os fenómenos de *overfitting* e *underfitting* afetam o desempenho da solução e desta forma progredir de forma a diminuir o erro.

5.2.1 10 epochs

Para proceder ao treino da rede é necessário definir quais os hiperparâmetros a utilizar. Este tipo de valores define a forma como o processo de treino, iterações e conjunto de alterações a valores de entrada deverão ser processados. Os hiperparâmetros considerados foram: o tamanho dos lotes, número de *epochs*, número de *workers*, taxa de aprendizagem e dimensões da imagem.

Os valores anteriores foram fixados, à exceção do número de *epochs*. O tamanho dos lotes fixou-se em 2, tal como o número de *workers*, a taxa de aprendizagem em 0,0001 e as dimensões das imagens em 512x512. Relativamente às *epochs* iniciaram-se os testes com 10.

Os resultados obtidos para o processo de treino, ou seja, a evolução dos valores de perda de treino e de perda de validação podem ser observados na Figura 5.2

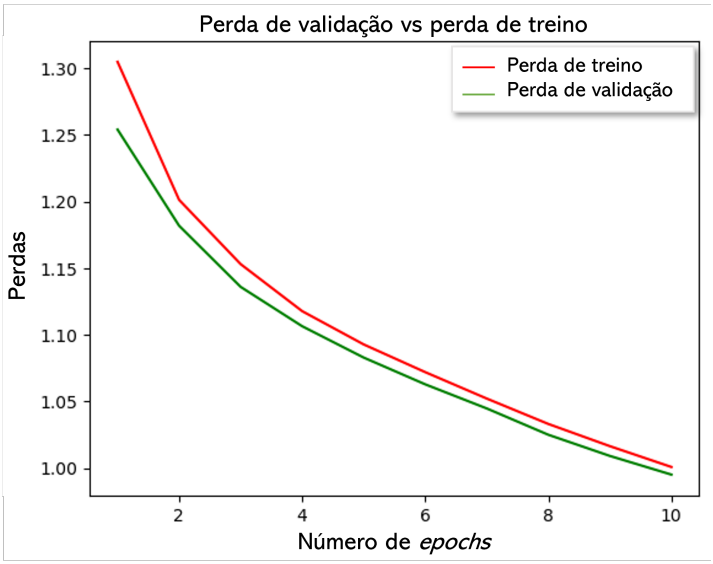


Figura 5.2: Evolução dos valores de perda de validação e perda de treino ao longo de 10 *epochs* com um lote de tamanho dois.

De acordo com o gráfico anterior, é possível verificar que em todas as iterações de treino houve uma melhoria em ambos os parâmetros em questão. Nota-se que o valor de perda de validação indica a capacidade de ajuste do modelo a novos

dados, enquanto o valor de perda de treino indica o quão bem o modelo se ajusta aos dados utilizados para o treinar.

Também é de notar que não existe uma zona onde ambas as curvas se encontrem estabilizadas, tal indicador realça o facto de que o modelo ainda possui uma capacidade de melhoria.

Na Figura 5.3 é possível observar a imagem original, a máscara de segmentação manual, e a máscara de segmentação prevista pelo modelo.

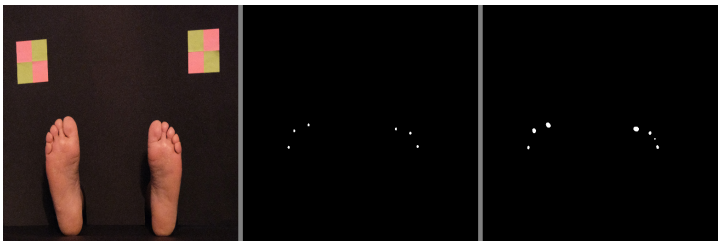


Figura 5.3: Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 10 *epochs*.

É possível observar que, apesar de os 3 pontos em análise terem sido identificados, existe um ponto extra que foi também identificado pelo modelo, que não pertence à máscara original. De forma a melhor entender a que zona do pé corresponde este ponto inesperado, foi executado outro algoritmo de testes que, em vez de gerar imagens das máscaras por si só, gera as máscaras por cima da imagem original. O resultado deste processo pode ser contemplado na Figura 5.4.

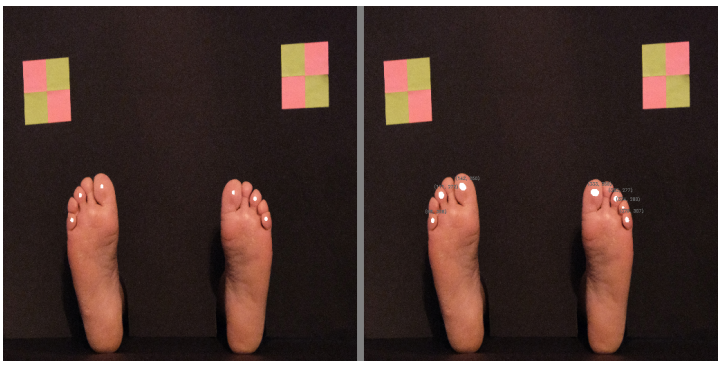


Figura 5.4: Sobreposição da máscara manual e da máscara gerada pelo modelo de 10 iterações, sobre a imagem de origem.

Olhando agora para esta figura, nota-se que o ponto extra, detetado pelo algoritmo, pertence a outro dedo do pé. Desta forma, o modelo identificou um dedo a mais para esta imagem em particular com os hiperparâmetros definidos.

No que toca às métricas obtidas para esta versão do modelo é possível consultar as mesmas nos ficheiros de *metrics.txt* gerados pelos algoritmos de teste.

Relativamente a valores em concreto, para a pontuação de Jaccard obteve-se 0,1894, para a pontuação de F1 0,3178, para o Recall 0,2853, para a Precision 0,3608 e para a Accuracy 0,9976.

As métricas personalizadas desenvolvidas não se aplicam para este modelo preditivo, pelo que não foram geradas.

5.2.2 20 epochs

Para o modelo seguinte os hiperparâmetros foram mantidos iguais aos do modelo anterior, à exceção do número de *epochs* que foi aumentado para 20. Este aumento foi efetuado de forma a tentar perceber se um maior número de iterações de treino produz valores de perda de treino e perda de validação mais baixos que o modelo anterior.

Os resultados para o processo de treino podem ser consultados na Figura 5.5. Nesta, é possível ver que o progresso do treino da rede a partir das 10 *epochs* ainda é notável. O modelo apresenta, aparentemente, um comportamento de estabilização entre as 10 e 13 *epochs*, contudo, a partir desse mesmo ponto, existe uma melhoria nos valores relativos a ambos os parâmetros em análise.

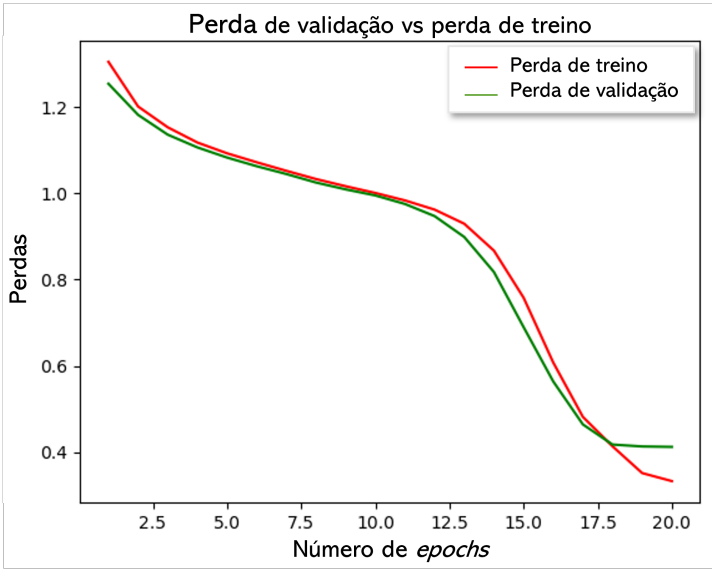


Figura 5.5: Evolução dos valores de perda de validação e perda de treino ao longo de 20 *epochs* com um lote de tamanho dois.

Os resultados gerados pelo novo modelo podem ser observados na Figura 5.6. Na mesma observa-se que, face à máscara gerada pelo modelo anterior, existe agora uma concordância maior entre a máscara obtida de forma manual e a obtida pelo modelo. Acrescenta-se ainda que, de acordo com a Figura 5.7, consegue-se verificar que os 3 pontos detetados pelo algoritmo se encontram localizados na

zona correta do pé, e que nenhum destes pontos foi colocado num dedo errado, ao contrário do sucedido com o modelo anteriormente visto.

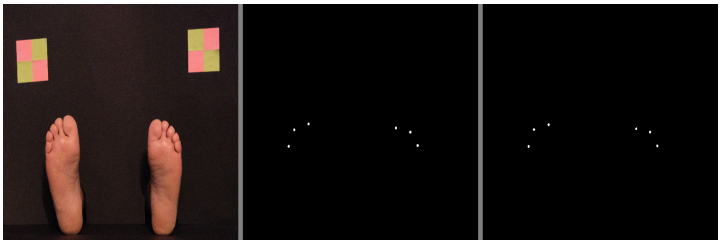


Figura 5.6: Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 20 *epochs*.

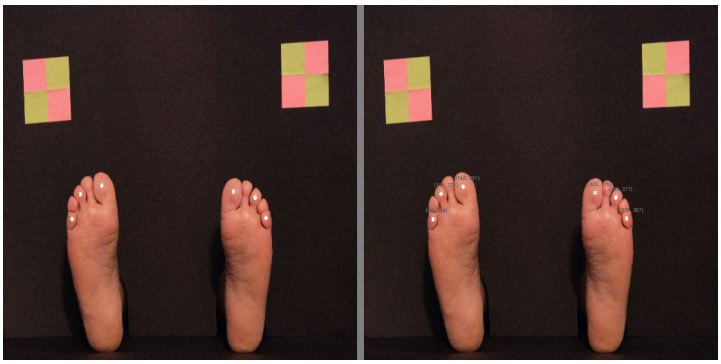


Figura 5.7: Sobreposição da máscara manual e da máscara gerada pelo modelo de 20 iterações, sobre a imagem de origem.

Considerando novamente as métricas extraídas das imagens de validação, foi possível perceber que houve uma melhoria em certos valores, contudo, outros não demonstraram um desenvolvimento tão positivo, sendo que em certos casos ocorreu uma diminuição do valor face ao modelo anterior.

Assim sendo, os valores encontrados foram os seguintes: para a pontuação de Jaccard 0,1700, para a pontuação de F1 0,2898, para o Recall 0,1800, para a Precision 0,7455, e para a Accuracy 0,9982. Pode-se ver que existiu uma ligeira diminuição nos valores de Jaccard, F1 e Recall, mas que houve um aumento dos valores de Precision e Accuracy.

5.2.3 50 epochs

O novo modelo gerado manteve os hiperparâmetros dos dois anteriores, com a exceção do número de *epochs* que desta vez foi aumentado substancialmente para 50. Novamente, os resultados para a sequência de treino podem ser vistos na Figura 5.8.

Do modelo anterior para este, a mudança mais significativa deve-se ao facto de que entre as 20 e 30 *epochs* a curva de perda de validação deixa de acompanhar a curva de perda de treino. Enquanto a curva de perda de treino diminui o seu valor de perda até ao final das 50 iterações, a curva de perda de validação fica fixa no valor de cerca de 0,4 até ao fim do processo.

Geralmente, quando a curva de perda de validação se encontra num valor acima do da curva de perda de treino, é sinal de que poderá estar a ocorrer um fenómeno de *underfitting*. Isto ocorre quando o modelo não consegue modelar com exatidão os dados de treino, o que pode ocorrer devido a vários fatores como: iterações de treino insuficientes ou falta de quantidade e diversidade de dados.

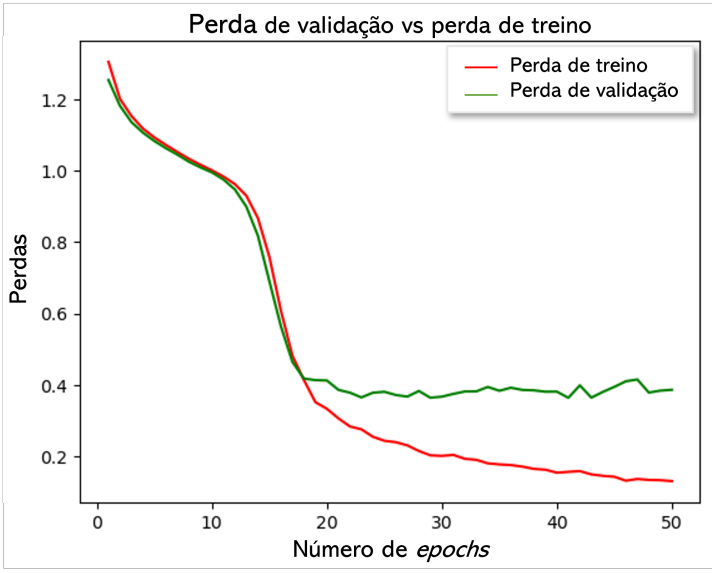


Figura 5.8: Evolução dos valores de perda de validação e perda de treino ao longo de 50 *epochs* com um lote de tamanho dois.

Os resultados relativos à máscara gerada pelo modelo estão presentes na Figura 5.9. Aparentemente não existe nenhuma alteração significativa face ao modelo visto anteriormente. Para se poder ver exatamente onde é que o modelo prevê os pontos na imagem original, mais uma vez, foi gerada a imagem com as máscaras sobrepostas na imagem original, a qual se pode ver na Figura 5.10.

De acordo com a Figura, nota-se que os pontos identificados pertencem às zonas corretas dos pés, o que seria expectável face aos resultados do modelo anterior e de não ter ocorrido *overfitting* com o incremento de iterações.

As métricas obtidas para este modelo mostraram-se mais dececionantes quando comparadas com o modelo anterior de 20 *epochs*. Assim, os resultados para a pon-

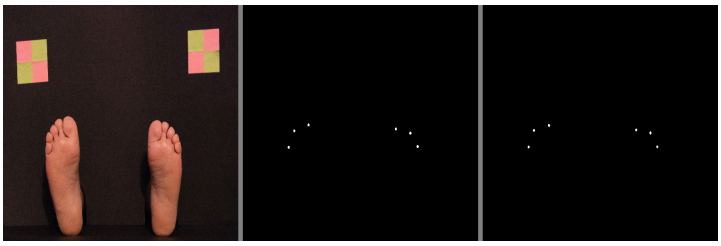


Figura 5.9: Imagem original, máscara de segmentação obtida manualmente, e máscara de segmentação gerada pelo modelo de 50 *epochs*.

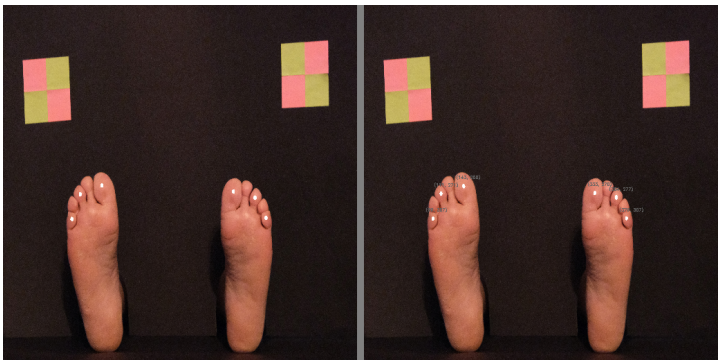


Figura 5.10: Sobreposição da máscara manual e da máscara gerada pelo modelo de 50 iterações, sobre a imagem de origem.

tuação de Jaccard foram de 0,1668, para a pontuação de F1 de 0,2853, para o Recall 0,1783, para a Precision 0,7162, e para a Accuracy de 0,9982.

Neste modelo nitidamente se conclui que os valores de todas as métricas, à exceção do valor de Accuracy, pioraram.

Sumariando a evolução do modelo face às alterações no número de iterações de treino, foi gerado o gráfico de barras que pode ser consultado na Figura 5.11.

5.3 Modelo de análise de 9 pontos

O passo seguinte consistiu em replicar as mesmas condições de treino dos modelos anteriores, mas agora, para o modelo que realmente se pretende obter, ou seja, o modelo de 9 pontos. Neste modelo a localização dos pontos está definida como contendo 3 pontos nas extremidades dos dedos, 3 pontos na zona do metatarso logo abaixo dos pontos situados nas extremidades dos dedos, 2 pontos a meio do pé e, por fim, 1 ponto no calcanhar. Esta nova máscara de segmentação apresenta uma dificuldade acrescida para a previsão por parte do algoritmo, uma vez que para além de conter um maior número de pontos que é necessário localizar, as zonas onde os mesmos se encontram são relativamente ambíguas, dado que não

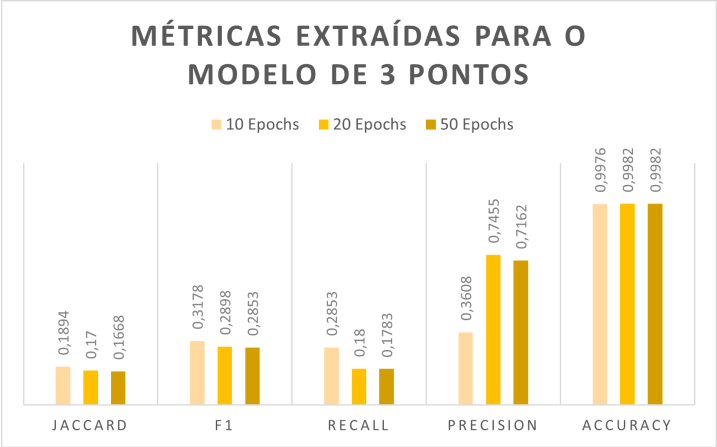


Figura 5.11: Valores de métricas extraídas em cada modelo.

existe uma localização específica onde o ponto deva ser marcado, mas sim uma área na qual é aceitável marcar o ponto. Analogamente ao modelo anterior, a análise será focada no numero de *epochs* utilizadas, uma vez que é um dos hiperparâmetros com mais impacto no resultado final do algoritmo e serve como termo de comparação com os modelos da secção anterior. Também permitirá visualizar se os fenómenos de *overfitting* e *underfitting* encontrados anteriormente se verificam para esta segmentação mais complexa.

5.3.1 10 epochs

Tal como no modelo de apenas 3 pontos, a primeira execução de treino foi efetuada apenas para 10 *epochs*. Os hiperparâmetros restantes (dimensão do lote, número de *workers*, dimensões das imagens e taxa de aprendizagem) foram mantidos com os valores definidos anteriormente. A Figura 5.12 mostra os resultados obtidos para esta primeira execução.

Consegue-se notar que o comportamento das curvas de perda de validação e treino é semelhante ao do modelo com 3 pontos, onde no fim das 10 iterações de treino ainda se verifica que o modelo requer mais treino de forma a obter melhores valores para a perda de validação e perda de treino.

Tal como nos modelos anteriores, a melhor forma de verificar qual o estado do modelo é verificando quais os resultados que o mesmo se encontra a gerar para as imagens de teste. Estes mesmos resultados podem ser consultados na Figura 5.13. Nesta imagem nota-se que todos os pontos da máscara manual foram gerados, também, pela máscara do modelo. Contudo, existem mais 5 pontos gerados sem razão aparente, e que não pertencem ao conjunto de localizações em estudo.

Observando agora a Figura 5.14, percebe-se que, tal como no modelo de 3 pontos e 10 *epochs*, houve novamente a criação de um ponto num dos dedos

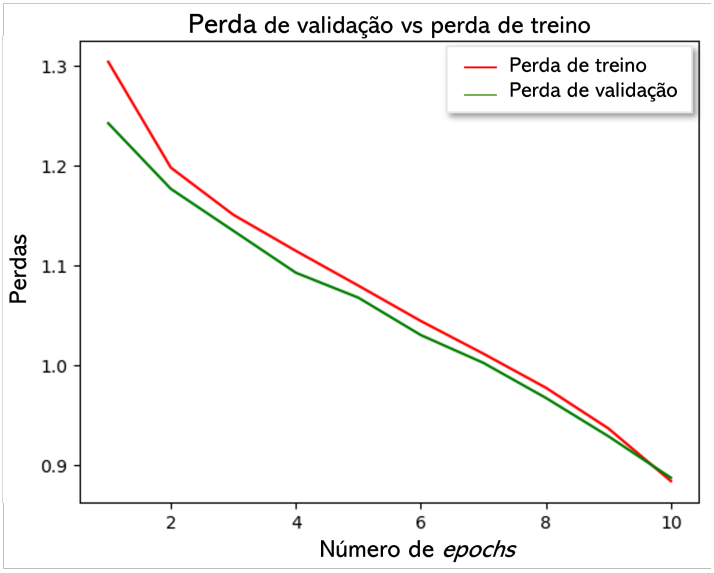


Figura 5.12: Evolução dos valores de perda de validação e perda de treino ao longo de 10 *epochs*, com um lote de tamanho dois, para o modelo de 9 pontos.

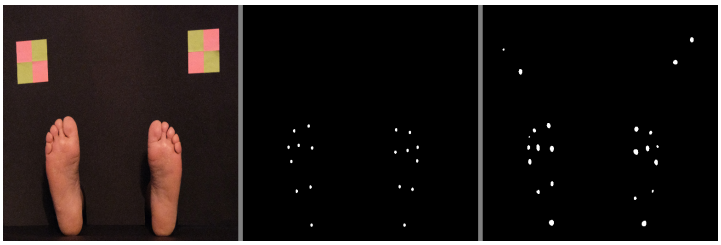


Figura 5.13: Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 10 *epochs* e 9 pontos.

errados, e que os outros 4 restantes pontos a mais não pertencem sequer ao pé e assim sendo, estão completamente errados.

Curiosamente, estes pontos errados foram identificados nos quadrados cuja cor é semelhante à do pé, o que não aconteceu para o modelo de 3 pontos. Apesar de ser uma premissa precipitada, poderá indicar que o conjunto de dados de teste não é suficientemente diversificado para o problema em questão.

As métricas padrão extraídas para este modelo foram as seguintes: a pontuação de Jaccard foi de 0,2507, a pontuação de F1 foi 0,3995, o valor de Recall foi de 0,6277, a Precision foi de 0,2962 e a Accuracy foi 0,9962.

No que toca a este modelo de 9 pontos, foi possível também extrair informação de métricas personalizadas ao problema. A Figura 5.15 representa os valores encontrados ao longo das 20 imagens de avaliação do modelo. Nota-se que a

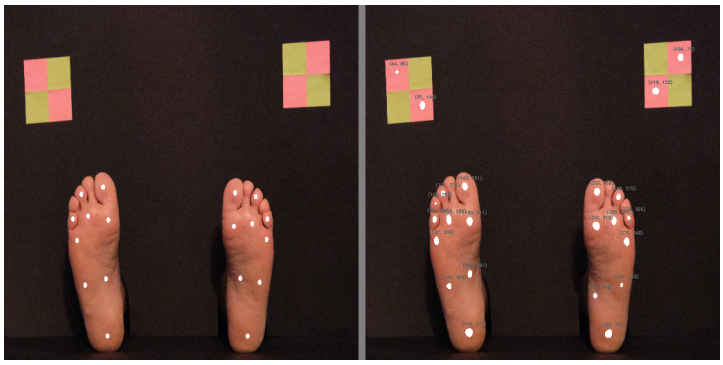


Figura 5.14: Sobreposição da máscara manual, e da máscara gerada pelo modelo de 10 iterações e 9 pontos, sobre a imagem de origem.

dimensão do pé é calculada como a distância entre a coordenada y com menor valor e maior valor. Desta forma, os pontos identificados fora do pé irão causar elevada interferência nas áreas que são posteriormente analisadas pelo algoritmo.

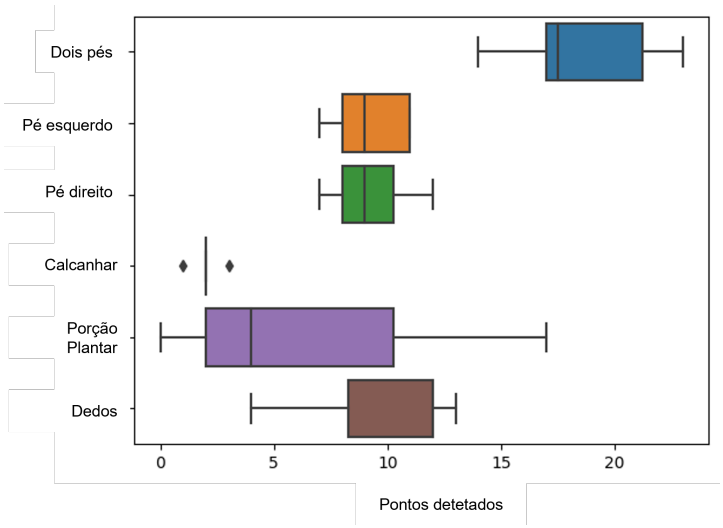


Figura 5.15: Número de pontos detetados para diferentes áreas dos pés ao longo das 20 imagens de avaliação do modelo de 10 iterações.

Na Figura 5.15, ao longo do eixo das ordenadas consegue-se ver as diferentes áreas onde o algoritmo procura identificar quantos pontos foram marcados. As zonas analisadas são: a imagem na totalidade, o pé esquerdo, o pé direito, o calcanhar, a zona plantar e a área dos dedos.

5.3.2 20 epochs

De natureza igual aos exemplos anteriores, procedeu-se ao incremento do número de iterações de treino para 20. Os hiperparâmetros foram novamente mantidos

iguais. Os resultados são visíveis na Figura 5.16, e neste caso a curva de perda de validação e a curva de perda de treino separam-se muito mais cedo do que nos modelos anteriores. Este aspeto significa que o modelo em questão entra num fenómeno de *underfitting* mais cedo que o anterior.

Na Figura 5.17, reconhecem-se os resultados para a máscara gerada para o modelo com 20 iterações de treino. Analisando os mesmos, verifica-se que, ao contrário do modelo anterior, este modelo para a imagem apresentada não gerou pontos não pertencentes ao pé. Contudo, falhou no que toca à identificação de todos os 18 pontos que o algoritmo é suposto localizar.

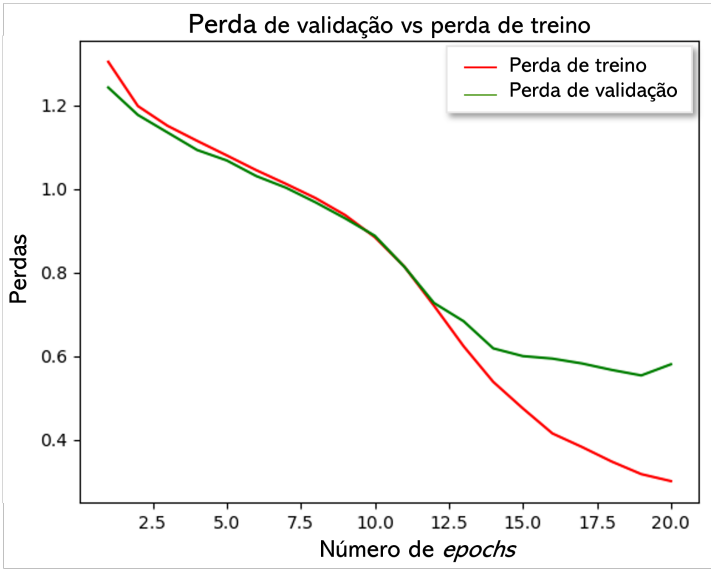


Figura 5.16: Evolução dos valores de perda de validação e perda de treino ao longo de 20 *epochs*, com um lote de tamanho dois, para o modelo de 9 pontos.

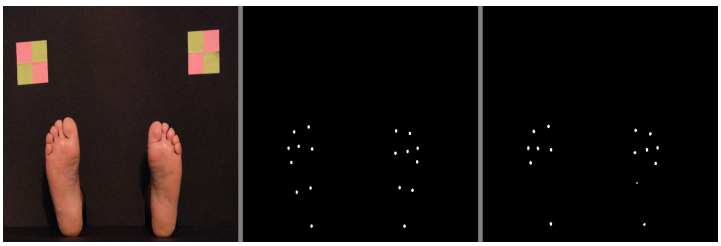


Figura 5.17: Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 20 *epochs* e 9 pontos.

A Figura 5.18 mostra a máscara gerada pelo algoritmo de treino sobre a imagem original, onde se reconhece que os pontos que não foram detetados são aqueles pertencentes à zona plantar do pé.

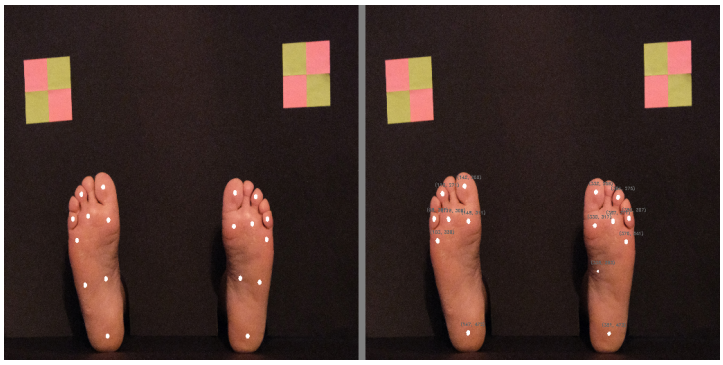


Figura 5.18: Sobreposição da máscara manual e da máscara gerada pelo modelo de 20 iterações e 9 pontos, sobre a imagem de origem.

Os valores para as métricas extraídas para este exemplo foram as seguintes: 0,3178 para a pontuação de Jaccard, 0,4799 para a pontuação de F1, 0,4876 para o valor de Recall, 0,4753 para o valor de Precision e 0,9979 para o valor de Accuracy.

Repara-se que houve um aumento do valor de todas as métricas à exceção do valor de Recall, o que pode dever-se ao modelo anterior estar a prever um maior número de pontos, mesmo esses pontos estando errados.

As métricas personalizadas para o número de pontos encontrados em cada região do pé podem ser visualizadas na Figura 5.19.

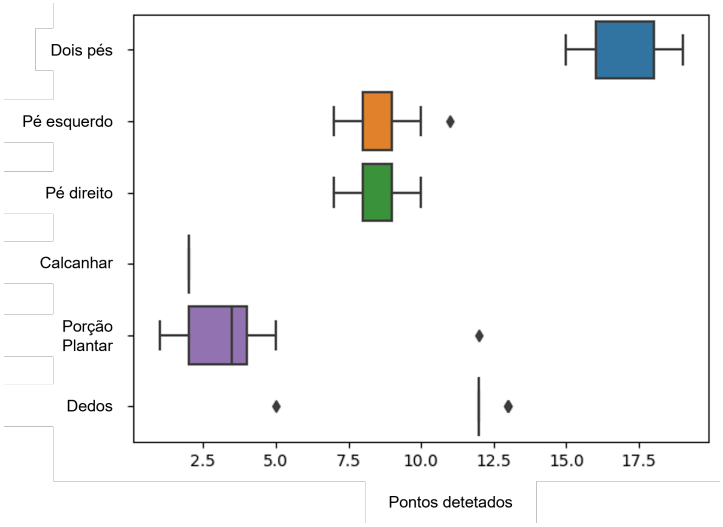


Figura 5.19: Número de pontos detetados para diferentes áreas dos pés ao longo das 20 imagens de avaliação do modelo de 20 iterações.

5.3.3 50 epochs

De forma a manter o padrão utilizado para a versão anterior do modelo, procedeu-se então a um incremento do número de *epochs* para 50, mantendo constantes os restantes hiperparâmetros. Na Figura 5.20 é possível ver os resultados obtidos ao longo destas 50 iterações de treino. O comportamento é semelhante ao modelo de 3 pontos, quando treinado para 50 *epochs*, contudo com uma separação das curvas de perda muito mais cedo.

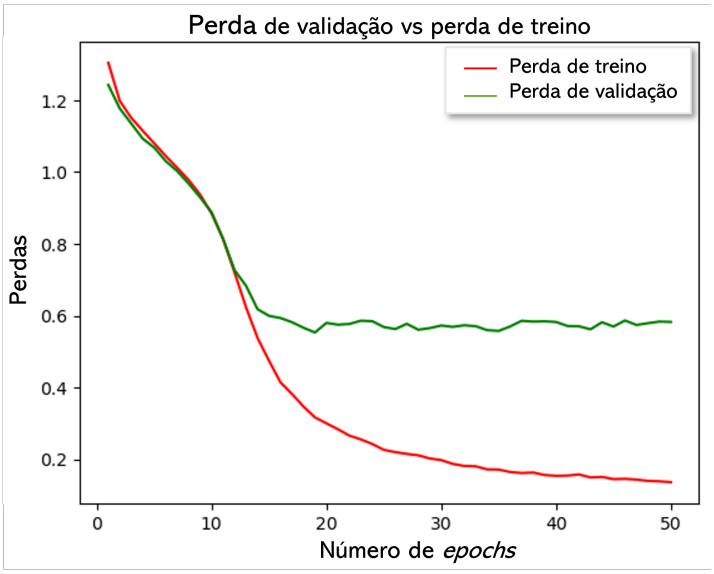


Figura 5.20: Evolução dos valores de perda de validação e perda de treino ao longo de 50 *epochs*, com um lote de tamanho dois, para o modelo de 9 pontos.

As Figuras 5.21 e 5.22 mostram os resultados obtidos a nível de máscaras geradas pelo algoritmo. As mesmas representam bem os resultados obtidos no gráfico anterior, onde apesar de se ter uma diminuição da curva de perda de treino, a curva de perda de validação mantém-se estática. Desta forma, não é possível perceber se houve uma melhoria no comportamento preditivo do algoritmo para este exemplo.

As métricas encontradas para o modelo apontam para aquilo que já se foi a entender ao longo deste exemplo. Desta forma, os valores das métricas correspondem exatamente aos mesmo valores enunciados para o modelo de 9 pontos e 20 iterações. O que permite concluir que não existe nenhum tipo de melhoria ao treinar o conjunto de dados em questão ao longo de 50 iterações. De forma meramente representativa, a Figura 5.23 mostra os resultados para as métricas personalizadas, que vão de encontro aos valores obtidos também para o modelo anterior.

O gráfico da Figura 5.24 expõe comparativamente os valores obtidos para as

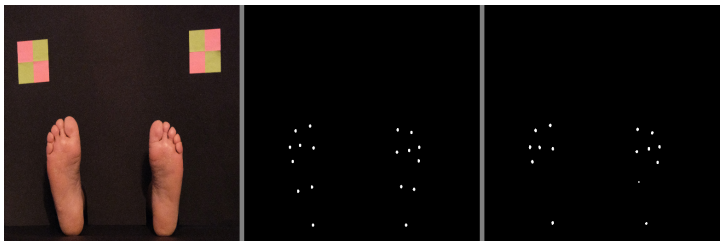


Figura 5.21: Imagem original, máscara de segmentação obtida manualmente e máscara de segmentação gerada pelo modelo de 50 *epochs* e 9 pontos.

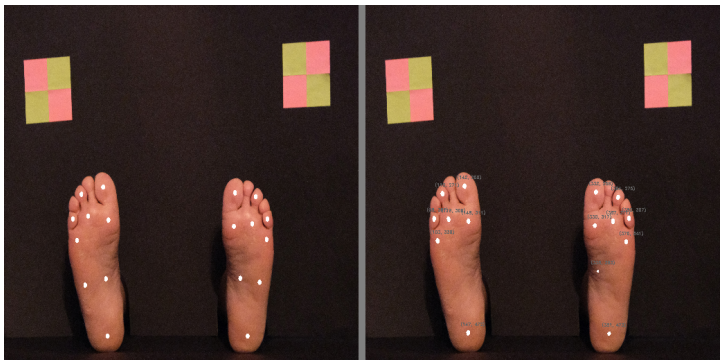


Figura 5.22: Sobreposição da máscara manual e da máscara gerada pelo modelo de 50 iterações e 9 pontos, sobre a imagem de origem.

métricas dos 3 exemplos gerados, de forma a facilitar a visualização da progressão, ou regressão, de cada modelo.

5.4 Ajustes ao modelo com melhor desempenho

Os resultados obtidos para os modelos de 9 pontos e 20 ou 50 iterações já apresentam um aspeto bastante próximo daquilo que é pretendido. Contudo, foram executadas mais algumas tentativas e ajustes nos parâmetros da rede de forma a perceber de que forma se poderiam baixar os valores de perda de validação. O objetivo seria produzir um modelo cuja curva de perda de validação acompanhasse a curva de perda de treino.

5.4.1 Expansão do conjunto de dados

Uma das primeiras tentativas aplicadas foi a de ampliar ainda mais o conjunto de dados disponíveis através do aumento da extensão de manipulação das imagens originais.

Acrescentaram-se então mais 3 manipulações que não tinham sido previamente executadas: um ajuste de contraste/luminosidade aleatório, um desvio

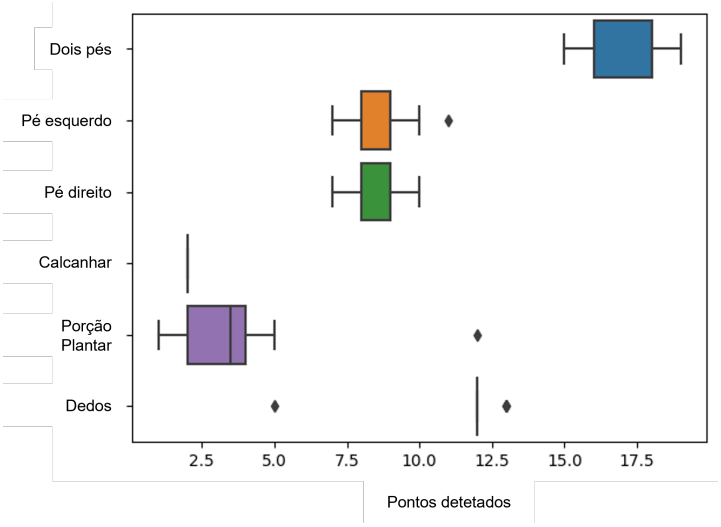


Figura 5.23: Número de pontos detetados para diferentes áreas dos pés ao longo das 20 imagens de avaliação do modelo de 50 iterações.

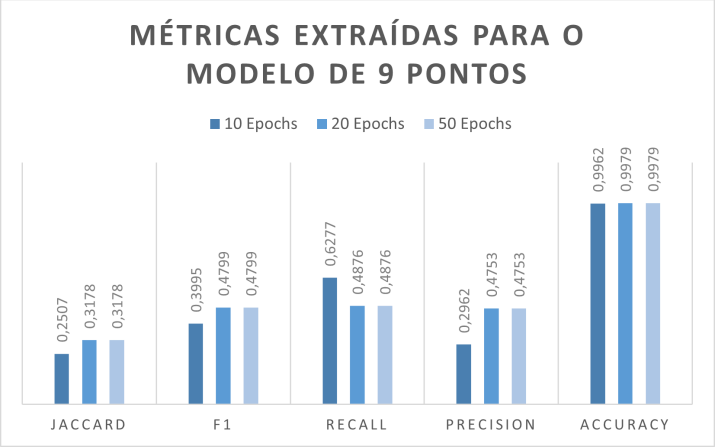


Figura 5.24: Valores de métricas extraídas em cada modelo de 9 pontos.

dos padrões de cor e uma distorção elástica.

Desta forma, o conjunto de dados passou de 276 imagens de treino para um total de 414. Com estes dados, o modelo foi novamente treinado para 50 *epochs*. Foi selecionado este número de iterações, em vez das 20, para haver espaço de melhoria ao longo do processo de treino.

A Figura 5.25 mostra os resultados que foram obtidos para esta situação. Percebe-se que entre estes resultados e os da Figura 5.20 não existem grandes diferenças. Neste modelo, contudo, verifica-se que os mesmos valores de perda de validação são atingidos em menos iterações de treino, uma vez que a curva de perda de validação acompanha a curva de perda de treino durante mais tempo.

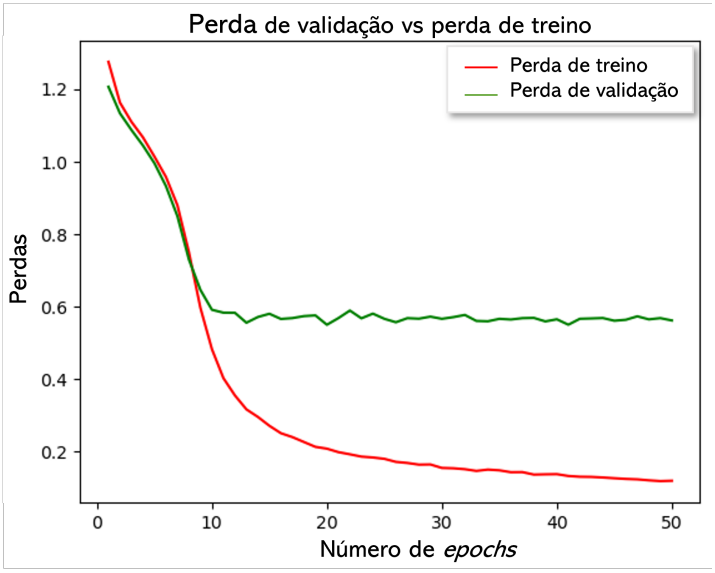


Figura 5.25: Evolução dos valores de perda de validação e perda de treino ao longo de 50 epochs, com um lote de tamanho dois, para o modelo de 9 pontos com um conjunto de dados expandido.

Ao comparar as métricas obtidas para este modelo com as métricas obtidas para o modelo anterior, verifica-se que, apesar dos gráficos aparentemente indicarem que os dois modelos são semelhantes ao nível de perdas, existe uma ligeira diminuição nos valores deste novo exemplo. O gráfico da Figura 5.26 mostra exatamente essa relação.

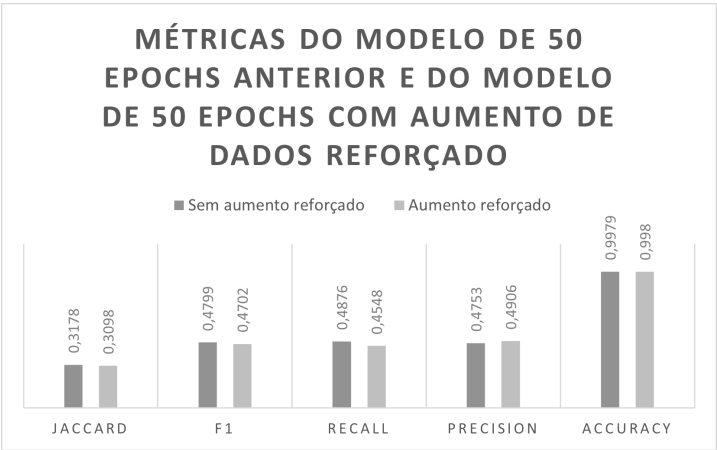


Figura 5.26: Métricas extraídas para o modelo de 50 *epochs* e 9 pontos e para o modelo de 50 *epochs* e 9 pontos com o conjunto de dados expandido.

5.4.2 Aumento do tamanho de lote

Outro hiperparâmetro que se ajustou de forma a procurar obter melhores resultados foi o tamanho de lote. Esta variável por si só representa o número de amostras pelo qual o algoritmo itera e executa as previsões para saídas, sendo que de seguida as compara com as saídas reais. Este processo é executado antes de o modelo ser atualizado.

Numa primeira abordagem tentou-se aumentar o tamanho de lote de 2 para 4. Contudo, rapidamente se teve de abortar esta ideia, dado que a GPU do computador utilizado para executar o treino da rede não possuía memória suficiente para executar todo o processo.

Seguidamente decidiu tentar-se com um tamanho de lote de 3, valor para o qual foi possível realizar o processo com o *hardware* em questão. Para este modelo apenas foram consideradas 30 iterações de treino devido às referências obtidas previamente.

Para este novo modelo notou-se que houve uma diminuição na amplitude dos valores, tanto de perda de treino, como de perda de validação, sugerindo uma eficiência inferior face ao modelo com tamanhos de lote de valor dois. Não obstante, os valores finais para ambas as curvas são bastante próximos daqueles observados para os modelos anteriores. Os resultados do treino encontram-se sintetizados na Figura 5.27.

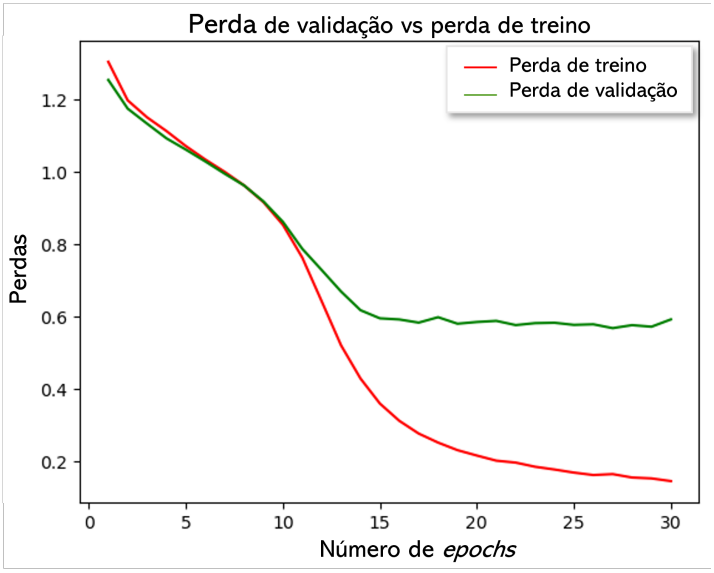


Figura 5.27: Evolução dos valores de perda de validação e perda de treino ao longo de 30 *epochs* com um lote de tamanho três para o modelo de 9 pontos.

Respetivamente às métricas deste mesmo modelo, os resultados, tal como

esperado, não foram melhores que os obtidos com o modelo anterior, nem do que os obtidos com o modelo original de 50 iterações. Os resultados obtidos, face aos resultados do melhor modelo até à data, podem ser observados na Figura 5.28. Novamente, apesar de os valores se encontrarem próximos, na globalidade, o modelo de 50 iterações leva a vantagem.

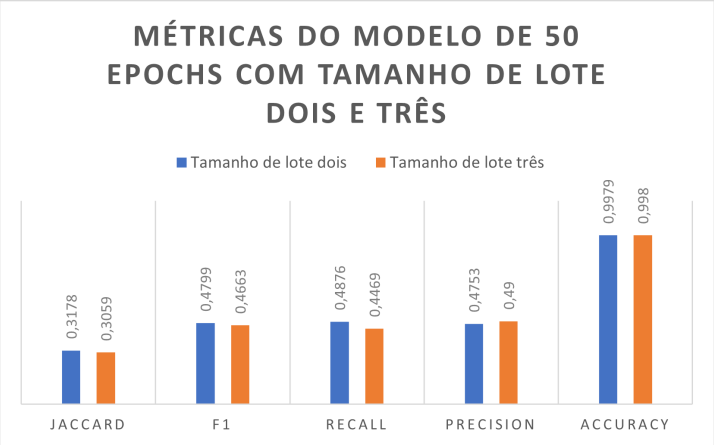


Figura 5.28: Métricas extraídas para o modelo de 50 *epochs* e 9 pontos, e para o modelo de 30 *epochs* e 9 pontos com tamanho de lote 3.

5.5 Filtro para a deteção de pontos

Ao longo deste capítulo, várias imagens foram mostradas na secção de resultados do modelo preditivo. Estas imagens contêm já as coordenadas em pixeis para os pontos detetados. Contudo, o modelo de segmentação apenas se limita a assinalar os pontos nos lugares certos do pé.

Para se saber quantos pontos o algoritmo detetou, e quais as coordenadas dos mesmos, é necessário proceder a um conjunto de operações designado por " blob detection". Este método permite analisar uma imagem de forma a encontrar manchas/bolhas com características idênticas, através da ativação ou inativação de um conjunto de propriedades de triagem.

Para o caso em estudo, as propriedades de filtragem definidas foram as de cor, para filtrar somente pontos brancos, e as propriedades de área, para garantir que apenas bolhas com mais de 4 pixeis de diâmetro eram consideradas. As restantes propriedades foram desativadas, como a inércia, circularidade e convexidade.

De forma a conseguir entender melhor os resultados obtidos por este filtro, criou-se um pequeno *script* para desenhar sobre a imagem da máscara círculos vermelhos em volta dos pontos que o algoritmo conseguiu recolher da mesma. O resultado desse processo pode ser observado na Figura 5.29. As bolhas detetadas

pelo filtro são depois armazenadas num objeto com atributos de posição em x, posição em y, e diâmetro da mancha. Seguidamente é possível sobrepor a máscara com as coordenadas de cada ponto, sobre a imagem original.

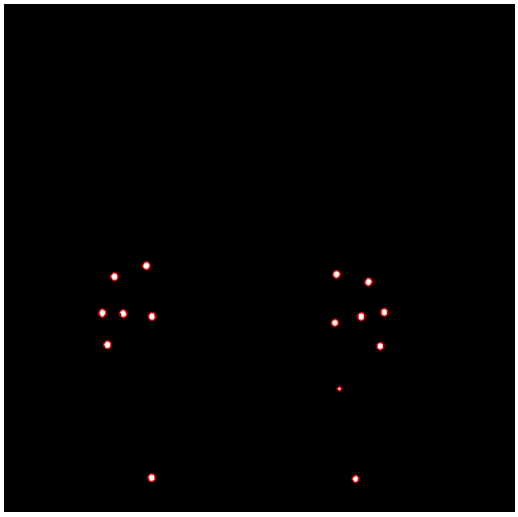


Figura 5.29: Representação visual de quais os pontos encontrados pelo algoritmo de recolha de pontos para uma das máscaras geradas pelo modelo preditivo.

5.6 Obtenção de coordenadas reais

A secção anterior explica de que forma se conseguiu obter as coordenadas em pixeis para cada um dos pontos segmentados pelo algoritmo preditivo. Contudo, quando se tenta extrapolar estes resultados para aplicações práticas, percebe-se que posições relativas a pixeis não cumprem os requisitos de outro tipo de dispositivos ou *software*. De tal forma é necessário aplicar um processo que as transforme em coordenadas reaism, com unidades de medida padronizadas.

O problema foi pensado em duas partes: primeiramente pretendia-se fazer conversão dos pixeis para centímetros das coordenadas x e y, e seguidamente obter a coordenada relativa à profundidade, ou seja, z.

Para a conversão das unidades em píxel para centímetro fez-se uso dos quadrados coloridos (amarelo e rosa) presentes no fundo de cada foto dos pés. Mediram-se as dimensões de um dos quadrados amarelos manualmente, para a qual se obteve os valores de 3,8 cm por 3,8 cm. Para obter as dimensões do mesmo quadrado, mas em pixeis, aplicou-se um filtro *Hue Saturation Value* (HSV) utilizando como limites de segmentação os limiares da cor amarelo. Esta aplicação permite segmentar unicamente os quadrados amarelos da imagem original. Por cima da máscara obtida aplicaram-se ainda duas operações morfológicas, uma dilatação seguida de uma erosão, tal processo permite obter segmentações mais perfeitas.

Por fim, aplicando uma função de extração de contornos, foi possível extrair quais as dimensões, em pixeis, de cada quadrado. O seguimento do processamento de imagem pode ser observado na Figura 5.30.

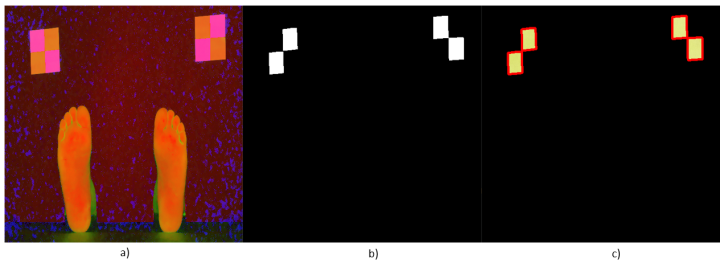


Figura 5.30: Processo de segmentação dos quadrados amarelos. a) Filtro HSV, b) Segmentação após dilatação e erosão, c) Resultado da segmentação.

A obtenção destes dados permite, agora, estabelecer um termo de conversão para o cálculo das coordenadas reais. Dividindo a dimensão real de um lado do quadrado amarelo pela dimensão em pixeis desse mesmo lado, obtém-se o número de centímetros por píxel. Multiplicando este termo pelas coordenadas obtidas na secção anterior obtém-se o valor das coordenadas de cada ponto em centímetros. Os novos valores podem ser observados na Figura 5.31.

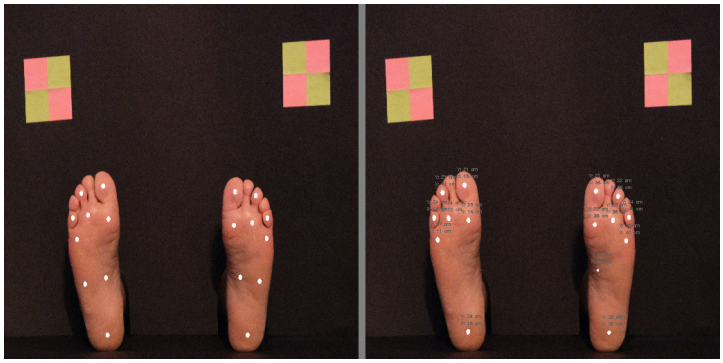


Figura 5.31: Imagem original com a máscara manual e imagem original com as coordenadas reais dos pontos segmentados pelo modelo preditivo.

A captura da coordenada em profundidade, z, teve de ser obtida recorrendo a outro método. Para tal, utilizou-se um dispositivo OAK-D [59] (Figura 5.32), o qual contém duas câmaras *stereo* que produzem uma imagem em profundidade da secção enquadrada no *frame*, e uma câmara 4K, central, que oferece uma imagem RGB desse mesmo *frame*.

Tal como referido anteriormente, a imagem em profundidade é conseguida através de um par de câmaras *stereo*, que se encontram localizadas em ambas as extremidades do equipamento. Cada câmara vai capturar uma imagem única do

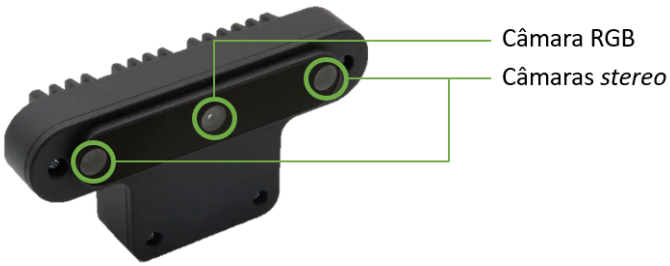


Figura 5.32: Dispositivo OAK-D e as suas diferentes câmaras.

objeto na sua frente, contudo, uma vez que estão em posições distintas, as imagens obtidas por cada uma não serão exatamente iguais. Esta diferença chama-se disparidade e não é igual ao longo de toda a imagem. Ao se sobrepor estas duas imagens, consegue-se prever qual o valor estimado para a profundidade em que um determinado objeto se encontra. Nota-se que este processo simula o processo de visão humana, no que toca à noção de profundidade, e pode ser observado na Figura 5.33.

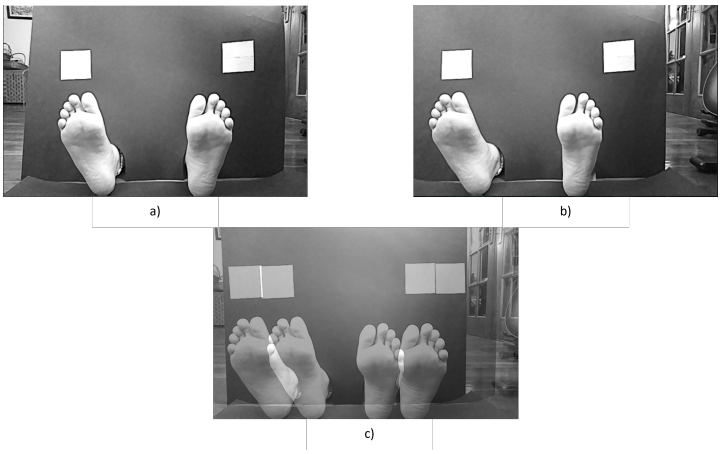


Figura 5.33: Imagens obtidas pelas câmaras *stereo*. a) captura da câmara esquerda, b) captura da câmara direita, c) sobreposição das duas imagens, dando origem à imagem de disparidade.

Uma vez que os valores de profundidade são difíceis de extrair através da análise do resultado não processado das imagens sobrepostas, utilizaram-se os valores de disparidade obtidos dessa mesma sobreposição para se gerar um mapa de cores da imagem, o qual pode ser observado na Figura 5.34.

Já na Figura 5.35 é possível verificar que é definida uma região de interesse (ROI), que se encontra delimitada a verde, dado que, a Application Programming

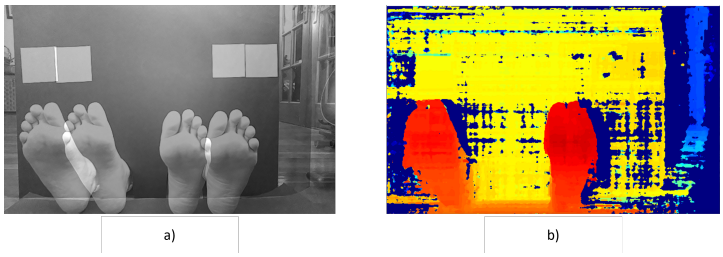


Figura 5.34: a) disparidade resultante da sobreposição das camaras *stereo*, b) introdução do mapa de cores sobre os valores de disparidade.

Interface (API) utilizada para comunicar com a câmara implica a criação de uma ROI para se poder extrair o valor numérico para a posição z. Sendo o pé uma superfície aproximadamente plana, que estará numa posição vertical durante todo o exame, é possível assumir que os valores de profundidade para todos os pontos do pé serão bastante próximos.

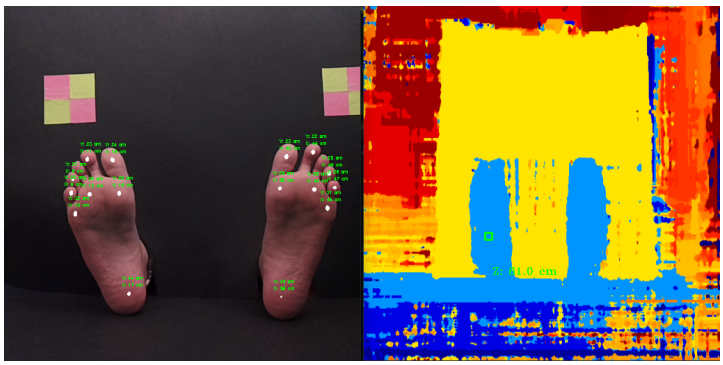


Figura 5.35: Resultado de uma previsão em tempo real com a imagem de profundidade para a extração do valor de z.

5.7 Conclusão do capítulo 5

Quando o tema passa por processamento de imagem, existem sempre diversos problemas que têm de ser pensados para que o desenvolvimento corra de forma fluida. Este capítulo oferece uma perspetiva de quais as decisões tomadas ao longo do desenvolvimento, reações face aos resultados que foram surgindo, análise de métricas próprias para problemas de segmentação, e métricas personalizadas ao problema em mão. Torna-se também importante refletir no conjunto de resultados obtidos e analisar de que forma os mesmos evoluíram. Finalmente, para que o algoritmo possa integrar aplicações reais, foi necessário a obtenção das coordenadas reais no espaço para cada um dos pontos segmentados pelo algoritmo preditivo.

Capítulo 6

Conclusões

Ao longo deste capítulo será feita uma reflexão relativamente a todo o trabalho desenvolvido durante a unidade de Tese/Dissertação. É importante parar e olhar para aquilo que foi feito de maneira a retirar conclusões, que sejam úteis, tanto para trabalhos futuros, como para o desenvolvimento pessoal. É igualmente importante analisar o que correu bem, o que correu mal, e o que poderia ter sido feito para mitigar este último. Assim, este capítulo será dividido em duas secções principais: conclusão e trabalhos futuros. Ao longo da conclusão tem-se uma dissecação de tudo o que foi produzido com o trabalho desenvolvido. Incluindo tudo aquilo que se conseguiu obter de acordo com o expectável para o trabalho, e também os pontos que ficaram comprometidos devido à falta de tempo. Na secção dos trabalhos futuros serão expostas todas as implementações e adições novas que, de uma ou outra forma, enriqueceriam o projeto e o seu objetivo final.

6.1 Conclusão

O culminar dos últimos meses de trabalho, pesquisa e implementação de código fundem-se, agora, nesta secção final do trabalho, as conclusões e pensamentos finais. O trabalho foi iniciado com um grande objetivo em vista, o de introduzir inteligência artificial numa tarefa de segmentação. Apesar da iteração anterior do trabalho, utilizando técnicas de processamento de imagem e segmentação, ter sido sucedida, a revolução tecnológica e os avanços ao nível da computação continuam cada vez mais robustos. Desta forma, partiu-se para uma solução mais sofisticada do ponto de vista tecnológico, a aplicação de processos de segmentação com recurso a algoritmos de *machine learning* e aprendizagem profunda.

A aplicação de tais conceitos permite ao utilizador uma abstração dos processos inerentes à segmentação de imagem, pelo que leva simultaneamente a uma

previsão mais robusta do *output* final. O modelo em estudo foi o das redes de segmentação U-Net, famosas pela sua capacidade de produzir excelentes resultados face a uma dimensão reduzida de dados de treino. O modelo desenvolvido foi treinado com o recurso a um subconjunto de dados de treino, e outro subconjunto de dados de avaliação. Nota-se que, em ambas as partições existia um par imagem - máscara em que a partição de treino foi expandida utilizando processos de aumento de dados, como distorções elásticas, morfológicas e mudanças de saturação.

Para os treinos da rede, foram definidos hiperparâmetros específicos, para os quais se fez variar maioritariamente o valor das *epochs* entre iterações. Após o treino de seis redes, verificou-se que o modelo que apresentava melhor comportamento a nível de qualidade de saída e desempenho foi o de 20 *epochs*. Mais tarde, foram ainda gerados mais dois modelos com vista a tentar atingir resultados ainda melhores, procedendo a um aumento do número de dados de treino, e a um aumento do número de *workers*. Contudo, nenhum destes obteve resultados melhores que o anterior.

De forma a ser possível avaliar cada modelo gerado, foram definidas um conjunto de métricas de avaliação, sendo estas definidas em duas categorias: métricas gerais para problemas de segmentação, e métricas especificas ao problema em questão. Os resultados para as métricas gerais não foram tão animadores como o esperado, contudo, os resultados das específicas encontram-se de acordo com as máscaras geradas pelos modelos preditivos. Isto deve-se ao facto de, não só os cálculos das métricas gerais, como a pontuação de Jaccard e a pontuação de F1, serem realizados com base na posição em pixeis de cada zona segmentada, como também pelo facto de a zona segmentada ser de reduzida dimensão.

Apesar de os resultados obtidos com o modelo preditivo gerado satisfazerem as necessidades de projeto, é possível afirmar que ainda existe espaço para melhoria, quer seja a nível de *design* do modelo em si, ou até mesmo do processo de treino na totalidade. Um dos pontos que deixou a desejar foi os bloqueios técnicos face à máquina onde o processo de treino foi executado.

Finalmente, um dos temas que não se conseguiu atingir foi o de complementar o algoritmo de segmentação com um processo mecânico que realmente conseguisse executar o exame à neuropatia periférica. Apesar de todo o algoritmo se encontrar pronto para integração com a parte robótica, através da extração das coordenadas reais de cada um dos pontos segmentados, esta parte final não foi possível de implementar.

Concluindo, o trabalho apresentou resultados bastante interessantes na medida da segmentação, conseguindo mesmo extrapolar quais as coordenadas reais para cada um dos pontos detetados pelo modelo. Todos estes avanços proporcionam uma boa perspetiva futura para a continuação do projeto.

6.2 Trabalhos futuros

O desenvolvimento deste trabalho abre múltiplas possibilidades para aplicações práticas do mesmo. Uma dessas aplicações, a qual não foi possível realizar devido à falta de tempo, é a integração da informação extraída pelo algoritmo de inteligência artificial com um braço robótico. O objetivo seria dotar o manipulador robótico de capacidades que lhe permitissem realizar o exame da neuropatia periférica de forma autónoma.

Através de uma API de ligação em rede de baixo nível seria possível criar uma "interface" entre o código em Python e os comandos aceites pelo robô. Isto permite ao utilizador injetar valores externos, como coordenadas, nos comandos utilizados pelo *software* nativo do equipamento.

Uma aplicação interessante estaria também no controlo de força exercido pelo equipamento. Assim, para além de ser possível aplicar uma força específica num determinado ponto do pé, seria também capaz de extrair quais os valores de força para os quais o paciente sentiu contacto com monofilamento. Esta funcionalidade seria extremamente interessante, dada a possibilidade de analisar o nível de sensibilidade de cada zona.

Complementar à funcionalidade anterior, também seria de esperar a implementação de um *time out*, ou seja, um valor de tempo para o qual o algoritmo procuraria por algum tipo de *feedback* do paciente, quer seja por intervenção direta com o equipamento, como o pressionar de um botão, ou até mesmo por algum tipo de sinal esperado pelo algoritmo de visão artificial.

Do ponto de vista do *software*, algo interessante seria comparar os resultados obtidos com os resultados obtidos por outro tipo de modelo de rede, como, por exemplo, a ResU-Net. A possibilidade de analisar qual o desenvolvimento no que toca ao desempenho de treino de uma rede para uma máquina com uma capacidade de processamento gráfico superior, também é um ponto importante.

Adicionalmente, seria interessante proceder a uma recolha de um maior número de imagens de pés e da segmentação dos mesmos. Uma amostra de treino maior, certamente, levaria a resultados mais interessantes. Alterando as configurações do modelo preditivo, poder-se-ia gerar máscaras coloridas, como aquelas apresentadas no Capítulo 4. A utilização deste tipo de máscaras permitiria obter uma noção mais profunda da qualidade preditiva do algoritmo, uma vez que se poderia visualizar se, para além de todos os pontos previstos terem sido gerados corretamente, se cada um destes foi gerado na posição adequada ao mesmo, e assim validar se o algoritmo estaria a fazer a correta distinção entre, por exemplo, os vários dedos dos pés.

Referências Bibliográficas

[1] OMS, "Diabetes." https://www.who.int/news-room/fact-sheets/detail/diabetes, Último acesso: 30 de Março 2022. [cited on p. vii, ix, 1, 2]

[2] A. J. Boulton, "Management of Diabetic Peripheral Neuropathy," *Clinical Diabetes*, vol. 23, pp. 9–15, 01 2005. [cited on p. vii, ix, 2]

[3] C. Price and C. Nester, "Foot dimensions and morphology in healthy weight, overweight and obese males," *Clinical Biomechanics*, vol. 37, pp. 125–130, aug 2016. [cited on p. v, 43]

[4] W. V. Gonzales, A. T. Mobashsher, and A. Abbosh, *The progress of glucose monitoring—A review of invasive to minimally and non-invasive techniques, devices and sensors*, vol. 19. sensors, 2019. [cited on p. 1]

[5] M. S. Anjali D Deshpande, Marcie Harris-Hayes, "Diabetes-Related Complications," *Physical Therapy & Rehabilitation Journal*, vol. 88, no. 11, 2008. [cited on p. 1, 2]

[6] K. M. Gillespie, "Type 1 diabetes: pathogenesis and prevention," *CMAJ*, vol. 175, pp. 165–170, July 2006. [cited on p. 1]

[7] K. Barrell and A. G. Smith, "Peripheral neuropathy," *Medical Clinics of North America*, vol. 103, no. 2, pp. 383–397, 2019. Neurology for the Non-Neurologist. [cited on p. 2]

[8] M. Y. Shaheen, "Applications of Artificial Intelligence (AI) in healthcare: A review," *ScienceOpen Preprints*, sep 2021. [cited on p. 3]

[9] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism: Clinical and Experimental*, vol. 69, pp. S36–S40, 2017. [cited on p. 7, 13, 14, 15]

[10] M. Neelam, *Aspects of Artificial Intelligence In Karthikeyan.J, Su-Hie Ting and Yu-Jin Ng*. Ordine Nuovo Publication, 01 2022. [cited on p. 7]

[11] A. N. Ramesh, C. Kambhampati, J. R. Monson, and P. J. Drew, "Artificial intelligence in medicine," *Annals of the Royal College of Surgeons of England*, vol. 86, no. 5, pp. 334–338, 2004. [cited on p. 8]

[12] A. Krogh, "What are artificial neural networks?," *Nature Biotechnology*, vol. 26, no. 2, pp. 195–197, 2008. [cited on p. 8, 9]

[13] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996. [cited on p. 9]

[14] G. Zhang, M. Y. Hu, B. E. Patuwo, and D. C. Indro, "Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis," *European Journal of Operational Research*, vol. 116, no. 1, pp. 16–32, 1999. [cited on p. 9]

[15] M. Haenlein and A. Kaplan, "A brief history of artificial intelligence: On the past, present, and future of artificial intelligence," *California Management Review*, vol. 61, no. 4, pp. 5–14, 2019. [cited on p. 9, 10]

[16] A. Benko and C. Lanyi, "History of Artifical Inteligence," *IGI global*, pp. 3–4, 2009. [cited on p. 9]

[17] Y. Shin, "The Spring of Artificial Intelligence in Its Global Winter," *IEEE Annals of the History of Computing*, vol. 41, no. December, pp. 71–82, 2019. [cited on p. 9]

[18] A. Jean, "A brief history of artificial intelligence," *Medecine/Sciences*, vol. 36, no. 11, pp. 1059–1067, 2020. [cited on p. 10]

[19] J. Frankemfield, "Weak AI Definition." https://www.investopedia.com/terms/w/weak-ai.asp, Último acesso: 24 de Maio 2022. [cited on p. 11]

[20] J. Borana, "Applications of Artificial Intelligence & Associated Technologies.," *IEEE Region 5 Conference*, no. March, pp. 1–4, 16. [cited on p. 11]

[21] "What is Supervised Learning? | IBM." https://www.ibm.com/cloud/learn/supervised-learning, Último acesso: 25 de Maio 2022. [cited on p. 11, 12]

[22] IBM, "What is Unsupervised Learning? | IBM." https://www.ibm.com/cloud/learn/unsupervised-learning, 2022. [cited on p. 11, 12, 13]

[23] P. Dayan and Y. Niv, "Reinforcement learning: The Good, The Bad and The Ugly," *Current Opinion in Neurobiology*, vol. 18, no. 2, pp. 185–196, 2008. [cited on p. 11, 13]

[24] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, 2018. [cited on p. 12, 20, 21]

[25] Z. Ghahramani, "LNAI 3176 - Unsupervised Learning," *Machine Learning*, pp. 72–112, 2004. [cited on p. 12, 13]

[26] Y. Mintz and R. Brodie, "Introduction to artificial intelligence in medicine," *Minimally Invasive Therapy and Allied Technologies*, vol. 28, no. 2, pp. 73–81, 2019. [cited on p. 13, 14]

[27] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain Intelligence: Go beyond Artificial Intelligence," *Mobile Networks and Applications*, vol. 23, no. 2, pp. 368–375, 2018. [cited on p. 13, 15, 16, 17]

[28] M. N. O. Sadiku, T. J. Ashaolu, A. Ajayi-Majebi, and S. M. Musa, "Artificial Intelligence in Social Media," *International Journal Of Scientific Advances*, vol. 2, no. 1, 2021. [cited on p. 14, 16]

[29] K. H. Yu, C. Zhang, G. J. Berry, R. B. Altman, C. Ré, D. L. Rubin, and M. Snyder, "Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features," *Nature Communications*, vol. 7, pp. 1–10, 2016. [cited on p. 14]

[30] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 4633–4658, 2021. [cited on p. 15]

[31] G. Skinner and T. Walmsley, "Artificial intelligence and deep learning in video games a brief review," *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, pp. 404–408, 2019. [cited on p. 16]

[32] M. Leo, A. Furnari, G. G. Medioni, M. Trivedi, and G. M. Farinella, *Deep learning for assistive computer vision*, vol. 11134 LNCS. Springer International Publishing, 2019. [cited on p. 19, 20]

[33] IBM, "What is Computer Vision? | IBM." https://www.ibm.com/topics/computer-vision, Último acesso: 4 de Junho 2022. [cited on p. 20]

[34] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, *Deep Learning vs. Traditional Computer Vision BT - Advances in Computer Vision*. Springer International Publishing, 2020. [cited on p. 20, 21]

[35] Olaf Ronneberger, Philipp Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Springer International Publishing Switzerland 2015*, vol. 9351, pp. 12–20, 2015. [cited on p. 21, 22, 35]

[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to digit recognition," *AT&T Bell Laboratories Holdmodel*, vol. 1, no. 4, pp. 541–551, 1989. [cited on p. 21]

[37] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Xplore*, nov 2014. [cited on p. 21]

[38] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, p. 22, 2018. [cited on p. 22, 23]

[39] D. Lee and Y. Park, "Vision-based remote control system by motion detection and open finger counting," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2308–2313, 2009. [cited on p. 23]

[40] N. Singla, "Motion Detection Based on Frame Difference Method," *International Journal of Information & Computation Technology*, vol. 4, no. 15, pp. 1559–1565, 2014. [cited on p. 23, 24]

[41] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001. [cited on p. 23, 24]

[42] D. Khemasuwan, J. S. Sorensen, and H. G. Colt, "Artificial intelligence in pulmonary medicine: Computer vision, predictive model and covid-19," *European Respiratory Review*, vol. 29, no. 157, pp. 1–16, 2020. [cited on p. 24, 25, 26]

[43] O. Wieben, *Virtual and augmented reality in medicine.* Elsevier Inc., 2016. [cited on p. 24, 27]

[44] J. Gao, Y. Yang, P. Lin, and D. S. Park, "Editorial: Computer vision in healthcare applications," *Journal of Healthcare Engineering*, vol. 2018, 2018. [cited on p. 24, 25]

[45] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado, D. P. Naidich, and S. Shetty, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," *Nature Medicine*, vol. 25, no. 6, pp. 954–961, 2019. [cited on p. 25]

[46] S. L. Walsh, L. Calandriello, M. Silva, and N. Sverzellati, "Deep learning for classifying fibrotic lung disease on high-resolution computed tomography: a case-cohort study," *The Lancet Respiratory Medicine*, vol. 6, no. 11, pp. 837–845, 2018. [cited on p. 25]

[47] A. J. Sweatt, H. K. Hedlin, V. Balasubramanian, A. Hsi, L. K. Blum, W. H. Robinson, F. Haddad, P. M. Hickey, R. Condliffe, A. Lawrie, M. R. Nicolls, M. Rabinovitch, P. Khatri, and R. T. Zamanian, "Discovery of Distinct Immune Phenotypes Using Machine Learning in Pulmonary Arterial Hypertension," *Circulation Research*, vol. 124, no. 6, pp. 904–919, 2019. [cited on p. 26]

[48] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020. [cited on p. 29, 33]

[49] "robot-arm-and-feet-detection." https://github.com/RuiMesquita/robot-arm-and-feet-detection, Último acesso: 17 de Outubro 2022. [cited on p. 30]

[50] T. GIMP, "GIMP - GNU Image Manipulation Program." https://www.gimp.org/, 23 de Julho 2022. [cited on p. 31]

[51] D. R. Cornblath, "Diabetic neuropathy: Diagnostic methods," *Advanced Studies in Medicine*, vol. 4, no. 8 A, pp. 650–661, 2004. [cited on p. 31]

[52] E. L. Feldman and M. J. Stevens, "Clinical Testing in Diabetic Peripheral Neuropathy," *Canadian Journal of Neurological Sciences / Journal Canadien des Sciences Neurologiques*, vol. 21, no. S4, pp. S3–S7, 1994. [cited on p. 31]

[53] Adobe, "GIF files: How to create, edit and open them | Adobe." https://www.adobe.com/creativecloud/file-types/image/raster/gif-file.html, 23 de Julho 2022. [cited on p. 31]

[54] N. Ketkar and J. Moolayil, *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch, 2nd Edition.* Nikhil Ketkar, Jojo Moolayil, second edi ed., 2021. [cited on p. 34, 35]

[55] "PyTorch documentation — PyTorch 1.12 documentation." https://pytorch.org/docs/stable/index.html, Último acesso: 24 de Julho 2022. [cited on p. 34]

[56] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015. [cited on p. 37]

[57] "Loss Function Library - Keras & PyTorch | Kaggle." https://www.kaggle.com/code/bigironsphere/loss-function-library-keras-pytorch/notebook, Último acesso: 9 de Agosto 2022. [cited on p. 37]

[58] "torch.nn.functional.binary_cross_entropy — PyTorch 1.12 documentation." https://pytorch.org/docs/stable/generated/torch.nn.functional.binary_cross_entropy.html#torch.n Último acesso: 9 de Agosto 2022. [cited on p. 37]

[59] "OAK-D — DepthAI Hardware Documentation 1.0.0 documentation," Último acesso: 17 de Outubro 2022. [cited on p. 62]