Universidade do Minho
Escola de Engenharia

Américo Filipe Moreira Oliveira

Otimização de Arquiteturas de *Deep Learning* para Segmentação de Vasos Sanguíneos em Imagens de Fundo Retiniano

Dissertação de Mestrado
Mestrado Integrado em Engenharia Biomédica
Ramo de Eletrónica Médica

Trabalho efetuado sob orientação do
Professor Doutor Carlos Alberto Batista Silva

Outubro 2017

Declaração

Nome: Américo Filipe Moreira Oliveira

Endereço eletrónico: americofmoliveira@gmail.com

Número de Cartão de Cidadão: 14380768

Título de Dissertação: Otimização de Arquiteturas de *Deep Learning* para Segmentação de Vasos Sanguíneos em Imagens de Fundo Retiniano

Orientador:

Professor Doutor Carlos Alberto Batista Silva

Ano de conclusão: 2017

Designação do Mestrado: Mestrado Integrado em Engenharia Biomédica
Ramo: Eletrónica Médica

Universidade do Minho, ___ / ___ / ___

Assinatura: _____

Agradecimentos

Foram incontáveis as vezes em que me imaginei a escrever estas linhas ao longo deste ano. Desde logo, porque elas simbolizam o fim de uma etapa que sinto ser altura de fechar. Depois, e sobretudo, porque supus que estas fossem as linhas mais fáceis de escrever. Chegada a hora da verdade, confesso que incluir neste espaço todas as pessoas que, ao longo destes cinco anos, contribuíram para a minha formação enquanto Homem e Engenheiro é uma tarefa condenada ao fracasso. Ainda assim, reservo algumas palavras para os que não poderia deixar de mencionar.

Em primeiro lugar, quero agradecer ao Professor Carlos Silva, meu orientador, pela oportunidade de trabalhar numa área que me fascina e, acima de tudo, pela dedicação, pela disponibilidade a qualquer altura e por todos os conselhos e orientações ao longo deste ano.

Ao Sérgio e ao Adriano, quero agradecer as imensas vezes em que me prestaram ajuda. Foi um privilégio poder aprender convosco e estou seguro que vos esperam carreiras de sucesso.

Aos meus inigualáveis Engenheiros uma palavra de gratidão por tudo o que simbolizaram para nós ao longo desta aventura. O meu obrigado por todos os "zero!"'s, pelos ensinamentos e pelas memórias. Ao meu padrinho, Sr. Engenheiro Zamith, um obrigado especial por teres sido sempre um exemplo e por estares sempre presente.

Aos meus caloiros, o meu obrigado pelo melhor ano que passei na universidade. Obrigado pela humildade, pela coragem para pensar pela própria cabeça e pela oportunidade de vos ver crescer. O nosso (vosso) bloco foi o melhor que já vi. Às minhas afilhadas, Tampas, Arsénio e Puré, obrigado por um dos dias mais felizes da minha passagem por Braga. Saí dali a achar que tinha as melhores bestas do bloco, e tenho.

Aos que partilharam comigo todos estes anos, um obrigado enorme por me terem feito crescer. Ao Braga, um obrigado por todos os episódios hilariantes que vamos poder recordar mais tarde. À Bia e ao Rei, um obrigado por me terem feito sentir que somos irmãos à séria. Ao Bito, um obrigado especial por todos os desafios que enfrentamos juntos nesse período mítico que é o 4º ano de Eletrónica Médica. A todos vós, acima de tudo, um obrigado pela amizade.

À Catarina, obrigado por todas as vezes em que ouviste os meus problemas, as minhas frustrações e desabafos. Obrigado por teres sido, tantas vezes, a minha fuga. Obrigado pelo amor e pela amizade. Sem ti, não teria concluído esta dissertação.

Aos meus pais, obrigado pela oportunidade para me formar. E à minha Mãe, obrigado por tudo o que sou. Obrigado pelo teu amor incondicional, pela dedicação, pela preocupação e por todas as vezes em que te continuas a sacrificar por mim. O que poderei ser na vida, a ti se deve. Obrigado por seres a mulher mais forte que conheço.

Resumo

A retina é a única parte do corpo humano onde a corrente sanguínea pode ser observada de forma direta e não invasiva. Atualmente, existem claras evidências de que as alterações fisiológicas da árvore vascular estão relacionadas com várias doenças que afetam milhões de pessoas a nível mundial, como a diabetes e a hipertensão. Contudo, a avaliação da condição vascular da retina exige a segmentação de várias imagens que, quando executada manualmente, se traduz num processo moroso e propenso a erros até para os especialistas mais experientes. Neste contexto, as metodologias de segmentação automática ganham cada vez mais relevância e são vistas como alternativas capazes de reduzir a carga de trabalho dos especialistas e de fornecer resultados robustos e reprodutíveis, em tempo útil.

O objetivo desta dissertação passa pelo desenvolvimento de um método de segmentação automática de vasos retinianos em imagens de fundo. Ao longo dos anos, várias técnicas têm sido utilizadas para abordar este problema. Esta dissertação foca-se em métodos de *deep learning* e, em particular, nas redes neuronais convolucionais. Este tipo de modelos tem ganho especial destaque nos últimos anos, graças à sua capacidade para aprender, de forma automática, uma hierarquia de *features* cada vez mais complexas a partir dos dados de entrada. Isto parece sugerir uma mudança de paradigma segundo a qual o foco está agora na otimização de arquiteturas e não na tradicional fase de *feature engineering*, que é provavelmente a etapa mais crítica nos métodos convencionais.

O método proposto nesta dissertação foi avaliado em três bases de dados públicas e amplamente reconhecidas na área da segmentação de vasos retinianos: a DRIVE, a STARE e a CHASE_DB1.

Durante as várias etapas deste trabalho, estudaram-se diversas variantes procurando otimizar os resultados obtidos. O melhor modelo assenta numa rede neuronal completamente convolucional com várias escalas. Além disto, faz uso de operações de rotação para fins de aumento artificial de dados e beneficia da informação codificada pela transformada de wavelet não decimada. Em termos quantitativos, obtiveram-se valores médios de *accuracy* de 0.9576, 0.9694 e 0.9653 e valores médios de área abaixo da curva ROC de 0.9821, 0.9905 e 0.9855, na DRIVE, STARE e CHASE_DB1, respetivamente. Para mais, o método revelou-se robusto face à mudança do conjunto de treino e à presença de patologias, o que mostra o seu potencial para aplicações do mundo real.

Abstract

The retina is the only part of the human body where the bloodstream can be directly and non-invasively observed. There is now clear evidence that physiological changes in the vascular tree are related to several diseases that affect millions of people worldwide, such as diabetes and hypertension. However, evaluating the retinal vascular condition requires the segmentation of several images which, done manually, is a time-consuming and error prone task even for experienced experts. In this context, automatic segmentation methodologies are becoming increasingly important to reduce the workload of specialists, and provide robust and reproducible results in a timely manner.

The aim of this thesis is to develop an automatic vessel segmentation method in fundus images. Over the years, several works have been proposed to approach vessel segmentation. This thesis focuses on deep learning methodologies, especially on convolutional neural networks. These models have gained particular prominence in recent years, due to their ability to automatically learn an increasingly complex hierarchy of features from the input data. This hints a paradigm shift according to which the focus is now on optimizing architectures rather than on the traditional feature engineering phase, which is probably the most critical step in conventional methods.

The proposed method was evaluated on three publicly available databases, widely used in the retinal vessel segmentation area: DRIVE, STARE, and CHASE_DB1.

During the various stages of this work, several variants were studied in order to improve the results. The best model is based on a fully convolutional neural network with several scales. In addition, it uses rotation operations for data augmentation, and benefits from the information encoded by the stationary wavelet transform. In quantitative terms, the method achieved average accuracy of 0.9576, 0.9694, and 0.9653 and average area under the ROC curve of 0.9821, 0.9905, and 0.9855 on the DRIVE, STARE, and CHASE DB1 databases, respectively. It also appears to be robust to the training set and to the presence of pathological signs, which shows its potential for real-world applications.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

1D Unidimensional.

2D Bidimensional.

3D Tridimensional.

AMD *Age-Related Macular Degeneration*.

ANN *Artificial Neural Network*.

AUC *Area Under the ROC Curve*.

CAR *Cancer Associated Retinopathy*.

CHASE_DB1 *Child Heart and Health Study in England*.

CLAHE *Contrast-Limited Adaptive Histogram Equalization*.

CNN *Convolutional Neural Network*.

COSFIRE *Combination of Shifted Filter Responses*.

CRF *Conditional Random Field*.

CWT *Continuous Wavelet Transform*.

DoOG *Difference of Offset Gaussians*.

DRIVE *Digital Retinal Images for Vessel Extraction*.

DWT *Discrete Wavelet Transform*.

FA *Fluorescein Angiography*.

FC *Fully Connected*.

FCN *Fully Convolutional Neural Network*.

FOV *Field of View*.

GMM *Gaussian Mixture Model*.

GPU *Graphics Processing Unit*.

IDWT *Inverse Discrete Wavelet Transform*.

k-NN *k-Nearest Neighbor*.

MLP *Multilayer Perceptron*.

OCT *Optical Coherence Tomography*.

ReLU *Rectified Linear Unit*.

RF *Random Forest*.

RGB *Red, Green, Blue*.

RNN *Recurrent Neural Network*.

ROC *Receiver Operating Characteristic*.

SD-OCT *Spectral Domain Optical Coherence Tomography*.

SGD *Stochastic Gradient Descent*.

SLO *Scanning Laser Ophthalmoscopy*.

STARE *Structured Analysis of the Retina*.

SVM *Support Vector Machine*.

SWT *Stationary Wavelet Transform*.

VEGF *Vascular Endothelial Growth Factor*.

Capítulo 1

Introdução

Neste capítulo, apresenta-se a motivação e os principais objetivos associados ao trabalho desenvolvido nesta dissertação. Além disto, apontam-se também as contribuições que resultaram deste trabalho e clarifica-se a estrutura dos capítulos seguintes.

1.1 Motivação

A retina é uma membrana de tecido nervoso, disposta em camadas, que reveste o interior do olho e é responsável por converter a luz em sinais elétricos que são, posteriormente, processados pelo córtex visual do cérebro. Como este processo pressupõe que os raios luminosos se propaguem através do globo ocular para que a formação da imagem seja possível, as estruturas que intervêm no processo visual precisam ser transparentes. Por este motivo, a retina é vista como a única parte do corpo humano em que a corrente sanguínea pode ser observada de forma direta e não invasiva [1]. Mas, afortunadamente, este não é o único aspeto que torna a retina particularmente interessante em termos imagiológicos. A sua função faz com que este tecido seja muito ativo metabolicamente, sendo irrigado por duas redes sanguíneas distintas; assim, a árvore vascular da retina reúne todas as condições para que se possa avaliar, de forma indireta, a condição da circulação noutras zonas do organismo [1, 2].

Nas últimas décadas, a crescente investigação em imagens vasculares da retina, aliada à evolução dos meios imagiológicos, trouxe claras evidências que as alterações fisiológicas da árvore vascular estão relacionadas com uma grande variedade de doenças que afetam milhões de pessoas a nível mundial, como a diabetes e a hipertensão [2]. No entanto, a avaliação clínica da condição vascular retiniana exige que as imagens sejam manualmente segmentadas, tarefa que se revela cansativa, demorada e suscetível a erros, especialmente em ambiente de rastreio. A investigação desenvolvida por vários grupos a nível mundial visa reduzir a carga de trabalho dos especialistas e fornecer uma alternativa capaz de gerar resultados robustos e reprodutíveis, com aplicabilidade prática no dia a dia das unidades clínicas. As metodologias de segmentação automática de imagens da retina são, assim, vistas como ferramentas de diagnóstico e monitorização cada vez mais importantes e procuradas [1, 2].

1.2 Objetivos

O potencial da análise automática de imagens retinianas é, sem dúvida, muito vasto. Desde logo, este tipo de sistema poderá dotar as unidades clínicas duma maior capacidade para analisar grandes conjuntos de imagens, com reduções substanciais em termos de tempo e custos financeiros. Depois, metodologias de segmentação automática robustas abrem portas para a criação de novos sistemas de rastreio, que podem reformular por completo o diagnóstico e tratamento de diversas doenças.

O problema que se coloca é que, apesar do seu potencial, desenvolver uma metodologia de segmentação automática configura uma tarefa complexa. Em primeira instância, é preciso lidar com as variações abruptas nos atributos (tamanho, forma, níveis de intensidade) e arranjo (ramificação, cruzamento) dos vasos. Depois, as imagens vasculares da retina possuem, normalmente, baixa qualidade [1, 3]. Quando se pensa em imagens com sinais patológicos, surge um campo ainda mais amplo de fatores a endereçar, sendo crucial desenvolver modelos robustos e confiáveis nestas condições.

Neste trabalho, procura-se desenvolver um método automático capaz de segmentar eficazmente imagens de diferentes bases de dados, mantendo a robustez e atuando num período de tempo razoável.

O desenvolvimento desta metodologia combina diversas áreas de conhecimento, com destaque para o processamento de imagem, para as áreas de *machine learning* em geral e de *deep learning* em particular e, claro, para a área das ciências da saúde.

1.3 Contribuições

Ao longo desta dissertação foram desenvolvidos alguns métodos que, no conhecimento do autor, podem ser considerados contribuições originais, nomeadamente:

- O estudo de diferentes estratégias de aumento artificial de dados que se revelam cruciais em métodos de *deep learning* e que têm particular importância em imagem médica, devido à dificuldade para obter novas imagens e, principalmente, novas anotações manuais.
  - Base do artigo da quinta conferência portuguesa IEEE em Bioengenharia [4].
- Um método de segmentação automática de vasos retinianos, baseado numa rede neuronal convolucional, onde se destaca a fusão de informação em diferentes escalas, para lidar com a natureza multi-escala da vasculatura.
- Uma nova estratégia de segmentação que assenta na apresentação de vários exemplos à rede e na obtenção da segmentação única através do cálculo da média entre as segmentações de cada exemplo.
- A combinação entre *features* geradas pelo próprio classificador e *features* obtidas manualmente, para avaliar os efeitos da inclusão de conhecimento de domínio em métodos de *deep learning*.
  - Pontos principais do artigo submetido ao jornal *IEEE Transactions on Biomedical Engineering*, que se encontra submetido e sob revisão.

1.4 Estrutura da Dissertação

Os restantes capítulos desta dissertação estão organizados da seguinte forma. No capítulo 2, faz-se uma contextualização clínica que começa por se focar no processo visual humano e evolui, depois, para a descrição dos vários meios imagiológicos da retina e para as principais doenças que se manifestam através da árvore vascular. No capítulo 3, analisam-se alguns dos principais trabalhos de segmentação automática de vasos retinianos do estado da arte. No capítulo 4, introduzem-se os conceitos teóricos associados aos métodos de *deep learning*, com especial enfâse nas redes neuronais convolucionais e no treino destes modelos. No capítulo 5, descrevem-se as diversas fases da metodologia de segmentação automática proposta. No capítulo 6, reportam-se os principais resultados, começando pela validação do modelo base e passando depois à comparação do melhor modelo com outros métodos do estado da arte. Por último, no capítulo 7 destacam-se as principais conclusões do trabalho desenvolvido e sugerem-se possíveis áreas de trabalho futuro.

Capítulo 2

Contexto Clínico

A visão é, provavelmente, o sentido mais influente na nossa forma de percecionar o ambiente que nos rodeia e de reagir aos seus estímulos. A maioria dos sinais que recolhemos são processados através dos nossos olhos e, por isso, dependemos da visão mais do que de qualquer outro sentido, sendo o olho o nosso órgão mais sensível [5]. Cada estrutura ocular contribui de forma específica para o processo visual e a harmonia estrutural é crucial para que o olho possa desempenhar uma vasta gama de funções desde a perceção de forma, tamanho ou cor até à captura de distância ou profundidade [5, 6].

Neste capítulo, fornece-se uma descrição simplificada do processo visual humano, enquanto se caracterizam algumas das mais importantes estruturas que nele intervêm – em especial, a retina. A análise foca-se, depois, nos principais meios imagiológicos usados para observar a retina e nas mais relevantes patologias que se manifestam através dela.

2.1 Fisiologia da Visão Humana

O sistema visual humano é composto por duas partes: o olho e parte do cérebro. O cérebro executa todo o processamento de imagem, enquanto o olho atua como o equivalente biológico de uma câmera [5]. Desde que a luz atinge o olho até que os sinais nervosos sejam processados pelo cérebro, várias etapas se sucedem. Uma das mais relevantes ocorre num grupo especial de células da retina: os fotorrecetores. Assim, torna-se oportuno dividir o processo visual nas duas fases seguintes.

2.1.1 Da Córnea aos Fotorrecetores

A figura 2.1a representa uma secção transversal do olho humano. Quando um raio de luz atinge o olho, passa sucessivamente pela córnea, humor aquoso, pupila, cristalino e humor vítreo, até convergir, finalmente, na retina [5]. Quase toda a superfície do olho é revestida por uma camada rígida e branca: a esclera. A córnea é a camada transparente, que cobre a restante área, através da qual a luz entra e é refratada, como numa lente convexa. Os raios de luz passam, depois, por uma câmara que contém humor aquoso, um líquido incolor que nutre a córnea e regula a pressão intra-ocular. De seguida, os raios progridem até ao cristalino através da pupila. A abertura da pupila é regulada pela íris, que atua como

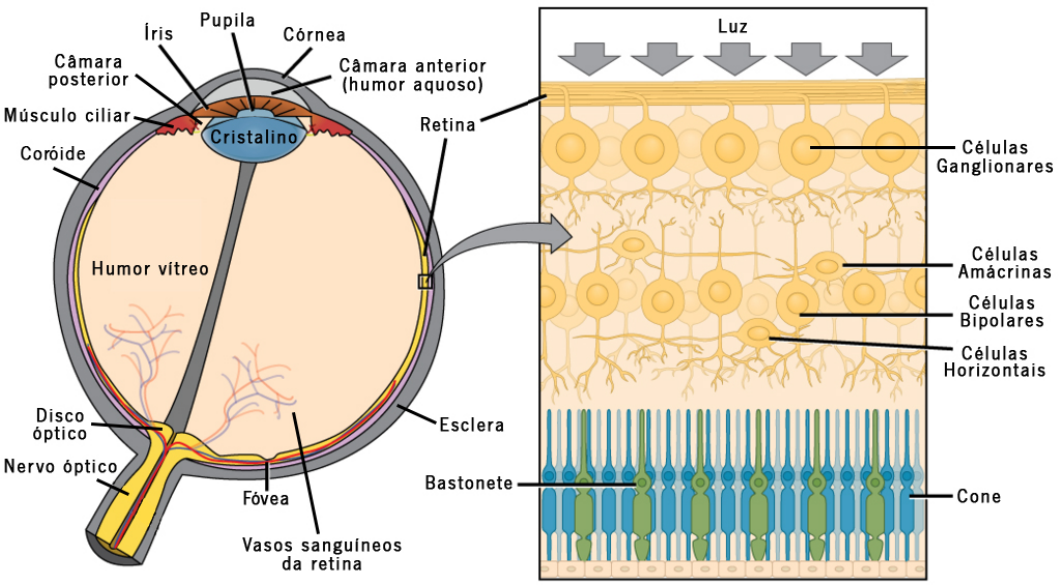


Figura 2.1: Anatomia do olho humano: (a) Secção transversal; (b) Detalhe da retina. Adaptado de [7].

um diafragma fotográfico. Num ambiente escuro, a pupila dilata, captando tanta luz quanto possível; num ambiente luminoso, ela retrai, evitando lesões devido ao excesso de luminosidade. A principal função do cristalino é focar a luz que nele incide, fazendo-a convergir. A potência focal é regulada pelos músculos ciliares, que o tornam mais ou menos espesso perante a focagem de objetos a pequenas ou grandes distâncias, respetivamente. Este processo – chamado de acomodação – é semelhante à focagem fotográfica. Até atingirem a retina, os raios propagam-se no interior do olho onde abunda uma massa transparente e gelatinosa que minimiza a dispersão e atenuação da luz: o humor vítreo [5].

A retina é uma membrana de tecido nervoso, disposta em camadas (figura 2.1b), que cobre a parte posterior do olho e que, na prática, opera como uma extensão cerebral [5]. As suas exigências metabólicas não são uniformes e, por isso, ela é irrigada por duas redes sanguíneas distintas: a primeira – visível nas imagens de fundo – dá resposta a cerca de um terço das necessidades, alimentando a parte interna; a segunda – geralmente não detetável – supre a restante porção, irrigando a parte externa e a coróide. A retina é composta por dois tipos de células fotorrecetoras: os bastonetes e os cones [5, 6].

2.1.2 Dos Fotorrecetores ao Córtex Visual

Os bastonetes são facilmente ativados por um número reduzido de fotões, enquanto os cones são menos sensíveis à luz e estão desenhados para distinguir entre cores. Em ambientes escuros, só os primeiros se encontram ativos. Por isso, é impossível distinguir cores durante a noite (visão escotópica). Em ambientes luminosos, os segundos despertam. Com eles, regressam as diversas cores (visão fotópica). A visão central forma-se na mácula, onde se concentra a maioria dos cones. A fóvea consiste numa depressão no centro dessa região, onde convergem os raios responsáveis pela acuidade visual[1] máxima. Por sua vez, os bastonetes distribuem-se pela periferia e contribuem para a visão periférica [5].

[1] Aptidão do olho para distinguir entre dois pontos muito próximos.

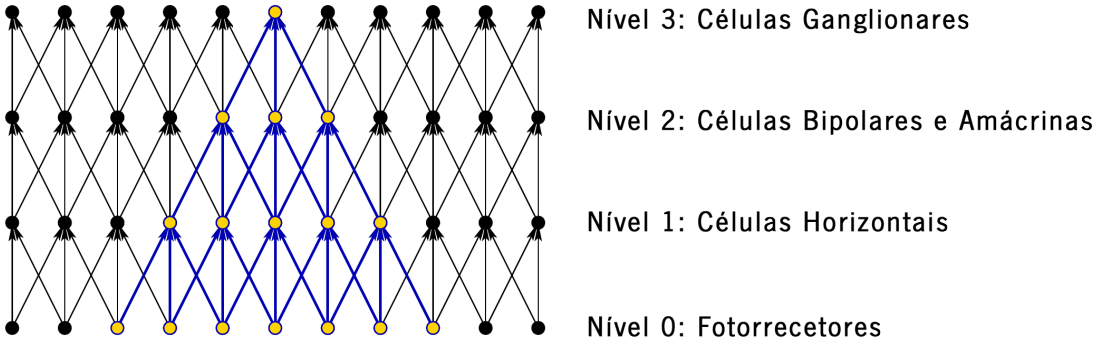


Figura 2.2: Quatro níveis de uma hierarquia neuronal. Cada círculo representa uma célula e as setas indicam o fluxo dos impulsos nervosos. Neste esquema, extremamente simplificado, cada célula liga-se apenas a três células do nível seguinte. A azul destacam-se todas as células que influenciam a saída da célula do último nível e que constituem o seu *receptive field*. Adaptado de [8].

As células fotorrecetoras operam como transdutores que convertem estímulos luminosos em sinais elétricos ou impulsos nervosos. Estes passam depois pelas células horizontais, bipolares e amácrinas até atingirem as células ganglionares que realizam operações de filtragem simples baseadas em variações espaciais, temporais ou espectrais dos sinais enviados pelos fotorrecetores [5, 6]. Note-se que, na figura 2.1b, a luz parece entrar na direção errada; isto é, passa primeiro pelas células nervosas antes de atingir as células fotorrecetoras. Isto só é possível porque as primeiras são transparentes. Ainda assim, este arranjo impede que as células ganglionares se liguem diretamente ao nervo óptico, que é responsável por enviar os sinais ao cérebro. Esta ligação estabelece-se, externamente, através de uma abertura na retina: o disco óptico. Como nessa região não existem fotorrecetores, gera-se um ponto cego ou *blindspot* [5].

Uma vez no nervo óptico, os sinais são propagados, de forma hierárquica, até ao córtex visual, num processo homólogo ao que leva a luz até ao sensor de uma máquina fotográfica. A figura 2.2 ilustra um modelo simplificado. Neurónios de níveis mais baixos estabelecem comparações simples entre sinais provenientes de fotorrecetores vizinhos. À medida que se atingem níveis mais altos, cada neurónio é influenciado por um número cada vez maior de fotorrecetores. Este conceito – vulgarmente conhecido como *receptive field* – será resgatado no capítulo 4. Eventualmente, quando os impulsos nervosos atingirem níveis neuronais mais elevados (não representados na figura), a informação alocada na memória poderá ser fundida com aquela que provém dos fotorrecetores. O complexo processamento cerebral permitir-nos-á, então, reconhecer uma face ou avaliar o tamanho de um objeto [6].

2.2 Métodos Imagiológicos da Retina

Entre todas as especialidades médicas, a oftalmologia é, talvez, a mais orientada para a tecnologia. Os recentes avanços na óptica clínica e na imagem médica têm-se revelado fundamentais para melhorar a compreensão e tratamento de diversas doenças. Numa clínica de oftalmologia, os dispositivos imagiológicos da retina continuam a ser, essencialmente, usados no diagnóstico e monitorização de doenças oculares [9, 10]. No entanto, o nível de detalhe com que o olho pode ser observado, de forma não invasiva, abre novas possibilidades de investigação em doenças crónicas e sistémicas [2].

De seguida, analisam-se as principais modalidades de imagem para a retina.

2.2.1 Retinografia

A retinografia gera uma imagem bidimensional (2D) da superfície interna tridimensional (3D) do olho (figura 2.3a). É realizada através de um sistema constituído por um microscópio especializado de baixa potência e uma câmera anexa [9]. O paciente é colocado com o queixo em repouso e a testa em contacto com uma barra, enquanto o operador foca e alinha a câmera antes de disparar o *flash* e criar a imagem. Esta imagem é uma versão ampliada do fundo retiniano com ângulos de visão típicos de 30°, 45°, ou 60° e com uma ampliação que, normalmente, ronda os 32.5, mas que está dependente da óptica do sistema. Algumas modificações a estes parâmetros podem ser introduzidas através de lentes auxiliares. Por exemplo, uma lente de 15° fornece uma ampliação maior, enquanto uma lente de 140° capta mais área. Um campo de visão (FOV[2]) maior pode ser atingido através da composição de múltiplas imagens adquiridas em diferentes pontos de fixação [2, 9]. Além disso, imagens de maior qualidade podem ser obtidas através da dilatação da pupila, de antemão, com gotas midriáticas. As resoluções atuais rondam os 3000×3000 pixeis [2].

Normalmente, a retina é examinada a cores através de iluminação com luz branca. No entanto, a luz pode ser filtrada para remover determinados componentes vermelhos, criando uma imagem *red-free* com contraste melhorado dos vasos sanguíneos retinianos e de outras estruturas [2]. Alternativamente, na angiografia fluoresceínica (FA[3]), o indivíduo recebe uma injeção intravenosa de um corante fluorescente, enquanto a retina é iluminada a um comprimento de onda específico para que se possam distinguir os locais onde o corante está presente. Isto permite gerar uma imagem de alto contraste e identificar áreas de dano onde o corante escapa para o tecido circundante. Paralelamente, a dinâmica do fluxo e as patologias associadas podem ser reveladas através da captura de uma sequência cronometrada de imagens que ilustre a progressão do corante a nível vascular (figuras 2.3b e 2.3c) [2, 9, 10].

As imagens de fundo retiniano são amplamente usadas na triagem de diversas doenças, como a retinopatia ou o glaucoma; estão validadas clinicamente como ferramenta de diagnóstico, identificando estados pré-sintomáticos onde ações de tratamento atempadas podem proteger a visão [2, 9, 10].

2.2.2 Oftalmoscopia de Exploração a Laser

Na oftalmoscopia de exploração a *laser* (SLO[4]), como na retinografia, gera-se uma imagem 2D da superfície interna do olho (figura 2.3d). Porém, o *flash* de luz branca é substituído por um *laser* que varre a superfície 3D da retina [9]. Trata-se de uma técnica de imagem confocal; ou seja, a retina é varrida, ponto a ponto, por um raio *laser* focado e a luz refletida é capturada através de uma pequena abertura: o *pinhole*. Assim, suprime-se a luz que é refletida para fora do plano focal, que poderia afetar a qualidade da imagem. Em simultâneo, espelhos de varredura são usados para direcionar o *laser* para regiões inacessíveis a uma câmera de fundo [9]. As imagens obtidas são, tipicamente, retângulares devido ao padrão de varredura do *laser* e o ângulo de visão pode variar, de acordo com o instrumento, desde valores semelhantes aos das câmeras de fundo, até 200° no chamado *ultra-widefield* [2, 9].

[2] Do inglês: *Field of View*.
[3] Do inglês: *Fluorescein Angiography*.
[4] Do inglês: *Scanning Laser Ophthalmoscopy*.

Ao contrário da retinografia, a SLO não captura imagens a cores, porque apenas um ou dois comprimentos de onda são usados. No entanto, a iluminação a *laser* combinada com o sistema óptico confocal produz imagens de alto contraste e de detalhe fino. As resoluções atuais rondam os 3000×2800 pixeis [2]. Além disso, a estreita faixa de comprimentos de onda proporciona uma excitação de fluorescência mais eficiente do que a iluminação de *flash* usada na câmera de fundo. O uso combinado de SLO e FA permite ainda medir a velocidade do fluxo capilar; estas medições fornecem informações importantes sobre o estado da circulação vascular [2, 9, 10].

Embora vários dispositivos de SLO adquiram uma área maior e captem o fundo periférico, a resolução das imagens é, como já se fez notar, geralmente menor do que numa câmera de fundo [2, 9, 10]. O custo e a complexidade dos dispositivos levam, também, a que a SLO esteja, na maioria das vezes, confinada a clínicas de oftalmologia [2]. No entanto, os fabricantes reconhecem que a facilidade de uso e a acessibilidade para profissionais de outras áreas fazem dela uma alternativa real às câmeras de fundo, que têm capacidades limitadas em termos de FOV. Além disto, a SLO pode ser mais confortável para o paciente, visto que várias imagens podem ser capturadas rapidamente e sem cansaço [2, 9, 10].

2.2.3 Tomografia de Coerência Óptica

A tomografia de coerência óptica (OCT[5]) é um método não-invasivo capaz de gerar imagens transversais, *in vivo*, das estruturas internas da retina. Consiste numa técnica de varrimento óptico que emprega luz do infravermelho próximo e que pode ser entendida como "ultra-som óptico" no que diz respeito ao procedimento e à própria interpretação das imagens [2, 9, 10]; embora com uma profundidade limitada a 2 mm de tecido [2]. O procedimento consiste em direcionar um feixe óptico para o tecido-alvo e analisar os sinais refletidos por interferometria. Como é habitual nesta técnica, o feixe de varrimento é dividido, através de um divisor de feixe, entre o tecido-alvo e um espelho de referência; ambos são depois refletidos, recombinados no acoplador e, posteriormente, direcionados para o detetor [2, 9, 10].

Quando a distância para o espelho de referência é igual à distância para o alvo, ocorre interferência que é usada para inferir a profundidade do tecido onde a luz foi refletida. Na OCT de tempo convencional, o espelho de referência é colocado a diferentes distâncias da fonte, permitindo identificar tecidos a diferentes profundidades. Este movimento mecânico, relativamente lento, limita a quantidade de dados adquirida e a qualidade das imagens. Na SD-OCT[6], uma variante mais recente, a imagem de profundidade é calculada analisando o sinal de interferência com base no comprimento de onda da luz. A necessidade de um espelho de referência em movimento é eliminada, o que melhora a velocidade de aquisição e a resolução [2, 9, 10].

Cada vez mais, a OCT é combinada com outras técnicas como a retinografia ou a SLO para criar imagens multimodais (figuras 2.3e e 2.3f). Estas, além de mostrarem a superfície da retina, permitem ao operador localizar a secção transversal adquirida pela OCT. Assim, há um registo de localização preciso e garante-se que a mesma área da retina é digitalizada em aquisições posteriores [2, 9, 10].

[5] Do inglês: *Optical Coherence Tomography*.
[6] Do inglês: *Spectral Domain Optical Coherence Tomography*.

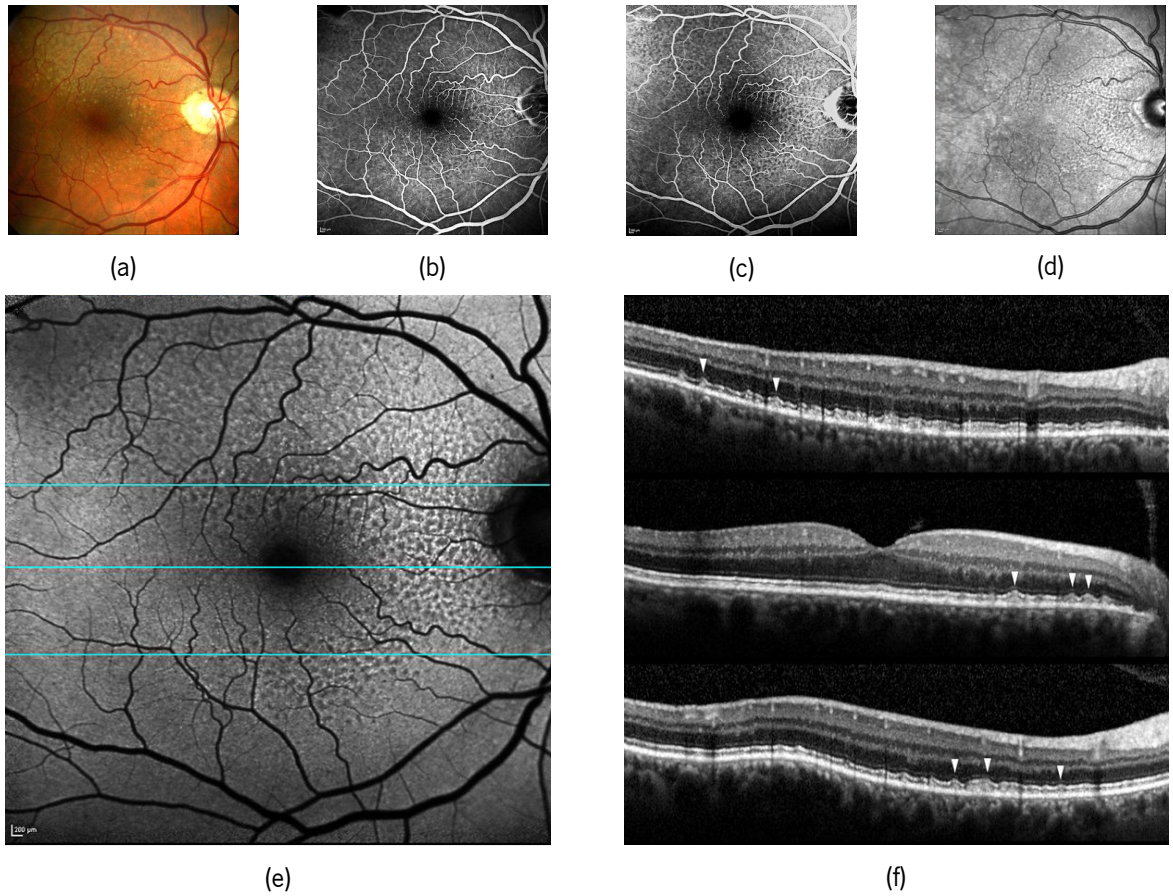


Figura 2.3: Diferentes modalidades de imagem da retina: (a) Retinografia; (b) FA em fase inicial; (c) FA em fase tardia; (d) SLO; (e) SLO com marcações relativas aos cortes transversais de OCT; (f) Cortes de OCT correspondentes. Adaptado de [11].

2.3 A Retina como Biomarcador Ocular e Sistémico

O equilíbrio funcional e estrutural do olho pode ser afetado por diversas doenças. Além de patologias oculares, capazes de provocar lesões e mudanças estruturais *per se*, também várias patologias sistémicas podem induzir, colateralmente, alterações nas estruturas oculares, resultando numa vasta gama de sinais clínicos. Isto, aliado aos recentes avanços imagiológicos que permitem um rastreio mais eficaz e uma compreensão clínica mais profunda, torna possível interpretar estes sinais como biomarcadores, que podem ser úteis para acelerar o diagnóstico e acompanhar a evolução de várias doenças [2, 12].

No caso particular da retina, o desenvolvimento vascular saudável tem um papel crucial na eficiência da irrigação sanguínea. Desvios a este estado de harmonia revelam-se, também aqui, através de um conjunto de sinais que importa saber como identificar. Quando se observam imagens de fundo – que são o alvo de estudo desta dissertação –, especial atenção deve ser dada a fatores como o estado do nervo óptico, a geometria e o ângulo de bifurcação dos vasos, a tortuosidade[7], os diâmetros arteriolares e venulares[8] e a condição da mácula (figura 2.4). Regra geral, padrões geométricos anormais reduzem a eficiência do transporte metabólico, enquanto diâmetros adequados e uma arquitetura de ramificação ótima são cruciais para a saúde vascular [2].

[7] Frequência com que os vasos exibem curvas mais ou menos acentuadas.
[8] A distinção entre arteríolas e vénulas é importante, porque estas são afetadas de forma distinta por várias doenças.

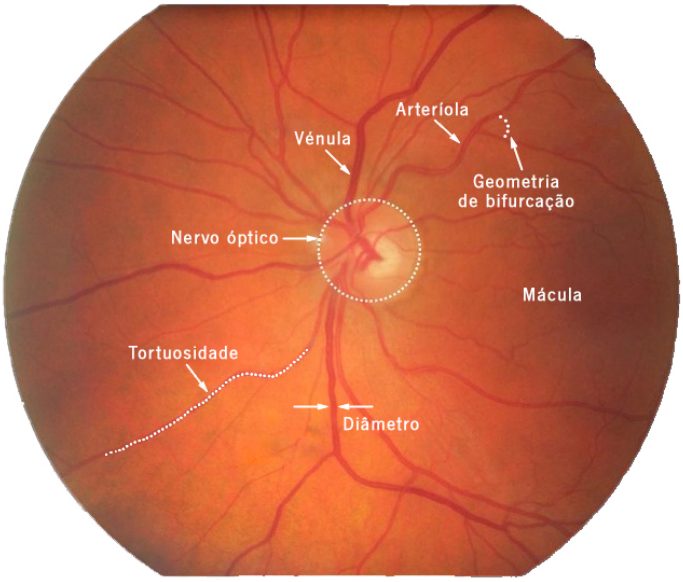


Figura 2.4: Focos de interesse numa imagem de fundo retiniano. Adaptado de [2].

De seguida, descrevem-se algumas das doenças mais relevantes que se manifestam através da retina e identificam-se os principais sinais clínicos associados a cada uma delas.

2.3.1 Degeneração Macular Relacionada à Idade

A degeneração macular relacionada à idade (AMD[9]) é a principal causa de cegueira em pessoas com idade superior a 50 anos, no mundo ocidental [13]. Trata-se de uma doença ocular que afeta a mácula e que, por isso, tem efeitos sobre a visão central. Existem duas formas de AMD: a "seca", ou não-neovascular, representa a fase inicial da doença e é a mais comum; a "húmida", ou neovascular, tem efeitos mais graves e engloba cerca de 10 % dos casos [12, 13].

A principal causa da AMD é a acumulação de proteínas e gordura (drusas) numa fina camada celular que se encontra sob a retina: a membrana de Bruch. Na AMD seca, a presença destes depósitos amarelados é a característica mais visível (figura 2.5a). Nesta fase, a regressão visual é suficientemente lenta para que a visão se mantenha inalterada durante longos períodos de tempo. Tipicamente, as drusas não resultam em cegueira por si sós; interferem em atividades quotidianas como a condução, a escrita ou qualquer outra onde a visão de detalhe seja importante. No entanto, o crescimento e aumento do número destes depósitos pode, eventualmente, agravar a doença. Na AMD húmida, assiste-se ao aparecimento de vasos patológicos, geralmente provenientes da coróide, na região da mácula. Estas neovascularizações causam uma diminuição repentina, muitas vezes bastante acentuada, da acuidade visual, além de distorções devido ao vazamento de fluidos e sangue (figura 2.5b). Em fases avançadas, o paciente poderá sofrer uma redução irreversível da visão, parcial ou total [12, 13].

Entre os principais fatores de risco da AMD encontram-se – para além da idade – a herança genética, a hipertensão e a etnia (caucasianos parecem ser mais propensos que negros ou latinos). Atenção especial deve ser dada aos principais fatores de índole comportamental: o tabagismo e a obesidade [13].

[9] Do inglês: *Age-related Macular Degeneration*.

Até à data, não existe cura para a AMD [12, 13]. Os tratamentos limitam-se a minimizar, ou retardar, os efeitos da doença. Uma dieta saudável, rica em ómega 3, tem sido associada a um menor risco de incidência. Em pacientes com AMD seca, a ingestão de uma combinação de antioxidantes e zinco parece ser capaz de reduzir o risco de progressão para a AMD húmida. Já na AMD húmida, o panorama era, até à última década, pouco animador [13]. Porém, avanços recentes trouxeram novas perspetivas. Pensa-se que os vasos sanguíneos patológicos, característicos desta fase, surgem devido a uma substância química – o fator de crescimento do endotélio vascular (VEGF[10]) –, que está naturalmente presente na retina, mas que surge em maiores quantidades na presença da doença; as recém descobertas injeções intravítreas anti-VEGF atuam bloqueando a ação deste composto [12].

2.3.2 Glaucoma

O glaucoma é a segunda causa mais grave de cegueira em todo o mundo, só superado pelas cataratas [14]. É uma doença que afeta o nervo óptico, com consequências, sobretudo, na visão periférica. Entre as várias formas de glaucoma, podem destacar-se duas: a de ângulo aberto é gradual, silenciosa e mais frequente; a de ângulo fechado manifesta-se de forma súbita, provoca vários sintomas – fortes dores de cabeça, vermelhidão ocular ou visão turva –, mas é menos comum [12, 14].

O grande agente municiador do glaucoma é o desequilíbrio entre as taxas de secreção e drenagem de humor aquoso, no interior do olho. Em condições normais, a produção – pelo corpo ciliar – é compensada pelo escoamento – através da malha trabecular e do fluxo uveoscleral – e o líquido flui livremente da câmara posterior para a anterior. Na presença da doença, porém, as vias de escoamento são obstruídas e a taxa de drenagem é afetada. Esta obstrução pode ser suficientemente gradual para que, quando examinados, os canais não pareçam comprometidos – "canal aberto" – ou ser repentina ao ponto de gerar um bloqueio visível – "canal fechado". Em ambos os casos, o humor aquoso não é expelido à taxa que deveria, mas continua a ser produzido ao mesmo ritmo. Assim, a pressão intra-ocular aumenta e, num ponto crítico, ocorre a lesão do nervo óptico (figura 2.5c). Como a forma mais comum do glaucoma é totalmente assintomática até um estado avançado e, além disso, as alterações na visão periférica são menos notórias, estima-se que grande parte dos afetados não esteja ciente da sua condição [12, 14].

Entre as características que têm sido associadas a maiores taxas de incidência destacam-se a idade, os antecedentes familiares, o sexo feminino, a etnia (negros e asiáticos parecem ser mais propensos) e o tamanho dos olhos (volumes oculares maiores oferecem mais resistência ao aumento da pressão). Há, ainda, indícios que o uso de corticoesteróides[11] favorece a evolução da doença [12, 14].

Apesar das lesões associadas ao glaucoma serem irreparáveis e de não se conhecer cura, existem diversos tratamentos que permitem retardar a perda visual. Os colírios são os mais comuns e, por norma, os primeiros a serem utilizados; tratam-se de medicamentos, de uso tópico, que têm um efeito hipotensor e que podem ser classificados em diferentes grupos consoante a sua composição. Se o tratamento inicial se revelar ineficaz, parte-se normalmente para uma intervenção cirúrgica a *laser*: a trabeculoplastia – usada em glaucomas de ângulo aberto – consiste na criação de pequenos furos na malha trabecular para

[10] Do inglês: *Vascular Endothelial Growth Factor*.

[11] Anti-inflamatórios usados no tratamento de doenças como a artrite reumatóide ou a asma.

repor a taxa de escoamento; a iridotomia – indicada para glaucomas de ângulo fechado – visa criar um pertuito, através da íris, entre as câmaras anterior e posterior para aliviar a diferença de pressão [12].

2.3.3 Diabetes

A diabetes é a principal causa de cegueira entre pessoas em idade profissionalmente ativa, no mundo ocidental [15]. Trata-se de uma doença metabólica na qual se verificam elevados níveis de glicose no sangue (hiperglicemia) durante longos períodos de tempo. Este panorama é causado por baixos níveis sanguíneos de insulina, uma hormona que reduz a glicemia ao promover a entrada de glicose nas células. Consoante as causas deste défice, a diabetes pode classificar-se em dois tipos: no tipo I – mais comum e, normalmente, associado a fatores genéticos – há uma destruição auto-imune das células produtoras de insulina; no tipo II – menos frequente e, na maioria das vezes, provocado por má alimentação ou falta de atividade física – existe uma resistência anormal a ela por parte do organismo [12].

Uma das complicações mais comuns da diabetes – tanto de tipo I como de tipo II – ocorre a nível ocular. A retinopatia diabética é uma patologia causada por uma microangiopatia[12] induzida pela glicemia. A nível capilar, há proliferação e agregação de células endoteliais com espessamento membranar, redução de calibre e alterações de consistência, elasticidade e permeabilidade; isto provoca uma diminuição na irrigação sanguínea e gera hipóxia[13]. Na fase inicial da doença – retinopatia não proliferativa – são visíveis várias alterações vasculares como microaneurismas (figura 2.5d), hemorragias (figura 2.5e), exsudados (figura 2.5f) e manchas algodonosas (figura 2.5g). Com o acentuar do bloqueio capilar, atinge-se um estado de isquemia[14], que induz a libertação de fatores angiogénicos[15] como o VEGF. Na fase mais avançada da doença – retinopatia proliferativa – assiste-se, assim, ao surgimento de novos vasos em torno das zonas ocluídas, numa tentativa do metabolismo para restaurar a perfusão sanguínea. Estes neovasos (figura 2.5h) são frágeis e, muitas vezes, provocam sangramentos na cavidade vítrea, levando a uma diminuição repentina e acentuada da acuidade visual. Se não controlados, os vasos podem evoluir para cicatrizes fibrovasculares e posterior descolamento da retina (figura 2.5i). Em simultâneo, com o aumento do vazamento vascular, pode ocorrer o espessamento da mácula. Este é o principal indicador de edema macular, que – apesar de não ser visível em imagens de fundo – é a principal causa de cegueira em pacientes com retinopatia diabética [12, 15].

O fator de risco mais preponderante da retinopatia diabética continua a ser a duração da diabetes. Pacientes diabéticos há mais de 10 anos parecem ser muito mais propensos. Além disto, há indícios de maior risco nos períodos de gravidez e pós-puberdade. Em termos de fatores modificáveis, a obesidade é o maior destaque [12].

À semelhança da diabetes, a retinopatia diabética não tem cura. O controlo regular da glicemia continua a ser visto como o tratamento mais eficaz, ainda que existam recomendações diferenciadas consoante o estado evolutivo da doença. Em pacientes com retinopatia não-proliferativa, as intervenções terapêuticas não são, geralmente, necessárias; em alternativa, exigem-se exames de controlo regulares.

[12] Problema que afeta os vasos finos.
[13] Baixo teor de oxigénio.
[14] Presença de um fluxo de sangue e oxigénio inadequado às necessidades do organismo.
[15] Substâncias químicas que induzem a formação de novos vasos.

Logo que neovascularizações sejam detetadas, sinalizando a fase proliferativa da doença, um tratamento a *laser* – fotocoagulação – é aconselhado. O objetivo desta terapia é induzir a involução[16] dos novos vasos, reduzindo o risco de hemorragia vítrea e de desprendimento traumático da retina. Nos casos em que a doença não pode ser controlada através da terapia a *laser*, recorre-se, geralmente, a uma intervenção cirúrgica – vitrectomia – que permite extraír parte do humor vítreo contaminado com sangue [12, 15].

2.3.4 Hipertensão

A hipertensão é uma doença cardiovascular crónica na qual se verificam elevados níveis de pressão arterial de forma persistente e que afeta mais de um terço da população mundial [16, 17].

O efeito mais notório da hipertensão no olho é a retinopatia hipertensiva, uma patologia que engloba uma série de mudanças vasculares na retina. Numa fase inicial, a elevação da pressão arterial leva à constrição das arteríolas retinianas e subsequente estreitamento, numa tentativa de controlar o volume de sangue recebido pelo leito capilar. Se a pressão arterial permanecer cronicamente elevada, dá-se a compressão das vénulas, por ação das mudanças estruturais nas arteríolas, com o surgimento de entalhes arteriovenosos (figura 2.5j). Numa fase mais grave, há progressão para um estado "exsudado" em que são observadas hemorragias e manchas algodonosas; e, em última instância, para um estado "maligno", com papiledema (figura 2.5k) e edema macular, que pode levar à cegueira [16, 17].

De forma homóloga à relação que se estabelece entre a retinopatia diabética e a diabetes, os principais fatores de risco da retinopatia hipertensiva relacionam-se com as causas da hipertensão. Entre elas, incluem-se o consumo excessivo de sal, a obesidade, o tabagismo, o alcoolismo, o histórico familiar, o stress e a etnia (negros parecem ser mais propensos) [16, 17].

Também neste caso, as lesões provocadas pela doença são irreversíveis e não se conhece cura. O tratamento mais eficaz continua a ser a monitorização cuidada dos níveis de pressão arterial. Em pessoas afetadas, as principais recomendações médicas passam por manter um estilo de vida saudável, que alie atividade física regular com alimentação adequada, e por exames de monitorização frequentes. Em certos casos, à semelhança do que ocorre na retinopatia diabética, pacientes em estado grave podem ser sujeitos a vitrectomias para minimizar os efeitos das hemorragias [16, 17].

2.3.5 Cancro

O cancro é a segunda causa mais grave de morte a nível mundial [18]. Trata-se de uma doença genética onde se observa uma proliferação celular anormal, com capacidade para invadir outras partes do corpo a partir de um foco inicial (metastização) [18].

O primeiro registo de perda de visão devido a um efeito remoto de cancro apareceu na década de setenta. Desde então, vários outros casos foram descritos e a retinopatia associada ao cancro (CAR[17]) foi devidamente identificada. A principal causa da CAR é a produção de auto-antigénios capazes de invadir a retina e provocar apoptose[18]. O alvo é, aparentemente, a recoverina, uma proteína dos fotorrecetores

[16] Regressão evolutiva.
[17] Do inglês: *Cancer Associated Retinopathy*.
[18] Morte celular.

que intervém na transdução da luz. O quadro clínico clássico é o de um paciente com perda visual, que progride ao longo de dias ou semanas, precedida ou acompanhada de fotopsias[19] frequentes. Na fase inicial, apenas se observam mudanças discretas como o revestimento de vasos periféricos. Depois, podem surgir constrições das arteríolas retinianas e alterações pigmentares (figura 2.5l). A CAR evolui progressivamente e afeta tanto a visão central como a visão periférica. Na grande maioria dos pacientes, o diagnóstico de tumor maligno ainda não está estabelecido quando os sintomas visuais surgem [12, 19].

O principal fator de risco da CAR prende-se com a herança genética, porque apesar da maioria das mutações cancerígenas ser de origem não hereditária as suas causas ainda não são conhecidas. O cancro causador mais comum é o carcinoma de células pequenas do pulmão; no entanto, outros tipos foram já implicados, incluindo o cancro endometrial, da mama e da próstata [12, 19].

Embora uma deteção prévia possa ser benéfica, os tratamentos são ainda limitados e o dano visual é, geralmente, irreparável. Ainda assim, pensa-se que os agentes imunossupressores[20] e a plasmaférese[21] podem abrandar, ou mesmo estabilizar, a perda visual [12, 19].

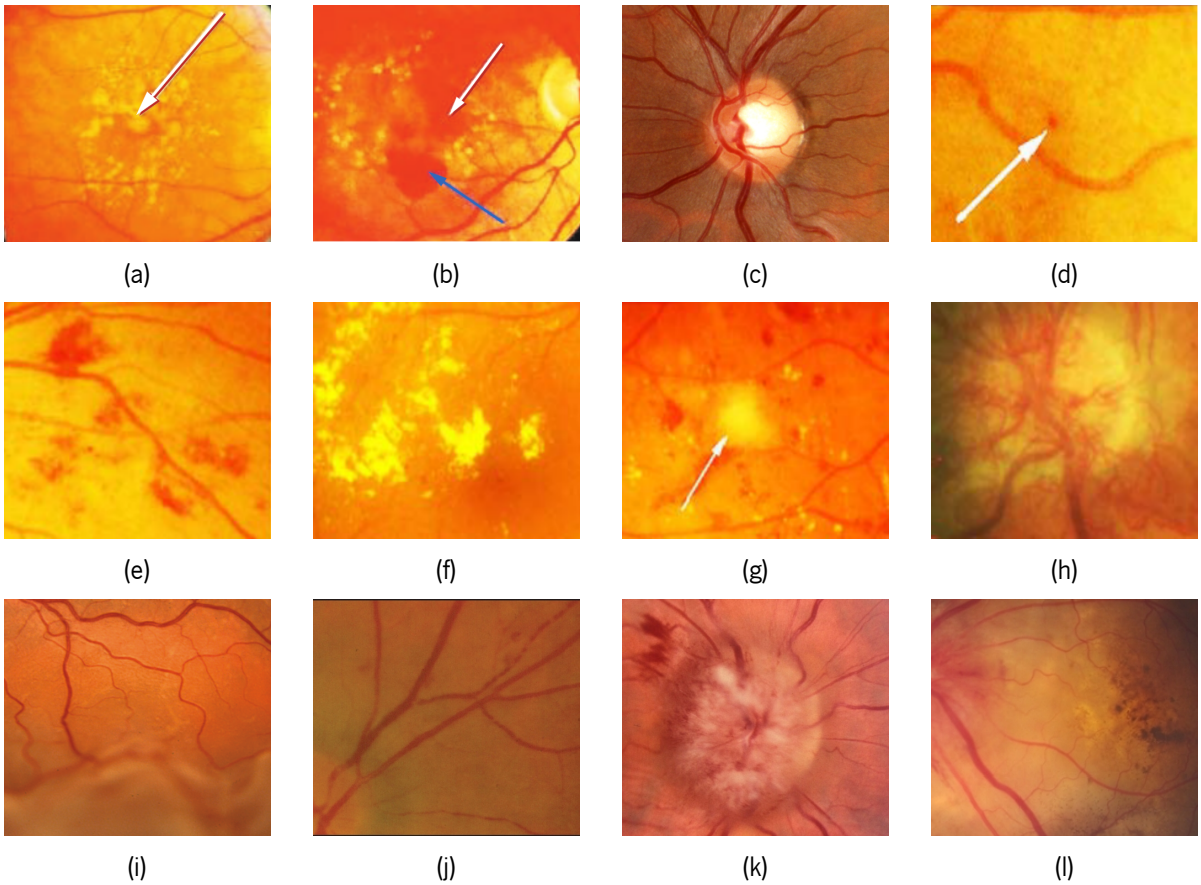


Figura 2.5: Manifestações patológicas em imagens de fundo retiniano: (a) Drusas; (b) Vazamentos sanguíneos; (c) Lesão do nervo óptico; (d) Microaneurismas; (e) Hemorragias; (f) Exsudados; (g) Manchas algodonosas; (h) Neovasos; (i) Descolamento da retina; (j) Entalhes arteriovenosos; (k) Papiledema; (l) Alterações pigmentares. Adaptado de [13, 20–24].

[19] Perceção de *flashes* de luz.
[20] Fármacos que visam reduzir a atividade do sistema imunitário.
[21] Técnica utilizada para remover elementos potencialmente patológicos do plasma.

2.4 Sumário

Desde que a luz entra no olho até que uma imagem seja formada no cérebro, várias estruturas oculares são chamadas a intervir. A retina é uma camada de tecido nervoso, formada por células fotorrecetoras, onde ocorre um dos eventos mais marcantes do processo visual: a conversão da luz em impulsos nervosos.

Devido à evolução tecnológica, a oftalmologia dispõe, hoje, de várias técnicas que permitem obter imagens da retina. Entre elas, destacam-se a retinografia, a SLO e a OCT. Cada técnica tem as suas particularidades e, por isso, é indispensável identificar a modalidade mais indicada para cada situação. No caso particular dos vasos sanguíneos da retina, as imagens de retinografia – também chamadas de imagens de fundo retiniano – são amplamente reconhecidas como uma das mais valiosas ferramentas de diagnóstico e monitorização. De facto, várias manifestações patológicas podem ser observadas através de imagens de fundo. Desde pequenas hemorragias ou microaneurismas até ocorrências mais graves como vazamentos sanguíneos, entalhes arteriovenosos ou neovasos, estes sinais podem ser provocados por doenças oculares como a AMD ou o glaucoma, mas podem também representar efeitos colaterais de doenças como a diabetes, a hipertensão ou até o cancro.

Embora nem todos os sinais patológicos observáveis nas imagens de fundo estejam diretamente relacionados com a árvore vascular da retina, a geometria de ramificação e a própria condição da vasculatura são valiosos indicadores do estado de saúde dos pacientes. Além disto, a segmentação automática constitui um passo intermédio no cálculo de diversos indicadores relacionados com as características morfológicas da árvore vascular, que se tornam particularmente úteis em programas de rastreio. Por isso, não surpreende que vários trabalhos de segmentação de vasos retinianos em imagens de fundo tenham sido propostos ao longo dos anos; este é, precisamente, o ponto de partida do próximo capítulo.

Capítulo 3

Estado da Arte

Para que se possam avaliar as características morfológicas dos vasos sanguíneos da retina, a sua segmentação é indispensável. No entanto, quando realizada manualmente, esta tarefa revela-se morosa e propensa a erros até para os especialistas mais experientes. Assim, é globalmente aceite pela comunidade científica que a segmentação automática dos vasos retinianos constitui, no caso de diversas doenças oftalmológicas e sistémicas, o primeiro passo para o desenvolvimento de um sistema de diagnóstico assistido por computador. Apesar da inclusão dos sistemas automáticos de segmentação ainda ser pouco comum na prática clínica, vários trabalhos têm apresentado resultados robustos e demonstrado potencial para reduzir os recursos humanos e a variabilidade associados à segmentação manual. Além disto, a segmentação automática, cada vez mais rápida e eficaz, tem ganho importância no planeamento cirúrgico e/ou terapêutico [1, 3, 25].

Neste capítulo, identificam-se as principais etapas de um sistema automático de segmentação de vasos retinianos. Para cada uma delas, são, depois, apresentados alguns dos trabalhos mais relevantes – com natural destaque para os aplicados em imagens de fundo –, bem como as vantagens e desvantagens associadas a cada abordagem. Especial ênfase é dada à etapa de classificação, onde os métodos de *machine learning* são divididos em três grupos: não supervisionados, supervisionados e de *deep learning*. Neste contexto, ressalva-se que, formalmente, todos os métodos do último grupo se enquadram num dos anteriores. No entanto, por conveniência, foi-lhes reservada uma secção particular.

3.1 Sistema Típico de Segmentação de Vasos Retinianos

Entre os algoritmos de segmentação de vasos retinianos, podem distinguir-se, tipicamente, três fases: pré-processamento, classificação e pós-processamento. A inclusão de uma etapa de pré- ou pós-processamento não garante, necessariamente, uma segmentação mais eficaz; tudo depende do método desenvolvido e da aplicação para a qual ele foi pensado. Dito isto, além de perceber o tipo de processamento a adotar, é também importante avaliar se a hipotética complexidade adicionada ao método, em termos de carga computacional e de tempo de execução, traz melhorias efetivas que a justifiquem.

De seguida, parte-se para a análise de cada uma das fases enunciadas.

3.1.1 Pré-processamento

A segmentação automática dos vasos retinianos é, desde logo, uma tarefa complexa devido às constantes variações nos atributos (comprimento, largura e tortuosidade) e arranjo (cruzamentos e bifurcações) da vasculatura. Porém, além destas complicações naturais, existem outros fatores como o ruído, a luminosidade não uniforme e o contraste reduzido dos vasos finos, que são herdados durante o processo de aquisição e que dificultam, ainda mais, a tarefa [1, 3, 25]. Assim, antes de recorrer a uma técnica de classificação, muitas vezes, são aplicadas algumas etapas de pré-processamento para suavizar estes problemas e melhorar a qualidade das imagens.

Grosso modo, podem destacar-se dois passos de pré-processamento mais comuns.

3.1.1.1 Correção do Contraste

As imagens de fundo retiniano exibem, frequentemente, variações no contraste do *background* devido à iluminação não uniforme durante o processo de aquisição. Entre as várias técnicas propostas na literatura para mitigar este problema, a equalização do histograma é, provavelmente, a mais popular [3]. No entanto, apresenta algumas limitações. Desde logo, só pode ser aplicada a imagens em níveis de cinza – ainda que na segmentação de imagens de fundo isto seja menos relevante porque, na maioria das vezes, apenas um dos canais (o verde) é considerado. Depois, é indiscriminada; ou seja, a mesma transformação é aplicada a toda a imagem, independentemente da distribuição de intensidades de cada região [25, 26]. Outra técnica bem conhecida é a correção gama que, na sua versão mais simples, consiste em mapear a intensidade de cada pixel segundo: $o = i^{\frac{1}{\gamma}}$, onde i e o são, respetivamente, as intensidades inicial e final e γ é o parâmetro que controla a transformação. Esta correção é apropriada para lidar com imagens muito escuras ou brilhantes; porém, os resultados parecem ser demasiado dependentes da imagem [25].

Um método idealizado para a correção de luminosidade e contraste em imagens retinianas foi proposto por Foracchia *et al.* [27]. Posto de forma simples, a ideia é calcular as variações de luminosidade e contraste com base apenas nos pixeis do *background* e, depois, usá-las para normalizar toda a imagem. A técnica pode ser aplicada a cada imagem, mas também permite atuar sobre um conjunto de imagens.

Fraz *et al.* [28] propuseram um método alternativo que se baseia em subtrair uma estimativa do *background*, obtida através de um filtro de média aritmética, da imagem original.

Um método cada vez mais usual é a equalização adaptativa do histograma com contraste limitado (CLAHE[1]) [26]. Nesta abordagem, a imagem é dividida em várias regiões que são normalizadas, individualmente, usando a equalização do histograma. Isto torna o algoritmo mais robusto perante regiões muito escuras ou claras. Além disto, a amplificação de cada região é limitada por um valor pré-definido (o chamado *clip limit* ou *clip level*), que permite suavizar o ruído em regiões homogéneas. Zhao *et al.* [29], Azzopardi *et al.* [30] e Yin *et al.* [31], por exemplo, adotaram esta técnica.

Finalmente, uma das técnicas mais recentes para a correção do contraste em imagens retinianas foi apresentada por Zhao *et al.* [32, 33]. Nesta proposta, cada imagem é modelada como sendo a multiplicação de dois componentes: a refletância e a iluminação. Segundo os autores, o componente de refletância pode ser visto como uma versão normalizada da imagem original.

[1] Do inglês: *Contrast-limited Adaptive Histogram Equalization*.

3.1.1.2 Realce dos Vasos

Uma imagem de fundo retiniano apresenta diversas estruturas anatómicas que diferem entre si em termos de tamanho, forma e orientação, como os vasos sanguíneos, o disco óptico e a mácula. O maior problema da segmentação de vasos é a complexidade das estruturas que podem estar próximas dos vasos finos e as regiões de bifurcação ou cruzamento. Por isso, muitas vezes, as estruturas vasculares são previamente realçadas.

A maioria dos filtros propostos na literatura para o realce dos vasos baseia-se no perfil de intensidades. Além dos convencionais *matched filters* [34, 35], encontram-se também filtros baseados em valores próprios [36], operadores lineares multi-escala [37], filtros gaussianos de segunda ordem [38], wavelets [39], filtros de Gabor [40] e filtros COSFIRE[2] [30, 41]. Cada tipologia visa realçar características distintas. No entanto, todas partilham do mesmo problema: a vulnerabilidade face à não homogeneidade das intensidades. Em alternativa, os filtros de informação de fase [32, 33] constituem uma técnica emergente capaz de aliviar os problemas enfrentados pelo grupo anterior. Estes filtros baseiam-se nas propriedades espectrais da imagem e parecem manter a consistência nas zonas de bifurcação e cruzamento [25].

Outra técnica bem conhecida é o uso de operações de morfologia matemática como a transformação *top-hat* [42, 43]. Neste caso, calcula-se uma estimativa do *background* próximo através de uma operação de abertura morfológica e esta é, depois, subtraída da imagem original.

Por fim, uma das técnicas propostas mais recentemente passa pela aplicação de filtros multi-wavelet. Wang *et al.* [44] apontam estes filtros, onde a tradicional wavelet mãe é substituída por um vetor de wavelets, como sendo particularmente eficazes para lidar com vasos tortuosos ou com áreas brilhantes localizadas (como lesões).

3.1.2 Classificação

A tabela 3.1 sumaria todos os trabalhos analisados ao longo desta secção. Como já se referiu, dependendo da forma como os seus parâmetros são obtidos, todos os métodos de segmentação de vasos propostos na literatura podem ser enquadrados em um de dois grupos: métodos supervisionados e não supervisionados. Os métodos de *deep learning* não são exceção, visto que mesmo entre eles se distinguem estratégias supervisionadas e não supervisionadas. Não obstante, dado o seu relevo atual – e o âmbito desta dissertação –, eles serão agrupados num terceiro grupo.

3.1.2.1 Métodos Não Supervisionados

Os métodos não supervisionados procuram estruturar um dado conjunto de dados, sem recorrer a anotações manuais (ou *labels*). Por outras palavras, o objetivo passa por distinguir os diferentes grupos que constituem o conjunto, com base nas características dos dados. Em geral, estes algoritmos podem ser interpretados como técnicas baseadas em regras e, com base nelas, dividem-se em *matched filtering*, morfologia matemática, *vessel tracking*, técnicas multi-escala e técnicas baseadas em modelos [3].

[2] Do inglês: *Combination of Shifted Filter Responses.*

3.1.2.1.1 Matched Filtering

As técnicas de *matched filtering* para a deteção dos vasos retinianos convolvem um filtro 2D com a imagem de fundo. O filtro é projetado para identificar uma determinada característica de interesse (*feature*), cuja posição e orientação são desconhecidas, com o resultado da operação a refletir a tendência para a *feature* codificada estar presente. A técnica assenta em três ideias: os vasos têm, geralmente, uma curvatura limitada e podem ser aproximados por pequenos segmentos lineares; o diâmetro dos vasos diminui à medida que estes se deslocam, radialmente, para fora do disco óptico; e o perfil de intensidades desses pequenos segmentos lineares pode ser aproximado através de uma curva gaussiana [3]. Apesar destes pressupostos serem razoáveis, o filtro responde de forma ótima nas regiões onde o desvio padrão dos segmentos vasculares é o mesmo da função gaussiana por ele especificada. Além disto, a presença de patologias aumenta o número de falsas respostas porque estas podem exibir os mesmos atributos locais que os vasos. Devido a estas limitações, na maioria das vezes, estes métodos só apresentam resultados razoáveis quando combinados com técnicas de processamento adicionais [3, 45].

Chaudhuri *et al.* [34] foram os primeiros a publicar um trabalho em segmentação de vasos da retina usando *matched filters*. A proposta baseia-se na modelação gaussiana do perfil de intensidades da vasculatura, através de um filtro linear 2D. O filtro é rodado várias vezes, com incrementos de 15°, para lidar com vasos em diferentes orientações. Para cada pixel, a orientação com a resposta mais elevada é selecionada e, na fase final, a imagem binária é obtida através de um *threshold* fixo.

Hoover *et al.* [35] aperfeiçoaram a técnica proposta por Chaudhuri *et al.* [34], introduzindo restrições relacionadas com a vizinhança de cada pixel. O método também se baseia na análise da resposta ao *matched filter*, mas prevê a posterior divisão desta imagem em diferentes regiões. Segue-se um processo iterativo ao longo do qual vários valores de *threshold* são testados. Em cada iteração, compara-se a classe atribuída a cada pixel com a classe dos pixeis envolventes, numa dada região. Se o pixel estiver demasiado desenquadrado é tomado como um possível falso candidato e, na iteração seguinte, avaliado noutra região.

Um método mais recente assenta na aplicação dos filtros COSFIRE propostos por Azzopardi *et al.* [30]. A característica chave destes filtros é a resposta seletiva a qualquer estrutura em forma de barra, como os vasos. Além disto, como o método prevê a utilização de várias respostas desfasadas entre si, é possível aplicar filtros com diferentes escalas e, assim, lidar com vasos de largura variável que se revelavam problemáticos nas propostas anteriores.

Kovács e Hadju [46] propuseram uma técnica de auto-calibração que pode ser usada para adaptar um modelo previamente treinado, num determinado conjunto de imagens de fundo, a um novo grupo de imagens com diferente resolução, FOV ou nível de ruído. O método procura atingir esta correspondência usando filtros de Gabor e estratégias de reconstrução de contornos.

Recentemente, Zhang *et al.* [47] propuseram uma nova técnica que se baseia na maximização da resposta a um filtro de derivada gaussiana, de segunda ordem, num domínio matemático particular a que os autores chamaram de *orientation scores*. O método parece mostrar um desempenho promissor em regiões de cruzamento e, também, em vasos estreitos.

3.1.2.1.2 Morfologia Matemática

O termo morfologia matemática engloba uma série de operações usadas para extrair certas características das imagens como bordas, esqueletos ou componentes ligados. As operações morfológicas são, geralmente, utilizadas em imagens binárias, mas podem ser estendidas a imagens em níveis de cinza. As duas principais operações morfológicas são a dilatação e a erosão. A dilatação expande os objetos através de um dado elemento estruturante, preenchendo "furos" e conectando regiões disjuntas. A erosão, pelo contrário, faz uso desse elemento estruturante para encolher objetos e separar regiões inicialmente unidas. Compondo as duas operações básicas é possível obter duas outras operações muito comuns: a abertura e o fecho morfológicos. Sabe-se, *a priori*, que a morfologia da vasculatura pode ser vista como um conjunto de segmentos lineares conectados. Por isso, a morfologia matemática pode extrair longos segmentos vasculares de forma rápida e robusta ao ruído. A principal desvantagem de confiar exclusivamente em métodos morfológicos reside no facto de estes não explorarem a secção transversal conhecida do vaso; isto é, o facto dos vasos se tornarem mais estreitos à medida que se afastam radialmente do nervo óptico. Além disto, o uso de um elemento estruturante excessivamente grande pode dificultar a identificação de vasos altamente tortuosos [3, 45].

A aplicação de operações morfológicas para a segmentação de vasos da retina foi introduzida por Zana e Klein [48], através de uma proposta que prevê a combinação de filtros morfológicos com a avaliação da curvatura dos vasos. Segundo os autores, enquanto a morfologia matemática explora o facto dos vasos poderem ser vistos como segmentos lineares e conectados, a avaliação de curvatura visa identificar áreas de elevada tortuosidade para que se saiba quando ajustar o tamanho do elemento estruturante.

Mais tarde, Mendonça e Campilho [42] propuseram-se a extrair a vasculatura da retina através da combinação de um filtro de diferença de desfasamento de gaussianas (DoOG[3]) com uma reconstrução morfológica multi-escala. Neste método, as linhas centrais dos vasos são extraídas aplicando o filtro DoOG e a vasculatura é, depois, realçada através de um operador *top-hat* modificado, com elementos estruturantes circulares de tamanho variável, para lidar com vasos de diferentes larguras. Assim, são obtidos quatro mapas binários em escalas diferentes, usando reconstruções morfológicas com parâmetros distintos. Isto permite obter a imagem final através de um processo iterativo de crescimento de regiões que faz uso das linhas centrais dos vasos e dos vários mapas binários gerados.

Por sua vez, Miri e Mahloojifar [49] propuseram um método que combina morfologia matemática com a transformada de curvelet. A aplicação da transformada visa melhorar o contraste dos vasos, enquanto o uso das operações morfológicas está relacionado com a deteção de bordas. Na fase final, distingue-se uma etapa de redução de falsas deteções através de uma abertura morfológica.

Mais recentemente, Imani *et al.* [50] debruçaram-se sobre o tradicional problema da segmentação de vasos na presença de anormalidades. Nesta proposta, operações morfológicas são combinadas com a transformada de wavelet de Morlet para separar os vasos de outras estruturas, como lesões e hemorragias, que são frequentes na avaliação de casos patológicos.

[3] Do inglês: *Difference of Offset Gaussians.*

3.1.2.1.3 Vessel Tracking

Os métodos de *vessel tracking* segmentam um vaso entre dois pontos e operam ao nível do vaso em vez de se focarem em toda a vasculatura. Na fase inicial, selecionam-se, de forma manual ou automática, alguns pontos de partida (*seed points*), que geralmente se localizam nas bordas e zonas de cruzamento. Depois, a ideia passa por seguir as linhas centrais dos vasos, tentando encontrar o "caminho" que melhor combina com o perfil típico dos vasos sanguíneos. As principais vantagens destas abordagens assentam na capacidade para determinar a largura dos vasos de forma precisa e, também, para fornecer informações sobre cada vaso individualmente. Sabendo que os vasos estão conectados entre si, é possível seguir uma árvore inteira sem gastar tempo na análise exaustiva do *background*. Além disto, raramente há incoerências de conectividade ou problemas em zonas de ramificação. Por outro lado, este tipo de procedimento acarreta algumas complicações. Desde logo, a perda de segmentos vasculares que não tenham sido inicialmente marcados através de um ponto de partida é muito comum. Depois, quando um dado segmento vascular não é detetado, há tendência para descartar sub-árvores completas [3, 45].

O trabalho de Liu e Sun [51] foi o primeiro a utilizar uma estratégia de *vessel tracking* para segmentar os vasos da retina. Nesta proposta, as trajetórias locais dos vasos são estimadas depois de se fixar, manualmente, um ponto de partida dentro de cada vaso. Sempre que um segmento vascular é identificado, ele é excluído da imagem inicial. O processo é, então, repetido até que a árvore vascular tenha sido extraída por completo.

Mais tarde, Yin *et al.* [52, 53] apresentaram outro método deste tipo. Nesta proposta, os pontos de borda do vaso são inicialmente detetados usando os níveis de intensidade e propriedades de continuidade. Depois, um esquema de amostragem é adotado para selecionar os candidatos mais fortes entre os pontos marcados inicialmente. Os perfis de intensidade dos vasos são estimados, localmente, por uma curva gaussiana. Finalmente, um método bayesiano com o critério MAP[4] é usado para identificar a estrutura dos vasos e os pontos de borda definitivos.

Um dos mais recentes trabalhos de *vessel tracking* foi apresentado por Bekkers *et al.* [54]. Nele, propõe-se uma análise de conectividade, segundo várias orientações, perto das regiões de bifurcação e cruzamento. Segundo os autores, esta abordagem é particularmente eficaz para avaliar com sucesso a largura dos vasos, sobretudo em zonas tradicionalmente problemáticas.

3.1.2.1.4 Técnicas Multi-escala

Como já se referiu, a largura de um vaso diminui, gradualmente, à medida que este viaja radialmente para fora do disco óptico. Assim, um vaso pode ser visto como um padrão contrastado, composto por vários segmentos localmente lineares e com largura cada vez menor. A ideia por detrás das técnicas multi-escala passa por agrupar esta informação relativa às diferentes larguras, em várias escalas. Desta forma, procura-se resolver um dos principais problemas associados à segmentação de vasos: tratar, com eficácia semelhante, vasos grossos e vasos finos [3, 45].

[4] Do inglês: *Maximum a Posteriori*.

Frangi *et al.* [36] propuseram-se a desenvolver um filtro capaz de realçar a vasculatura, através da análise das derivadas parciais de segunda ordem da imagem (matriz Hessiana). A análise dos valores próprios desta matriz permite descobrir as direções de maior e menor curvatura dos vasos; isto permite lidar de forma diferente com vasos grossos e finos, porque se admite que os primeiros são, por norma, mais tortuosos que os segundos. Esta foi a primeira proposta deste género e abriu portas ao desenvolvimento de vários outros trabalhos.

Fathi e Nilchi [39] apresentaram um método baseado na utilização da transformada de wavelet para decompor os vasos em diferentes escalas. Os parâmetros da transformada são otimizados para detetar os vasos em várias direções, permitindo distingui-los de outras bordas. Na fase final, a árvore vascular é obtida aplicando um método de *threshold* baseado na equalização do histograma.

Por sua vez, Nguyen *et al.* [37] introduziram o conceito de detetor de linha multi-escala para a segmentação de vasos. Esta proposta foi a extensão do detetor de linha básico proposto em [55]. Face à proposta inicial, os autores concluíram que este novo método é mais eficaz em zonas que contenham vasos estritamente paralelos, em pontos de cruzamento e em vasos com reflexo central[5].

Mais tarde, Yin *et al.* [31] modificaram a ideia de Nguyen *et al.* [37] e introduziram um detetor de linha sensível à orientação. O método é capaz de identificar a orientação dos vasos, sem pressupostos iniciais, fazendo uso da densidade espectral da transformada de Fourier. Além disto, enquanto os vasos grossos são extraídos pelo detetor de linha, os vasos finos são identificados através de filtros de Gabor, visto que os detetores de linha encontram, frequentemente, problemas em escalas menores.

3.1.2.1.5 Técnicas Baseadas em Modelos

Como o próprio nome indica, as técnicas baseadas em modelos aplicam modelos explícitos para extrair a vasculatura. Estes podem, por norma, dividir-se em dois grupos: modelos do perfil dos vasos[6] e modelos deformáveis [3, 45].

3.1.2.1.5.1 Modelos do Perfil dos Vasos

Os modelos do perfil dos vasos assentam no conhecimento de algumas características da árvore vascular. Uma delas é o perfil transversal que, por norma, é aproximado através de uma curva gaussiana ou de uma mistura de gaussianas. Ocasionalmente, aplicam-se também *splines* cúbicas ou polinómios de Hermite. Outras características relevantes residem no facto dos vasos poderem ser vistos como segmentos que possuem uma determinada largura e que estão conectados em árvore. A principal limitação associada a este grupo é a dificuldade para incluir *features* que não estejam diretamente relacionadas com os vasos. Deste modo, a presença de sinais patológicos é especialmente limitadora [3, 45].

Vermeer *et al.* [56] apresentaram um dos primeiros trabalhos deste género, aplicando um modelo laplaciano para extrair as bordas e usando um algoritmo de preenchimento pensado para contornar o problema do reflexo central. O método prevê uma fase de remoção de falsas deteções devido ao ruído.

[5] Fenómeno onde a luz é refletida na parte interna do vaso, levando, na maioria das vezes, à deteção de dois vasos finos em lugar do vaso grosso inicial.

[6] Do inglês: *Vessel Profile Models*.

Lam e Yan [57] propuseram um novo algoritmo de segmentação de vasos com base na divergência de campos vetoriais. Neste método, as linhas centrais dos vasos são detetadas usando o campo do vetor gradiente e, numa fase posterior, servem de referência para filtrar objetos que estejam excessivamente distantes e que tenham sido, erroneamente, detetados como vasos.

Mais tarde, Lam *et al.* [58] estenderam a proposta apresentada por Lam e Yan [57], tentando mitigar o principal problema revelado nessa altura: a incapacidade para lidar, eficazmente, com imagens patológicas. Assim, surgiu um novo método que se baseia na utilização de medidas de concavidade distintas para distinguir os vasos sanguíneos das áreas de lesão.

3.1.2.1.5.2 Modelos Deformáveis

Os modelos deformáveis podem, ainda, dividir-se em dois sub-grupos: modelos de contornos ativos e modelos geométricos.

Nos modelos de contornos ativos, informalmente conhecidos como *snakes*, são definidas curvas, no domínio da imagem, que podem mover-se sob a influência de forças internas (inerentes à própria curva) e externas (derivadas dos dados da imagem). Enquanto as primeiras produzem tensão e restringem o comportamento da curva, as segundas podem ser ajustadas por um processo automático de supervisão, ou por um utilizador humano, de forma a ir de encontro à *feature* que se pretende modelar. A principal vantagem destes métodos recai na sua autonomia na procura por um estado de energia mínimo [3, 45].

Um dos trabalhos de segmentação de vasos mais relevantes, dentro deste grupo, foi proposto por Al-Diri *et al.* [59]. A proposta assenta na utilização de um modelo de contornos ativos que prevê o uso de um par – ou dois, na presença do reflexo central – de contornos para capturar cada borda do vaso e manter a consistência da largura. Assim, consegue-se localizar as bordas com precisão, mesmo em condições desafiadoras como a presença de vasos estritamente paralelos ou vasos muito finos.

Zhao *et al.* [33], por seu lado, propuseram um método de segmentação baseado num modelo de contornos ativos de perímetro infinito. A proposta faz, também, uso de diferentes tipos de informação para caracterizar cada região, combinando informações de intensidade com a aplicação de filtros de fase.

Passando aos modelos geométricos, aqui a base assenta na teoria de evolução de curvas. Estes modelos são, geralmente, implementados através do método *level-set* (um algoritmo numérico usado para rastrear formas). A vantagem destes métodos é o facto das computações numéricas poderem ser realizadas numa grelha cartesiana fixa, sem que haja a necessidade de recorrer à parametrização dos objetos de interesse [3, 45].

Xiao *et al.* [60] aplicaram o método *level-set* para minimizar uma função de energia de forma a extrair a vasculatura. O método assume que a probabilidade de cada pixel é dependente da dos pixeis vizinhos, exercendo, assim, restrições espaciais.

Zhao *et al.* [29], alternativamente, combinaram duas técnicas para extrair vasos grossos e finos, a partir de imagens de fundo. Nos vasos grossos, é aplicado o método *level-set* para minimizar uma função de energia, à semelhança do método de Xiao *et al.* [60]; nos vasos finos, é usada uma estratégia de crescimento de regiões.

3.1.2.2 Métodos Supervisionados

Os métodos supervisionados requerem um conjunto manualmente anotado de imagens para aprender um modelo de classificador capaz de rotular cada pixel como vaso ou *background*. Durante a fase de treino, um algoritmo de otimização analisa um conjunto de *features* e a classe associados a cada exemplo de treino e infere um conjunto de parâmetros que o classificador usa para lidar com novos exemplos, não vistos até então, na fase de teste. A fase crítica destas abordagens prende-se com a extração do conjunto de *features* que melhor descreve as condições do problema, visto que é com base nele que o classificador estabelece relações. Como os métodos supervisionados são projetados com base em dados pré-classificados, o seu desempenho é, geralmente, melhor do que o dos métodos não supervisionados, principalmente em imagens que não evidenciem sinais patológicos [3, 45].

Niemeijer *et al.* [61] apresentaram um dos primeiros métodos supervisionados para a extração dos vasos retinianos em imagens de fundo. Neste trabalho, extrai-se um vetor de *features* composto pelas intensidades do canal verde e pelas respostas a um filtro gaussiano e às suas derivadas de primeira e segunda ordem tomadas em várias escalas. Depois, um classificador k-NN[7] é usado para estimar a probabilidade de cada pixel e a imagem binária é obtida pela aplicação de um *threshold* fixo.

Staal *et al.* [62] estenderam o método de Niemeijer *et al.* [61] apresentando uma metodologia que explora o facto dos vasos serem estruturas alongadas. A técnica baseia-se na extração de bordas, que são indicadores naturais dos vasos e que podem ser usadas para sinalizar as linhas centrais. A imagem é dividida em várias regiões convexas, atribuindo cada pixel à linha central mais próxima. Depois, são extraídas *features* sobre essas regiões, mas também sobre cada pixel de forma individual. Na etapa final, recorre-se, novamente, a um classificador k-NN.

A combinação de uma classificação supervisionada com a wavelet 2D de Gabor foi, pela primeira vez, introduzida por Soares *et al.* [40]. Nesta proposta, cada pixel é representado por um vetor de *features* composto pelo nível de intensidade do canal verde e pelas respostas à transformada de wavelet de Gabor tomada em várias escalas. Para classificar cada pixel como vaso ou *background* recorre-se a um classificador GMM[8]. O algoritmo usa informação de contexto, mas descarta pressupostos sobre as formas e estruturas presentes na imagem. Assim, gera, com frequência, falsas deteções na borda do disco óptico e na presença de sinais patológicos.

Xu e Luo [63], por sua vez, fizeram uso de várias técnicas de processamento de imagem e recorreram a um classificador SVM[9]. Durante o pré-processamento, é obtida uma imagem binária através de uma técnica de *threshold* adaptativo. Depois, através de uma análise de componentes ligados, os vasos grossos são separados do disco óptico, isolados e retirados da imagem binária, restando apenas dois grupos: os vasos finos e outros segmentos mal classificados. Nesta altura, um conjunto de *features* obtido através da transformada de wavelet é fornecido ao classificador SVM, que é responsável por separar os vasos finos dos falsos candidatos. Os vasos finos validados são, ainda, sujeitos a uma etapa de crescimento de regiões e, na fase final, a árvore vascular é obtida reunindo-os com os vasos grossos detetados previamente.

[7] Do inglês: *k-Nearest Neighbor*.
[8] Do inglês: *Gaussian Mixture Model*.
[9] Do inglês: *Support Vector Machine*.

O classificador SVM foi também incluído na proposta de You *et al.* [64]. Nesta abordagem, opta-se por sinalizar previamente os vasos finos através de projeções radiais e, depois, aplica-se a transformada de wavelet para compilar um conjunto de *features* que são passadas ao classificador. O algoritmo revela-se muito eficaz na deteção de vasos de baixo contraste, mas a propensão para falsas deteções, na presença de lesões, continua a representar uma limitação importante.

Marin *et al.* [65] apresentaram um método baseado no convencional percetrão multi-camada (MLP[10]). A rede é composta por uma camada de entrada com sete neurónios, por três camadas ocultas com quinze neurónios cada e por uma camada de saída com um neurónio único. Para detetar os vasos, que não têm largura uniforme e podem estar orientados de várias formas, é utilizado um conjunto de *features* composto pelo nível de intensidade do canal verde e por descritores de forma invariantes à translação, à rotação e à mudança de escala. O método revela-se robusto perante sinais patológicos e imagens pertencentes a outros conjuntos de dados.

Fraz *et al.* [43] propuseram um método inovador baseado no uso de várias técnicas de extração de *features*, como a análise do vetor gradiente, transformações morfológicas e respostas a filtros de Gabor. A probabilidade de cada pixel é obtida através de um conjunto[11] de árvores de decisão.

Mais tarde, Roychowdhury *et al.* [66] desenvolveram um trabalho que engloba três fases. Na primeira fase, obtêm-se duas imagens binárias, através de filtragem passa-alto e reconstrução morfológica; isto permite extrair os vasos grossos através das porções comuns entre as duas imagens. Depois, numa etapa homóloga à descrita por Xu e Luo [63], um classificador GMM é usado para separar os vasos finos dos falsos candidatos, através de um conjunto de *features* baseado nas intensidades dos pixeis vizinhos e nos gradientes de primeira e segunda ordem. Na fase final, os vasos grossos são combinados com os vasos finos validados pelo classificador, originando a imagem segmentada.

Num dos mais recentes trabalhos de segmentação de vasos em imagens de fundo retiniano, Zhang *et al.* [67] desenvolveram o seu anterior trabalho [47] e, desta feita, apresentaram uma abordagem supervisionada. O método engloba a utilização de *features* extraídas através da transformada de wavelet, *features* derivadas do domínio de *orientation scores* e, também, *features* obtidas através de filtros gaussianos. Para classificar cada pixel como vaso ou *background* foi usado um classificador RF[12]. Segundo os autores, este novo método revela-se mais robusto que a abordagem não supervisionada proposta inicialmente.

3.1.2.3 Métodos de Deep Learning

Os métodos de *deep learning* são um sub-conjunto de métodos de *machine learning* que tem ganho grande destaque nos últimos anos, devido à sua capacidade para aprender, de forma automática, uma hierarquia de *features* cada vez mais complexas a partir dos dados de entrada. Isto sugere uma mudança de paradigma de acordo com a qual o foco está agora no design e otimização de arquiteturas e não na extração manual de *features*, que podem ser dependentes do problema e exigir conhecimento especializado [68].

[10] Do inglês: *Multilayer Perceptron*.
[11] Do inglês: *Ensemble*.
[12] Do inglês: *Random Forest*.

Mesmo que a origem das redes neuronais profundas possa ser traçada até ao século passado [69, 70], elas ressurgiram em força recentemente, tendo sido aplicadas em várias áreas, inclusive na segmentação de vasos retinianos. Li *et al.* [71], por exemplo, transformaram a tarefa de segmentação num problema de mapeamento de vasos e utilizaram uma rede neuronal profunda para aprender a função de mapeamento. Devido à dificuldade para treinar este tipo de rede diretamente, através do convencional algoritmo de *backpropagation*, a primeira camada é pré-treinada usando um *autoencoder*.

Entre os métodos de *deep learning*, as redes neuronais convolucionais (CNNs[13]) começaram por ganhar destaque ao serem usadas, com enorme sucesso, em desafios de reconhecimento de objetos [72, 73]. Desde aí, devido aos grandes avanços que as CNNs trouxeram no ramo de visão por computador, têm surgido, também, outros trabalhos que as aplicam em imagem médica e, em particular, na segmentação de vasos retinianos em imagens de fundo.

Um dos primeiros trabalhos deste tipo foi proposto por Melinščak *et al.* [74]. Nesta proposta, faz-se uso de uma CNN composta por dez camadas (quatro camadas convolucionais, quatro camadas de *pooling* e duas camadas densas) e cada pixel é segmentado de forma individual.

Liskowski e Krawiec [75], por seu turno, testaram um conjunto compreensivo de arquiteturas, entre as quais uma rede composta por sete camadas (quatro camadas convolucionais e três camadas densas) foi a que mais se destacou. O método prevê, ainda, o uso de um esquema de *structured prediction* que permite adicionar informação de contexto e segmentar uma janela de 5×5 pixeis, fazendo com que a última camada densa seja composta por 25 neurónios.

Recentemente, Fu *et al.* [76] formularam a segmentação de vasos como um problema de deteção de bordas e aplicaram uma CNN composta por catorze camadas (onze camadas convolucionais e três camadas de *pooling*). Na fase final, um campo aleatório condicional (CRF[14]), reformulado como uma rede neuronal recorrente (RNN[15]), é utilizado para modelar as interações entre pixeis a longas distâncias.

3.1.3 Pós-processamento

Ao contrário da fase de pré-processamento, onde algumas práticas padrão podem ser facilmente identificadas, a etapa de pós-processamento é, tipicamente, mais dependente de cada abordagem. Ainda assim, quando aplicada, podem destacar-se dois grandes objetivos: preencher "furos" em vasos detetados e/ou eliminar segmentos ou pixeis isolados.

No primeiro caso, recorre-se, por norma, a um algoritmo de preenchimento através do qual qualquer pixel com um número limite de pixeis vizinhos classificados como vaso, é também tomado como vaso. You *et al.* [64] e Marin *et al.* [65] fixaram esse limite em quatro e seis pixeis, respetivamente.

Já no segundo caso, destacam-se três alternativas. Soares *et al.* [40], por exemplo, aplicaram a convencional abertura morfológica para remover pixeis isolados associados ao ruído. Marin *et al.* [65] e Zhao *et al.* [29], por sua vez, optaram por uma análise de componentes ligados, na qual todos os segmentos vasculares com menos de vinte e cinco pixeis são descartados e considerados *background*. Em alternativa, Whang *et al.* [44] aplicaram um *threshold* correspondente a quatro quintos da intensidade

[13] Do inglês: *Convolutional Neural Network*.
[14] Do inglês: *Conditional Random Field*.
[15] Do inglês: *Recurrent Neural Network*.

máxima da imagem e usaram-no para identificar possíveis zonas de lesão (alta intensidade). Depois, usando a operação morfológica *convex hull* é possível delimitar estas regiões e, assim, descartar todos os pixeis isolados que possam estar contidos nelas.

Tabela 3.1: Estado da arte da segmentação de vasos retinianos em imagens de fundo. As últimas colunas avaliam a presença (√) ou ausência (-) de: Métricas – avaliação quantitativa; Cruzado – avaliação num conjunto de dados diferente do de treino; Lesões – avaliação em imagens patológicas; Vasos Finos – avaliação exclusiva nos vasos finos; *Inter-rater* – avaliação face às anotações de um especialista diferente.

Tipo	Técnica	Ano	Método	Métricas	Cruzado	Lesões	Vasos Finos	Inter-rater
Não Supervisionado	*Matched Filtering*	1989	Chaudhuri *et al.* [34]	-	-	-	-	-
		2000	Hoover *et al.* [35]	-	-	-	-	-
		2013	Wang *et al.* [44]	√	-	-	-	-
		2015	Azzopardi *et al.* [30]	√	-	-	-	-
		2016	Kovács e Hadju [46]	√	√	-	-	-
		2016	Zhang *et al.* [47]	√	-	-	-	-
	Morfologia Matemática	2001	Zana e Klein [48]	-	-	-	-	-
		2006	Mendonça e Campilho [42]	√	-	√	-	-
		2011	Miri e Mahloojifar [49]	√	-	-	-	-
		2015	Imani *et al.* [50]	√	-	-	-	-
	Vessel Tracking	1993	Liu e Sun [51]	-	-	-	-	-
		2013	Yin *et al.* [53]	√	-	√	-	-
		2014	Bekkers *et al.* [54]	-	-	-	-	-
	Multi-escala	1998	Frangi *et al.* [36]	-	-	-	-	-
		2013	Fathi e Nilchi [39]	√	-	-	-	-
		2013	Nguyen *et al.* [37]	√	-	-	-	-
		2015	Yin *et al.* [31]	√	-	-	-	√
	Modelos do Perfil do Vaso	2004	Vermeer *et al.* [56]	-	-	-	-	-
		2008	Lam e Yan [57]	√	-	-	-	-
		2010	Lam *et al.* [58]	√	-	√	-	-
	Modelos Deformáveis	2009	Al-Diri *et al.* [59]	√	-	-	-	√
		2013	Xiao *et al.* [60]	√	-	-	-	-
		2014	Zhao *et al.* [29]	√	-	-	-	-
		2015	Zhao *et al.* [33]	√	-	-	-	-
Supervisionado		2004	Niemeijer *et al.* [61]	√	-	-	-	-
		2004	Staal *et al.* [62]	√	-	-	-	-
		2006	Soares *et al.* [40]	√	-	-	-	-
		2010	Xu e Luo [63]	√	-	-	-	-
		2011	You *et al.* [64]	√	-	-	-	-
		2011	Marin *et al.* [65]	√	√	-	-	-
		2012	Fraz *et al.* [43]	√	√	√	-	-
		2015	Roychowdhury *et al.* [66]	√	√	√	-	-
		2017	Zhang *et al.* [67]	√	√	-	-	-
Deep Learning		2015	Melinščak *et al.* [74]	√	-	-	-	-
		2016	Li *et al.* [71]	√	√	√	-	-
		2016	Liskowski e Krawiec [75]	√	-	√	√	-
		2016	Fu *et al.* [76]	√	-	-	-	-

3.2 Sumário

O problema da segmentação de vasos retinianos tem sido vastamente estudado nas últimas décadas. Prova disso é o grande número de trabalhos descritos na literatura, entre os quais se distinguem três fases estruturantes: pré-processamento, classificação e pós-processamento.

As fases de pré- e pós-processamento visam, respetivamente, melhorar a qualidade das imagens iniciais e refinar as segmentações produzidas durante a etapa de classificação. Apesar destas etapas apresentarem, naturalmente, diferenças conforme a metodologia de cada trabalho, é possível associá-las a algumas práticas padrão. No pré-processamento, a correção do contraste e o realce dos vasos são bastante comuns. No pós-processamento, destacam-se algumas estratégias que visam preencher "furos" em vasos detetados e/ou eliminar segmentos isolados.

Dependendo da forma como cada método obtém os seus parâmetros durante a etapa de classificação, os vários trabalhos propostos na literatura podem enquadrar-se em um de dois grupos. Os métodos não supervisionados operam sem *labels* e baseiam-se em regras. Como não são obtidos com base numa amostra da população, apresentam uma capacidade de generalização potencialmente boa. Contudo, revelam alguma dependência face aos contornos específicos de cada problema.

Nos métodos supervisionados é necessário um conjunto manualmente anotado de imagens para aprender um modelo de classificador, durante a fase de treino. Este é, depois, usado para lidar com novas imagens, na fase de teste. Apesar de, por norma, este grupo obter melhores resultados que o anterior, o seu desempenho é fortemente afetado pela fase de *feature engineering* que pode ser morosa e exigir conhecimento especializado.

Por último, ainda que mesmo entre eles se possam distinguir abordagens supervisionadas e não supervisionadas, os métodos de *deep learning* constituem um grupo especial que se destaca pela sua capacidade para aprender, de forma automática, uma hierarquia de *features* cada vez mais complexas. Isto permite obviar o principal inconveniente dos métodos supervisionados convencionais e potencia uma nova mentalidade onde o foco está na construção e otimização de arquiteturas e não na extração manual de *features*. Dentro deste grupo, as CNNs têm sido o foco de maior destaque na área da segmentação de vasos. Será, em grande parte, sobre elas que a análise do próximo capítulo se debruçará.

Capítulo 4

Fundamentos Teóricos

O destaque que os métodos de *deep learning* ou de aprendizagem profunda reuniram junto da comunidade científica, nos últimos anos, é surpreendente. Também por isso, é curioso perceber que as fronteiras exatas do termo não são facilmente identificáveis. Goodfellow *et al.* [77] sugeriram que estes métodos se distinguem dos demais sob duas perspetivas distintas. A primeira foca-se na sua capacidade única para aprender a melhor representação possível para um conjunto de dados, partindo da ideia de que qualquer conceito pode ser expresso em relação a outros mais simples. A segunda olha para a profundidade destes modelos como um meio para aprender um programa de computador mais complexo, onde cada camada de representação pode ser vista como o estado que se atinge depois de executar um novo conjunto de instruções [77].

Informalmente, os métodos de *deep learning* podem ser entendidos como o campo particular de *machine learning* que lida com redes neuronais profundas (*deep*) – ainda que, mais uma vez, o limite que distingue uma rede profunda das demais não seja claro [77]. Assim, para que se possa perceber estes métodos da melhor forma, importa conhecer alguns conceitos básicos de *machine learning* e de redes neuronais em particular. Neste capítulo, apresentam-se os fundamentos teóricos mais importantes para a compreensão da metodologia de segmentação automática que será apresentada no capítulo seguinte. A análise foca-se, primeiro, nas redes neuronais artificiais – em especial nas CNNs – e prossegue, depois, para o treino destes modelos. Por fim, apresentam-se alguns conceitos sobre a transformada de wavelet.

4.1 Machine Learning e Reconhecimento de Padrões

Arthur Samuel definiu *machine learning* como sendo "a disciplina que confere aos computadores a habilidade para aprender sem que sejam explicitamente programados para tal" [78]. Mais tarde, Tom Mitchell clarificou este conceito de aprendizagem dizendo que "um programa de computador aprende através de uma experiência (E), com respeito a uma tarefa (T) e a uma medida de desempenho (P), se o seu desempenho em T, avaliado por P, melhora com E" [79]. De forma simplificada, *machine learning* ou aprendizagem automática é uma área que se ocupa da criação de algoritmos ou modelos capazes de extrair conhecimento a partir de um conjunto de dados e que se popularizou entre a comunidade de visão por computador. O reconhecimento de padrões é, muitas vezes, visto como uma faceta diferente da mesma área, mas com origens na engenharia [80].

A análise exaustiva dos conceitos de experiência, medida de desempenho e tarefa extrapola os objetivos desta dissertação. Ainda assim, em relação a esta última, é conveniente fazer uma análise breve para que se possa enquadrar o problema que aqui se discute.

4.1.1 O Problema de Classificação Binária

Apesar de existirem várias outras, a regressão e a classificação são amplamente conhecidas como as tarefas mais comuns em *machine learning* [79, 80].

Num problema de regressão, o modelo produz um valor numérico para um dado conjunto de entrada. O mesmo é dizer que o objetivo passa por aprender uma função $f : \mathbb{R}^n \to \mathbb{R}$, onde n simboliza o número de *features*. Um exemplo típico deste problema é prever o valor de venda de uma casa [79, 80].

Num problema de classificação, o modelo enquadra cada elemento do conjunto de entrada em uma de k classes ($k \geq 2$). Assim, é gerada uma função $f : \mathbb{R}^n \to \mathbb{R}^k$. Mediante o número de classes a discriminar, a modalidade da classificação é distinta. Quando $k = 2$, está-se perante um problema de classificação binária; quando $k > 2$, surge um problema de classificação multi-classe [79, 80].

No contexto da segmentação de vasos retinianos, estuda-se um problema de classificação binária. Em específico, dado um conjunto de pixeis (uma imagem), o objetivo é classificar cada pixel como vaso ou *background*. Na figura 4.1, cada pixel é representado por um círculo e a cor de cada círculo sinaliza a classe a que esse pixel pertence (a correspondência exata não é importante neste exemplo). A linha verde que separa os pontos de cada classe é chamada de fronteira de decisão (ou *decision boundary*). O objetivo dos modelos é encontrar boas fronteiras de decisão; isto é, fronteiras que permitam separar tão bem quanto possível as duas classes [79, 80]. Em problemas de classificação, é comum interpretar a saída do modelo como a probabilidade de um dado pixel pertencer a cada uma das classes[1]. Se a classe 0 simbolizar o *background* e a classe 1 disser respeito aos vasos, a saída do modelo será (y_0, y_1), com $y_0 + y_1 = 1$. A fronteira de decisão sinaliza, assim, os pontos onde $y_0 = y_1 = 0.5$. Isto significa que quando $y_1 > y_0$ o pixel será classificado como vaso e vice-versa [80].

O exemplo da figura 4.1 é meramente conceptual e não faz jus à dificuldade dos problemas reais. Na verdade, na maioria das vezes, os modelos são confrontados com problemas que exigem a identificação de fronteiras de decisão arbitrariamente complexas. Nesses casos, a escolha do modelo apropriado torna-se ainda mais crucial [79, 80].

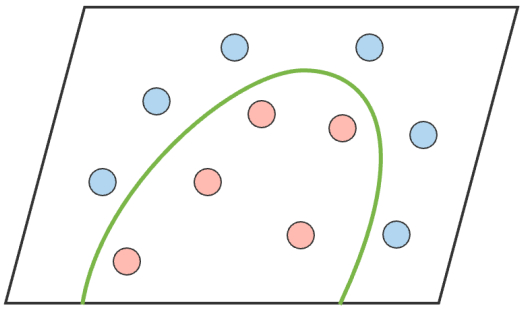


Figura 4.1: Problema genérico de classificação binária. Reproduzido de [81].

[1] Ainda que existam exceções, como nas SVMs.

4.1.2 Redes Neuronais Artificiais

As redes neuronais artificiais (ANNs[2]) – ou, simplesmente, redes neuronais – são modelos computacionais inspirados pelo sistema nervoso [82]. Informalmente, uma ANN é um conjunto de neurónios artificiais, que interagem entre si e têm uma estrutura baseada nos neurónios biológicos (figura 4.2).

Há várias décadas, unidades de investigação por todo o mundo tentam compreender o complexo funcionamento do cérebro humano. Estas pesquisas têm revelado informações importantes, ainda que o que se conhece seja uma ínfima amostra do que está por revelar. Estima-se que o cérebro contenha, em média, 86 biliões de neurónios [83]. Juntos, eles formam uma rede de complexidade inimaginável. Mesmo que, por ventura, houvesse um conhecimento exaustivo do funcionamento cerebral, a tecnologia atual não seria capaz de dar resposta a uma arquitetura desta dimensão [77]. Ainda assim, quando dotadas de um bom compromisso tamanho/desempenho, as ANNs são já capazes de obter ótimos resultados em várias tarefas de *machine learning*, liderando o estado da arte, sobretudo, em problemas de classificação [77, 79, 80].

Devido ao seu potencial, é fácil pensar nas ANNs como algo complicado. De facto, arquiteturas de grande envergadura foram já utilizadas com resultados surpreendentes. No entanto, os elementos estruturantes são bastante simples. Nesta secção, começa-se por analisar a rede neuronal elementar e passa-se, sucessivamente, a modelos mais complexos até culminar nas CNNs.

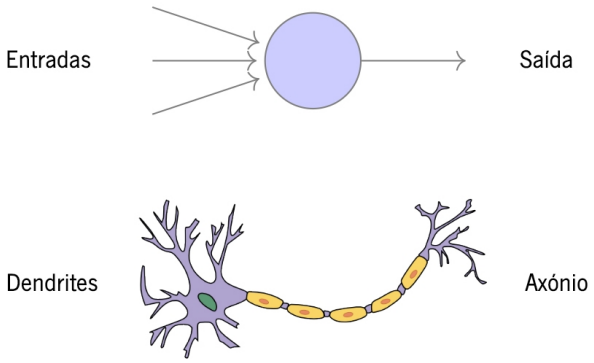


Figura 4.2: Homologias entre os neurónios biológico e artificial. Adaptado de [84].

4.1.2.1 Percetrão Simples

O percetrão simples (figura 4.3) é a rede neuronal mais trivial: consiste num único neurónio artificial. Na prática, recebe n entradas representadas por um vetor $x \in \mathbb{R}^n$ e retorna uma ativação $y \in \mathbb{R}$. Cada entrada i, $1 \leq i \leq n$, tem um peso associado $w_1, ..., w_n$. Para gerar y, as entradas são combinadas linearmente e transformadas através de uma função de ativação (φ). O *bias* (b) é um parâmetro adicional, que também pode ser visto como um peso que atua sobre uma entrada de valor unitário ($x_0 = 1$, $w_0 = b$)[3]. Formalmente, este conjunto de operações pode ser representado por [80]:

$$y = \varphi(z) = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right). \tag{4.1}$$

[2] Do inglês: *Artificial Neural Network*.

[3] Por este motivo, nas ANNs é frequente usar os termos parâmetro e peso de forma indiscriminada.

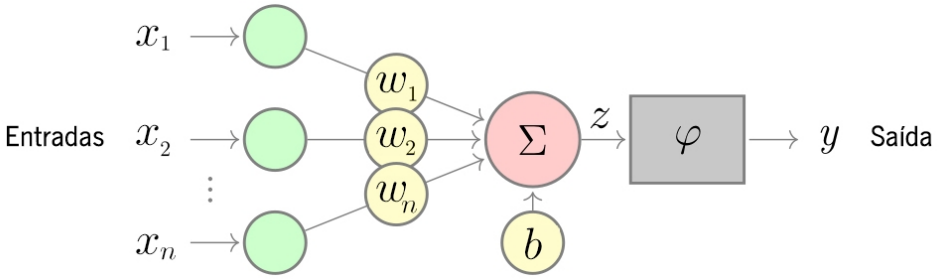


Figura 4.3: Percetrão simples. Adaptado de [84].

As fronteiras de decisão que um único neurónio é capaz de identificar estão limitadas a hiperplanos e, por isso, o percetrão simples é demasiado limitado para fazer face a tarefas mais complexas (cinge-se, por exemplo, à réplica de portas lógicas elementares) [79, 80]. No entanto, quando usado como unidade básica, este modelo torna possível o surgimento de outros mais versáteis.

4.1.2.2 Percetrão Multi-camada

Numa rede com vários neurónios, as ligações podem ser estabelecidas arbitrariamente. Assim, há várias topologias. Aqui analisam-se apenas redes neuronais *feedforward*, ou não recorrentes, onde a informação flui num só sentido (da entrada para a saída) [79, 80]. O exemplo típico é o MLP (figura 4.4).

Em geral, um MLP é composto por um grupo de neurónios distribuídos por c camadas mutuamente exclusivas $L^{(1)}, ..., L^{(c)}$. A rede resulta da agregação de cada camada $L^{(l)}$ na camada anterior $L^{(l-1)}$, através da ligação de cada neurónio da primeira a todos os neurónios da segunda. Assim, $L^{(1)}$ é a camada de entrada, $L^{(c)}$ a camada de saída e $L^{(2)}, ..., L^{(c-1)}$ as camadas ocultas ou escondidas [80].

O tamanho de $L^{(1)}$ e $L^{(c)}$ é, normalmente, projetado de acordo com as condições do problema: $L^{(1)}$ deve ter n neurónios correspondentes a n *features* ($x \in \mathbb{R}^n$); $L^{(c)}$ deve possuir k neurónios para que k classes possam ser avaliadas ($y \in \mathbb{R}^k$). A exceção mais comum ocorre na classificação binária onde, muitas vezes, $y \in \mathbb{R}$. Quanto às camadas ocultas, o seu número e tamanho podem variar mesmo em problemas idênticos e são hiperparâmetros com crucial importância no desempenho da rede [79, 80].

O conjunto de operações que permite obter a saída de uma ANN dadas as suas entradas e um conjunto de parâmetros é chamado de propagação direta [79, 80]. Num MLP, isto traduz-se num processo sequencial de multiplicação de matrizes. Para o descrever formalmente, considere-se a seguinte notação. Cada neurónio j da camada $L^{(l)}$ está associado a um *bias* $b^{(l)}$ e produz uma ativação $a_j^{(l)}$ que é o j-ésimo elemento de $a^{(l)}$. Além disto, $w^{(l-1)}$ é a matriz de pesos que mapeia $L^{(l-1)}$ em $L^{(l)}$, com ${w_{ji}}^{(l-1)}$ a ser o peso que liga o neurónio i da primeira ao neurónio j da segunda. Agora tome-se a figura 4.4 como exemplo. Em $L^{(1)}$, vem diretamente $a^{(1)} = x$. Em $L^{(2)}$, tem-se $a^{(2)} = [a_1^{(2)}, ..., a_5^{(2)}]^T$, com $a_1^{(2)} = \varphi(\sum_{i=1}^{n} {w_{1i}}^{(1)} x_i + b^{(2)})$, ..., $a_5^{(2)} = \varphi(\sum_{i=1}^{n} {w_{5i}}^{(1)} x_i + b^{(2)})$. Como na última camada se tem apenas um neurónio, a saída da rede vem $y = a^{(3)} = a_1^{(3)} = \varphi(\sum_{i=1}^{5} {w_{1i}}^{(2)} a_i^{(2)} + b^{(3)})$.

Num MLP, a inclusão de camadas ocultas torna as fronteiras de decisão arbitrariamente complexas e permite resolver uma vasta gama de problemas. Quando novas camadas são introduzidas, surgem redes mais profundas e daqui advém o termo *deep learning* [77]. Assim, importa realçar que a principal característica destes modelos é a sua profundidade e não uma topologia em particular – esta pode variar das RNNs às CNNs, passando pelos *autoencoders* e por várias outras que extrapolam esta dissertação.

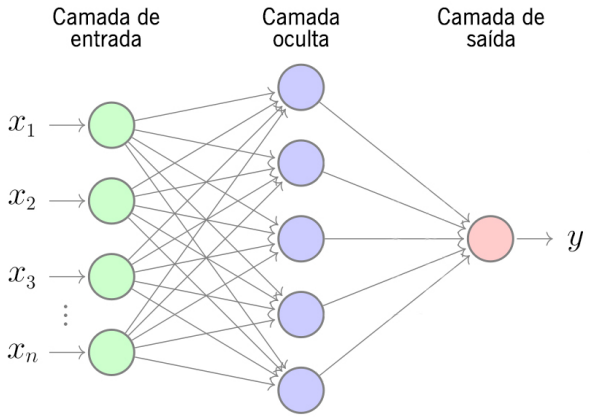


Figura 4.4: Percetrão multi-camada. Adaptado de [84].

4.1.2.3 Redes Neuronais Convolucionais

As CNNs [69] (figura 4.5) são redes *feedforward* especialmente concebidas para lidar com dados em forma de *array*, como séries temporais (1D), imagens (2D) ou vídeos (3D) [77]. São inspiradas no trabalho de Hubel e Wiesel sobre o sistema visual dos gatos [85] e foram os primeiros modelos de *deep learning* a apresentar resultados significativos, muito antes da sua aceitação global. O nome deriva do uso da operação de convolução. Assim, uma CNN é toda a ANN que utilize convolução em vez da convencional multiplicação de matrizes em pelo menos uma das camadas [77].

Novas arquiteturas de CNNs surgem praticamente todas as semanas. No entanto, todas elas são construídas com base num número limitado de camadas que se analisam de seguida.

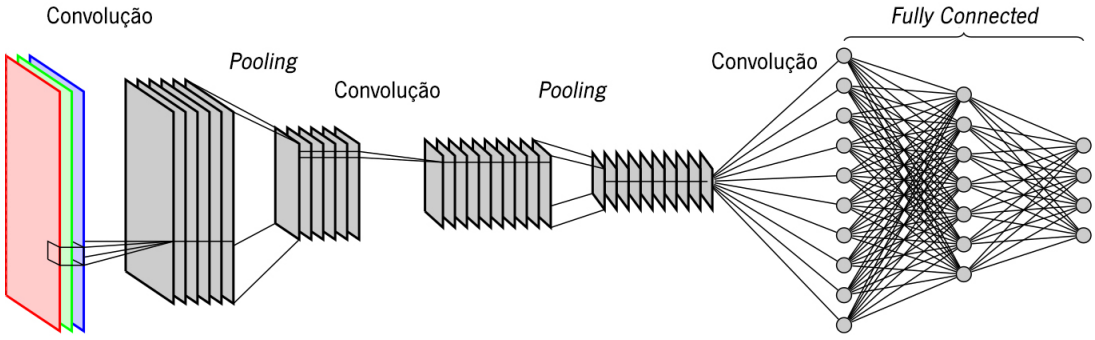


Figura 4.5: Rede neuronal convolucional. Adaptado de [84].

4.1.2.3.1 Camada Convolucional

Suponha-se que, por hipótese, se dispõe de um sensor de posição capaz de gerar um sinal contínuo $x(t)$. Para minimizar potenciais efeitos do ruído, pretende-se obter uma média entre medições adquiridas em vários instantes diferentes. No entanto, deseja-se que medições recentes tenham maior preponderância, utilizando uma função $w(\tau)$ em que τ representa o tempo passado desde a aquisição. Matematicamente, isto pode ser traduzido pela operação de convolução [77]:

$$f(t) = (x * w)(t) = \int_{-\infty}^{\infty} x(\tau) w(t - \tau) d\tau. \tag{4.2}$$

Numa situação real, a ideia de um sensor capaz de produzir um sinal contínuo é ilusória. Na prática, ele forneceria medições em intervalos de tempo regulares. Assim, se n tomar apenas valores discretos, pode definir-se convolução discreta [77]:

$$f[n] = (x * w)[n] = \sum_{\tau=-\infty}^{\infty} x[\tau]w[n-\tau]. \tag{4.3}$$

Estendendo este raciocínio para 2D, vem [77]:

$$F(i,j) = (I * K)(i,j) = \sum_{p}\sum_{q} I(p,q)K(i-p, j-q). \tag{4.4}$$

Como a convolução exibe propriedade comutativa, pode ainda escrever-se [77]:

$$F(i,j) = (K * I)(i,j) = \sum_{p}\sum_{q} I(i-p, j-q)K(p,q). \tag{4.5}$$

No contexto das CNNs, I designa a entrada, K é o *kernel* ou filtro e F o *feature map*. Uma CNN contém várias camadas convolucionais, que recebem um certo número de entradas e retornam um número variável de *feature maps*. Cada *feature map* resulta de convoluir a entrada com um só filtro e uma camada convolucional pode ter vários filtros. Note-se que os filtros de uma camada têm de possuir o mesmo número de canais que a entrada. Por exemplo, se a entrada da primeira camada for uma imagem RGB de $a \times b \times 3$ pixeis, então todos os filtros dessa camada terão $c \times d \times 3$ pesos [77].

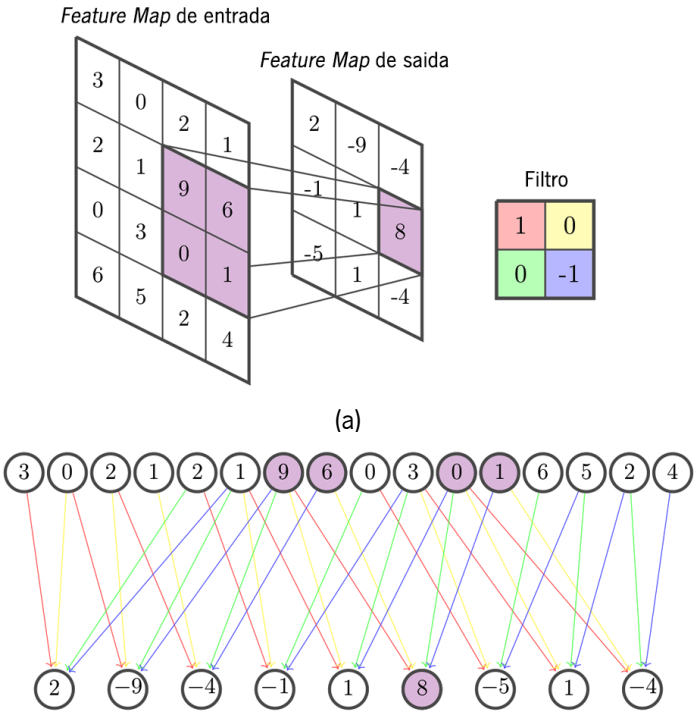


Figura 4.6: Operação de convolução: (a) Visão simples; (b) Visão no contexto das ANNs. Cada neurónio de saída liga-se apenas a um grupo restrito de neurónios de entrada (conectividade local) e ligações com a mesma cor têm o mesmo peso associado (partilha de parâmetros). Adaptado de [84]

A figura 4.6a ilustra um exemplo de computação básico, onde a entrada e o filtro possuem apenas um canal. Considerando os pixeis assinalados a roxo, vem: $9\times1 + 6\times0 + 0\times0 + 1\times(-1) = 8$. Da perspetiva das ANNs convencionais, cada uma destas unidades é a ativação de um neurónio que captura *features* particulares numa zona restrita da entrada ou da camada anterior. Essa zona é chamada de *receptive field* – ver capítulo 2 – e é definida pelo tamanho do filtro. Quando uma unidade apresenta um valor elevado, então o filtro encontrou uma correspondência forte e a *feature* codificada por ele (como uma borda ou uma curva) está, pelo menos parcialmente, presente. Caso contrário, esta correspondência é mais fraca. À medida que novas camadas são utilizadas, redes mais profundas aparecem e é possível reconhecer *features* mais complexas (como padrões ou objetos) nos *feature maps* [77].

O facto de *features* cada vez mais complexas surgirem à medida que novas camadas convolucionais são adicionadas faz com que a etapa de *feature engineering* dos métodos convencionais possa ser obviada e permite olhar para as CNNs como um modelo capaz de aprender a melhor representação dos dados de entrada – motivo pelo qual os métodos de *deep learning* são também vistos como um grupo particular de métodos de *representation learning* [77].

4.1.2.3.1.1 Motivação

A operação de convolução é motivada por dois princípios base que ajudam a explicar o sucesso das CNNs face às ANNs convencionais: a conectividade local e a partilha de parâmetros (figura 4.6b) [77].

As ANNs convencionais (como o MLP) assentam na multiplicação por uma matriz de pesos, pelo que todas as unidades de camadas consecutivas se encontram ligadas entre si. O mesmo é dizer que cada neurónio de uma camada interage com todos os neurónios da camada anterior. Nas CNNs, pelo contrário, cada neurónio de uma camada está apenas ligado aos neurónios que constituem o seu *receptive field*. Esta propriedade é conhecida como conectividade local ou ligações dispersas[4]. Isto significa que a rede é composta por menos pesos e, além de reduzir os requisitos de memória, torna as CNNs mais fáceis de treinar [77].

A partilha de parâmetros[5] refere-se ao uso de um mesmo conjunto de pesos para regiões diferentes da entrada. Numa ANN convencional, cada peso é multiplicado por um elemento de entrada e não mais é usado. Numa CNN, o mesmo conjunto de pesos (o filtro) é convoluído pelas várias localizações da entrada. O mesmo é dizer que em vez de aprenderem um conjunto diferente de parâmetros para cada localização, estes modelos aprendem um conjunto único para todas as localizações. Isto, além de também reduzir o número total de pesos, dota as CNNs de uma propriedade única: a invariância ou equivariância à translação [77]. Suponha-se que se pretende detetar um vaso numa imagem de fundo. Em geral, esta propriedade significa que o modelo deve ser capaz de identificar o vaso quer ele esteja no centro ou nos limites da imagem e que, além disso, seja qual for a localização do vaso a dificuldade da tarefa deve ser semelhante. A convolução não é naturalmente invariante a outras transformações, como as rotações [77]. Outras estratégias podem ser utilizadas para lidar com estes casos – ver proposta na secção 5.2.3 e respetivos efeitos na secção 6.5.7.

[4] Do inglês: *Sparse Connections*.
[5] Do inglês: *Shared Weights*.

4.1.2.3.1.2 Variantes e Hiperparâmetros

No contexto das CNNs, o conceito de convolução nem sempre corresponde exatamente à descrição formal de convolução discreta. Na verdade, por conveniência ou fatores relacionados com a própria implementação, podem introduzir-se ligeiras nuances [77]. Assim, torna-se pertinente discutir estes aspetos e apresentar os hiperparâmetros a ter em conta no projeto de um modelo deste tipo.

Desde logo, quando se diz que uma CNN se baseia na operação de convolução, em rigor, deveria dizer-se que ela assenta em várias operações de convolução em paralelo – daí a importância que as unidades de processamento gráfico (GPUs[6]) tiveram na afirmação destes métodos [77]. Isto ocorre porque cada filtro só pode extrair uma única *feature*, ainda que em várias localizações. Como cada camada convolucional deve ser capaz de extrair diferentes *features*, utilizam-se, como já se referiu, vários filtros por camada. Este é o primeiro hiperparâmetro a ter em conta numa CNN. Em geral, para representar uma entrada de maiores dimensões pode ser necessário recorrer a um maior número de *features* e, por isso, o passo lógico passa por usar mais filtros por camada. No entanto, este pressuposto é amplamente subjetivo e, *per se*, não garante um modelo mais eficaz [77].

Adicionalmente, a entrada de uma camada convolucional não é, por norma, um *array* 2D. Na prática, lida-se com *arrays* 3D em que duas das dimensões representam a localização espacial (largura e altura) e a outra o número de canais. Assim, as estruturas de dados 3D que fluem através das CNNs são, por vezes, chamadas de tensores [77]. Visto que as operações de convolução se estendem a vários canais, pode ser imperativo reduzir os custos computacionais. Uma das formas de o fazer – a troco de uma extração menos minuciosa da *feature* codificada – é "saltar" algumas posições da entrada, fazendo com que o filtro se mova de um certo número de posições entre cada cálculo, o *stride* (st) [77].

Um recurso essencial ao implementar uma CNN é a possibilidade de fazer *zero padding* à entrada; isto é, preencher as bordas com um número variável de zeros (p). Perante isto, há duas opções típicas. Na convolução válida, abdica-se do *zero padding* fazendo com que o filtro só visite localizações em que esteja totalmente contido na entrada. Assim, todas as unidades de saída são baseadas, exclusivamente, em unidades de entrada e o seu comportamento pode ser mais regular. No entanto, o tamanho dos *feature maps* vai diminuindo à medida que novas camadas convolucionais são usadas e, por isso, o número de camadas é limitado [77]. Na convolução *same*, aplica-se *zero padding* para que o tamanho do *feature map* de saída (s_o) seja igual ao do de entrada (s_i). Aqui não há limite teórico para o número de camadas da rede. Porém, as unidades de saída que "olham" para unidades de entrada situadas nas bordas são menos viáveis que aquelas que se focam nas unidades de entrada próximas do centro [77].

Independentemente da estratégia de *padding* adotada, o tamanho do filtro (s_k) é outro hiperparâmetro a ter em conta. Por um lado, um s_k elevado leva a um de dois efeitos homólogos. Na convolução válida, a redução das dimensões dos *features maps* torna-se mais abrupta e o número limite de camadas convolucionais reduz-se. Na convolução *same*, a quantidade de zeros necessária para assegurar a manutenção destas dimensões é maior e, assim, os seus efeitos notam-se mais rapidamente [77]. Por outro lado, o mesmo s_k elevado parece assegurar um *receptive field* maior [77] – ainda que haja uma forma de compensar este efeito, como se discute na secção 5.2.2.2.2.

[6] Do inglês: *Graphics Processing Unit*.

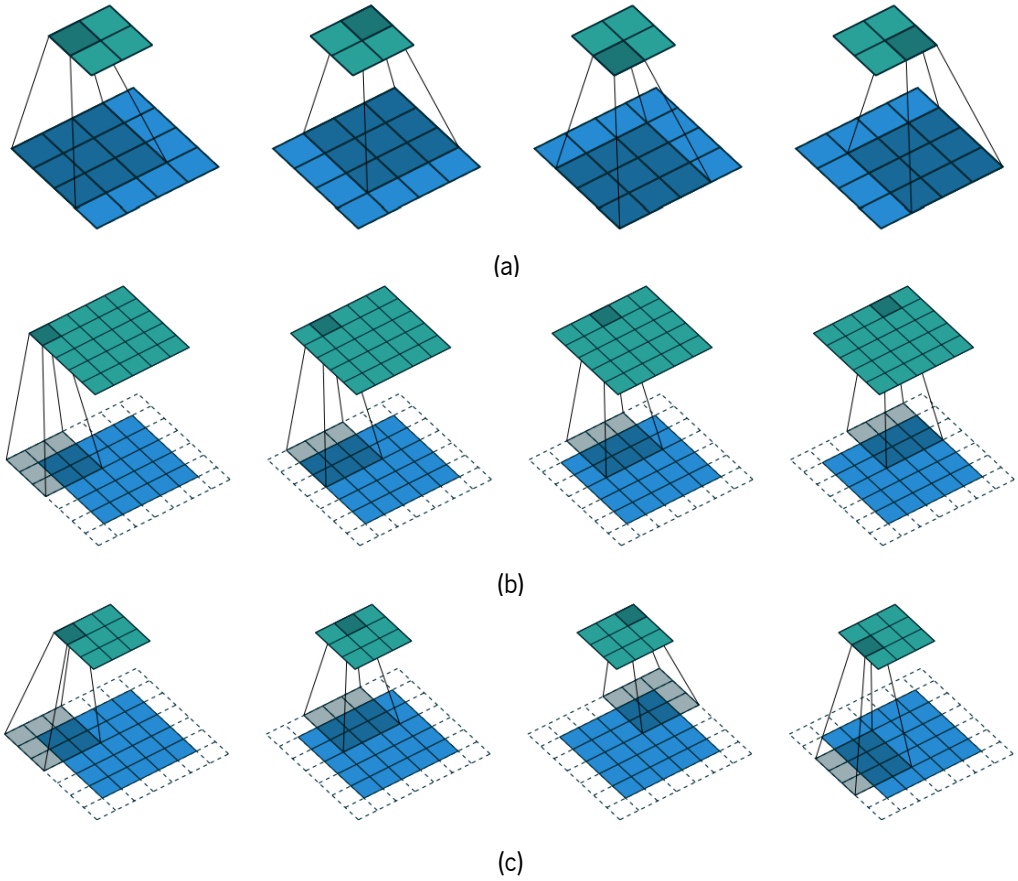


Figura 4.7: Variantes da operação de convolução: (a) Convolução válida c/ *stride* unitário ($p = 0$, $st = 1$); (b) Convolução *same* c/ *stride* unitário ($p = 1$, $st = 1$); (c) Convolução c/ *zero padding* e *stride* não unitário ($p = 1$, $st = 2$). Adaptado de [86].

Em síntese, todos os hiperparâmetros supramencionados podem influenciar o desempenho de uma CNN e acabam por ser projetados conjuntamente. Dados s_i, s_k, p e st, é possível obter s_o [86]:

$$s_o = \frac{s_i + 2p - s_k}{st} + 1. \tag{4.6}$$

A figura 4.7 ilustra três variantes da operação de convolução. Usando a equação acima, é possível validar o tamanho dos *feature maps* de saída para 4.7a, 4.7b e 4.7c, respetivamente:

$$s_o = \frac{4 + 2 \times 0 - 3}{1} + 1 = 2.$$

$$s_o = \frac{5 + 2 \times 1 - 3}{1} + 1 = 5.$$

$$s_o = \frac{5 + 2 \times 1 - 3}{2} + 1 = 3.$$

Teoricamente, nada impede que os *feature maps* e os filtros de uma CNN tenham dimensões arbitrárias. No entanto, por conveniência, estes são quase sempre quadrados [77]. Aqui opta-se por manter essa convenção, pelo que s_i, s_o e s_k bastam para definir por completo as dimensões destas entidades.

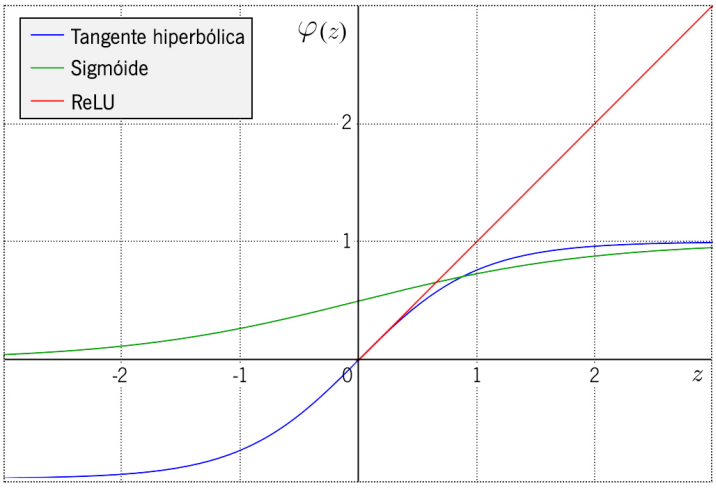


Figura 4.8: Funções de ativação mais comuns. Adaptado de [87].

4.1.2.3.2 Camada de Ativação

Na secção 4.1.2.1, introduziu-se o conceito de função de ativação (φ). Uma camada de ativação não é mais do que a aplicação de φ a todas as unidades de um *feature map*. De forma simples, uma função de ativação é uma transformação não linear. O objetivo é fazer com que a rede não esteja limitada a combinações lineares e, assim, possa identificar fronteiras de decisão arbitrariamente complexas [77]. A definição formal de percetrão simples prevê que φ seja uma função degrau [80]. No entanto, em ANNs com várias camadas (como os MLPs ou as CNNs) isto não é possível. Nestes casos, o treino da rede assenta no mecanismo de *backpropagation* – ver secção 4.1.3.1.3 – e φ deve ser diferenciável [77].

A figura 4.8 compara as funções de ativação mais comuns em ANNs multi-camada. A sigmóide e a tangente hiperbólica são opções tradicionais. Porém, recentemente, a ReLU[7] [88], definida por:

$$\varphi(z) = \max(0, z), \tag{4.7}$$

tornou-se mais usual. A ReLU exibe duas propriedades convenientes. Por um lado, quando $z > 0$, o gradiente é unitário e torna-se mais fácil contornar o fenómeno de *vanishing gradient* (que se relaciona, como será verá mais tarde, com a *backpropagation*). Este é um problema comum nas funções tradicionais, visto que o seu gradiente se torna cada vez mais reduzido à medida que z aumenta. Por outro lado, quando $z < 0$, a ativação resultante da ReLU é nula. Isto cancela algumas ligações, tornando-as mais dispersas e acelerando a convergência [77].

4.1.2.3.3 Camada de Pooling

Numa CNN típica, cada bloco estrutural é composto por três fases. Na primeira, uma camada convolucional realiza várias convoluções em paralelo e gera um mapa de pré-ativações lineares. Depois, cada uma delas é não linearmente transformada por uma função de ativação gerando um *feature map*. Finalmente, este é, ainda, transformado por uma camada de *pooling* [77].

[7] Do inglês: *Rectified Linear Unit*.

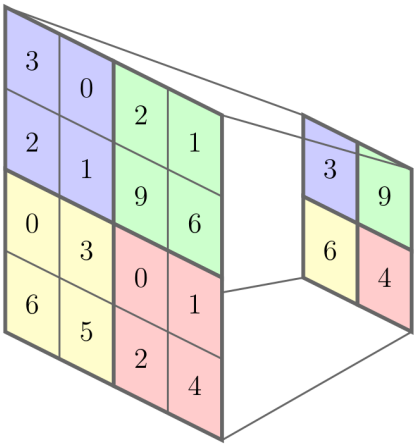


Figura 4.9: Princípio de *max pooling* ($s_k = st = 2$). Reproduzido de [84].

Uma função de *pooling* atua sobre um *feature map* substituindo todas as unidades de uma dada região por uma representação estatística dessa região. Por exemplo, no *max pooling* [89] (figura 4.9) utiliza-se o valor máximo. Noutras variantes comuns como o *average pooling* e o *sum pooling* retornam-se, respetivamente, a média e o somatório das unidades da região. Independentemente da variante, o *pooling* reduz as dimensões dos *feature maps* e visa tornar a representação invariante a pequenas translações. Isto pode ser particularmente útil se a deteção de uma dada *feature* for mais importante que a sua localização exata. Simultaneamente, além de reduzir os custos computacionais, o *pooling* também permite aumentar o *receptive field* de unidades mais profundas [77].

À semelhança da camada convolucional, a camada de *pooling* mantém os hiperparâmetros s_k e st. O primeiro delimita a região-alvo. O segundo restringe as posições que o filtro pode tomar.

4.1.2.3.4 Camada Densa

Nas CNNs convencionais, as últimas camadas são densas (ou FC[8]); isto é, na etapa final da CNN pode identificar-se um MLP (rever figura 4.5). Assim, neste tipo de arquitetura, é possível distinguir duas fases. A primeira, onde atuam as camadas convolucionais e de *pooling*, corresponde à extração de *features*. A segunda, da responsabilidade das camadas densas, diz respeito à classificação [77]. Além do natural aumento do número de parâmetros, as camadas densas acarretam outras condicionantes que levaram à sua substituição nas arquiteturas mais recentes. Este aspeto é discutido na secção 4.1.2.4.

4.1.2.3.5 Camada Softmax

A camada final de uma CNN passa, regra geral, pela aplicação de uma função como a *softmax* [77]:

$$y_i = \frac{e^{\tilde{y}_i}}{\sum_{j=1}^{k} e^{\tilde{y}_j}}. \tag{4.8}$$

Esta operação permite normalizar as k saídas da rede para que a sua soma seja 1. Assim, como já se referiu, a saída y_i pode ser vista como a probabilidade da entrada pertencer à classe i.

[8] Do inglês: *Fully Connected*.

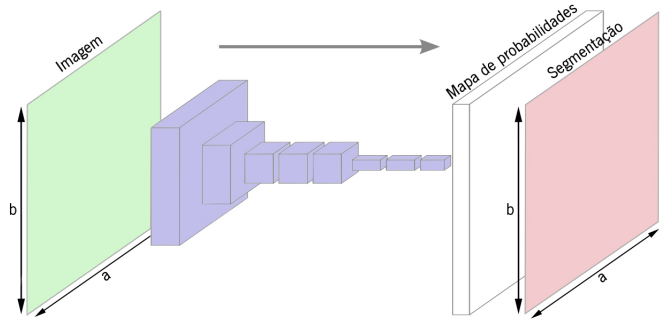


Figura 4.10: Rede neuronal completamente convolucional. Adaptado de [90].

4.1.2.4 Redes Neuronais Completamente Convolucionais

Retomando o tema introduzido na secção 4.1.2.3.4, todas as tradicionais CNNs de reconhecimento de objetos, como a LeNet [69], a AlexNet [72] ou a VGGNet [73], faziam uso de camadas densas. Isto, além de as tornar computacionalmente mais pesadas, dotava-as de algumas características desfavoráveis. Desde logo, como a camada de entrada de um MLP é projetada para lidar com um número particular de *features*, estas redes eram forçadas a usar entradas de tamanho fixo. Depois, como cada pixel era segmentado individualmente, a computação da função de custo não tinha em conta os pixeis vizinhos.

Perante estas limitações, Long *et al.* [90, 91] aperceberam-se que as camadas densas podem ser vistas como camadas convolucionais com $s_k = 1$. Reduzindo o tamanho do filtro a um simples peso, haverá um peso distinto para cada unidade de entrada como se passava inicialmente. Assim, surgiram as redes neuronais completamente convolucionais (FCNs[9]) (figura 4.10). Uma FCN pode tomar uma imagem de qualquer tamanho como entrada e é capaz de retornar um mapa de probabilidades com as mesmas dimensões iniciais, segmentando um *patch* completo de uma só vez e contemplando o custo associado a todos os pixeis que o compõem [90, 91].

Devido à sua versatilidade e eficiência computacional, as FCNs começam a ganhar preponderância sobre as CNNs convencionais e são já amplamente utilizadas em imagem médica. Por isso, este é o modelo base para o trabalho desenvolvido ao longo desta dissertação.

4.1.3 Treino de Redes Neuronais Artificiais

A habilidade para aprender é uma propriedade chave nas ANNs. Uma ANN pode ser usada num contexto de aprendizagem supervisionada ou não supervisionada. Aqui a análise foca-se apenas na primeira, onde se dispõe de um conjunto de M exemplos de treino $(x^{(m)}, d^{(m)})$, com $x^{(m)}$ sendo o m-ésimo vetor de *features* e $d^{(m)}$ a respetiva *label* ou saída desejada. O objetivo da fase de treino passa por encontrar o conjunto de parâmetros ótimo para resolver um dado problema. Trata-se de um processo iterativo onde um algoritmo de otimização compara a *label* $d^{(m)}$ com a solução $y^{(m)}$ gerada pela rede e, com base no gradiente de uma função de custo, ajusta os pesos para melhorar o desempenho. Por norma, a rede diz-se treinada quando se atinge um valor-alvo de desempenho durante a etapa de treino [77, 80].

A fase de treino engloba vários conceitos que se relacionam entre si e que importa conhecer para que se possa treinar um modelo de forma eficaz.

[9] Do inglês: *Fully Convolutional Neural Network*.

4.1.3.1 Otimização

A maioria dos métodos de *machine learning* envolve otimização de alguma forma. Matematicamente, otimizar consiste em minimizar ou maximizar uma função $f(x)$, atuando em x. A função que se pretende minimizar ou maximizar é chamada de função objetivo. Numa tarefa de minimização, em particular, surge o termo função de custo [77, 80].

4.1.3.1.1 Função de Custo

Considere-se um problema de classificação onde se pretende distinguir entre k classes. Uma possível função de custo vem [80]:

$$J(w) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{k}\left[d_j^{(m)} ln(y_j^{(m)}) + (1 - d_j^{(m)}) ln(1 - y_j^{(m)})\right]. \tag{4.9}$$

Esta função é chamada de entropia cruzada (ou *cross-entropy*). Note-se que $y^{(m)}$ é naturalmente dependente do conjunto de pesos w da rede. Por isso, pode escrever-se J como $J(w)$ [77, 80].

Um tema crucial no projeto de uma ANN é que o gradiente da função de custo deve manter-se suficientemente elevado para que o algoritmo de otimização possa convergir. Funções que saturam facilmente dificultam esse objetivo, porque reduzem o gradiente até valores muito baixos. Numa ANN, isso pode ocorrer por ação das funções de ativação das unidades ocultas ou pelas funções com efeito normalizador das unidades de saída. A função de custo é, normalmente, projetada para lidar com as últimas. A entropia cruzada, devido ao logaritmo natural, é particularmente eficaz perante uma transformação exponencial, como a função *softmax*, que pode saturar quando o seu argumento é muito negativo [77].

4.1.3.1.2 Gradient Descent

Esqueça-se, por momentos, a função de entropia cruzada e suponha-se agora uma função $j : \mathbb{R} \to \mathbb{R}$ com $j(w) = \frac{1}{2}w^2$. A derivada desta função $j'(w)$ ou $\frac{dj}{dw}$ traduz o declive de $j(w)$ no ponto w. Por outras palavras, ela especifica o fator através do qual uma pequena variação (ϵ) na entrada deve ser escalada para produzir a variação correspondente na saída: $j(w + \epsilon) \approx j(w) + j'(w)\epsilon$ [77].

Assim, a derivada pode ser usada para minimizar uma função através de sucessivas pequenas reduções. Em particular, é possível minimizar $j(w)$ "movendo" w na direção oposta ao sinal da derivada. Este algoritmo de otimização é conhecido como *gradient descent* (figura 4.11) [77, 80]. Quando $j'(w) = 0$, o algoritmo termina. Atinge-se um ponto crítico que pode ser um mínimo (ou máximo) global ou local. Há até pontos que, sendo críticos, não são mínimos nem máximos (pontos de sela) [77, 80].

O objetivo óbvio durante um problema de minimização é obter um mínimo global. Porém, a não linearidade das ANNs torna as funções de custo mais interessantes não convexas. Assim, principalmente em *deep learning*, é frequente otimizar funções com inúmeros mínimos locais e vários pontos de sela. Tudo isto dificulta a tarefa, especialmente face a uma entrada multidimensional. Por isso, o treino geralmente cessa quando se atinge um valor suficientemente baixo para a função de custo, mas não necessariamente mínimo no sentido formal [77, 80].

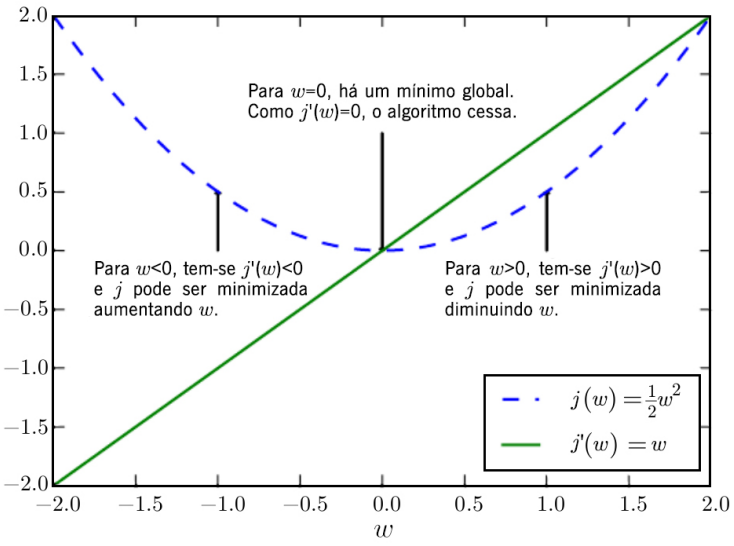


Figura 4.11: *Gradient descent*. Adaptado de [77].

Voltando ao contexto inicial, na esmagadora maioria dos problemas reais otimizam-se funções de custo, como a entropia cruzada $J(w)$, em que $w \notin \mathbb{R}$. w é sim um conjunto de cardinalidade variável conforme a arquitetura da ANN. Aquilo que, invariavelmente, se mantém é que $J(w) \in \mathbb{R}$ – caso contrário, o termo "minimização" deixaria de fazer sentido [77].

Nestas condições, o conceito de derivada é substituído pelo de gradiente $\nabla_w J(w)$, que engloba as várias derivadas parciais de $J(w)$ em ordem a cada elemento de w. Posto isto, é possível descrever formalmente o algoritmo de *gradient descent* como [77]:

$$w \leftarrow w - \alpha \nabla_w J(w), \tag{4.10}$$

onde $\leftarrow$ simboliza a operação de atribuição e $\alpha (\in]0, 1])$ é um hiperparâmetro de grande importância conhecido como *learning rate*. Informalmente, α controla a extensão do "passo" que se dá em cada iteração. Do ponto de vista teórico, é sempre possível fazer uma ANN convergir se α for suficientemente reduzido; no entanto, no limite, isto levaria a uma etapa de treino interminável. Por outro lado, um α demasiado elevado pode fazer com que a ANN não convirja ou, até, que divirja [80].

4.1.3.1.3 Backpropagation

Como se exemplificou na secção 4.1.2.2, quando se utiliza uma ANN para produzir uma saída $y^{(m)}$, dados uma entrada $x^{(m)}$ e um conjunto de pesos w, está-se perante um processo de propagação direta. Durante o treino, $y^{(m)}$ e $d^{(m)}$ permitem obter o custo através de $J(w)$. O algoritmo de *backpropagation*, também chamado de *backprop* ou retro-propagação, é o processo através do qual a informação flui no sentido contrário (da saída para a entrada) de forma a que se possa calcular $\nabla_w J(w)$ [77, 80].

O termo *backpropagation* é, frequentemente, confundido com o algoritmo de otimização que permite treinar uma ANN. Na verdade, este processo diz apenas respeito à computação do gradiente, enquanto outro algoritmo, como o *gradient descent*, é responsável por atualizar os pesos com base nele [77].

O algoritmo de *backpropagation* pode ser entendido com base num só exemplo de treino $(x^{(m)}, d^{(m)})$ e na respetiva saída $y^{(m)}$ [77, 92]. Deixando cair os sobrescritos, por simplicidade, tem-se (x, d) e y.

Para descrever o processo formalmente, relembre-se alguma da notação já utilizada. Dada uma ANN com c camadas, $L^{(1)}$, ..., $L^{(c)}$, $a_j^{(l)}$ e $z_j^{(l)}$ representam, respetivamente, a ativação e pré-ativação linear do neurónio j da camada $L^{(l)}$. Desta forma, a equação 4.1 pode ser generalizada para qualquer neurónio:

$$a_j^{(l)} = \varphi(z_j^{(l)}) = \varphi\left(\sum_i {w_{ji}}^{(l-1)} a_i^{(l-1)} + b^{(l)}\right). \tag{4.11}$$

Recorde-se, também, que $b^{(l)}$ é o *bias* da camada $L^{(l)}$ e ${w_{ji}}^{(l-1)}$ ($\in \mathbb{R}$) representa o peso que liga o neurónio i da camada $L^{(l-1)}$ ao neurónio j da camada $L^{(l)}$. Anteriormente, definiu-se *gradient descent* para um conjunto total de pesos w. Por conveniência, é possível fazê-lo para cada ${w_{ji}}^{(l-1)}$:

$${w_{ji}}^{(l-1)} \leftarrow {w_{ji}}^{(l-1)} - \alpha \frac{\partial J(w)}{\partial {w_{ji}}^{(l-1)}}. \tag{4.12}$$

Assim, o objetivo passa agora por calcular a derivada parcial $\frac{\partial J(w)}{\partial {w_{ji}}^{(l-1)}}$, que traduz a variação de $J(w)$ em relação a ${w_{ji}}^{(l-1)}$. Utilizando a regra da cadeia, pode escrever-se [92]:

$$\frac{\partial J(w)}{\partial {w_{ji}}^{(l-1)}} = \frac{\partial J(w)}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial {w_{ji}}^{(l-1)}}. \tag{4.13}$$

Paralelamente, Haykin [92] define o potencial do neurónio j na camada $L^{(l)}$ como:

$$\delta_j^{(l)} = \frac{\partial J(w)}{\partial z_j^{(l)}} = \frac{\partial J(w)}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}}. \tag{4.14}$$

Com base nesta convenção, a equação 4.13 torna-se simplesmente [92]:

$$\frac{\partial J(w)}{\partial {w_{ji}}^{(l-1)}} = \delta_j^{(l)} \frac{\partial z_j^{(l)}}{\partial {w_{ji}}^{(l-1)}}. \tag{4.15}$$

Recorrendo à equação 4.11, o último termo da equação anterior pode ser escrito como [92]:

$$\frac{\partial z_j^{(l)}}{\partial {w_{ji}}^{(l-1)}} = \frac{\partial \left(\sum_i {w_{ji}}^{(l-1)} a_i^{(l-1)} + b^{(l)}\right)}{\partial {w_{ji}}^{(l-1)}} = a_i^{(l-1)}. \tag{4.16}$$

Então, a equação 4.15 pode tornar-se ainda mais simples [92]:

$$\frac{\partial J(w)}{\partial {w_{ji}}^{(l-1)}} = \delta_j^{(l)} a_i^{(l-1)}. \tag{4.17}$$

A equação 4.17 permite expressar a derivada parcial de $J(w)$ em relação a qualquer ${w_{ji}}^{(l-1)}$, com $l \in \{2, ..., c\}$. Informalmente, o potencial $\delta_j^{(l)}$ pode ser interpretado como o erro associado ao neurónio j da camada $L^{(l)}$, por isso, $\delta^{(1)}$ não está definido (visto que $a^{(1)} = x$) [92].

A chave do algoritmo de *backpropagation* está em entender que $\delta_j^{(l)}$ não pode ser obtido da mesma forma independentemente da camada $L^{(l)}$.

Quando $l = c$, está-se na camada de saída $L^{(c)}$ e aplicando a equação 4.14, vem [92]:

$$\delta_j^{(c)} = \frac{\partial J(w)}{\partial z_j^{(c)}} = \frac{\partial J(w)}{\partial a_j^{(c)}} \frac{\partial a_j^{(c)}}{\partial z_j^{(c)}} = \frac{\partial J(w)}{\partial y_j} \varphi'(z_j^{(c)}). \tag{4.18}$$

Neste caso, a obtenção de $\delta_j^{(c)}$ é direta porque existe um valor desejado d_j para cada y_j e, portanto, é possível derivar a função de custo $J(w)$ em ordem a y_j [92].

Quando $l \neq c$, está-se numa camada oculta e isto deixa de ser verdade. Veja-se que, agora, já não há um valor desejado para a ativação $a_j^{(l)}$ [92]. Assim, o raciocínio precisa ser diferente. Se $L^{(l)}$ é uma camada oculta, então existe $L^{(l+1)}$. Com isto em mente, continue a chamar-se j a cada neurónio da camada $L^{(l)}$ e introduza-se h para denominar cada neurónio da camada $L^{(l+1)}$ (figura 4.12).

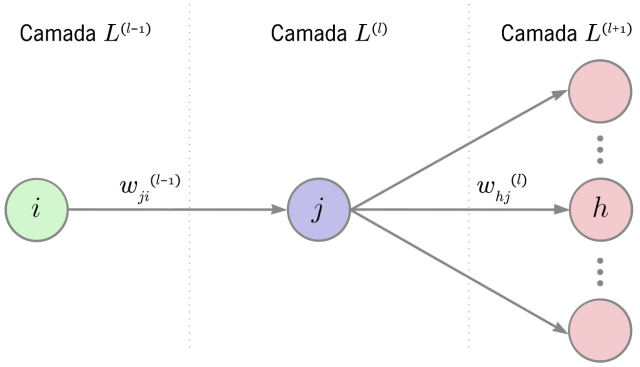


Figura 4.12: Três camadas consecutivas de uma ANN e respetivos neurónios i, j e h.

Isto permite escrever [92]:

$$\frac{\partial J(w)}{\partial a_j^{(l)}} = \sum_h \frac{\partial J(w)}{\partial z_h^{(l+1)}} \frac{\partial z_h^{(l+1)}}{\partial a_j^{(l)}}. \tag{4.19}$$

Recuperando a definição de $\delta_j^{(l)}$ dada na equação 4.14, a equação anterior torna-se [92]:

$$\frac{\partial J(w)}{\partial a_j^{(l)}} = \sum_h \delta_h^{(l+1)} \frac{\partial z_h^{(l+1)}}{\partial a_j^{(l)}}. \tag{4.20}$$

E sabe-se que [92]:

$$\frac{\partial z_h^{(l+1)}}{\partial a_j^{(l)}} = \frac{\partial \left(\sum_j {w_{hj}}^{(l)} a_j^{(l)} + b^{(l+1)} \right)}{\partial a_j^{(l)}} = {w_{hj}}^{(l)}. \tag{4.21}$$

Então, a equação 4.20 pode ser ainda mais simplificada [92]:

$$\frac{\partial J(w)}{\partial a_j^{(l)}} = \sum_h \delta_h^{(l+1)} {w_{hj}}^{(l)}. \tag{4.22}$$

Finalmente, combinando as Equações 4.14 e 4.22, pode escrever-se [92]:

$$\delta_j^{(l)} = \left(\sum_h \delta_h^{(l+1)} {w_{hj}}^{(l)} \right) \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} = \left(\sum_h \delta_h^{(l+1)} {w_{hj}}^{(l)} \right) \varphi'(z_j^{(l)}). \tag{4.23}$$

Esta é, talvez, a expressão que melhor representa a ideia por detrás do algoritmo de *backpropagation*. Em particular, dados os potenciais (ou erros) $\delta_h^{(l+1)}$ de todos os neurónios h da camada $L^{(l+1)}$ é possível encontrar o potencial (ou erro) $\delta_j^{(l)}$ do neurónio j da camada anterior $L^{(l)}$. Assim, partindo dos potenciais da última camada que podem ser calculados diretamente, é possível obter o potencial de qualquer neurónio da rede através de um processo que se dirige da saída para a entrada [77, 92].

Por último, combinando as Expressões 4.12 e 4.17, é possível voltar ao objetivo inicial e explicitar uma atualização ao peso ${w_{ji}}^{(l-1)}$ segundo o método de *gradient descent*:

$$ {w_{ji}}^{(l-1)} \leftarrow {w_{ji}}^{(l-1)} - \alpha \delta_j^{(l)} a_i^{(l-1)}, \tag{4.24} $$

onde $l \in \{2, ..., c\}$ e, com base nas Equações 4.18 e 4.23, se tem [92]:

$$ \delta_j^{(l)} = \begin{cases} \frac{\partial J(w)}{\partial y_j} \varphi'(z_j^{(l)}), & l = c \\ \left(\sum_h \delta_h^{(l+1)} {w_{hj}}^{(l)} \right) \varphi'(z_j^{(l)}). & l \neq c \end{cases} \tag{4.25} $$

Note-se a presença de φ', pelo que a função de ativação tem de ser diferenciável. Mais do que isso, veja-se que φ' atua como um fator de escala no cálculo de cada gradiente. O mesmo é dizer que quando este termo toma valores muito baixos os gradientes são significativamente diminuídos. Como o gradiente de uma camada depende dos da camada seguinte, gradientes de camadas mais próximas da entrada poderão tornar-se sucessivamente menores. Este fenómeno é conhecido como *vanishing gradient*. O grande obstáculo que surge daqui reside no facto das atualizações aos pesos das camadas iniciais se tornarem demasiado ténues e impedirem que se acondicione devidamente os dados que são fornecidos à entrada. Assim, além de uma inicialização cuidada – ver secção 4.1.3.2 –, é importante assegurar que a função de ativação usada exibe um gradiente que se mantém entre valores convenientes (em especial, que não se torna muito inferior à unidade) [77, 92].

A descrição que aqui se apresentou aplica-se diretamente a uma ANN convencional (como o MLP) e pretende dar alguma intuição sobre o funcionamento deste algoritmo. Numa CNN, os conceitos e limitações mantêm-se (havendo nuances relacionadas com cada topologia) [77].

4.1.3.1.4 Stochastic Gradient Descent

Um problema recorrente em *deep learning* é o facto de um número de exemplos de treino elevado ser favorável para obter melhor capacidade de generalização, mas tornar o processo de otimização computacionalmente mais pesado [77].

Como se viu na secção 4.1.3.1.1, uma função de custo $J(w)$ pode ser, geralmente, decomposta como uma soma que percorre cada exemplo de treino $(x^{(m)}, d^{(m)})$. Assim, o algoritmo de *gradient descent* requer o cálculo de [77]:

$$ \nabla_w J(w) = \frac{1}{M} \sum_{m=1}^{M} \nabla_w J(d^{(m)}, y^{(m)}, w). \tag{4.26} $$

O mesmo é dizer que o gradiente deve ser calculado, através do algoritmo de *backpropagation*, para cada exemplo de treino e que, na etapa final, se toma a média entre todos os exemplos. Por outras palavras, em cada época – uma passagem por todos os M exemplos de treino – faz-se apenas uma iteração (ou seja, uma atualização aos pesos). Nestas condições, à medida que M aumenta, o tempo necessário para que o *gradient descent* execute uma única iteração torna-se proibitivo [77].

O princípio base do *stochastic gradient descent* (SGD) consiste em aproximar o gradiente usando um número reduzido de exemplos [77, 80]. Em particular, cada iteração baseia-se num conjunto (*minibatch*) de exemplos, $\mathbb{B} = \{(x^{(1)}, d^{(1)}), ..., (x^{(\hat{M})}, d^{(\hat{M})})\}$, selecionados aleatoriamente [77]. $\hat{M}$ representa o número de elementos do *minibatch* e é visto como um hiperparâmetro. Em geral, um $\hat{M}$ elevado implica maior custo computacional, mas confere uma estimativa mais fidedigna do gradiente. Por outras palavras, um conjunto com mais exemplos permite reduzir a variância entre as várias aproximações do gradiente e, em teoria, obter uma direção segundo a qual é possível dar um "passo" maior em cada iteração, acelerando a convergência (ainda que se cumpram menos iterações por época). No entanto, a certo ponto, a estimativa deixará de beneficiar de um maior número de exemplos, pelo que a computação extra será desperdiçada [77, 80]. Formalmente, esta estimativa do gradiente é dada por [77]:

$$\nabla_w \hat{J}(w) = \frac{1}{\hat{M}} \sum_{m=1}^{\hat{M}} \nabla_w J(d^{(m)}, y^{(m)}, w). \tag{4.27}$$

E, assim, o algoritmo SGD traduz-se em [77]:

$$w \leftarrow w - \alpha \nabla_w \hat{J}(w). \tag{4.28}$$

Em resumo, a característica mais importante do SGD reside no facto do tempo de computação por iteração não depender do número de exemplos de treino. Assim, é possível atingir a convergência, num período razoável, mesmo quando o conjunto de treino se torna muito vasto [77].

4.1.3.1.4.1 Momento

Apesar de muito popular, a aprendizagem através do SGD traz alguns inconvenientes. O facto da seleção do *minibatch* ser feita aleatoriamente torna possível que, durante várias iterações sucessivas, a função de custo praticamente não decresça, atingindo-se um *plateau*. Além disso, pontualmente, algumas iterações podem até levar a que a função de custo cresça. Isto retarda a convergência [77].

O método do momento [93] visa mitigar estas limitações, tornando a otimização mais estável e rápida, através da introdução de uma variável v, que é atualizada em cada iteração:

$$v \leftarrow \mu v - \alpha \nabla_w \hat{J}(w). \tag{4.29}$$

O termo v significa velocidade e acumula a estimativa do gradiente de iterações anteriores. Quanto maior o momento (μ) em relação a α, mais forte se torna a influência das estimativas anteriores sobre a direção da iteração atual [77]. Já quanto à regra de atualização, vem simplesmente [77]:

$$w \leftarrow w + v. \tag{4.30}$$

4.1.3.1.4.2 Momento de Nesterov

O momento de Nesterov [94, 95] mantém a regra de atualização do momento convencional, mas introduz uma ligeira nuance no que diz respeito a v:

$$v \leftarrow \mu v - \alpha \nabla_w \hat{J}(w + \mu v). \tag{4.31}$$

A diferença entre as duas alternativas reside na avaliação do gradiente que, neste caso, só ocorre após a aplicação da velocidade atual. Assim, este método pode ser visto como uma tentativa de adicionar um fator de correção ao anterior [77].

4.1.3.2 Inicialização

Alguns algoritmos de otimização são não iterativos e podem resolver-se diretamente em ordem a uma solução ótima. Outros, ainda que iterativos por natureza, convergem para soluções aceitáveis, quando convenientemente aplicados, independentemente da inicialização [80]. No treino de modelos de *deep learning*, desafortunadamente, nenhuma destas premissas se verifica. Algoritmos como o SGD são, de facto, iterativos e requerem uma inicialização para que se possa começar a iterar. Todavia, a complexidade da tarefa torna a escolha inicial preponderante no desenrolar do processo [77].

Em casos extremos, os parâmetros iniciais podem mesmo impedir o modelo de convergir. Na maioria dos casos, isto não acontece; porém, a inicialização permanece determinante no tempo de convergência e no custo, mais ou menos elevado, da própria solução [77].

A maioria das estratégias de inicialização visam adquirir algumas propriedades alvo quando a ANN é inicializada. No entanto, o entendimento sobre esta matéria é, ainda, algo limitado visto que não se sabe exatamente que propriedades são mantidas durante a aprendizagem e, quando mantidas, em que condições isso se verifica [77, 80].

Uma propriedade – talvez a única [77] – que se toma como imperativa é que os parâmetros iniciais devem "quebrar a simetria" entre neurónios. Se dois neurónios, com a mesma função de ativação, estão ligados às mesmas entradas e possuem parâmetros iguais, então o mesmo algoritmo de otimização aplicado à mesma função de custo irá atualizá-los, sempre, da mesma forma. Isto reduz a versatilidade do modelo e, ainda que haja estratégias para mitigar este problema, é globalmente aceite que cada neurónio deve ser inicializado de forma a que compute uma função diferente dos demais [77, 80].

Este objetivo motiva a inicialização aleatória dos parâmetros. Tipicamente, os *biases* são inicializados com uma dada constante, enquanto nos pesos se recorre a uma distribuição uniforme ou gaussiana. A escolha do tipo de distribuição parece secundária e tem sido pouco debatida; já a escala dessa distribuição revela-se crucial. Pesos iniciais maiores têm maior capacidade para "quebrar a simetria" entre neurónios e parecem contribuir para que a informação flua, no sentido direto e reverso, através do aumento das pré-ativações (z) de cada neurónio. No entanto, quando estas pré-ativações se tornam extremas podem gerar a saturação das funções de ativação e potenciar o *vanishing gradient*. Assim, o objetivo passa por encontrar o melhor compromisso entre estes fatores [77, 80].

4.1.3.2.1 Inicialização de Xavier

A maioria dos métodos de inicialização reside em estratégias heurísticas. Uma das mais comuns é inicializar os pesos de uma camada com i entradas e j saídas extraíndo-os de $U(-\frac{1}{\sqrt{i}}, \frac{1}{\sqrt{i}})$, onde U representa a distribuição uniforme [80].

A inicialização de Xavier [96] propõe algo diferente:

$$W \approx U\left(-\frac{\sqrt{6}}{\sqrt{i+j}}, \frac{\sqrt{6}}{\sqrt{i+j}}\right). \tag{4.32}$$

De forma muito simplicada, esta estratégia visa fazer com que os pesos sejam altos o suficiente para manter a variância entre ativações de camadas diferentes, sem que se tornem tão elevados que impeçam que isso aconteça no caso dos gradientes. A proposta pressupõe que a rede consiste apenas numa cadeia de operações lineares. Nas ANNs reais, obviamente, isso não acontece. No entanto, os efeitos positivos parecem manter-se quando as não linearidades são tidas em conta [77].

4.1.3.3 Generalização e Capacidade

Até aqui, tem-se descrito o treino de uma ANN como uma etapa cujo principal objetivo é otimizar uma função de custo, de forma a minimizar o erro global num conjunto de amostras cujas soluções se conhecem à partida. Referiu-se, por exemplo, que um algoritmo de otimização como o SGD procura encontrar, num período de tempo razoável, uma solução ótima com um custo (ou erro de treino) mínimo.

Porém, um problema de *machine learning* em geral, e de *deep learning* em particular, não é meramente um problema de otimização. O desafio central é criar modelos capazes de obter bons resultados num conjunto de exemplos não analisados até então – e não apenas naqueles em que foram treinados. Por outras palavras, o que se pretende minimizar, em primeira instância, é o erro de teste. Porém, na ausência de uma alternativa direta, tenta-se atingir este objetivo, indiretamente, minimizando o erro de treino. A consequência que emerge é fácil de explicar (mas difícil de contornar): o facto de um modelo apresentar um ótimo desempenho no conjunto de treino não implica que o mesmo se passe no conjunto de teste. Aliás, como se verá, desempenhos demasiadamente favoráveis no primeiro implicam, na maioria das vezes, resultados surpreendentemente negativos no segundo [77].

Tendo em conta esta necessidade de exibir um desempenho coerente no conjunto de teste, face ao que se passava no conjunto de treino, surge o conceito de capacidade de generalização. Note-se que este conceito é tão importante que reformula a forma como se olha para o processo de treino. Por exemplo, voltando à discussão inicial, uma solução ótima não é simplesmente aquela que leva à obtenção do menor custo; é sim a que, além disso, permite ao modelo generalizar melhor. Posto de forma simples, duas soluções podem ter exatamente o mesmo custo no conjunto de treino e, no entanto, uma delas ser significativamente melhor que a outra no conjunto de teste [77].

Em resumo, os fatores que definem a efetividade de um modelo são a sua habilidade para: (1) minimizar o erro de treino; e (2) manter a diferença entre os erros de treino e teste tão pequena quanto possível. Por isso, sem surpresa, cada um deles introduz um problema chave em *machine learning* [77].

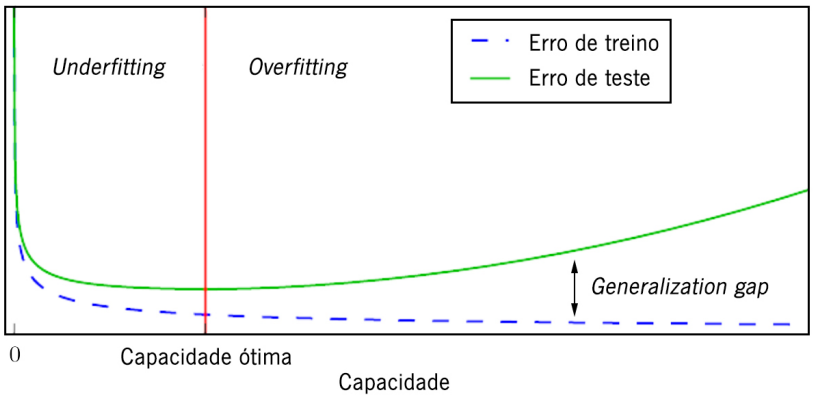


Figura 4.13: Relação típica entre a capacidade de um modelo e os erros de treino e teste: regimes de *underfitting* e *overfitting*. Adaptado de [77].

4.1.3.3.1 Underfitting e Overfitting

Quando o desempenho de um modelo é pobre, a causa reside, por norma, em um de dois fenómenos: se o modelo não é capaz de reduzir suficientemente o erro de treino ocorre *underfitting*; se a diferença entre os erros de treino e teste é demasiado elevada existe *overfitting* [77].

A principal forma de alterar a tendência de um modelo para apresentar um destes problemas passa por ajustar a sua capacidade que, informalmente, não é mais do que a gama de funções que ele pode representar. Uma capacidade insuficiente gera problemas para obter desempenhos satisfatórios no conjunto de treino; já uma capacidade excessiva leva à extração de *features* que não trazem benefício perante um novo conjunto de exemplos. A figura 4.13 ilustra o comportamento típico. Para capacidades reduzidas, tanto o erro de treino como o de teste são elevados. Este é o regime de *underfitting*. À medida que a capacidade de um modelo aumenta, o erro de treino diminui, mas a diferença para o erro de teste torna-se maior. Eventualmente, esta distância (conhecida como *generalization gap*) pode tornar-se mais significativa que a redução do erro de treino e atinge-se o regime de *overfitting*. Assim, um dos maiores desafios no projeto de um modelo de *machine learning* consiste em encontrar a capacidade ótima [77].

Outro dos aspetos a considerar é a duração da etapa de treino. Até um modelo com a capacidade ótima para um dado problema pode entrar em *underfitting* ou *overfitting* se o treino se estender por um período demasiado curto ou longo, respetivamente [79, 80]. De facto, seria também possível interpretar o gráfico da figura 4.13 em ordem ao número de épocas de treino e este continuaria a fazer sentido.

Num problema real, o controlo destes fatores passa, tipicamente, pela avaliação do modelo num conjunto de validação. Este consiste num sub-conjunto de exemplos de treino, que não é usado para esse fim, e tem como finalidade aferir o comportamento do modelo perante amostras não vistas anteriormente, para que se possa induzir o que ocorrerá no conjunto de teste [79, 80].

Regra geral, enquanto a duração do treino pode ser controlada eficazmente comparando os erros de treino e validação, a obtenção da capacidade apropriada a um dado problema exige alguma sensibilidade. A principal forma de dotar um modelo de maior capacidade passa por aumentar o seu número de pesos. Numa CNN, isto traduz-se num maior número de camadas convolucionais ou no uso de mais filtros por camada. Por isso, quanto mais profunda for uma rede, maior é a sua capacidade. Dito isto, um modelo

de *deep learning* tem, por definição, grande capacidade e dá-se melhor quando o conjunto de treino disponível é considerável [77]. Por outro lado, se o objetivo for reduzir a capacidade do modelo existem alternativas para o fazer sem que se tenha, necessariamente, de reduzir o número de pesos [77, 80].

4.1.3.4 Regularização

A grande capacidade característica dos modelos de *deep learning* faz com que, na esmagadora maioria dos casos, se torne imperativo controlá-la. Por outras palavras, há várias estratégias desenhadas explicitamente para reduzir o erro de teste, ainda que isso implique aumentar o erro de treino. Estas estratégias são globalmente conhecidas como regularização [77]. Regra geral, as estratégias de regularização podem atuar de duas formas: a primeira passa por adicionar novos termos à função de custo; a segunda baseia-se na inclusão de restrições no próprio modelo [77].

4.1.3.4.1 Weight Decay

O *weight decay* (ou regularização L^2) é uma estratégia que visa diminuir o valor absoluto dos pesos, introduzindo um novo termo na função de custo. Em particular, dada uma função de custo $J(w)$ a sua versão regularizada vem [77]:

$$J_R(w) = J(w) + \lambda w^T w, \tag{4.33}$$

onde λ é um hiperparâmetro que controla a intensidade da regularização ou, por outras palavras, o quão grande é a preferência por pesos mais pequenos.

Minimizar $J_R(w)$ torna-se, assim, num compromisso entre obter um erro de treino reduzido e tornar o valor absoluto dos pesos mais pequeno. Na prática, isto faz com que o modelo seja menos versátil e tenha mais dificuldade para representar funções que exibam variações bruscas. O mesmo é dizer que a capacidade é diminuída e que, por isso, a tendência para fazer *overfitting* se torna menor [77].

A escolha de λ é crucial. Quando $\lambda = 0$, o efeito da regularização é nulo e a propensão para o *overfitting* mantém-se. Por isso, é imperativo selecionar um valor capaz de produzir efeitos práticos. Porém, quando λ se torna demasiado elevado o modelo começa a recorrer exclusivamente a pesos diminutos e vê a sua capacidade tão reduzida que não é sequer capaz de lidar com o conjunto de treino. Por outras palavras, passa-se do *overfitting* ao *underfitting* [77].

4.1.3.4.2 Dropout

O *dropout* [97, 98] é uma abordagem recente que interpreta a regularização de uma forma inovadora. A principal ideia desta estratégia consiste em anular (*drop*) algumas unidades e respetivas ligações durante a etapa de treino (figura 4.14). A escolha das unidades a anular é aleatória e cada unidade é anulada segundo uma probabilidade p independente das demais unidades. Na etapa de teste, todas as unidades são mantidas e os pesos associados a cada unidade são escalados por um fator de p [98].

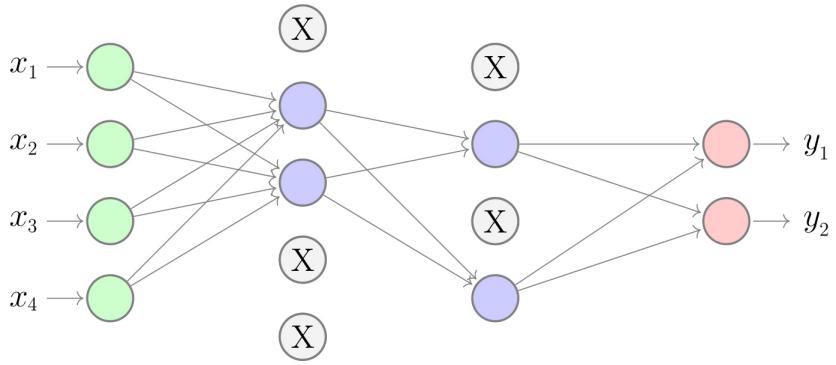


Figura 4.14: *Dropout*. Unidades anuladas são marcadas com 'X'. Adaptado de [84].

O principal objetivo do *dropout* passa por reduzir a coadaptação entre unidades vizinhas. Na ausência desta estratégia, a topologia da rede é mantida e as unidades são sempre rodeadas pelos mesmos vizinhos. Isto pode favorecer o aparecimento de uma certa codependência. Em alternativa, o *dropout* anula unidades, de forma aleatória, em cada iteração e força constantemente cada unidade a novos contextos. Isto pode levar a que as unidades aprendam *features* mais interessantes, de forma independente, porque os vizinhos nem sempre estão presentes para colaborar nesse processo. Assim, potencia-se uma capacidade de generalização maior e reduz-se a tendência para o *overfitting* [97, 98].

4.1.3.4.2.1 Dropout Espacial

O *dropout* espacial [99] é uma variante que surge no contexto das CNNs. O argumento é que mesmo que se cancelem algumas unidades de um mesmo *feature map*, as que permanecem ativas continuam a estar fortemente coadaptadas. Assim, a estratégia passa por anular *feature maps* por completo e não apenas algumas unidades. Desta forma, as unidades de um *feature map* partilham sempre o mesmo estado durante o treino, seja ele de ativação ou de inativação [99].

4.1.3.4.3 Aumento Artificial de Dados

Como se referiu anteriormente, por norma, os modelos de *deep learning* (e de *machine learning* em geral) generalizam melhor quando são treinados em mais dados. De forma muito simplificada, isto ocorre porque o modelo olha para mais pontos de referência e torna-se mais difícil representar uma função que se ajuste perfeitamente a todos eles. Assim, reduz-se a tendência para o *overfitting* [77].

O grande problema é que a quantidade de dados disponível é limitada, ainda mais na área médica. Assim, muitas vezes, recorre-se ao aumento artificial de dados (ou *data augmentation*), através do qual se criam novos exemplos de treino artificiais através da modificação dos exemplos disponíveis inicialmente. Operações como translações e rotações são bastante comuns. O principal desafio que surge na aplicação de transformações mais complexas reside na necessidade imperiosa de não descaracterizar os exemplos de treino iniciais e de manter uma correspondência coerente entre $x^{(m)}$ e $d^{(m)}$ [77].

4.2 Transformada de Wavelet

A transformada de wavelet encontra diversas aplicações em áreas como a matemática, a física e a engenharia. Trata-se, também, de uma ferramenta frequentemente usada em processamento de sinal e imagem [100]. Aqui, faz-se uma breve descrição de alguns dos conceitos básicos da transformada de wavelet de tempo contínuo e discreto e introduz-se a transformada de wavelet não decimada.

4.2.1 Transformada de Wavelet de Tempo Contínuo

Os sinais encontrados na maioria das aplicações práticas exibem componentes de alta frequência durante segmentos temporais muito curtos e componentes de baixa frequência em segmentos temporais mais longos. Perante este tipo de sinais, oriundos de sistemas não estacionários, técnicas como a transformada de Fourier não são a melhor alternativa porque apesar de identificarem as várias componentes espectrais de um sinal, não fornecem informação sobre os instantes em que elas ocorrem [100].

A transformada de wavelet de tempo contínuo (CWT[10]) permite que cada componente espectral possa ser visualizada com um compromisso diferente entre resolução temporal e espectral e, assim, colmata as limitações anteriores. Formalmente, a CWT de um sinal $x(t)$ pode ser descrita como [100]:

$$\Psi_{\tau,s}(t) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t-\tau}{s} \right) dt, \tag{4.34}$$

onde s representa a escala, τ o deslocamento no tempo e ψ^* o complexo conjugado da wavelet mãe.

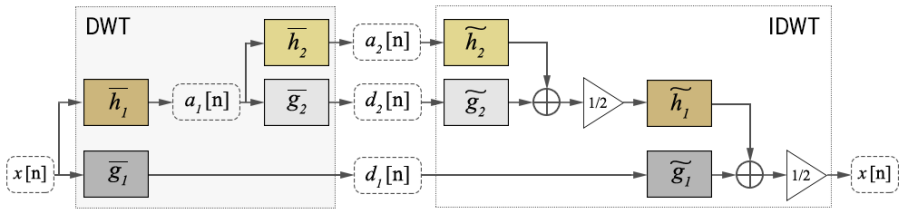
Em geral, uma wavelet não é mais do que uma função de energia finita e valor médio nulo. De forma homóloga ao que ocorre na transformada de Fourier, onde se decompõe um sinal em várias sinusóides de frequências distintas, na transformada de wavelet cada sinal é decomposto em várias versões escaladas e deslocadas de uma mesma wavelet (a wavelet mãe). Assim, o resultado da CWT é um conjunto de coeficientes, que dizem respeito a cada uma dessas versões. Os coeficientes permitem não só obter as wavelets constituintes do sinal através da multiplicação pela versão correspondente da wavelet mãe, mas também recuperar o sinal inicial através da CWT inversa [100].

Uma das várias aplicações destes coeficientes consiste, como se viu no capítulo 3, no realce dos vasos da retina. É nesse contexto que estes conceitos são úteis no âmbito desta dissertação.

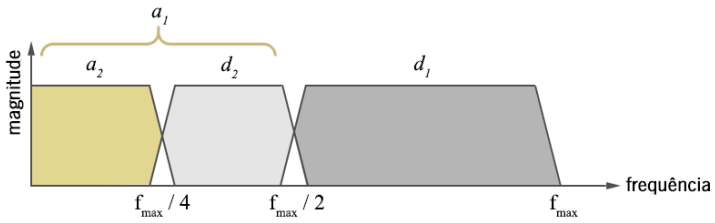
4.2.2 Transformada de Wavelet de Tempo Discreto

Armazenar os coeficientes da transformada de wavelet para todas as combinações de escala e deslocamento possíveis exigiria memória ilimitada. Assim, na prática, escolhe-se um número finito de combinações que, por norma, é uma potência de dois. Esta análise motiva o aparecimento da transformada de wavelet de tempo discreto (DWT[11]), que pode ser implementada de diversas formas [100]. A mais antiga e usual consiste na aplicação de um banco de filtros como proposto por Mallat [101].

[10] Do inglês: *Continuous Wavelet Transform*.
[11] Do inglês: *Discrete Wavelet Transform*.



(a)



(b)

Figura 4.15: Transformada de wavelet de tempo discreto: (a) Banco de filtros; (b) Análise espectral. f_{max} simboliza a componente espectral máxima de $x[n]$.

A figura 4.15 pretende dar alguma intuição sobre a operação de um banco de filtros. Por simplicidade, a análise centra-se numa decomposição com apenas duas escalas, em sinais 1D. Os termos $\overline{g_j}$ e $\overline{h_j}$ representam, respetivamente, os filtros passa-alto e passa-baixo de cada nível j. O primeiro retorna o coeficiente de detalhe d_j. O segundo transfere o coeficiente de aproximação a_j para o próximo nível $j+1$ para que seja posteriormente decomposto (figura 4.15a). Estes filtros atuam como um par que, em cada nível, divide o sinal em duas bandas espectrais (figura 4.15b). A cada nível $j+1$, a resposta ao impulso do filtro-passa baixo é obtida interpolando a resposta do filtro correspondente do nível j [100]:

$$\overline{h_{j+1}}[n] = \overline{h_j}[n] \uparrow = \begin{cases} \overline{h_j}[\frac{n}{2}], & n \text{ par} \\ 0, & n \text{ ímpar} \end{cases} \tag{4.35}$$

com a equação homóloga a ser aplicada para o filtro passa-alto. Cada sinal do nível $j+1$ é obtido convolvendo o sinal do nível anterior com $\overline{g_j}$ ou $\overline{h_j}$. Os filtros de reconstrução $\widetilde{g_j}$ e $\widetilde{h_j}$ são versões revertidas no tempo dos filtros de decomposição e permitem recuperar o sinal a partir dos seus coeficientes pela aplicação da transformada inversa de wavelet de tempo discreto (IDWT[12]).

No contexto do processamento de imagens, a transformada 2D é necessária. Isto implica que o processo seja aplicado sobre as direções vertical, horizontal e diagonal, resultando em três imagens de detalhe (dV_j, dH_j e dD_j) e mantendo uma imagem de aproximação (a_j), a cada nível j [100].

4.2.2.1 Transformada de Wavelet Não Decimada

A transformada de wavelet não decimada ou estacionária (SWT[13]) [102] é uma variante da DWT particularmente útil para lidar com imagens, porque não decima os coeficientes. Assim, é possível obter coeficientes com as mesmas dimensões da imagem inicial. No contexto de uma FCN, isto permite adicionar novos canais na entrada e tirar partido da informação multi-resolução codificada por eles.

[12] Do inglês: *Inverse Discrete Wavelet Transform.*
[13] Do inglês: *Stationary Wavelet Transform.*

4.3 Sumário

O problema da segmentação de vasos retinianos codifica uma tarefa de classificação binária. As ANNs são os modelos que lideram o estado da arte em tarefas de classificação, devido à sua capacidade de encontrar fronteiras de decisão arbitrariamente complexas. O termo *deep learning* surge quando se usam ANNs profundas (*deep*) para solucionar problemas de maior complexidade. Entre elas, destacam-se as CNNs que substituem a convencional multiplicação de matrizes pela operação de convolução, durante a propagação direta. As características que distinguem as CNNs das demais são a partilha de parâmetros e a conectividade local. Juntas, permitem reduzir o número de pesos e introduzir invariância à translação. As FCNs são uma variante das CNNs tradicionais onde as camadas densas são excluídas. Isto permite lidar com entradas de qualquer dimensão e aumentar a eficiência de computação.

Apesar do seu potencial, as ANNs profundas significam um desafio de otimização maior. De facto, o treino de um modelo de *deep learning* implica lidar com uma função de custo com inúmeros mínimos locais e pontos de sela. As ANNs em geral, e as CNNs em particular, são geralmente treinadas através do método de SGD que aplica o gradiente da função de custo, obtido pela *backpropagation*, para atualizar os pesos. Este mecanismo exige uma inicialização cuidada para que se possa convergir para um valor de custo razoável. Porém, a solução ótima não é apenas aquela que atinge um custo baixo; é, sobretudo, a que mantém boa capacidade de generalização. Um modelo demasiado potente, ou treinado durante um período excessivo, pode entrar em *overfitting* e ser incapaz de generalizar bem no conjunto de teste, mesmo que tenha ótimos resultados no conjunto de treino. As estratégias de regularização como o *weight decay*, o *dropout* e o aumento artificial de dados visam mitigar este problema.

Os conceitos da transformada de wavelet surgem porque esta é frequentemente utilizada para realçar os vasos da retina. A DWT é a versão discreta da CWT, que surge visto que os recursos de memória são limitados. A SWT tem a particulariedade de não decimar os coeficientes e, por isso, é útil quando se pretende que estes mantenham as dimensões da imagem inicial.

Capítulo 5

Trabalho Experimental

Apresentados os principais fundamentos teóricos, passa-se agora à análise do sistema de segmentação automática proposto. O trabalho desenvolvido ao longo desta dissertação pode ser analisado seguindo a estrutura típica dos métodos do estado da arte – descrita em pormenor ao longo do capítulo 3 –, que engloba as fases de pré-processamento, classificação e pós-processamento.

Neste capítulo, começa-se por fazer uma descrição breve das ferramentas de desenvolvimento do trabalho. Depois, parte-se para a apresentação do sistema proposto e das várias estratégias estudadas em cada etapa. Na fase final, descrevem-se ainda algumas operações que se aplicaram às imagens de forma a produzir análises de desempenho particulares que serão discutidas no próximo capítulo.

5.1 Ferramentas de Desenvolvimento

As funções usadas ao longo das diversas etapas deste trabalho foram desenvolvidas em *Python* [103], uma linguagem de programação de alto nível cada vez mais solicitada em *machine learning* e *data science*.

O *Python* disponibiliza inúmeros pacotes de manipulação de dados como o *numpy* [104], o *scipy* [105] e o *pandas* [106], pacotes de visualização como o *matplotlib* [107] e o *plotly* [108], pacotes genéricos para manipulação de imagens como o *scikit-image* [109], pacotes para a implementação de métodos de processamento de imagem como o *PyWavelets* [110] e, claro, pacotes específicos para o desenvolvimento de métodos de *deep learning* como o *tensorflow* [111] e o *keras* [112]. Estes recursos foram extremamente vantajosos porque possibilitaram uma implementação otimizada e mais rápida.

5.2 Descrição do Sistema

Nesta secção, apresentam-se as várias variantes estudadas em cada etapa, independentemente de trazerem, ou não, benefícios para o resultado final. Essa discussão fica reservada para o capítulo seguinte, onde se começará por fazer uma análise aos vários componentes do melhor sistema.

A figura 5.1 ilustra as várias etapas do sistema proposto. Podem destacar-se três fases. O pré-processamento englobou as etapas de correção do contraste das imagens, aplicação da SWT e normalização dos *patches*. A classificação foi feita através de uma FCN, com recurso ao aumento artificial de dados. Na fase final, as segmentações foram refinadas através de um processo de segmentação múltipla.

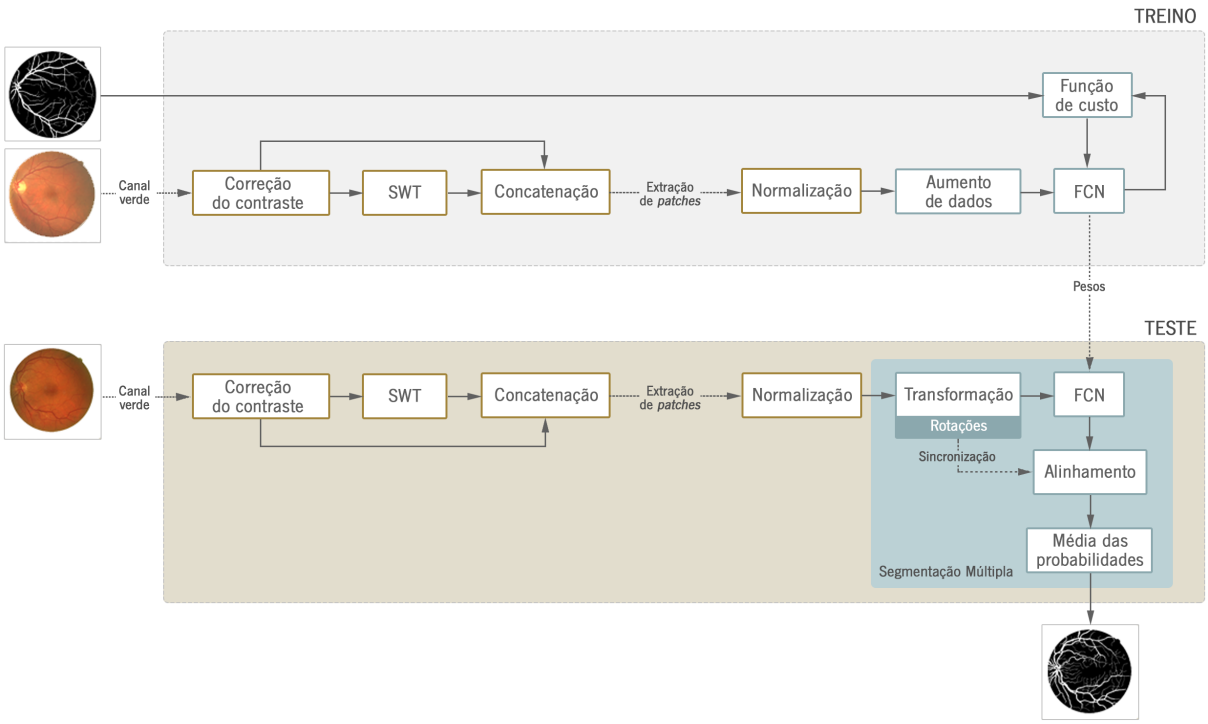


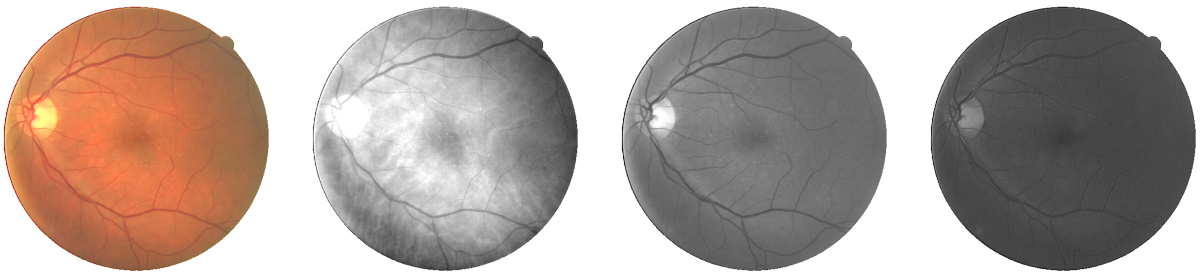
Figura 5.1: Visão geral do sistema de segmentação automática proposto.

5.2.1 Pré-processamento

O principal objetivo de um pré-processamento é uniformizar as características das imagens e, se possível, realçar as propriedades mais interessantes para que a tarefa de classificação possa ser facilitada. A estratégia de pré-processamento aplicada ao longo do trabalho dividiu-se em quatro fases.

5.2.1.1 Seleção de Canais

Uma imagem de fundo retiniano é adquirida a cores (figura 5.2a), podendo ser decomposta em três canais segundo o sistema RGB[1]: vermelho (figura 5.2b), verde (figura 5.2c) e azul (figura 5.2d). Os canais vermelho e azul são, tipicamente, mais ruidosos e apresentam menor gama dinâmica. Já o canal verde é aquele que exibe melhor contraste entre os vasos e o *background* [35]. Por isso, à semelhança da grande maioria dos métodos do estado da arte, ao longo deste trabalho operou-se apenas sobre o último.



(a) (b) (c) (d)

Figura 5.2: Exemplo de imagem de fundo retiniano: (a) Imagem RGB; (b) Canal vermelho; (c) Canal verde; (d) Canal azul.

[1] Do inglês: *Red, Green, Blue*.

5.2.1.2 Correção do Contraste

Como já se referiu, devido às variações das condições de iluminação durante a aquisição, as imagens de fundo apresentam variações no contraste do *background*. Este efeito não é detetável à vista desarmada, mas tem potencial para dificultar a classificação [3]. Deste modo, após a obtenção do canal verde, foram testadas duas alternativas para minimizar este problema.

5.2.1.2.1 CLAHE

A equalização do histograma é capaz de apresentar resultados favoráveis em diversos casos. Porém, como a mesma transformação é aplicada a toda a imagem, há uma dificuldade natural na presença de regiões muito escuras ou claras. A equalização adaptativa do histograma consiste no passo lógico seguinte: fazer com que cada região seja transformada de forma diferente. Na prática, calcula-se um histograma para cada região e, com base nele, faz-se um ajuste de contraste diferenciado [26].

O CLAHE é uma variante da equalização adaptativa que surge para dar resposta a regiões homogéneas: regiões que, além de muito brilhantes ou escuras, apresentam uma gama dinâmica reduzida. Nestas regiões, um aumento de contraste mínimo pode ser suficiente para que se perca informação relevante. Assim, a principal ideia por detrás do CLAHE é limitar esse aumento [26].

A variação do contraste pode ser vista como o declive da função que mapeia a intensidade de entrada na intensidade de saída. Um declive unitário é sinónimo de um ajuste nulo, enquanto declives cada vez maiores (ou menores) proporcionam aumentos de contraste cada vez mais (ou menos) significativos. Por outras palavras, limitar o aumento do contraste é sinónimo de restringir o declive da função de mapeamento [26]. Ora, esta função é proporcional ao histograma cumulativo em cada ponto. Como a derivada do histograma cumulativo é o histograma, então o declive da função de mapeamento pode ser limitado impedindo que o histograma ultrapasse um certo valor limite (o *clip limit*) (figura 5.3) [26].

Na implementação do CLAHE há dois parâmetros a ter em conta. O primeiro é o tamanho das várias sub-regiões. O segundo é o *clip limit*. Conceptualmente, o *clip limit* é o número máximo de pixeis que podem partilhar a mesma intensidade no histograma. Porém, em pacotes como o *scikit-image*, ele aparece normalizado entre 0 e 1. Por norma, a referência é o número de níveis de cinzento da imagem. Ou seja, numa imagem com 256 níveis, um *clip limit* de 0.5 corresponde a um limite de 128 pixeis para cada intensidade. Em resumo, quanto mais baixo for este parâmetro mais forte é a restrição introduzida.

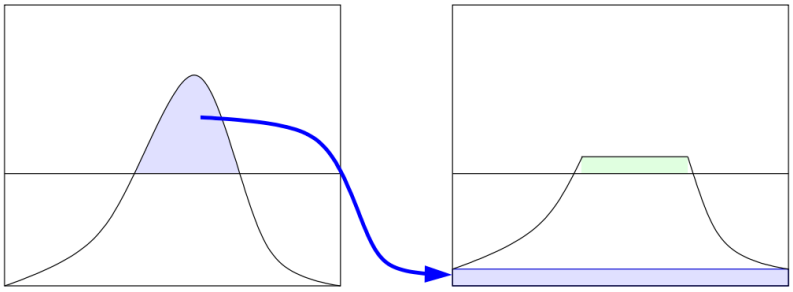


Figura 5.3: Restrição (*clipping*) do histograma no CLAHE. A linha horizontal cinzenta marca o *clip limit*. Reproduzido de [113].

(a) (b) (c)

Figura 5.4: Correção do contraste: (a) Canal verde; (b) Resultado após aplicação do CLAHE; (c) Resultado após aplicação do método de Nyúl e Udupa.

Ao longo deste trabalho, utilizaram-se regiões de 9×9 pixeis e um *clip limit* = 0.035 (≈ 9 pixeis). Estes valores foram obtidos manualmente, analisando os efeitos nas imagens. As figuras 5.4a e 5.4b ilustram, respetivamente, o canal verde inicial e o resultado da aplicação do CLAHE.

5.2.1.2.2 Método de Nyúl e Udupa

Ao contrário do CLAHE, que é frequentemente utilizado em imagens de fundo, o método de Nyúl e Udupa [114] foi desenvolvido para imagens de ressonância magnética. No entanto, visto que tem permitido obter resultados favoráveis na segmentação de tumores cerebrais [115, 116] e que o conceito pode ser estendido a qualquer tipo de imagem, este método foi também testado no contexto da segmentação de vasos. A ideia estruturante do método passa por transformar o histograma de uma dada imagem para que coincida com um histograma de referência. Isto é feito ao longo de duas etapas [114].

Na etapa de treino, definem-se os percentis do histograma de referência e as respetivas intensidades. Há dois tipos de percentis a estimar. Nos percentis intermédios, a intensidade é definida tomando a média das intensidades, do respetivo percentil, entre as várias imagens de treino. O número de percentis intermédios é variável. Neste trabalho, utilizou-se a seguinte distribuição:

$$L = \{pc_i, 10, 20, 30, 40, 50, 60, 70, 80, 90, pc_f\}\,. \tag{5.1}$$

Nos percentis inicial (pc_i) e final (pc_f), estimar as intensidades recaindo exclusivamente na média pode ser pouco prudente, pela possível presença de *outliers*. Assim, olha-se também para a variância entre percentis consecutivos. Por outras palavras, os percentis inicial e final devem ser escolhidos de forma a que as suas intensidades ($\hat{s}_i$ e $\hat{s}_f$, respetivamente) mantenham um comportamento aproximadamente linear face às dos percentis intermédios [114]. Neste caso, tomou-se $pc_i = 1.85$ e $pc_f = 95$.

Na etapa de transformação, o histograma de cada imagem é mapeado para a escala de referência (L). Em particular, o nível de intensidade inicial (x) é linearmente mapeado para o nível de intensidade respetivo ($\hat{x}$), segundo a expressão [114]:

$$\hat{x} = \hat{s}_j + \frac{x - s_j}{s_{j+1} - s_j}(\hat{s}_{j+1} - \hat{s}_j), \tag{5.2}$$

onde s_{j+1} e s_j designam, respetivamente, as intensidades do percentil mais baixo onde x está contido e daquele que o antecede. Note-se que, enquanto $\hat{s}_{j+1}$ e $\hat{s}_j$ representam intensidades relativas a percentis do histograma de referência, s_{j+1} e s_j dizem respeito a percentis do histograma da própria imagem [114]. Um aspeto chave nesta transformação é que a relação inicialmente existente entre as intensidades das várias estruturas é mantida [114]. O resultado da aplicação deste método é ilustrado na figura 5.4c.

5.2.1.3 Aplicação da SWT

Como se viu no capítulo 3, a transformada de wavelet é frequentemente utilizada para realçar os vasos da retina, devido à natureza multi-escala da vasculatura retiniana. Isto é, pelo facto do calibre dos vasos de uma árvore vascular ser amplamente variável.

Neste trabalho, aplicou-se a SWT no âmbito duma estratégia um pouco diferente da tradicional. A ideia é fundir as *features* extraídas pela SWT com aquelas que são aprendidas pela FCN. A principal motivação para esta abordagem reside numa tendência recente que defende que, apesar da sua grande capacidade para aprender boas representações dos dados *per se*, até os métodos de *deep learning* podem beneficiar da inclusão de conhecimento de domínio (ou *domain knowledge*) [117].

A estratégia usada neste trabalho passou por concatenar novos canais no canal verde, de forma a construir uma entrada composta que foi depois fornecida à FCN. Estes canais suplementares foram obtidos através da SWT, que se encontra implementada no *PyWavelets*. A escolha da wavelet mãe recaiu na wavelet de Haar [118] e utilizaram-se as decomposições de primeiro e segundo nível. Como em cada nível se extrai uma imagem de aproximação e três imagens de detalhe, foram gerados oito novos canais no total (figura 5.5). Quatro correspondentes ao primeiro nível (a_1 (5.5a), dV_1 (5.5b), dH_1 (5.5c) e dD_1 (5.5d)) e quatro referentes ao segundo (a_2 (5.5e), dV_2 (5.5f), dH_2 (5.5g) e dD_2 (5.5h)). Relembrando o que foi discutido na secção 4.2.2, as imagens de aproximação contêm informação de frequências mais baixas, enquanto as imagens de detalhe codificam informação de frequências mais altas. Em conjunto, estes canais fornecem informação multi-resolução, que pode ser útil na segmentação de imagens [100].

No sistema final, a entrada não é composta por todos os canais aqui descritos. Os efeitos dos canais extra são analizados na secção 6.6, onde se esclarece qual o sub-conjunto de canais que permitiu obter o melhor resultado final.

5.2.1.4 Normalização dos Patches

Sintetizando as etapas já descritas, começou-se por selecionar o canal verde da imagem e aplicaram-se duas técnicas para corrigir o seu contraste. Posteriormente, o canal verde foi concatenado com outros canais obtidos pela SWT.

Findadas estas etapas, seguiu-se uma fase de extração que permitiu obter os *patches* provenientes dos vários canais. Antes de ser apresentado à rede, cada *patch* foi normalizado para que tivesse média nula e desvio padrão unitário; ou seja, subtraiu-se a média das intensidades a cada pixel e dividiu-se o resultado pelo desvio padrão. Este procedimento, também conhecido como *feature standardization*, permite confinar as intensidades a uma gama mais uniforme e ajuda a acelerar a convergência num método de otimização como o SGD [79].

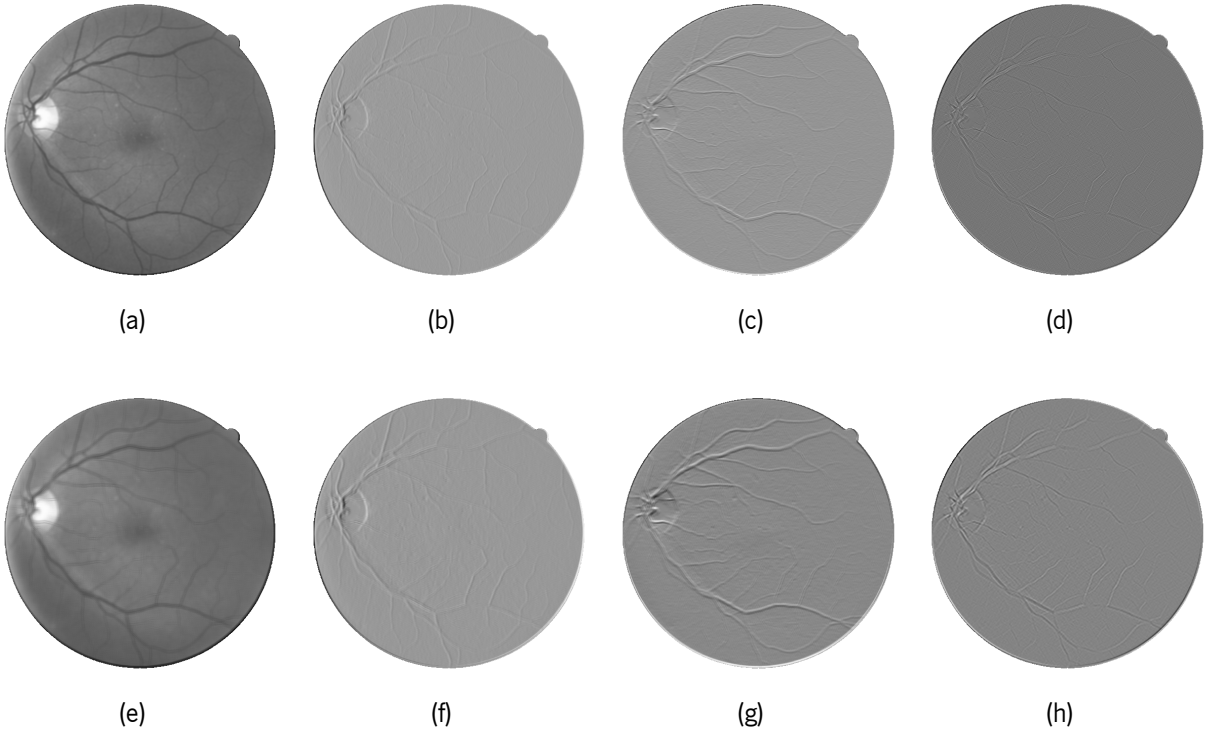


Figura 5.5: Canais obtidos pela aplicação da SWT: (a) a_1; (b) dV_1; (c) dH_1; (d) dD_1; (e) a_2; (f) dV_2; (g) dH_2; (h) dD_2.

5.2.2 Classificação

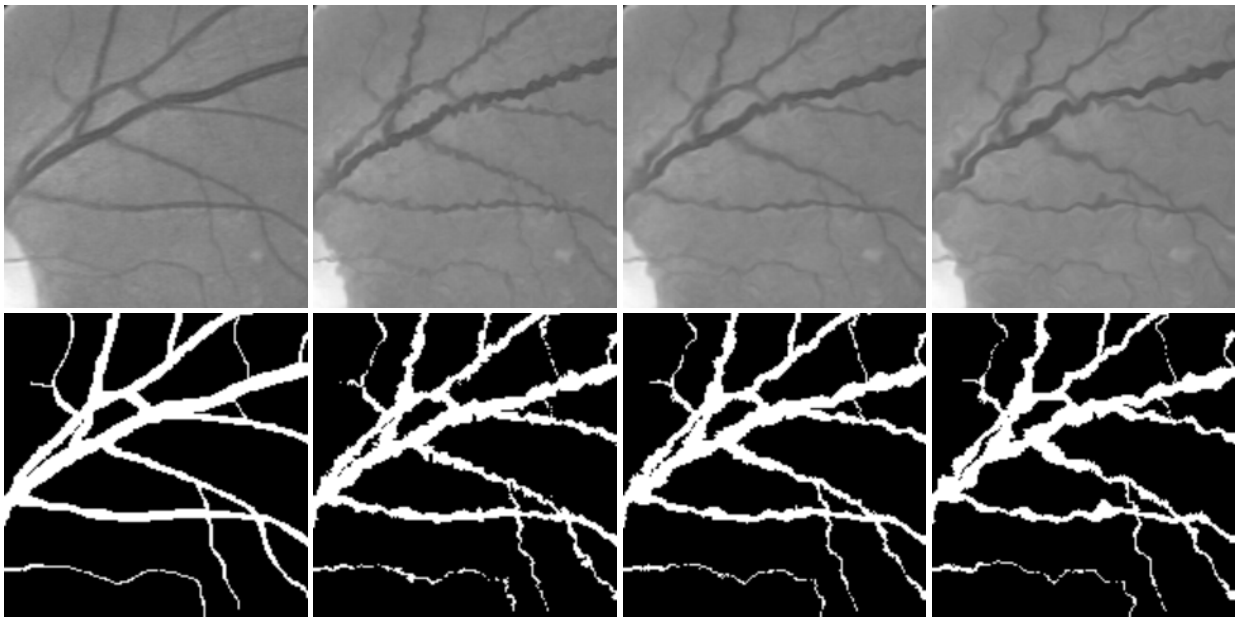
Como já se esclareceu, o modelo base deste trabalho foi a FCN. Aqui analisam-se todas as escolhas tomadas em termos de arquitetura e hiperparâmetros. Antes disso, começa-se por fazer uma descrição das técnicas de aumento artificial de dados usadas.

5.2.2.1 Aumento Artificial de Dados

Com base nos *patches* originais provenientes do canal verde, usaram-se duas estratégias distintas para gerar novos *patches* artificiais. Independentemente da abordagem, houve a necessidade de manter a correspondência entre o *patch* da imagem e o respetivo *patch* anotado pelo que cada transformação foi aplicada a ambos.

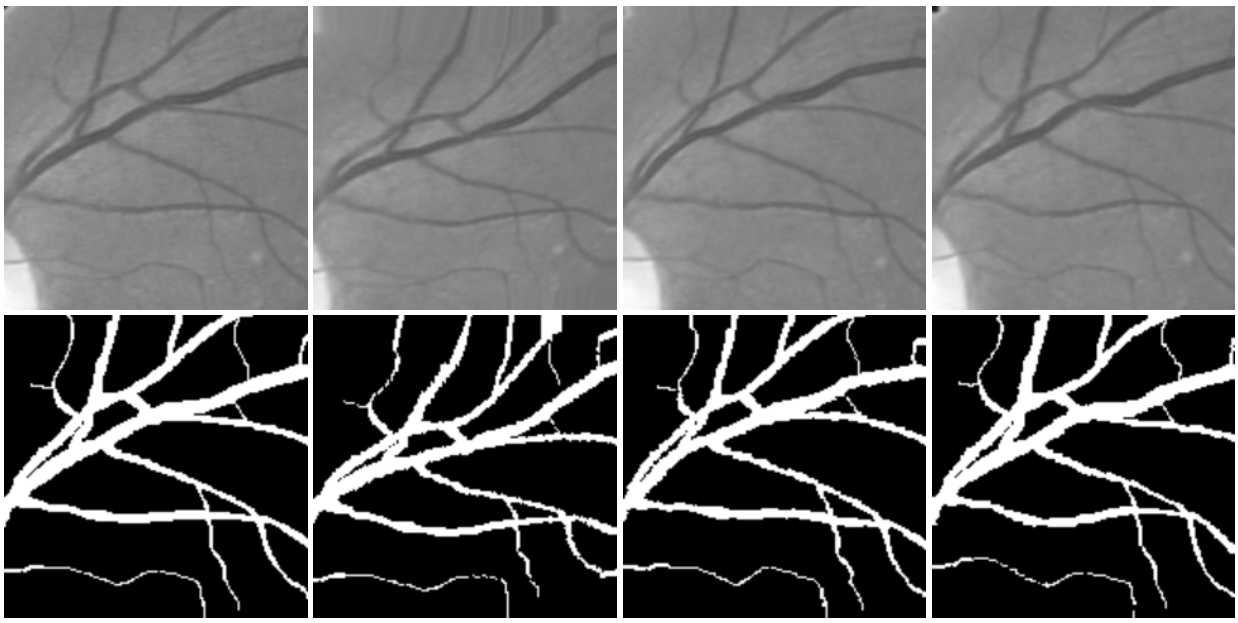
5.2.2.1.1 Rotações

A primeira estratégia residiu na aplicação de operações de rotação que são, como já se referiu, bastante comuns quando se pretende aumentar o tamanho do conjunto de treino. Em particular, cada *patch* foi submetido a três rotações de 90°, 180° e 270°. O mesmo é dizer que com base em cada *patch* original, se obteve um conjunto de quatro *patches* (o original e as três versões rodadas). Usaram-se apenas rotações por ângulos múltiplos de 90°, para evitar fazer interpolações. As operações de rotação foram a base para o esquema de segmentação múltipla que se apresenta na secção 5.2.3.



(a) (b) (c) (d)

Figura 5.6: Aumento artificial de dados através da transformação de Simard [119]: (a) *Patch* original; (b) *Patch* artificial c/ $\alpha = 8$ e $\sigma = 1.5$; (c) *Patch* artificial c/ $\alpha = 16$ e $\sigma = 2.5$; (d) *Patch* artificial c/ $\alpha = 32$ e $\sigma = 3$. A primeira linha ilustra as imagens, enquanto a segunda representa as respetivas anotações.



(a) (b) (c) (d)

Figura 5.7: Aumento artificial de dados através da transformação de Ronneberger [120]: (a) *Patch* original; (b) *Patch* artificial c/ $N = 3$ e $\sigma = 15$; (c) *Patch* artificial c/$N = 5$ e $\sigma = 4$; (d) *Patch* artificial c/ $N = 7$ e $\sigma = 2$. A primeira linha ilustra as imagens, enquanto a segunda representa as respetivas anotações.

5.2.2.1.2 Transformações Elásticas

A segunda estratégia assentou na utilização de transformações não lineares ou elásticas. Foram testadas duas alternativas [4].

5.2.2.1.2.1 Transformação de Simard

A primeira transformação foi inspirada na proposta de Simard *et al.* [119]. Neste caso, começou-se por gerar um campo de deslocamento inicial para cada pixel (x, y) do *patch* a replicar. Este campo consiste num vetor $u = (u_x, u_y)$, em que u_x e u_y são números aleatórios, gerados a partir de uma distribuição uniforme, que mapeiam a transformação segundo cada um dos eixos. Depois, os campos de deslocamento iniciais foram convoluídos com um filtro gaussiano com desvio-padrão σ e, por fim, multiplicados por um fator α. Assim, obtiveram-se os campos de deslocamento finais. De forma geral, valores elevados de σ provocam deformações subtis, enquanto valores reduzidos produzem deformações intensas. O objetivo passou por manter σ numa gama intermédia onde os efeitos da deformação fossem notórios, mas preservassem as características das imagens. Nesta gama intermédia, α foi usado para fazer um ajuste mais fino à intensidade da deformação. A figura 5.6 ilustra exemplos de novos *patches* artificiais obtidos através de diferentes combinações de σ e α.

5.2.2.1.2.2 Transformação de Ronneberger

A segunda transformação baseou-se na proposta de Ronneberger *et al.* [120]. Desta feita, o primeiro passo consistiu em definir um conjunto de $N \times N$ pontos de controlo. Para estes pontos, gerou-se então um campo de deslocamento u a partir de uma distribuição gaussiana com desvio-padrão σ. Na fase final, o campo de deformação foi interpolado para os restantes pixeis do *patch* original. Neste caso, os efeitos de sigma permanecem semelhantes ao que se descreveu anteriormente. O número de pontos de controlo foi ajustado, conforme o valor de σ, de forma a obter uma deformação coerente. Por outras palavras, para que os *patches* artificiais mantivessem as características dos originais, valores mais baixos de σ foram contrabalançados pelo uso de mais pontos de controlo. A figura 5.7 ilustra exemplos de novos *patches* artificiais obtidos através de diferentes combinações de σ e N.

5.2.2.2 Modelo Base

Apesar de uma CNN, e de uma FCN em particular, ser composta por um número limitado de blocos estruturais, é possível gerar uma diversidade enorme de arquiteturas distintas. Por esse motivo, neste trabalho definiu-se um modelo base que serviu de referência para avaliar os efeitos de adicionar, substituir ou remover certos componentes, como se verá no capítulo seguinte.

Nesta altura, importa discutir os princípios por detrás desse modelo base. Por isso, passa-se à descrição da motivação para a arquitetura proposta e também à discussão dos hiperparâmetros, que podem estar relacionados com a rede em si ou com a etapa de treino.

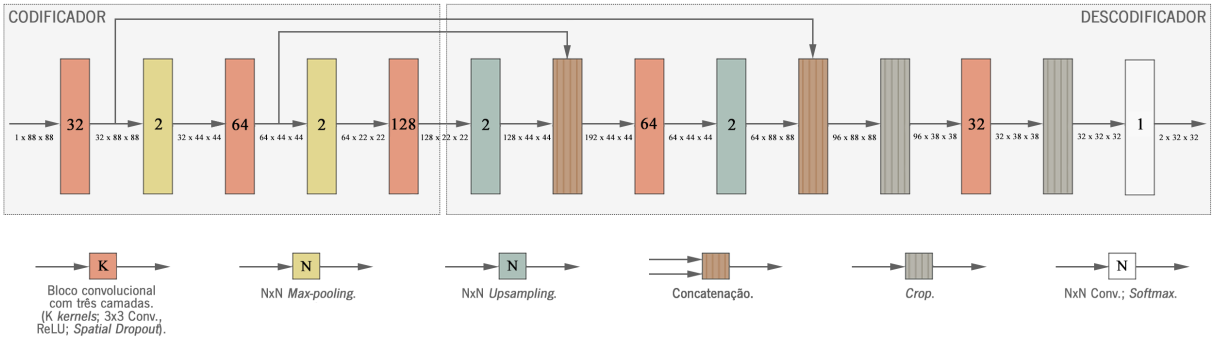


Figura 5.8: Arquitetura da FCN proposta.

5.2.2.2.1 Motivação

O modelo base deste trabalho consistiu numa FCN inspirada na U-net de Ronneberger *et al.* [120]. A arquitetura proposta (figura 5.8) é composta por dois blocos: um codificador e um descodificador. O primeiro, composto pelos três blocos convolucionais iniciais e pelas camadas de *pooling*, visa sumariar *features* vizinhas e criar representações de alto nível. O segundo, que surge a partir da primeira camada de *upsampling*[2], é responsável por combinar os *feature maps* que provêm do codificador, através de *skip connections*, para que se obtenham *feature maps* com as mesmas dimensões iniciais.

Como já se referiu, os operadores de *pooling* são altamente benéficos em problemas de reconhecimento de objetos e contribuem para aumentar o *receptive field* de unidades mais profundas. Porém, eles também fazem com que alguma informação de localização seja perdida. Assim, parece levantar-se um problema: a informação de alto nível revela *o quê*, enquanto a informação de baixo nível revela *onde* [90]. Uma forma de mitigar este problema passa por combinar os dois tipos de informação. Essa é a principal motivação para esta arquitetura, e para a estrutura do descodificador em particular, visto que as *skip connections* permitem fundir os *feature maps* superficiais, com grande dimensão e informação de baixo nível, com os *feature maps* profundos, com reduzida dimensão e informação de alto nível.

5.2.2.2.2 Hiperparâmetros da Arquitetura

No que diz respeito às camadas convolucionais, os hiperparâmetros mais relevantes são o tamanho dos filtros (s_k), o número de filtros por camada e o tipo de convolução (*padding* (p) e *stride* (st)).

Na arquitetura proposta, utilizaram-se apenas filtros com $s_k = 3$. Até ao surgimento da VGGNet [73], existia um grande debate sobre este aspeto. De facto, filtros maiores parecem ser sinónimo de *receptive fields* maiores, o que significa que cada neurónio capta informação de mais vizinhos. Porém, [73] veio mostrar que a utilização de filtros de 3×3 permite obter o mesmo *receptive field* de filtros maiores, utilizando menos pesos. Tome-se como exemplo a figura 5.9. Sobrepondo dois filtros de 3×3, cada unidade terá um *receptive* field de 5×5. Porém, enquanto neste caso se têm $2 \times 3 \times 3 = 18$ pesos, com um filtro de 5×5 ter-se-iam 25. Esta diferença torna-se cada vez mais importante à medida que mais filtros são usados. O número de filtros por camada também foi discutido em [73], onde se sugere que cada bloco convolucional deve ter um número de filtros múltiplo do anterior. Assim, utilizaram-se 32, 64 e 128 filtros por bloco, conforme o nível da arquitetura. O aumento sucessivo do número de filtros

[2] Operação inversa do *pooling*.

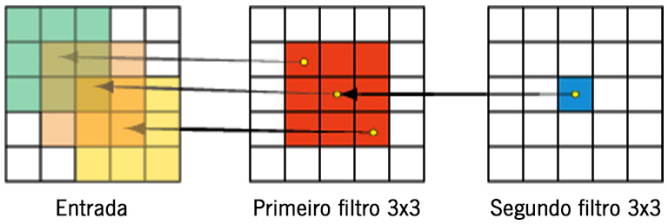


Figura 5.9: Observe-se que a sobreposição de dois filtros de 3×3 confere o mesmo *receptive field* que um filtro de 5×5. Adaptado de [121].

deve-se ao facto dos *feature maps* apresentarem níveis de abstração cada vez maiores. Quanto ao tipo de convolução, usaram-se apenas convoluções *same* ($p = st = 1$) para que se mantivessem as dimensões dos *feature maps* e não houvessem restrições quanto ao número de camadas convolucionais.

No que diz respeito às camadas de *pooling* e *upsampling*, utilizou-se $s_k = st = 2$. Isto significa que a primeira reduz as dimensões dos *feature maps* a metade, enquanto a segunda as duplica.

A função de ativação utilizada foi a ReLU, pelas vantagens apontadas na secção 4.1.2.3.2.

Por último, as dimensões dos *patches* de entrada e saída foram fixadas em $1 \times 88 \times 88$ e $2 \times 32 \times 32$, respetivamente. Vários outros tamanhos foram avaliados nas secções 6.5.3 e 6.5.4. Note-se que a primeira dimensão diz respeito ao número de canais. Na saída têm-se dois canais, correspondentes às duas classes (vaso e *background*). Na entrada tem-se apenas um, porque no modelo base não se consideram os canais suplementares posteriormente adicionados através da SWT.

5.2.2.2.3 Hiperparâmetros do Treino

Na etapa de inicialização, os pesos foram definidos recorrendo à inicialização de Xavier – secção 4.1.3.2.1 –, enquanto os *biases* foram inicializados a 0.1.

Em termos de regularização, a estratégia passou pelo uso do *dropout* espacial – secção 4.1.3.4.2.1 – com $p = 0.2$ em todos os blocos convolucionais, exceto no último onde se aplicou $p_{último} = 0.15$.

Na etapa de otimização, utilizou-se a função de entropia cruzada – secção 4.1.3.1.1 – como função de custo e aplicou-se o SGD – secção 4.1.3.1.4 – com momento de Nesterov – secção 4.1.3.1.4.2 – para minimizá-la. O tamanho do *mini-batch* foi fixado em 4, pelo que quatro *patches* são propagados através da rede em cada iteração. Vários outros tamanhos foram testados, como se discute na secção 6.5.6.

Cada treino foi prolongado por 20 épocas, porque se aferiu que a partir daí a função de custo estabilizava e não se tirava vantagem em termos de capacidade de generalização. O *learning rate* (α) e o momento (μ) foram alterados em épocas específicas segundo o calendário da tabela 5.1. Além disto, α foi sendo reduzido, entre estas alterações, através da expressão:

$$\alpha_i = \frac{\alpha_{i-1}}{1 + \eta \times i}, \tag{5.3}$$

onde α_i e α_{i-1} são, respetivamente, o *learning rate* nas iterações atual e anterior e η é um hiperparâmetro que controla a taxa de diminuição e que foi fixado em 1×10^{-6}. A diminuição sucessiva do *learning rate* surge para compensar a amostragem aleatória dos exemplos que compõem o *mini-batch*, que é vista como uma fonte de ruído que se mantém à medida que o algoritmo se aproxima de um mínimo [77].

Tabela 5.1: Calendário de atualização dos parâmetros de otimização

Parâmetro	Época	Valor
α	1	0.05
	10	0.02
	14	0.002
	18	0.0002
μ	1	0.2
	10	0.9
	14	0.99

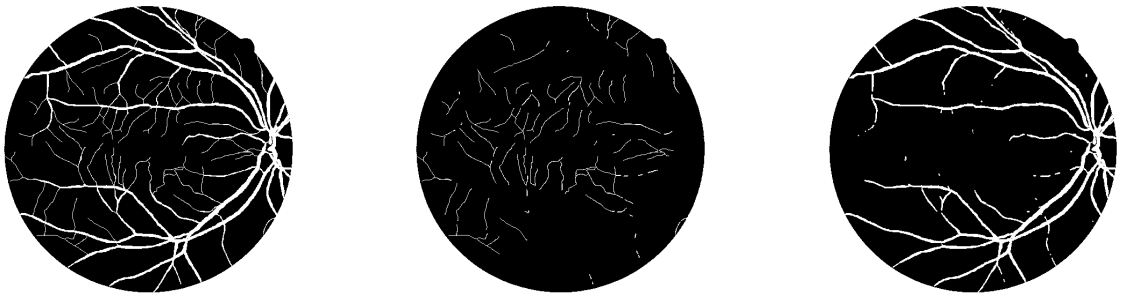
5.2.3 Segmentação Múltipla

Como se referiu no capítulo 3, uma etapa de pós-processamento é sempre muito dependente das características do modelo e das próprias segmentações. Neste trabalho, abdicou-se do pós-processamento morfológico porque não se obtiveram melhorias significativas através dessa estratégia. Em alternativa, as segmentações foram refinadas de outra forma. Conforme se verá na secção 6.5.2, a estratégia de aumento artificial de dados que permitiu obter melhores resultados foi aquela que se baseou exclusivamente em operações de rotação. Isto significa que a rede conseguiu extrair informação relevante a partir das cópias rodadas. Ao mesmo tempo, isto sugere que, na fase de teste, a rede pode acabar por classificar melhor um dado *patch* se ele se apresentar numa determinada orientação. Deste modo, introduziu-se um mecanismo de segmentação múltipla (recorde-se a figura 5.1) que visa tirar partido disto. Na fase inicial, cada *patch* a segmentar foi previamente rodado segundo as três direções descritas na secção 5.2.2.1.1 – transformação. Cada um dos quatro *patches* resultantes foi depois apresentado à rede e obtiveram-se os quatro mapas de probabilidades respectivos, para cada classe. De seguida, os mapas dos *patches* rodados foram rodados do mesmo ângulo inicial, mas na direção oposta – alinhamento. Isto permitiu que se calculasse a média entre os quatros mapas para cada pixel – média das probabilidades – e que se gerasse a segmentação final para o *patch* a segmentar.

5.3 Tratamento das Imagens

A esmagadora maioria dos métodos do estado da arte é já extremamente eficaz na segmentação dos vasos grossos. As diferenças que subsistem entre os métodos atuais residem, regra geral, na maior ou menor capacidade para evitar falsas deteções ou para detetar os vasos mais finos. Deste modo, é fundamental avaliar explicitamente o modelo nos vasos de menor calibre. Além disto, torna-se também importante caracterizar as regiões mais propensas a falsas deteções; ou seja, perceber até que ponto os pixeis mal classificados como vasos se encontram em zonas próximas de vasos reais.

Dito isto, para que fosse possível apresentar resultados quantitativos e debater estas questões no capítulo seguinte, foi necessário decompor as imagens em vasos finos e grossos e, paralelamente, identificar as vizinhanças próxima e distante desses vasos. Os procedimentos que permitiram efetuar estas operações são sucintamente descritos de seguida.



(a) (b) (c)

Figura 5.10: Decomposição em vasos finos e grossos: (a) Imagem inicial; (b) Imagem com vasos finos; (c) Imagem com vasos grossos.

(a) (b) (c)

Figura 5.11: Decomposição em vizinhanças próxima e distante: (a) Imagem inicial; (b) Imagem com vizinhanças próxima (a escuro) e distante (a claro); (c) Imagem final.

5.3.1 Decomposição em Vasos Finos e Grossos

A decomposição em vasos finos e grossos pode ser dividida em três etapas. Inicialmente, aplicou-se uma transformação *top-hat*. Isto permitiu obter os vasos finos, ainda que com bastantes pontos isolados. De seguida, para suprimir estes pontos, fez-se uma análise de componentes ligados e eliminaram-se todos os componentes com menos que 5 pixeis. Deste modo, obtiveram-se os vasos finos definitivos. Finalmente, para assegurar que todos os pixeis iniciais eram mantidos, os vasos grossos foram tomados como a diferença entre a imagem inicial e os vasos finos. A figura 5.10 ilustra um exemplo das imagens obtidas por este processo.

5.3.2 Decomposição em Vizinhanças Próxima e Distante

A identificação das vizinhanças próxima e distante dividiu-se em duas fases. Inicialmente, foram aplicadas sucessivas dilatações morfológicas à imagem inicial de forma a aumentar o calibre dos vasos. Depois, estes vasos aumentados foram subtraídos à máscara, dividindo-a em duas regiões. Note-se que o número de dilatações usadas determina o tamanho da vizinhança próxima. No exemplo da figura 5.11 foram utilizadas 10 dilatações.

Nas secções 5.3.1 e 5.3.2, as operações morfológicas foram implementadas com recurso ao pacote *morphology* do *scipy*.

5.4 Sumário

O sistema de segmentação automática apresentado neste capítulo pode ser dividido em três fases. A fase de pré-processamento iniciou-se com uma etapa de correção do contraste do canal verde que assentou na utilização de uma de duas estratégias: CLAHE ou método de Nyúl e Udupa. Depois, o canal verde foi concatenado com novos canais obtidos através da aplicação da SWT, para que se pudessem avaliar os efeitos de combinar *deep learning* e conhecimento de domínio. Finalmente, depois da fase de extração, os *patches* foram normalizados para terem média nula e desvio padrão unitário. A etapa de classificação assentou numa FCN cuja arquitetura foi motivada pela necessidade de fundir informação com diferentes níveis de abstração. Para aumentar o conjunto de treino, propuseram-se duas estratégias de aumento artificial de dados: rotações e transformações elásticas. Na etapa final, dispensou-se o típico pós-processamento morfológico e recorreu-se a um esquema de segmentação múltipla para refinar as segmentações.

Quando se comparam dois métodos do estado da arte, o comportamento nos vasos finos é particularmente importante porque é neles que residem as principais diferenças entre os métodos atuais. Além disso, torna-se também útil identificar as regiões da imagem mais propensas a falsas deteções. Este tipo de análise exigiu que se fizesse uma divisão das imagens em vasos finos e grossos e que se identificassem as vizinhanças próxima e distante.

Capítulo 6

Resultados

Neste capítulo, começa-se por descrever as bases de dados e métricas de avaliação utilizadas, a metodologia de testes adotada e o *software/hardware* de que se dispôs. Depois, passa-se à validação dos vários componentes do modelo base e à identificação do melhor modelo. Finalmente, o melhor modelo é comparado com outros métodos do estado da arte e avaliado em diferentes aspetos.

6.1 Bases de Dados

Neste trabalho utilizaram-se três bases de dados públicas e amplamente reconhecidas na área da segmentação de vasos retinianos. A tabela 6.1 resume algumas das suas características mais relevantes.

A DRIVE[1] [62] é composta por 40 imagens, obtidas no âmbito de um programa de rastreio de retinopatia diabética na Holanda, entre as quais 7 mostram sinais patológicos. As imagens foram adquiridas através de uma câmera não midriática[2] Canon CR5.

A STARE[3] [35] é oriunda de um projeto que surgiu nos Estados Unidos. É composta por 20 imagens, das quais 10 pertencem a indivíduos com evidências patológicas. As imagens foram adquiridas através de uma câmera midriática TopCon TRV-50.

A CHASE_DB1[4] [43] deriva de um estudo, em crianças, na Inglaterra. É composta por 28 imagens, das quais 14 pertencem ao olho direito e 14 ao olho esquerdo. As imagens foram adquiridas através de uma câmera não midriática Nidek NM-200-D.

Tabela 6.1: Principais características das bases de dados utilizadas

Base de dados	Ano	Imagens	Tamanho	Bits por canal	FOV	Divisão teste/treino
DRIVE [62]	2004	40	565×584	8	45°	$\surd$
STARE [35]	2000	20	605×700	8	35°	-
CHASE_DB1 [43]	2011	28	999×960	8	30°	-

[1] Do inglês: *Digital Retinal Images for Vessel Extraction*.
[2] Câmera que dispensa a aplicação de gotas midriáticas.
[3] Do inglês: *Structured Analysis of the Retina*.
[4] Do inglês: *Child Heart and Health Study in England*.

6.2 Avaliação do Desempenho

Como já se referiu, num problema de segmentação de vasos o objetivo é classificar cada pixel como vaso ou *background*. Assim, ao comparar as segmentações com as anotações manuais, cada pixel pode ser enquadrado em um de quatro grupos. Os pixeis que pertencem aos vasos ou ao *background* e que são bem classificados constituem, respetivamente, os verdadeiros positivos (VP) e os verdadeiros negativos (VN). Os pixeis do *background* classificados como vasos são os falsos positivos (FP). Já os pixeis dos vasos classificados como *background* são chamados de falsos negativos (FN) [122].

Com base nestes grupos, é possível definir três métricas recorrentes: sensibilidade (*Sen*), especificidade (*Esp*) e *accuracy* (*Acc*). Estas podem ser descritas pelas seguintes expressões [122]:

$$Sensibilidade = \frac{VP}{VP + FN}, \tag{6.1}$$

$$Especificidade = \frac{VN}{VN + FP}, \tag{6.2}$$

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}. \tag{6.3}$$

Em geral, a *Sen* traduz a capacidade do modelo para detetar os vasos, enquanto a *Esp* permite perceber até que ponto essa capacidade de deteção é obtida à custa da inclusão de falsas deteções. Por sua vez, a *Acc* resume o acerto global em todos os pixeis [122].

Estas medidas quantitativas partilham a desvantagem de dependerem do *threshold* usado para gerar as segmentações binárias a partir dos mapas de probabilidade. Para contornar isto, pode optar-se por reportar a área abaixo da curva ROC[5] (AUC[6]). A curva ROC (figura 6.1) é um gráfico xy, onde x e y correspondem, respetivamente, a $(1 - Esp)$ e à *Sen*. A AUC, como o próprio nome indica, é a área abaixo dessa curva, pelo que não depende do *threshold* usado para obter cada par sensibilidade/especificidade. Por outras palavras, a AUC engloba todos os *thresholds* possíveis [122].

O classificador perfeito obteria 1 em todas as métricas enunciadas.

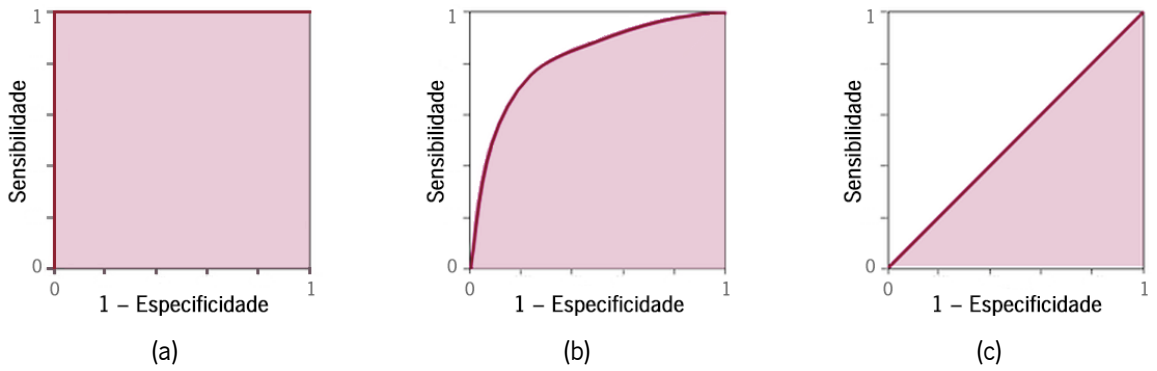


Figura 6.1: Curva ROC e respetiva AUC para o: (a) Classificador perfeito; (b) Classificador mediano; (c) Classificador aleatório.

[5] Do inglês: *Receiver Operating Characteristic*.
[6] Do inglês: *Area Under the ROC Curve*.

6.3 Metodologia de Testes

Como consta na tabela 6.1, a DRIVE possui uma divisão explícita entre os conjuntos de treino e teste. Em particular, cada um destes conjuntos é composto por 20 imagens. Deste modo, o modelo foi avaliado no conjunto de teste.

Nas outras duas bases de dados, esta divisão explícita não está disponível. Por este motivo, o modelo foi treinado usando uma metodologia de validação cruzada k-*fold* (figura 6.2). Neste método, o conjunto original é dividido em k sub-conjuntos (ou *folds*) com o mesmo tamanho. O processo de validação é, então, repetido k vezes, com cada *fold* a constituir o conjunto de teste e os restantes $k-1$ *folds* a serem usados para treino. Na fase final, é tomada a média dos resultados entre os vários *folds* de forma a produzir uma única estimativa. Na STARE, usou-se $k = 5$, obtendo-se 5 *folds* de 4 imagens. Além disso, cada *fold* foi estratificado garantindo que metade das imagens pertencia a indivíduos patológicos. No caso da CHASE_DB1, optou-se por $k = 4$, gerando-se 4 *folds* de 7 imagens. Desta feita, assegurou-se que cada *fold* continha 3 imagens de um olho e 4 imagens do outro.

Para cada base de dados, estão disponíveis duas anotações manuais de observadores humanos independentes. Para a DRIVE e a CHASE_DB1, usaram-se as anotações do primeiro observador humano como *labels*. Para a STARE, usaram-se as anotações de Hoover. O desempenho do segundo observador foi obtido comparando as anotações manuais produzidas por ele com as do primeiro observador.

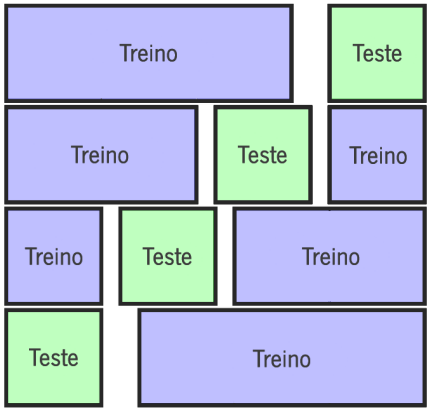


Figura 6.2: Metodologia de validação cruzada k-*fold* (com $k = 4$). Adaptado de [84].

6.4 Software e Hardware

O método proposto foi implementado usando *Keras* com o *backend* do *TensorFlow* e com recurso à NVIDIA cuDNN[7] 5.1.

Todos os testes foram realizados num computador com o sistema operativo Linux Mint 18, equipado com um GPU NVIDIA GeForce GTX 1070, um processador Intel® Core™ i7-6850k 3.6 GHz e 128 GB de memória RAM.

[7] Do inglês: *CUDA Deep Neural Network Library*.

6.5 Validação do Modelo Base

Na secção 5.2.2.2, apresentaram-se as principais justificações para a seleção dos hiperparâmetros do modelo base. Alguns desses hiperparâmetros, como o tamanho dos filtros ou o tipo de convolução, não foram alvo de estudos suplementares por serem já globalmente aceites. Nesta secção, validam-se as escolhas tomadas em relação a vários outros, tanto em termos da própria arquitetura como em termos de regularização e otimização. Além disto, no capítulo anterior introduziram-se várias estratégias para aumentar artificialmente o conjunto de dados e corrigir o contraste. Aqui revelam-se os procedimentos que se mostraram mais benéficos e descrevem-se os efeitos da estratégia de segmentação múltipla apresentada na secção 5.2.3.

Como já se referiu, o estabelecimento de um modelo base visou não só reduzir os custos computacionais associados a uma análise exaustiva, mas também avaliar o efeito de cada parâmetro ou variante de forma individualizada. Assim, os efeitos de cada componente foram avaliados através de testes de *ablation*, onde o componente em estudo foi removido ou substituído e os resultados recalculados. Para este efeito, recorreu-se à base de dados DRIVE e, por concisão, utilizaram-se as métricas *Acc* e AUC.

Devido à incerteza associada à distribuição dos dados, todos os testes de significância estatística foram computados através do teste de Wilcoxon [123], com nível de significância igual a 0.05. A hipótese nula avaliada supõe que não existem diferenças significativas entre cada variante e o modelo base.

6.5.1 Efeito da Correção do Contraste

Na secção 5.2.1.2, introduziram-se duas alternativas para corrigir as variações do contraste: o *CLAHE* e o método de *Nyúl e Udupa*. Aqui comparam-se as variantes que fizeram uso destes métodos com o *modelo base*, onde não se recorreu a qualquer estratégia deste tipo.

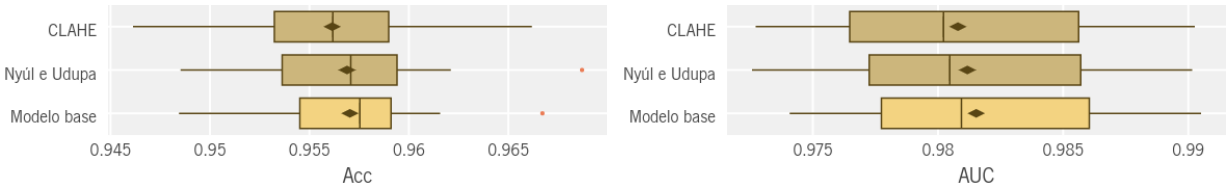
Analisando os valores médios da tabela 6.2, conclui-se que a inclusão do CLAHE gerou uma redução estatisticamente significativa em termos de *Acc* e AUC. No caso do método de Nyúl e Udupa, também se verificou uma redução estatisticamente significativa em termos de AUC, mas a diferença em termos de *Acc* revelou-se quase negligenciável.

Olhando para a figura 6.3, pode ainda aferir-se que, principalmente no que diz respeito à *Acc*, a inclusão destes métodos fez com que os resultados se tornassem mais dispersos. Houve, efetivamente, imagens que beneficiaram da correção, mas estes casos foram neutralizados por outros onde predominaram consequências negativas.

Em geral, o modelo não pareceu tirar partido destas estratégias e, por isso, estas não foram mantidas.

Tabela 6.2: Resultados dos testes relativos à correção do contraste, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Método de correção**	**Acc**	**AUC**
CLAHE	CLAHE	<u>0.9561</u>	<u>0.9808</u>
Nyúl e Udupa	Nyúl e Udupa	0.9569	<u>0.9812</u>
Modelo base	-	**0.9570**	**0.9815**



(a) (b)

Figura 6.3: *Boxplots* de cada variante apresentada na tabela 6.2 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

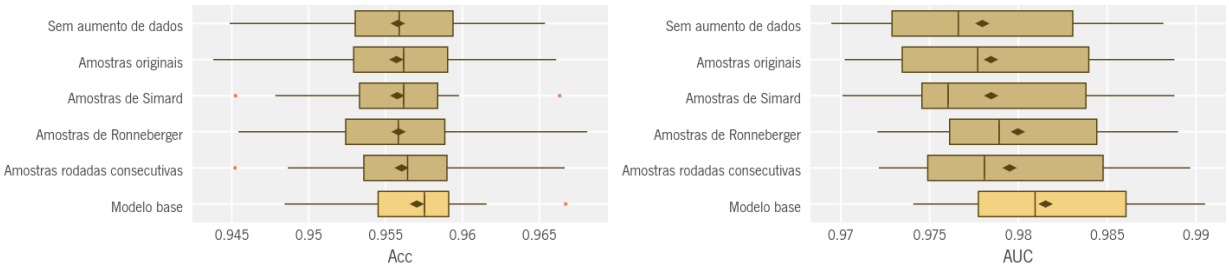
6.5.2 Efeito do Aumento Artificial de Dados

Para avaliar as técnicas de aumento artificial de dados apresentadas na secção 5.2.2.1, testaram-se diversas variantes. Na primeira, começou-se por extrair 3000 *patches* por imagem, sem recorrer a *patches* artificiais – *sem aumento de dados*. Nas variantes que se seguiram, introduziram-se diferentes estratégias para aumentar o número total de *patches* por imagem para 12000. Na segunda variante, os 12000 *patches* foram extraídos diretamente da imagem – *amostras originais*. Na terceira e quarta variantes, extraíram-se 3000 *patches* originais e os restantes 9000 foram obtidos artificialmente através da aplicação das transformações de Simard – *amostras de Simard* – e Ronneberger – *amostras de Ronneberger* –, respetivamente. Nestes casos, como cada transformação gera um *patch* de saída para cada *patch* de entrada, foi necessário usar três combinações diferentes de parâmetros (figuras 5.6 e 5.7). Na última variante, recorreu-se novamente a 3000 *patches* originais, mas os 9000 *patches* artificiais resultaram da aplicação de três rotações sobre cada *patch*. Os *patches* rodados resultantes foram fornecidos à rede de forma consecutiva – *amostras rodadas consecutivas*. Por fim, todas estas variantes foram comparadas com o *modelo base*, no qual o aumento artificial de dados também passou pelo uso de rotações, mas onde os *patches* rodados foram apresentados à rede de forma aleatória.

Considerando os valores médios da tabela 6.3, verifica-se que quando se utilizaram apenas 3000 *patches* por imagem os resultados deterioraram-se, consideravelmente, em termos de *Acc* e AUC. Além disto, quando se extraíram 12000 *patches* por imagem os resultados permaneceram muito semelhantes. Isto sugere que a partir de certa altura os *patches* originais começam a fornecer informação redundante. Relativamente às variantes que fizeram uso de amostras elásticas, subsistiram reduções consideráveis, em termos de *Acc* e AUC, face ao modelo base. Porém, o facto destas amostras terem permitido obter resultados idênticos ou até ligeiramente melhores aos obtidos com 12000 *patches* originais, recorrendo a apenas um quarto desses *patches* extraídos diretamente da imagem, é um indicador positivo. Por último, comparando a variante *amostras rodadas consecutivas* com o *modelo base*, é possível concluir que a ordem com que os *patches* são apresentados à rede tem grande influência no desempenho. De facto, quando os *patches* rodados foram apresentados à rede de forma consecutiva surgiram reduções praticamente da mesma ordem daquelas que se verificaram aquando da aplicação das amostras elásticas. Em resumo, realça-se que todas as variantes estudadas introduziram reduções estatisticamente significativas, em termos de *Acc* e AUC, face às rotações dispostas de forma aleatória aplicadas no *modelo base*.

Tabela 6.3: Resultados dos testes relativos ao aumento artificial de dados, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Amostras por imagem**				**Ordem**	**Acc**	**AUC**
	Originais	Rodadas	Elásticas	Total			
Sem aumento de dados	3000	-	-	3000	Aleatória	0.9557	0.9780
Amostras originais	12000	-	-	12000	Aleatória	0.9557	0.9785
Amostras de Simard	3000	-	9000	12000	Aleatória	0.9558	0.9785
Amostras de Ronneberger	3000	-	9000	12000	Aleatória	0.9558	0.9800
Amostras rodadas consecutivas	3000	9000	-	12000	Consecutiva	0.9560	0.9795
Modelo base	3000	9000	-	12000	Aleatória	**0.9570**	**0.9815**



(a) (b)

Figura 6.4: *Boxplots* de cada variante apresentada na tabela 6.3 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

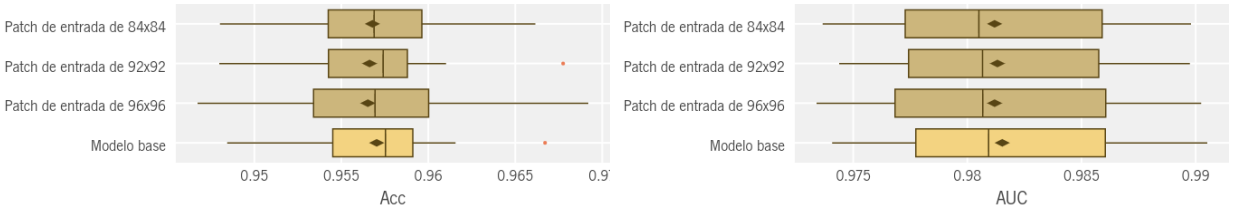
Analisando a figura 6.4, percebe-se que a dispersão dos resultados do *modelo base*, tanto em termos de *Acc* como de AUC, foi semelhante ou inferior à das restantes variantes. Em termos de *Acc*, os primeiro e segundo quartis do *modelo base* surgem visivelmente mais à frente dos das variantes alternativas. Isto significa que a estratégia de aumento artificial de dados usada, que assenta em rotações dispostas de forma aleatória, permitiu melhorar a eficácia de segmentação nas imagens mais desafiantes. Em termos de AUC, a tendência anterior manteve-se e houve também benefícios claros nos terceiro e quarto quartis. O mesmo é dizer que os efeitos positivos se fizeram sentir na globalidade das imagens: não só nas mais desafiantes, mas também nas teoricamente mais acessíveis.

6.5.3 Efeito do Tamanho do Patch de Entrada

Na secção 4.1.2.4, referiu-se que as principais vantagens de uma FCN são a eficiência computacional associada à segmentação por blocos e o facto da rede poder tomar uma imagem de qualquer tamanho como entrada. Posteriormente, na secção 5.2.2.2.2, mencionou-se que as dimensões espaciais do *patch* de entrada do *modelo base* são de 88×88 pixeis. Nesta secção, quantificam-se os efeitos que a alteração destas dimensões tem no desempenho da rede. Desta forma, o *modelo base* foi comparado com três variantes onde o tamanho do *patch* de entrada se fixou em 84×84, 92×92 e 96×96 pixeis, respetivamente. O tamanho do *patch* de saída, que será avaliado na secção seguinte, foi mantido em 32×32 pixeis tanto no *modelo base* como nas variantes.

Tabela 6.4: Resultados dos testes relativos ao tamanho do *patch* de entrada, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Patch de entrada**	**Patch de saída**	**Acc**	**AUC**
Patch de entrada de 84×84	84×84	32×32	0.9568	0.9812
Patch de entrada de 92×92	92×92	32×32	0.9566	0.9813
Patch de entrada de 96×96	96×96	32×32	<u>0.9565</u>	<u>0.9812</u>
Modelo base	88×88	32×32	**0.9570**	**0.9815**



(a) (b)

Figura 6.5: *Boxplots* de cada variante apresentada na tabela 6.4 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

A tabela 6.4 compara os valores médios das várias variantes com os do *modelo base*. Como se pode verificar, todas as variantes introduziram reduções ligeiras em termos de *Acc* e AUC. Quando se usaram *patches* de entrada de 84×84 e 92×92 pixeis, as diferenças não se revelaram estatisticamente significativas. Quando o tamanho passou a ser de 96×96 pixeis, os resultados mantiveram-se semelhantes em termos absolutos, mas as diferenças significativas já se fizeram sentir.

No que diz respeito à distribuição dos resultados, a figura 6.5 sugere que a variante que fez uso de *patches* de entrada de 92×92 pixeis foi a que mais se aproximou do *modelo base*, tanto em termos de *Acc* como de AUC. Quanto às variantes com 84×84 e 96×96 pixeis, percebe-se que os resultados se tornaram mais dispersos e, principalmente no segundo caso, os primeiro e segundo quartis situaram-se mais abaixo, pelo que houve uma dificuldade acrescida para lidar com as imagens mais desafiantes.

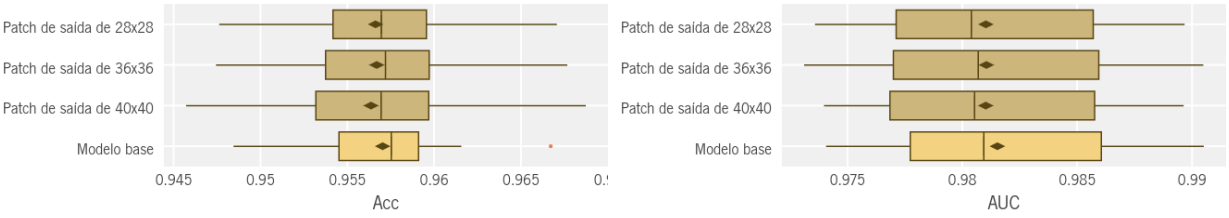
Em resumo, os melhores resultados foram obtidos utilizando *patches* de entrada de 88×88 pixeis e, por isso, este tamanho foi mantido. No que diz respeito às variantes, pareceu haver uma tendência para que os resultados se deteriorassem à medida que o tamanho dos *patches* de entrada se afastava deste valor de referência.

6.5.4 Efeito do Tamanho do Patch de Saída

Havendo fixado o tamanho do *patch* de entrada em 88×88 pixeis, passa-se agora ao estudo do tamanho do *patch* de saída. Desde logo, importa dizer que não existe uma forma inequívoca e globalmente aceite de definir o tamanho do *patch* de saída, em função do *patch* de entrada, numa FCN com múltiplas escalas como a que é usada no *modelo base*. O que se faz atualmente é estabelecer uma estimativa que, na maioria das vezes, visa restringir o *patch* de saída a unidades cujo *receptive field* não tenha sido afetado pelas sucessivas operações de *zero padding* que vão ocorrendo ao longo da rede. Com base nesta estimativa inicial, o tamanho do *patch* é depois ajustado através de testes suplementares.

Tabela 6.5: Resultados dos testes relativos ao tamanho do *patch* de saída, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Patch de entrada**	**Patch de saída**	**Acc**	**AUC**
Patch de saída de 28×28	88×88	28×28	0.9566	0.9810
Patch de saída de 36×36	88×88	36×36	0.9567	0.9810
Patch de saída de 40×40	88×88	40×40	0.9564	0.9810
Modelo base	88×88	32×32	**0.9570**	**0.9815**



(a) (b)

Figura 6.6: *Boxplots* de cada variante apresentada na tabela 6.5 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

Para que se possa entender o cálculo desta estimativa, recorde-se a figura 5.8 que ilustra a arquitetura do modelo base. A rede agrega três escalas. A primeira é composta pelos dois blocos convolucionais com 32 filtros, a segunda engloba os dois blocos convolucionais com 64 filtros e a terceira consiste no bloco convolucional com 128 filtros. Como cada bloco convolucional possui três camadas, na primeira e segunda escalas há seis operações de convolução, enquanto na terceira existem três. Para dimensionar o *patch* de saída é necessário estimar a quantidade de zeros que se acrescenta em cada convolução. No caso da primeira escala, como o tamanho do filtro é de 3×3 e se usam convoluções *same*, são acrescentadas duas colunas/linhas de zeros por convolução (relembre-se a figura 4.7b). À medida que surgem camadas de *pooling* este número duplica. Por outras palavras, cada operação de convolução na segunda escala equivale a quatro colunas/linhas de zeros e na terceira escala a oito. Dito isto, é possível obter uma estimativa do número total de linhas/colunas de zeros adicionadas ao longo da rede, multiplicando o número de convoluções em cada escala pelo número de zeros que cada convolução dessa escala acrescenta. Neste caso, tem-se: $6 \times 2 + 6 \times 4 + 3 \times 8 = 12 + 24 + 24 = 60$. Como o tamanho do *patch* de entrada foi fixado em 88×88, a estimativa para o tamanho do *patch* de saída vem $(88 - 60) \times (88 - 60) = 28 \times 28$ pixeis.

Nesta secção, compara-se o *modelo base*, que faz uso de um *patch* de saída de 32×32 pixeis, com três variantes. Na primeira, utiliza-se um *patch* de saída de 28×28 pixeis, com base na estimativa anterior. Na segunda e terceira variantes, usam-se *patches* de 36×36 e 40×40 pixeis, respetivamente.

Observando a tabela 6.5, percebe-se que o *modelo base* permitiu obter os melhores resultados, em termos de *Acc* e AUC. Face a ele, as variantes que fizeram uso de *patches* de saída de 28×28 e 36×36 pixeis geraram reduções estatisticamente significativas em termos de AUC. Quando o tamanho do *patch* de saída passou a ser de 40×40 pixeis, surgiram reduções estatisticamente significativas também em

termos de *Acc*. Este comportamento vem confirmar a dificuldade para estimar de forma inequívoca o tamanho ideal para o *patch* de saída. No entanto, as ligeiras diferenças entre os resultados obtidos pela estimativa (28 $\times$ 28 pixeis) e pelo valor otimizado utilizado no *modelo base* (32 $\times$ 32 pixeis) parecem conferir alguma credibilidade à estimativa inicial.

Através da análise da figura 6.6, é ainda possível perceber que a variante de 28 $\times$ 28 pixeis foi a que mais se aproximou do *modelo base* relativamente à dispersão dos resultados, tanto em termos de *Acc* como de AUC. Quanto às variantes que utilizaram tamanhos maiores (36 $\times$ 36 e 40 $\times$ 40 pixeis), parece ter havido uma dificuldade acrescida para classificar imagens mais desafiantes, visto que os primeiro e segundo quartis se situaram mais abaixo em ambas as métricas.

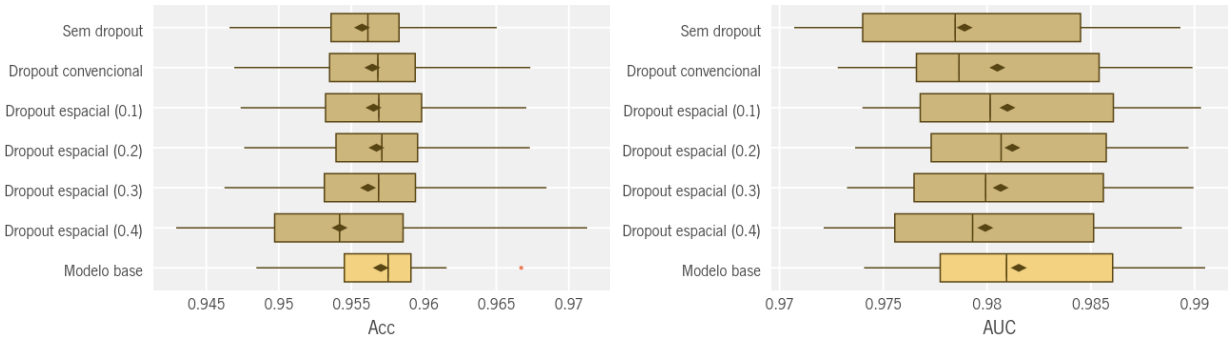
6.5.5 Efeito do Dropout

Definida a arquitetura do modelo, passa-se agora ao estudo da regularização. Na secção 5.2.2.2.3, referiu-se que o *modelo base* faz uso do *dropout* espacial com $p = 0.2$ em todos os blocos convolucionais, exceto no último onde se tem $p_{último} = 0.15$. Nesta secção, esta estratégia é comparada com seis outras variantes. Na primeira, abdica-se de qualquer tipo de *dropout* – *sem dropout*. Na segunda, faz-se uso dos mesmos valores do *modelo base*, mas aplica-se a versão tradicional onde apenas algumas unidades de cada *feature map* são canceladas – *dropout convencional*. Nas restantes variáveis, emprega-se *dropout* espacial com valores sucessivamente maiores de p – *dropout espacial (*p*)*.

Examinando a tabela 6.6, verifica-se que a variante *sem dropout* trouxe diminuições consideráveis e estatisticamente significativas em termos de *Acc* e de AUC. Na variante *dropout convencional*, também se verificaram diferenças estatisticamente significativas, ainda que mais moderadas em termos absolutos. As variantes *dropout espacial (0.1)* e *dropout espacial (0.2)* parecem ter sido as que mais se aproximaram do modelo base, com a segunda a ser inclusivamente a única onde não foram encontradas diferenças significativas. Ainda ssim, com base nela, foi possível comprovar que a utilização de $p_{último} = 0.15$ é benéfica. Por fim, as variantes *dropout espacial (0.3)* e *dropout espacial (0.4)* vêm mostrar que a utilização do *dropout* espacial com uma probabilidade excessiva é capaz de deteriorar os resultados, visto que reduções consideráveis e estatisticamente significativas foram encontradas. No caso da última variante, os resultados são inclusivamente comparáveis aos que se obtiveram na variante *sem dropout*.

Tabela 6.6: Resultados dos testes relativos à estratégia de regularização, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Tipo de dropout**	p	$p_{último}$	**Acc**	**AUC**
Sem *dropout*	-	-	-	0.9557	0.9789
Dropout convencional	Convencional	0.2	0.15	0.9564	0.9805
Dropout espacial (0.1)	Espacial	0.1	0.1	0.9565	0.9810
Dropout espacial (0.2)	Espacial	0.2	0.2	0.9567	0.9812
Dropout espacial (0.3)	Espacial	0.3	0.3	0.9562	0.9807
Dropout espacial (0.4)	Espacial	0.4	0.4	0.9542	0.9799
Modelo base	Espacial	0.2	0.15	**0.9570**	**0.9815**



(a) (b)

Figura 6.7: *Boxplots* de cada variante apresentada na tabela 6.6 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

Através da figura 6.7, pode ainda comparar-se o *modelo base* com as variantes alternativas, em termos de distribuição dos resultados. Tanto em termos de *Acc* como de AUC, é possível perceber que o efeito das diversas variantes se fez notar, sobretudo, através da descida dos primeiro e segundos quartis, pelo que houve uma dificuldade acrescida para lidar com imagens desafiantes. Além disto, principalmente em termos de *Acc*, a maioria das variantes parece ter tornado os resultados mais dispersos.

6.5.6 Efeito do Tamanho do Mini-batch

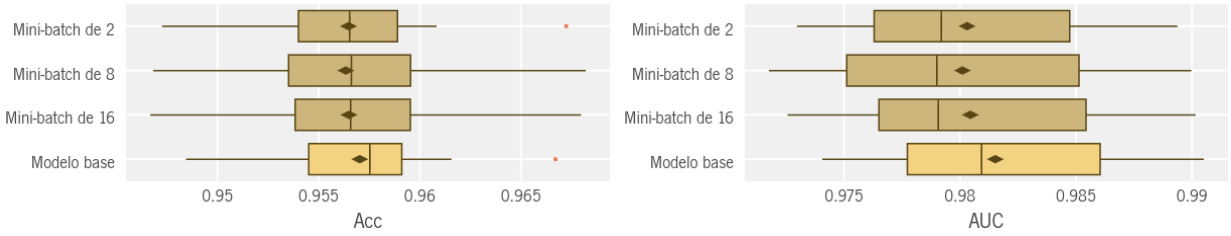
Como se referiu na secção 4.1.3.1.4, ao otimizar um modelo através de um método como o SGD, o tamanho do *mini-batch* ($\hat{M}$) é um hiperparâmetro de grande importância. Na secção 5.2.2.2.3, referiu-se que o *modelo base* faz uso de um *mini-batch* de 4. Nesta secção, o *modelo base* é comparado com três variantes onde $\hat{M}$ toma os valores 2, 8 e 16, respetivamente.

Observando a tabela 6.7, é possível concluir que todas as variantes alternativas introduziram reduções estatisticamente significativas em termos de *Acc* e AUC, ainda que em termos absolutos as diferenças para o *modelo base* sejam relativamente baixas. Para além disto, um aspeto importante residiu na duração da etapa de treino. Quando o tamanho do *mini-batch* foi fixado em 2, foram necessárias 35 épocas para que o modelo convergisse. Isto parece comprovar a teoria de que um *mini-batch* demasiado pequeno, apesar de permitir que se façam mais iterações por época, é sinónimo de uma estimativa menos precisa do gradiente e, por isso, torna a convergência mais lenta. Por outro lado, quando o tamanho do *mini-batch* foi de 4, 8 ou 16, o modelo convergiu ao fim de 20 épocas e o *mini-batch* de 4, usado no *modelo base*, foi o que levou a melhores resultados. Isto parece atestar que a partir de um certo limite, o aumento do tamanho do *mini-batch* deixa de ser benéfico para o processo de otimização.

Olhando para a figura 6.8, é possível perceber que a variante que fez uso de um *mini-batch* de 2 foi a que apresentou um comportamento mais semelhante ao *modelo base* no que diz respeito à distribuição dos resultados. Em termos de *Acc*, as três variantes provocaram uma descida nos primeiro e segundo quartis e uma subida nos terceiro e quarto. Por outras palavras, a eficácia de segmentação saiu prejudicada mas imagens mais desafiantes, mas beneficiada nas imagens menos problemáticas. Em termos de AUC, as diversas variantes provocaram uma descida generalizada em todos os quartis, pelo que houve uma dificuldade acrescida para lidar com todas as imagens.

Tabela 6.7: Resultados dos testes relativos ao tamanho do *mini-batch*, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	$\hat{M}$	**Épocas**	**Acc**	**AUC**
Mini-batch de 2	2	35	0.9565	0.9803
Mini-batch de 8	8	20	0.9563	0.9801
Mini-batch de 16	16	20	0.9565	0.9804
Modelo base	4	20	**0.9570**	**0.9815**



(a) (b)

Figura 6.8: *Boxplots* de cada variante apresentada na tabela 6.7 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

6.5.7 Efeito da Segmentação Múltipla

Nesta secção, analisam-se os efeitos da estratégia de segmentação múltipla apresentada na secção 5.2.3. Deste modo, compara-se o *modelo base*, que faz uso desta estratégia, com uma variante onde se utiliza a segmentação simples.

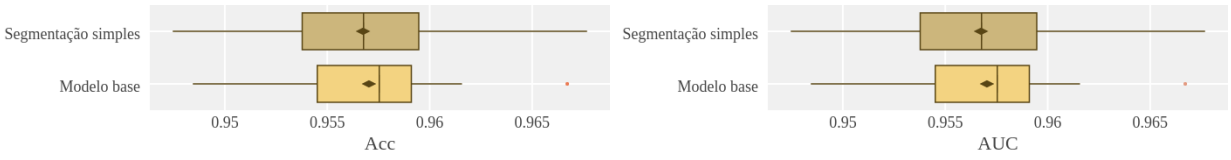
Com base na tabela 6.8, é possível perceber que, face à estratégia de segmentação simples, a segmentação múltipla trouxe melhorias estatisticamente significativas em termos de *Acc* e AUC.

Além disto, através da figura 6.9, pode ainda concluir-se que os resultados obtidos pela segmentação múltipla são menos dispersos. Em particular, houve uma subida nos primeiro e segundo quartis e uma descida nos terceiro e quarto. Isto significa que o modelo teve maior facilidade para lidar com imagens mais desafiantes, mas que não obteve resultados tão elevados nas imagens teoricamente mais acessíveis.

No cômputo geral, a inclusão da segmentação múltipla revelou-se positiva e melhorou a eficácia de segmentação. Isto significa que ao ver os mesmos pixeis, dispostos numa orientação espacial diferente, a rede foi capaz de extrair informação que não era acessível inicialmente. Isso pode dever-se ao facto dos filtros convolucionais apresentarem respostas diferentes ao processar *patches* com orientações distintas.

Tabela 6.8: Resultados dos testes relativos à estratégia de segmentação, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Segmentação**	**Acc**	**AUC**
Segmentação simples	Simples	0.9566	0.9805
Modelo base	Múltipla	**0.9570**	**0.9815**



(a) (b)

Figura 6.9: *Boxplots* de cada variante apresentada na tabela 6.8 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

Por outras palavras, as CNNs são invariantes à translação, mas não à rotação. Isto pode ajudar a explicar os motivos pelos quais a estratégia de aumento de dados baseada em rotações se revelou tão eficaz.

6.6 Obtenção do Melhor Modelo pela Aplicação da SWT

No capítulo anterior, a inclusão da SWT foi abordada na fase de pré-processamento visto que o objetivo passou por construir uma nova entrada para a FCN, numa etapa prévia à classificação. Apesar disto, durante o desenvolvimento do trabalho, a ideia de incluir canais extra na entrada só surgiu depois do *modelo base* estar estabelecido. Assim, nesta secção, descreve-se o processo através do qual se obteve o melhor modelo a partir do *modelo base*.

Como se descreveu na secção 5.2.1.3, a aplicação da SWT passou pela concatenação de novos canais na entrada. Nesta secção, compara-se o *modelo base (MB)*, que só faz uso do canal verde original, com diversas variantes onde se utilizaram diferentes conjuntos de canais. Para que se possa compreender melhor as variantes propostas, importa fazer um ponto prévio para esclarecer a notação utilizada: a_j representa o coeficiente de aproximação do nível j e corresponde a um só canal; d_j compreende o conjunto de coeficientes de detalhe (dV_j, dH_j e dD_j) do nível j e engloba três canais.

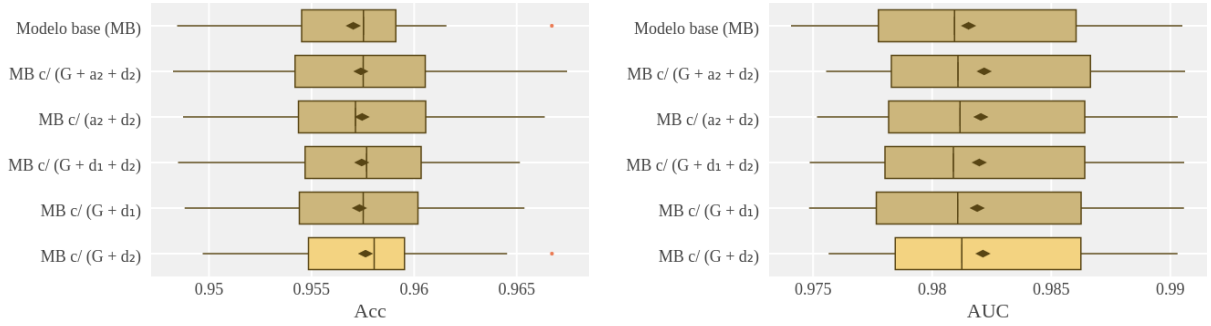
Nas três primeiras variantes, recorreu-se apenas a canais do segundo nível. Inicialmente, utilizaram-se todos os coeficientes do segundo nível e o canal verde (G) – *MB c/ (G + a_2 + d_2)*. Depois, o canal verde foi removido e mantiveram-se os coeficientes – *MB c/ (a_2 + d_2)*. Por fim, recorreu-se novamente ao canal verde e abdicou-se, desta feita, do coeficiente de aproximação – *MB c/ (G + d_2)*. Além destas três variantes, introduziram-se mais duas onde se avaliou a inclusão dos coeficientes de primeiro nível. Na variante *MB c/ (G + d_1 + d_2)*, utilizaram-se os coeficientes de detalhe dos primeiro e segundo nível, juntamente com o canal verde. Na variante *MB c/ (G + d_1)*, só os coeficientes de detalhe do primeiro nível foram mantidos.

Analisando a tabela 6.9, é possível verificar que todas as variantes introduziram melhorias face ao *modelo base*, tanto em termos de *Acc* como de AUC. Isto parece comprovar a teoria de que mesmo os métodos de *deep learning* podem beneficiar da inclusão de conhecimento de domínio. Começando pelas variantes que utilizaram canais do segundo nível, é possível aferir que a inclusão do coeficiente de aproximação não trouxe benefícios visto que a variante *MB c/ (G + d_2)* permitiu obter melhores resultados do que a variante *MB c/ (G + a_2 + d_2)*. Além disto, substituir o canal verde original pelo coeficiente de aproximação também se revelou prejudicial visto que a variante *MB c/ (G + d_2)* foi mais eficaz que a variante *MB c/ (a_2 + d_2)*. Por este motivo, quando se incluíram os canais do primeiro nível só se aplicaram os coeficientes de detalhe. Comparando as variantes *MB c/ (G + d_1)*, *MB c/ (G + d_1 + d_2)*

Tabela 6.9: Resultados dos testes relativos à inclusão de canais extra através da SWT, na base de dados DRIVE. Valores a negrito representam o melhor resultado entre todas as variantes. Valores sublinhados indicam diferenças estatisticamente significativas face ao Modelo base (p-*value* < 0.05)

Variante	**Número de canais**				**Acc**	**AUC**
	Originais	1° nível	2° nível	Total		
Modelo base (MB)	1	-	-	1	0.9570	0.9815
MB c/ $(G + a_2 + d_2^*)$	1	-	4	5	0.9574	0.9820
MB c/ $(a_2 + d_2^*)$	-	-	4	4	0.9575	0.9820
MB c/ $(G + d_1^* + d_2^*)$	1	3	3	7	0.9574	0.9820
MB c/ $(G + d_1^*)$	1	3	-	4	0.9573	0.9819
MB c/ $(G + d_2^*)$	1	-	3	4	**0.9576**	**0.9821**

* Por concisão de notação, d_j engloba os canais dV_j, dH_j e dD_j.



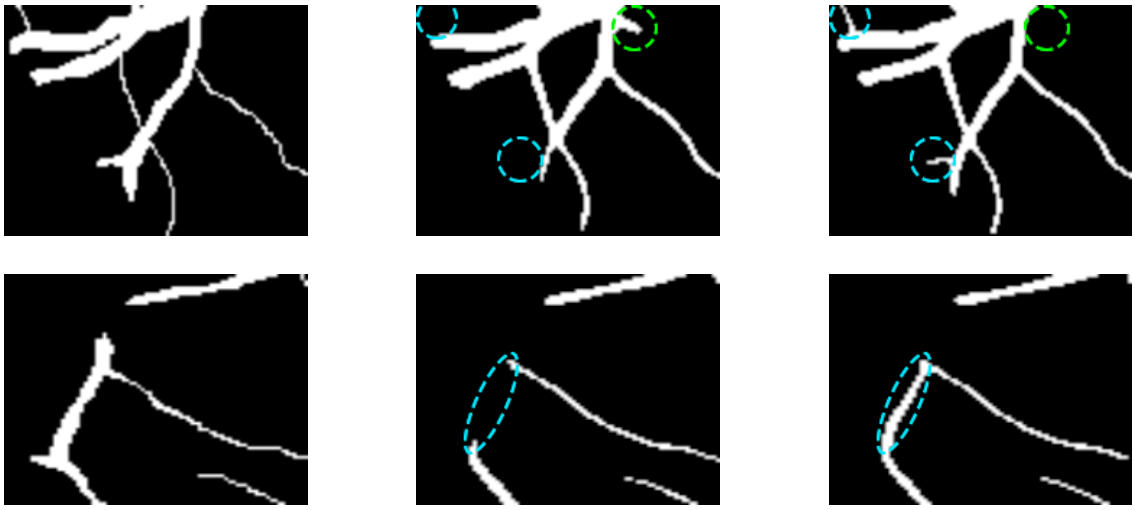
(a) (b)

Figura 6.10: *Boxplots* de cada variante apresentada na tabela 6.9 em termos de: (a) *Acc*; (b) AUC. A média e mediana de cada variante são representas pelo losango e linha vertical, respetivamente. Os pontos vermelhos representam *outliers*.

e *MB c/ (G + d_2)* é possível perceber que os coeficientes de detalhe do segundo nível foram os que permitiram obter melhores resultados. Sabe-se que os coeficientes de primeiro nível traduzem informação de frequências mais elevadas. Isto parece ter aumentado a propensão do modelo para introduzir falsas deteções, baixando o acerto global. A variante *MB c/ (G + d_2)* foi, inclusivamente, a única capaz de introduzir melhorias estatisticamente significativas em termos de *Acc* e AUC, visto que nos restantes casos as diferenças significativas se cingiram à AUC.

Observando a figura 6.10, é ainda possível comprovar que, em termos de *Acc*, a maioria das variantes tornou os resultados mais dispersos face ao que se passava no *modelo base*. Houve uma ligeira descida nos primeiro e segundo quartis, mas uma subida notória nos terceiro e quarto. Isto traduz um desempenho ligeiramente pior nas imagens mais desafiantes, mas bastante melhor nas imagens mais acessíveis. Assim, em termos médios, as diversas variantes acabaram por introduzir efeitos positivos. Relativamente à AUC, a maioria das variantes provocou uma subida generalizada em todos os quartis, o que significa uma melhor capacidade de segmentação na globalidade das imagens.

Em resumo, a variante *MB c/ (G + d_2)* assumiu-se como o melhor modelo, permitindo reduzir a combinação entre falsos positivos e falsos negativos, com efeitos visíveis nas imagens. A figura 6.11 ilustra dois exemplos desse comportamento, sendo possível observar zonas onde a rede passou a ignorar



(a) (b) (c)

Figura 6.11: Efeito da inclusão dos canais da SWT em dois casos distintos: (a) Anotação do 1º observador; (b) Segmentação do *modelo base*; (c) Segmentação do melhor modelo (*MB c/ (G + d_2)*). Marcações a verde indicam redução dos falsos positivos; marcações a azul indicam redução dos falsos negativos.

segmentos de vasos que haviam sido mal detetados e outras onde a rede foi capaz de detetar segmentos reais que haviam sido ignorados inicialmente.

Nas secções que se seguem, o melhor modelo é designado por *método proposto*.

6.7 Resultados de Segmentação

Nesta secção, fornece-se uma visão geral dos resultados obtidos pelo método proposto nas três bases de dados utilizadas. Além disto, comparam-se os resultados obtidos através da metodologia de segmentação automática com os resultados de um segundo observador humano. As segmentações binárias foram obtidas aplicando um *threshold* fixo de 0.5 aos mapas de probabilidades produzidos pela rede. Em todos os resultados apresentados nesta e em futuras secções, apenas se consideram pixeis dentro do FOV. Por outras palavras, os resultados dizem apenas respeito à área delimitada pela máscara de cada imagem. Isto é importante, porque avaliando o método em toda a imagem os resultados seriam inflacionados através do acerto massivo em pixeis do *background*.

Tabela 6.10: Comparação entre os resultados do método proposto e do segundo observador. Valores a negrito representam o melhor resultado entre os dois métodos

Base de dados	Método	Sen	Esp	Acc	AUC
DRIVE	2º Observador	0.7760	0.9725	0.9473	-
	Proposto	**0.8039**	**0.9804**	**0.9576**	**0.9821**
STARE	2º Observador	**0.8956**	0.9381	0.9346	-
	Proposto	0.8315	**0.9858**	**0.9694**	**0.9905**
CHASE_DB1	2º Observador	0.7705	0.9778	0.9561	-
	Proposto	**0.7779**	**0.9864**	**0.9653**	**0.9855**

Os resultados de segmentação do método proposto são comparados com os do segundo observador humano na tabela 6.10. Como se pode verificar, o método proposto apresentou melhores valores de *Acc* que o segundo observador em todas as bases de dados. Isto significa que, em termos globais, o modelo automático foi capaz de se aproximar mais das segmentações do primeiro observador do que um segundo observador humano independente. Em termos de *Esp*, a tendência anterior manteve-se, pelo que o modelo introduziu menos falsos positivos que o segundo observador. Em termos de *Sen*, o modelo superou o observador humano na DRIVE e na CHASE_DB1. Isto revela uma boa capacidade para detetar os vasos. Na STARE, apesar dos valores de *Sen* até terem sido mais elevados do que nas restantes bases de dados, o modelo não foi capaz de igualar o segundo observador. Isto deve-se à grande discrepância que existe entre os padrões de marcação dos primeiro e segundo observadores, nesta base de dados. Por outras palavras, na STARE, o segundo observador marca sistematicamente mais vasos que o primeiro. Isto permite-lhe obter um valor de *Sen* extremamente elevado, mas tem repercussões no valor de *Acc* devido às falsas deteções introduzidas. No que diz respeito à AUC, não é possível estabelecer esta comparação porque não existem mapas de probabilidades no caso do segundo observador humano.

Os valores de *Sen*, *Esp*, *Acc* e AUC do melhor caso, na DRIVE, foram 0.9119, 0.9742, 0.9667 e 0.9903, enquanto no pior se teve 0.7628, 0.9816, 0.9497 e 0.9786, respectivamente (figura 6.12). Na STARE, o melhor caso obteve 0.8527, 0.9936, 0.9837 e 0.9964, enquanto o pior apresentou 0.7231, 0.9827, 0.9503 e 0.9791 (figura 6.13). Já na CHASE_DB1, o melhor caso fixou-se em 0.8541, 0.9844, 0.9744 e 0.9909, enquanto no pior se obteve 0.8065, 0.9749, 0.9574 e 0.9808 (figura 6.14).

Em termos computacionais, o treino do modelo demorou cerca de 4 horas e cada imagem levou cerca de 2 segundos a ser segmentada.

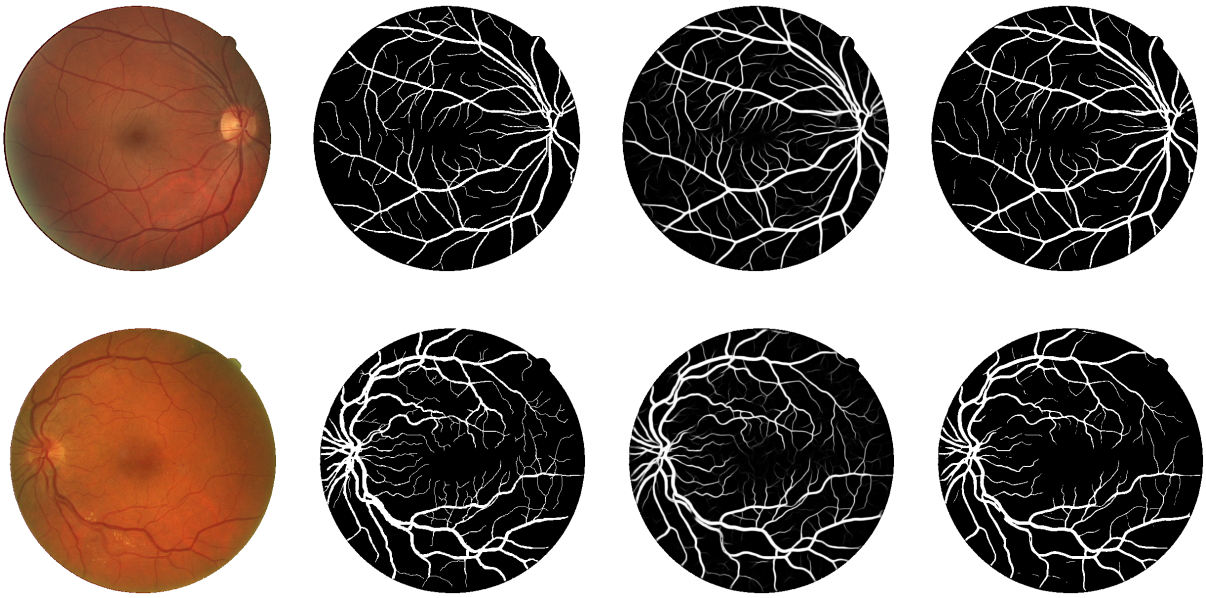


Figura 6.12: Exemplos de segmentação na base de dados DRIVE: (a) Imagem original; (b) Anotação do primeiro observador; (c) Mapa de probabilidades; (d) Segmentação binária. A primeira linha representa o melhor caso, enquanto a segunda ilustra o pior.

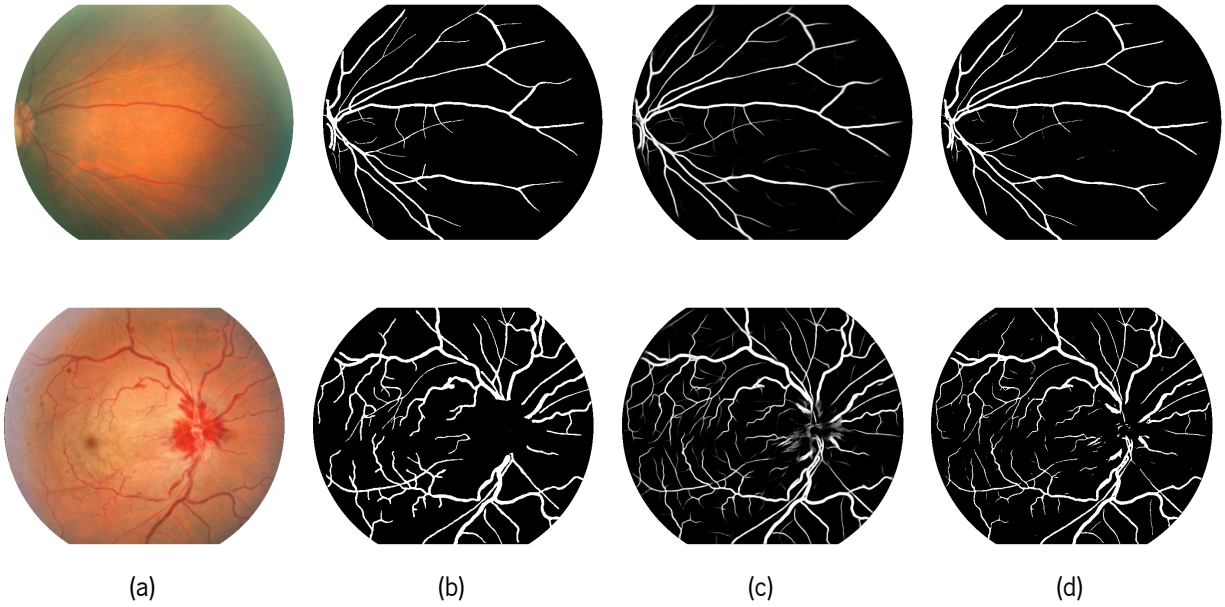


Figura 6.13: Exemplos de segmentação na base de dados STARE: (a) Imagem original; (b) Anotação do primeiro observador; (c) Mapa de probabilidades; (d) Segmentação binária. A primeira linha representa o melhor caso, enquanto a segunda ilustra o pior.

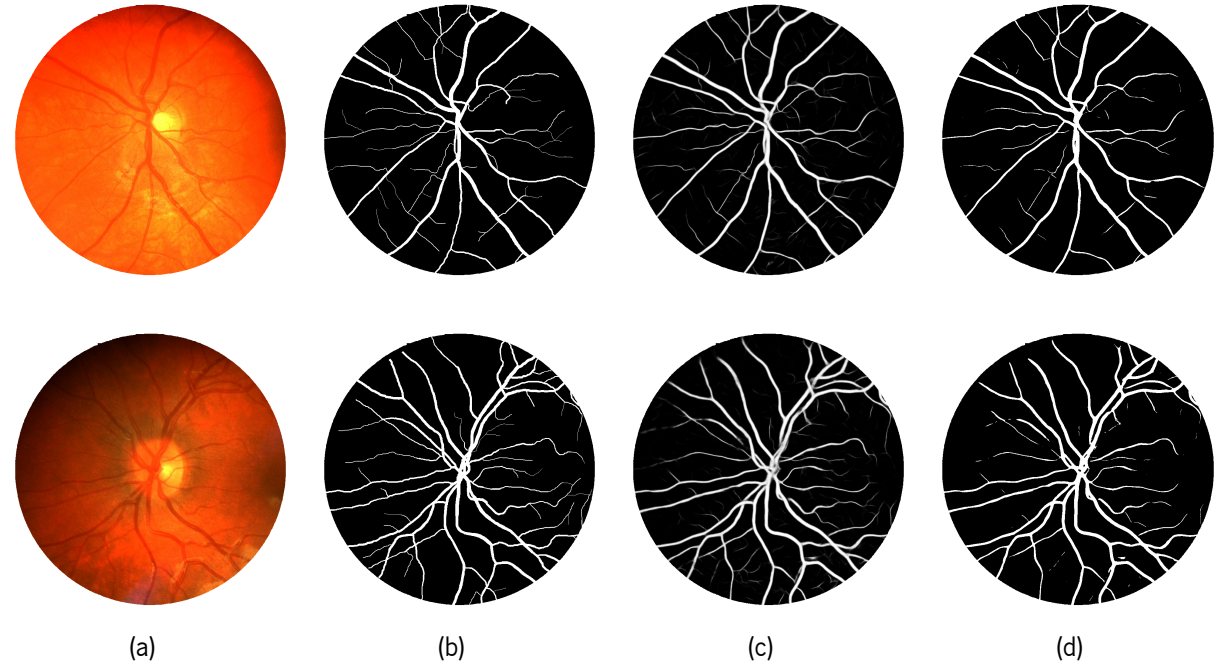


Figura 6.14: Exemplos de segmentação na base de dados CHASE_DB1: (a) Imagem original; (b) Anotação do primeiro observador; (c) Mapa de probabilidades; (d) Segmentação binária. A primeira linha representa o melhor caso, enquanto a segunda ilustra o pior.

Tabela 6.11: Resultados de segmentação na DRIVE, STARE e CHASE_DB1. Valores a negrito representam o melhor resultado entre todos os métodos

			DRIVE				**STARE**				**CHASE_DB1**			
Tipo	**Método**	**Ano**	**Sen**	**Esp**	**Acc**	**AUC**	**Sen**	**Esp**	**Acc**	**AUC**	**Sen**	**Esp**	**Acc**	**AUC**
Não Supervisionado	Mendonça e Campilho [42]	2006	0.7344	0.9764	0.9452	-	0.6996	0.9730	0.9440	-	-	-	-	-
	Lam *et al.* [58]	2010	-	-	0.9472	0.9614	-	-	0.9567	0.9739	-	-	-	-
	Wang *et al.* [44]	2013	-	-	0.9461	0.9543	-	-	0.9521	0.9682	-	-	-	-
	Azzopardi *et al.* [30]	2015	0.7655	0.9704	0.9442	0.9614	0.7716	0.9701	0.9497	0.9563	0.7585	0.9587	0.9387	0.9487
	Imani *et al.* [50]	2015	0.7524	0.9753	0.9523	0.9544	0.7502	0.9745	0.9590	0.9526	-	-	-	-
	Zhao *et al.* [33]	2015	0.7420	0.9820	0.9540	0.8620	0.7800	0.9780	0.9560	0.8740	-	-	-	-
	Kovács e Hadju [46]	2016	0.7450	0.9793	0.9494	0.9722	0.8034	0.9785	0.9610	0.9836	-	-	-	-
	Zhang *et al.* [47]	2016	0.7743	0.9725	0.9476	0.9636	0.7791	0.9758	0.9554	0.9748	0.7626	0.9661	0.9452	0.9606
Supervisionado	Staal *et al.* [62]	2004	-	-	0.9441	0.9520	-	-	0.9516	0.9614	-	-	-	-
	Soares *et al.* [40]	2006	0.7332	0.9782	0.9466	0.9614	0.7207	0.9747	0.9480	0.9671	-	-	-	-
	You *et al.* [64]	2011	0.7410	0.9751	0.9434	-	0.7260	0.9756	0.9497	-	-	-	-	-
	Marin *et al.* [65]	2011	0.7067	0.9801	0.9452	0.9588	-	-	-	-	-	-	-	-
	Fraz *et al.* [43]	2012	0.7406	0.9807	0.9480	0.9747	0.7548	0.9763	0.9534	0.9768	0.7224	0.9711	0.9469	0.9712
	Roychowdhury *et al.* [66]	2015	0.7249	**0.9830**	0.9519	0.9620	0.7719	0.9726	0.9515	0.9688	0.7201	0.9824	0.9530	0.9532
	Zhang *et al.* [67]	2017	0.7861	0.9712	0.9466	0.9703	0.7882	0.9729	0.9547	0.9740	0.7644	0.9716	0.9502	0.9706
Deep Learning	Melinscak *et al.* [74]	2015	0.7276	0.9785	0.9466	0.9749	-	-	-	-	-	-	-	-
	Li *et al.* [71]	2016	0.7569	0.9816	0.9527	0.9738	0.7726	0.9844	0.9628	0.9879	0.7507	0.9793	0.9581	0.9716
	Liskwoski e Krawiec [75]	2016	0.7520	0.9806	0.9515	0.9710	0.8145	**0.9866**	**0.9696**	0.9880	-	-	-	-
	Fu *et al.* [76]	2016	0.7603	-	0.9523	-	0.7412	-	0.9585	-	0.7130	-	0.9489	-
	Proposto	2017	**0.8039**	0.9804	**0.9576**	**0.9821**	**0.8315**	0.9858	0.9694	**0.9905**	**0.7779**	**0.9864**	**0.9653**	**0.9855**

6.8 Comparação com o Estado da Arte

O método proposto é comparado com outros métodos do estado da arte na tabela 6.11. Entre os vários trabalhos descritos no capítulo 3, incluem-se apenas os mais recentes ou com maior impacto.

Analisando a tabela 6.11, verifica-se que o método obteve a melhor *Sen* nas três bases de dados, evidenciando uma grande capacidade para detetar os vasos. Além disto, como a *Esp* se manteve elevada, esta capacidade não foi obtida à custa da inclusão de falsas deteções. Isto é particularmente importante na área médica, ainda mais na segmentação de vasos onde as classes são altamente desbalanceadas.

Avaliando conjuntamente as métricas *Sen*, *Esp* e *Acc*, o método apresentou o melhor desempenho na DRIVE e CHASE_DB1, estando bem classificado na STARE.

Começando pela DRIVE, o método apresentou as melhores *Sen* e *Acc*. Em termos de *Esp*, o método situou-se em sexto. No entanto, sabendo que a *Acc* engloba a informação da *Sen* e da *Esp*, pode concluir-se que o ganho nos verdadeiros positivos foi mais significativo que o aumento dos falsos positivos. Por outras palavras, há um compromisso notório entre *Sen* e *Esp*, com o método proposto a ter a *Sen* mais elevada e a apresentar o melhor balanço visto que também é o melhor em termos de *Acc*.

Em relação à STARE, o método apresentou a melhor *Sen* e a segunda melhor *Esp* e *Acc*, apenas superado por Liskowski e Krawiec [75]. Observando a segunda linha da figura 6.13, é possível verificar que, no pior caso, o método introduziu algumas falsas deteções na presença de hemorragias, o que pode ter afetado a *Esp*. Isto poderá estar relacionado com o uso da validação cruzada k-*fold*, que reduz o número de imagens patológicas usadas durante o treino. O principal motivo para a utilização desta estratégia face a outras, como a *leave-one-out* usada em [75], prendeu-se com a redução do número total de treinos a efetuar.

Por fim, na CHASE_DB1, o método obteve o melhor resultado em termos de *Sen*, *Esp* e *Acc*.

Comparando o método proposto com os restantes métodos de *deep learning*, realça-se que foi possível obter a melhor *Sen* em todas as bases de dados e a melhor *Acc* na DRIVE e CHASE_DB1 (ficando muito próximo de [75] na STARE). Na DRIVE, que é provavelmente a base de dados mais conceituada nesta área, percebe-se que até ao surgimento do método proposto nesta dissertação nenhum dos melhores trabalhos em termos de *Sen*, *Esp* ou *Acc* se baseava em *deep learning*.

Ao contrário da *Sen*, *Esp* e *Acc*, a AUC não depende do *threshold* utilizado para produzir as segmentações binárias. Para mais, ela é particularmente relevante na análise de classes desbalanceadas [124]. Com isto em mente, realça-se que o método proposto apresentou a melhor AUC em todas as bases de dados. Além disto, este parece ter sido o primeiro trabalho a reportar valores de AUC superiores a 0.98 nas três bases de dados mais utilizadas nesta área.

6.9 Robustez face ao Conjunto de Treino

Em contexto clínico, é impraticável retreinar um modelo sempre que existam novas imagens a segmentar. Além disto, um método fiável deve ser capaz de segmentar eficazmente diferentes imagens, independentemente do sistema responsável pela aquisição. Dito isto, a robustez face ao conjunto de treino é crucial para que um modelo tenha aplicabilidade prática.

Tabela 6.12: Resultados de segmentação do treino cruzado entre a DRIVE e a STARE. Valores a negrito representam o melhor resultado entre todos os métodos

Conjunto de Teste (Conjunto de Treino)	**Método**	**Sen**	**Esp**	**Acc**	**AUC**
DRIVE (STARE)	Soares *et al.* [40]	-	-	0.9397	-
	Marin *et al.* [65]	-	-	0.9448	-
	Fraz *et al.* [43]	0.7242	0.9792	0.9456	0.9697
	Roychowdhury *et al.* [66]	-	-	0.9494	-
	Li *et al.* [71]	0.7273	0.9810	0.9486	0.9677
	Kovács e Hadju [46]	**0.7274**	0.9812	0.9489	-
	Zhang *et al.* [67]	-	-	0.9447	0.9593
	Proposto	0.6706	**0.9916**	**0.9505**	**0.9748**
STARE (DRIVE)	Soares *et al.* [40]	-	-	0.9327	-
	Marin *et al.* [65]	-	-	0.9526	-
	Fraz *et al.* [43]	0.7010	0.9770	0.9495	0.9660
	Roychowdhury *et al.* [66]	-	-	0.9510	-
	Li *et al.* [71]	0.7027	0.9828	0.9545	0.9671
	Kovács e Hadju [46]	0.7700	**0.9830**	**0.9605**	-
	Zhang *et al.* [67]	-	-	0.9488	0.9676
	Proposto	**0.8453**	0.9726	0.9597	**0.9846**

Nesta secção, analisa-se o comportamento do modelo quando é treinado na DRIVE e testado na STARE e vice-versa. Estas são as bases de dados mais usadas para este efeito no estado da arte.

Comparando as tabelas 6.11 e 6.12, é possível perceber que os valores de *Acc* caíram de 0.9576 e 0.9694 para 0.9505 e 0.9597, na DRIVE e STARE, respetivamente. Isto traduz perdas aproximadas de 0.7 % e 1.0 %, face aos 0.2 % e 0.1 % de Roychowdhury *et al.* [66] e aos 0.05 % e 0.05 % de Kovács e Hadju [46], que apresentaram as quedas percentuais mais baixas. Em termos de AUC, os resultados baixaram de 0.9821 e 0.9905 para 0.9748 e 0.9846, pela mesma ordem. Isto significa quedas de aproximadamente 0.7 % e 0.6 %, contra os 0.5 % e 1.1 % de Fraz *et al.* [43].

A quantificação destas perdas em termos percentuais pode penalizar o método proposto face a outros que apresentem resultados mais baixos na situação padrão, onde o teste e o treino são realizados na mesma base de dados. Deste modo, é também importante olhar para a tabela 6.12 em termos absolutos. O método proposto obteve o melhor resultado em termos de *Esp*, *Acc* e AUC, quando foi treinado na STARE e testado na DRIVE. Na situação inversa, obtiveram-se os valores mais elevados de *Sen* e AUC. Isto significa que no primeiro caso se obteve o melhor desempenho entre todos os métodos e que no segundo o desempenho foi comparável ao de Kovács e Hadju [46].

Em termos globais, quando o modelo foi treinado na STARE e testado na DRIVE, a capacidade para detetar os vasos tornou-se menor, havendo uma queda na *Sen* e, subsequentemente, na *Acc*. No caso inverso, uma vez que nas imagens da DRIVE, tipicamente, se observam mais vasos finos do que nas da STARE, esta capacidade de deteção aumentou consideravelmente. No entanto, apesar dos valores de *Sen* terem melhorado, existiram mais falsas deteções o que resultou na descida da *Esp* e da *Acc*. Em resumo,

apesar do desempenho do modelo ter piorado, como seria expectável, o facto da descida em cada uma das métricas nunca ter ultrapassado 1 % e, além disto, o facto do método continuar bem posicionado face ao estado da arte, parecem ser sinais encorajadores em termos de aplicabilidade prática.

6.10 Robustez face à Presença de Sinais Patológicos

Como se referiu no capítulo 2, a presença de sinais patológicos como hemorragias, microaneurismas ou neovasos, pode ser um problema importante na segmentação de vasos retinianos. Nesta secção, compara-se o desempenho do método proposto com o dos restantes métodos do estado da arte, quando todas as imagens a segmentar evidenciam sinais patológicos. Para este efeito, selecionaram-se 10 imagens da base de dados STARE.

Analisando a tabela 6.13, é possível concluir que os resultados obtidos se revelaram competitivos face aos de Liskowski e Krawiec [75]. O método proposto obteve o segundo valor mais elevado em termos de *Esp* e *Acc* e o melhor valor em termos de *Sen* e AUC. Este era já o comportamento verificado quando todas as imagens da STARE foram consideradas.

Face a essa situação inicial, o método apresentou quedas de 0.9694 e 0.9905 para 0.9674 e 0.9880 em termos de *Acc* e AUC, respetivamente. Isto traduz perdas percentuais que rondam os 0.2 % e 0.3 %, face aos 0.2 % e 0.4 % de Liskowski e Krawiec [75] e aos 0.1 % e 0.3 % de Lam *et al.* [58].

Sumariando, o método apresentou quedas percentuais abaixo dos 0.5 % ao ser avaliado em imagens com sinais patológicos e, além disto, manteve-se competitivo face ao estado da arte.

Tabela 6.13: Resultados de segmentação em 10 imagens com sinais patológicos da STARE. Valores a negrito representam o melhor resultado entre todos os métodos

Método	Sen	Esp	Acc	AUC
Mendonça e Campilho [42]	0.6733	0.9669	0.9388	-
Lam *et al.* [58]	-	-	0.9556	0.9707
Fraz *et al.* [43]	0.7262	0.9764	0.9511	-
Roychowdhury *et al.* [66]	0.7428	0.9718	0.9453	0.9596
Li *et al.* [71]	0.7800	0.9805	0.9672	-
Liskowski e Krawiec [75]	0.7653	**0.9880**	**0.9679**	0.9843
Proposto	**0.7990**	0.9854	0.9674	**0.9880**

6.11 Desempenho nos Vasos Finos e Grossos

Como já se referiu, uma característica importante de qualquer método automático de deteção de vasos é a sua capacidade para detetar os vasos mais finos. Nesta secção, compara-se o desempenho do modelo quando a avaliação é feita na globalidade dos vasos ou apenas nos vasos de maior ou menor calibre. Para realizar esta análise, tanto as segmentações do modelo como as anotações do 1º observador foram manipuladas, como se descreveu na secção 5.3.1, de forma a separar os vasos finos e grossos.

Tabela 6.14: Resultados de segmentação na DRIVE, STARE e CHASE_DB1, consoante o tipo de vasos considerado. Valores a negrito representam o melhor resultado

Base de dados	Método	Sen	Esp	Acc
DRIVE	Global	0.8039	0.9803	0.9576
	Vasos Grossos	**0.8343**	0.9826	0.9670
	Vasos Finos	0.3690	**0.9915**	**0.9772**
STARE	Global	0.8315	0.9858	0.9694
	Vasos Grossos	**0.8388**	0.9878	0.9726
	Vasos Finos	0.3581	**0.9944**	**0.9878**
CHASE_DB1	Global	0.7779	0.9864	0.9653
	Vasos Grossos	**0.7905**	0.9864	0.9675
	Vasos Finos	0.3665	**0.9981**	**0.9937**

A tabela 6.14 sumaria os resultados obtidos. Em termos globais, é possível verificar que quando o modelo foi avaliado apenas nos vasos finos ou grossos, os valores médios de *Acc* e *Esp* subiram face ao que se passava na globalidade dos vasos. Este aumento parece considerável, mas deve ser relativizado porque, nexte contexto, o desbalanceamento entre as classes se tornou ainda mais abrupto. Por outras palavras, quando se considerou apenas parte dos vasos, a *Acc* e a *Esp* subiram devido ao acerto em mais pixeis do *background*. Comparando os valores de *Acc* e *Esp* nos vasos grossos e finos, é ainda possível perceber que, em todas as bases de dados, os valores mais elevados foram obtidos nos vasos finos. Isto parece comprovar que estas subidas se relacionam com o desbalanceamento entre as classes, que é obviamente mais forte quando os vasos grossos são descartados e apenas os vasos finos se mantêm.

A *Sen* adquire particular importância neste contexto, porque traduz a capacidade do modelo para detetar os diferentes tipos de vasos. Observando os resultados, é possível aferir que a capacidade do modelo para detetar os vasos finos se fixou em 36/37 %, em todas as bases de dados. Isto traduz a dificuldade associada à deteção dos vasos de menor calibre, que é talvez o aspeto mais desafiante que se coloca aos métodos atuais. Por outro lado, valores elevados de *Sen* foram obtidos nos vasos grossos, o que vem confirmar a ideia de que a deteção dos vasos de maior calibre é uma tarefa menos exigente. Comparando ainda os valores de *Sen* obtidos nos vasos grossos com os reportados na globalidade dos vasos, verifica-se que existiu uma subida generalizada. Isto seria expectável porque no primeiro caso não se consideram os vasos mais finos, cuja deteção é mais desafiante. Neste contexto, percebe-se que a maior subida se deu na DRIVE, o que parece confirmar a ideia de que as imagens desta base de dados possuem mais vasos finos, como se referiu na secção 6.9.

Sintetizando, apesar da *Sen* obtida nos vasos finos se ter revelado muito inferior à *Sen* global como seria expectável, a capacidade de deteção foi semelhante em todas as bases de dados, o que é positivo porque significa que o modelo foi capaz de lidar com imagens de resoluções bastante distintas com eficácia semelhante. Além disto, os valores de *Sen* nos vasos grossos mantiveram-se muito altos e, conjuntamente com os elevados valores de *Esp*, ajudam a explicar a grande eficácia global do modelo.

6.12 Localização de Falsas Deteções nas Vizinhanças Próxima e Distante

Além da habilidade para detetar os vasos finos, a capacidade para evitar falsas deteções é outro fator crucial na avaliação dos métodos atuais. Nesta secção, analisa-se a distribuição dos falsos positivos (FP) nas vizinhanças próxima e distante, consoante o número de dilatações morfológicas utilizadas para definir a primeira região (recorde-se a secção 5.3.2). Os resultados são apresentados na tabela 6.15.

Na DRIVE, em média, a máscara de cada imagem é composta por 226907 pixeis e o modelo introduziu 3896 falsos positivos por imagem. Entre estas falsas deteções, cerca de 87.6 % localizaram-se na área imediatamente adjacente aos vasos, delimitada por uma única dilatação morfológica. Quando se utilizaram 2, 5 e 10 dilatações, obtendo regiões próximas sucessivamente maiores, esta percentagem passou a 94.1 %, 95.9 % e 97.7 %, respetivamente.

Na STARE, em média, cada máscara tem 304847 pixeis e foram introduzidos 3897 falsos positivos por imagem. Entre eles, cerca de 72.7 % localizaram-se na área imediatamente adjacente. À medida que a região próxima foi sendo alargada, obtiveram-se valores de 84.7 %, 92.3 % e 94.8 %.

Na CHASE_DB1, em média, cada máscara possui 657147 pixeis e detetaram-se 8056 falsos positivos por imagem. Entre eles, cerca de 59.8 % localizaram-se na área imediatamente adjacente, com percentagens de 80.1 %, 90.7 % e 92.9 %, à medida que a região próxima se tornou mais alargada.

Analisando os resultados, observa-se que nas imagens de maior resolução foram detetados mais falsos positivos na vizinhança distante. Isto parece natural visto que, em termos relativos, o mesmo número de dilatações, com o mesmo elemento estruturante, equivale a uma área menor em imagens maiores. Em termos globais, os resultados parecem indicar que grande parte das falsas deteções introduzidas se situaram em regiões próximas dos vasos, o que é positivo pois significa que o modelo tem pouca propensão para introduzir falsas deteções em localizações isoladas.

Tabela 6.15: Quantificação de falsos positivos (FP) na região próxima (RP), consoante o número de dilatações morfológicas (it.) utilizadas para definir a sua extensão, na DRIVE, STARE e CHASE_DB1

Base de dados	Extensão RP (it.)	Máscara (pixeis)	FP Total (pixeis)	FP RP (pixeis)	FP RP (%)
DRIVE	1	226907	3896	3413	87.6
	2			3666	94.1
	5			3737	95.9
	10			3805	97.7
STARE	1	304847	3897	2834	72.7
	2			3302	84.7
	5			3598	92.3
	10			3693	94.8
CHASE_DB1	1	657147	8056	4816	59.8
	2			6457	80.1
	5			7307	90.7
	10			7482	92.9

6.13 Robustez face à Variabilidade Inter-observador

As anotações manuais são indispensáveis nas etapas de treino e teste dos métodos supervisionados. A experiência pode ajudar os especialistas a refinar as suas anotações, mas dois observadores diferentes terão sempre formas distintas de lidar com as imagens. Nem sempre há acordo entre os especialistas, porque alguns deles têm mais tendência para marcar vasos que outros. Além disto, mesmo quando a presença do vaso é clara, diferenças na avaliação do calibre são frequentes. Por estas razões, a variabilidade inter-observador está sempre presente na avaliação de um método de segmentação automática. Nesta secção, discutem-se os efeitos que esta variabilidade tem no desempenho do método proposto.

Todos os resultados reportados anteriormente neste capítulo foram obtidos usando as anotações do 1º observador como referência. Na tabela 6.16, contudo, a referência são as anotações do 2º observador. Com esta nova referência, o desempenho do método é então comparado com o do primeiro observador. No que diz respeito aos testes de significância estatística, a hipótese nula avaliada supõe que não existem diferenças significativas entre o método proposto e o 1º observador.

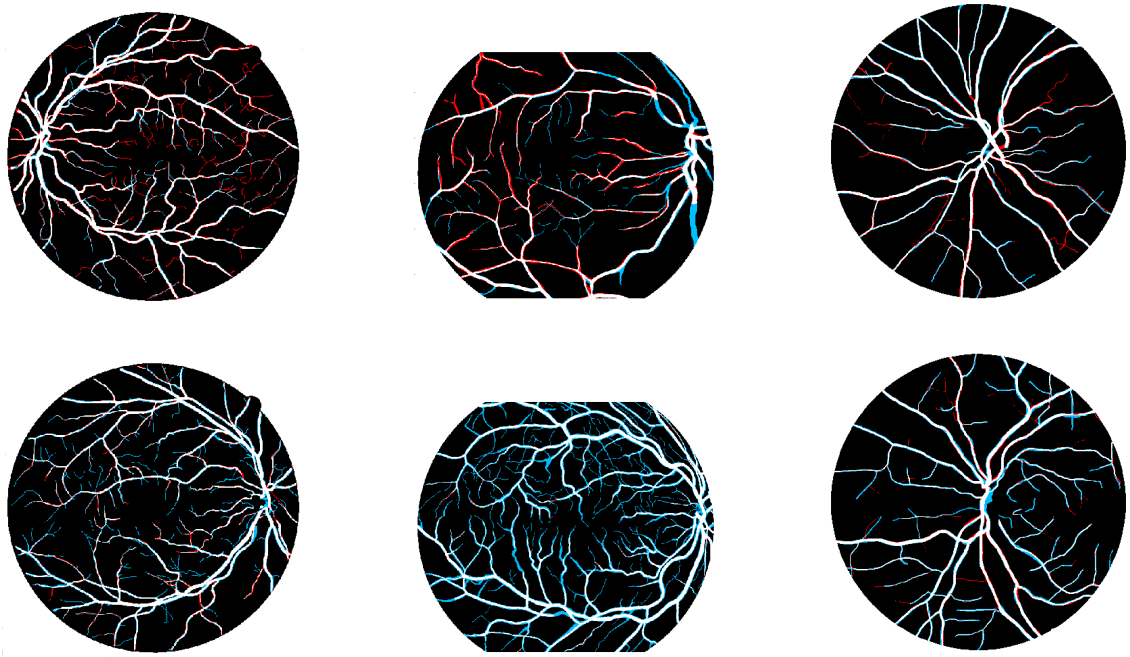
Analisando a tabela 6.16, percebe-se que o modelo superou o primeiro observador em termos de *Esp* e *Acc*, na STARE e CHASE_DB1. Isto significa que o modelo introduziu menos falsos positivos que o primeiro observador (em relação ao segundo). Por outras palavras, alguns pixeis marcados como *background* pelo modelo, e mal classificados segundo o primeiro observador, acabaram por ser validados pelo segundo. Na DRIVE, o modelo suplantou o primeiro observador em termos de *Sen*, *Esp* e *Acc*. Isto quer dizer que, além de apresentar menos falsos positivos, o modelo também obteve menos falsos negativos que o primeiro observador. Isto é, mesmo tendo sido treinado através das anotações do primeiro observador, o modelo foi capaz de marcar segmentos de vasos que só o segundo observador detetou. Conjuntamente, estes resultados parecem sugerir que mesmo quando o modelo foi treinado com base no primeiro observador, ele revelou-se mais consistente que o observador humano quando se consideraram as anotações de um segundo observador independente. Por outras palavras, as diferenças existentes entre as anotações de dois observadores humanos independentes revelaram-se maiores que as diferenças apresentadas pelo modelo quando se introduziu uma nova referência. Isto parece revelar uma consistência que pode não estar ao alcance de um ser humano, porque até um especialista pode oscilar de dia para dia e ser afetado por fatores como fadiga ou stress.

Tabela 6.16: Comparação entre os resultados do método proposto e do primeiro observador, tendo o segundo observador como referência. Valores a negrito representam o melhor resultado; valores sublinhados indicam diferenças estatisticamente significativas entre os dois métodos (p-*value* < 0.05)

Base de dados	Método	Sen	Esp	Acc
DRIVE	1º Observador	0.8066	0.9674	0.9473
	Proposto	**0.8405**	**0.9814**	**0.9639**
STARE	1º Observador	**0.6439**	0.9883	0.9346
	Proposto	0.6329	**0.9924**	**0.9365**
CHASE_DB1	1º Observador	**0.7974**	0.9736	0.9561
	Proposto	0.7731	**0.9813**	**0.9600**

Comparando as tabelas 6.11 e 6.16, podem ainda analisar-se os efeitos que a mudança de referência teve no desempenho do modelo. Na STARE e CHASE_DB1, os valores médios de *Acc* caíram de 0.9694 e 0.9653 para 0.9364 e 0.9600, respetivamente. Isto significa perdas aproximadas de 3.4% e 0.5%, que seriam expectáveis visto que o modelo foi treinado de acordo com as anotações do primeiro observador. No caso particular da STARE, a queda revelou-se bastante maior visto que, como já se referiu, existe uma grande diferença nas anotações dos dois observadores, com o segundo a marcar sistematicamente mais vasos que o primeiro. Já no caso da DRIVE, os resultados foram surpreendentes. Os valores médios de *Acc* subiram de 0.9576 para 0.9639, o que traduz um aumento aproximado de 0.7 %. Isto quer dizer que, apesar de ter sido treinado com base no primeiro observador, o modelo se aproximou mais do padrão de marcação do segundo.

Exceptuando o caso da STARE, nas restantes bases de dados não existe uma diferença clara no padrão de marcação dos especialistas. Isto é, há imagens em que o primeiro observador marca vasos que o segundo rejeita e vice-versa. Isto pode ser deduzido da figura 6.15, onde se comparam as anotações dos dois observadores. Este comportamento sugere que mesmo entre especialistas não é possível chegar a um consenso total numa tarefa de segmentação manual. Assim, e sabendo que os resultados desta área são cada vez mais elevados, é possível que a eficácia real dos métodos de segmentação automática esteja já a ser limitada pela potencial existência de imperfeições nas próprias anotações.



(a) (b) (c)

Figura 6.15: Comparação entre as anotações dos 1° e 2° observadores: (a) DRIVE; (b) STARE; (c) CHASE_DB1. A primeira linha ilustra as imagens onde o desempenho do modelo mais melhorou quando passou a ser avaliado face ao segundo observador; a segunda linha representa as imagens onde ele mais se ressentiu. Cada cor representa pixeis numa situação distinta: preto – ambos os observadores marcaram como *background*; branco – ambos marcaram como vaso; vermelho – só o 1° observador marcou como vaso; azul – só o 2° observador marcou como vaso.

6.14 Sumário

O método proposto nesta dissertação foi avaliado em três bases de dados públicas: DRIVE, STARE e CHASE_DB1.

Na primeira parte deste capítulo validaram-se os componentes do modelo base. Em termos gerais, comprovou-se que as estratégias de correção do contraste não trouxeram benefícios e que a estratégia de aumento de dados mais benéfica assentou na utilização de rotações dispostas de forma aleatória. Relativamente à arquitetura, utilizaram-se *patches* de 88×88 e 32×32 pixeis, na entrada e saída, respetivamente. A estratégia de regularização passou pelo uso do *dropout* espacial e, durante o processo de otimização, o tamanho do *mini-batch* foi fixado em 4. A estratégia de segmentação múltipla apresentada no capítulo anterior revelou-se positiva e o uso dos canais provenientes da SWT permitiu melhorar o desempenho do modelo base, obtendo-se o melhor modelo.

Em termos absolutos, o modelo proposto obteve *Acc* média de 0.9576, 0.9694 e 0.9653 e AUC média de 0.9821, 0.9905 e 0.9855, na DRIVE, STARE e CHASE_DB1, respetivamente. Face aos restantes métodos do estado da arte, o método obteve o melhor desempenho na DRIVE e CHASE_DB1, estando muito bem posicionado também na STARE. Além disto, o método revelou-se robusto face à mudança do conjunto de treino e à presença de sinais patológicos nas imagens, mantendo-se sempre competitivo face aos restantes trabalhos do estado da arte. Por fim, o método pareceu ter boa capacidade para detetar os vasos finos e evitar falsas deteções, sendo também robusto quando avaliado segundo anotações manuais de observadores diferentes.

Capítulo 7

Conclusões e Perspetivas Futuras

Neste capítulo final, sintetiza-se o trabalho desenvolvido ao longo desta dissertação e apresentam-se as conclusões mais relevantes. Antes de fechar, apontam-se também alguns aspetos que poderão ser alvo de trabalho futuro.

7.1 Conclusões

O principal objetivo desta dissertação passou pelo desenvolvimento de uma metodologia automática de segmentação de vasos retinianos, capaz de distinguir a vasculatura do *background*, em imagens de fundo. As características morfológicas da árvore vascular da retina fornecem informações de extremo relevo para o diagnóstico precoce e monitorização de diversas doenças como a diabetes, a hipertensão ou o glaucoma. Estas doenças têm um impacto enorme na qualidade de vida de milhões de pessoas a nível mundial, tendo sido esta a principal motivação para a realização deste trabalho.

Várias propostas de segmentação de vasos têm sido apresentadas nos últimos anos, com o objetivo último de substituir as segmentações manuais e contornar as desvantagens que lhes estão associadas, como a variabilidade intra- e inter-observador e os recursos humanos exigidos. Contudo, os métodos automáticos não estão, ainda, implementados em larga escala, pelo que existe um caminho a percorrer até que os resultados apresentados se revelem suficientemente robustos para que a aceitação global se torne uma realidade.

O trabalho realizado iniciou-se com a análise de dois métodos para a correção do contraste: o CLAHE e o método de Nyúl e Udupa. Estes métodos haviam sido aplicados com sucesso noutros trabalhos do estado da arte, mas neste estudo a sua inclusão não trouxe benefícios para a capacidade de generalização da rede. De facto, foi possível observar ganhos em algumas imagens, mas esta tendência foi sempre anulada pelo surgimento de perdas noutros casos. Isto resultou num efeito global negativo que, aliado ao custo computacional adicional, levou ao abandono destas estratégias.

Numa fase posterior, estudaram-se diferentes estratégias de aumento artificial de dados, com foco em transformações elásticas e rotações. O aumento artificial de dados é uma ferramenta de grande importância em métodos de *deep learning* e ganha particular relevo na área médica, onde a obtenção de anotações manuais é especialmente difícil. Neste contexto, os resultados obtidos revelaram-se muito

positivos, tendo sido possível obter ganhos consideráveis, quer através das amostras elásticas, quer por via das amostras rodadas. A utilização das rotações revelou-se particularmente eficaz, devido à capacidade dos filtros convolucionais para exibir respostas mais interessantes quando sujeitos a entradas dispostas de forma específica. Já as transformações elásticas parecem ter potencial para gerar imagens com características mais diversificadas, mas deformar não linearmente as imagens e as respetivas anotações, sem comprometer a correspondência entre elas, é uma tarefa de complexidade acrescida.

Durante a etapa de classificação, a escolha recaiu numa CNN. As CNNs, que são provavelmente o modelo que mais tem contribuído para a aceitação global dos métodos de *deep learning*, diferem das ANNs convencionais em dois aspetos principais. Em primeiro lugar, os neurónios de cada camada só interagem com uma região restrita da camada anterior e não com a sua totalidade. Em segundo lugar, todas as regiões da entrada são processadas pelos mesmos pesos. Estas características tornam as CNNs mais fáceis de treinar e, aliadas ao surgimento de unidades de computação paralela cada vez mais potentes, ajudam a explicar o impacto recente destes modelos.

No projeto de uma CNN, há uma série de hiperparâmetros que transitam das ANNs tradicionais, tanto em termos de regularização como de otimização. Estes aspetos foram estudados e devidamente validados ao longo deste trabalho. No entanto, adicionalmente, surgem outros focos de interesse. O maior reside no modo como as diversas camadas devem ser dispostas de forma a obter o melhor desempenho – ou, remetendo para o título desta dissertação, na otimização da arquitetura. Neste trabalho, utilizou-se uma FCN multi-escala que, além de beneficiar das vantagens computacionais associadas à segmentação por blocos, tem como principal característica a fusão de informação com diferentes níveis de abstração. Esta ideia surge porque apesar dos operadores de *pooling* serem altamente benéficos em problemas de reconhecimento de objetos, eles também podem fazer com que alguma informação de localização seja perdida. Assim, fundir a informação de alto nível, que revela *o quê*, com a informação de baixo nível, que revela *onde*, constitui uma solução com grande potencial.

Associada à popularidade crescente dos métodos de *deep learning* em geral, e das CNNs em particular, tem havido uma mudança de paradigma segundo a qual o foco de investigação está agora na otimização de arquiteturas e não na tradicional fase de *feature engineering*. Neste sentido, averiguaram-se também os efeitos de incluir novos canais, obtidos através da decomposição da SWT, na entrada da CNN. Os resultados obtidos parecem comprovar que a informação multi-resolução codificada por estes canais trouxe benefícios a todos os níveis, pelo que, acompanhando uma ideia que vem surgindo na comunidade científica, também este trabalho parece sugerir que o conhecimento de domínio permanece importante, mesmo entre os métodos de *deep learning*.

Por último, destaca-se ainda a estratégia de segmentação múltipla proposta que, conceptualmente, pode ser aplicada a qualquer tipo de imagem e que permitiu melhorar os resultados de forma estatisticamente significativa.

Em termos quantitativos, os resultados obtidos pelo método proposto nesta dissertação asseguraram uma segmentação de grande eficácia global e, acima de tudo, robusta perante fatores como a mudança do conjunto de treino ou a presença de lesões. Em resumo, quando comparada com outros métodos do estado da arte, a metodologia proposta obteve um desempenho bastante promissor e permitiu validar o enorme potencial dos métodos de *deep learning* na área de imagem médica.

7.2 Perspetivas Futuras

Nos tempos que correm, uma dissertação em *deep learning* significa desenvolver um trabalho numa área onde surgem novas abordagens semanalmente. Além disto, mesmo que os resultados do estado da arte sejam já elevadíssimos, o problema da segmentação de vasos não pode ser visto como fechado.

No que diz respeito a este trabalho, um dos fatores passível de desenvolvimento prende-se com o estudo aprofundado dos métodos de correção do contraste. Como se referiu, neste estudo estas estratégias não trouxeram benefícios. Porém, as diferenças revelaram-se muito ligeiras e é possível que uma otimização minuciosa dos parâmetros possa trazer melhorias. Paralelamente, poderá também ser interessante verificar se a aplicação destes métodos traz vantagens quando o modelo é confrontado com imagens diferentes das disponíveis durante o treino.

Outro ponto pertinente passa pela forma como a informação da SWT foi canalizada. Neste trabalho, os estudos realizados assentaram na concatenação de novos canais à entrada da CNN. Contudo, é possível que a rede possa beneficiar mais de determinado tipo de informação, se esta for fornecida em pontos específicos. Por outras palavras, poderá ser útil averiguar se existe vantagem em processar a informação codificada pelos diferentes níveis da SWT, em diferentes escalas da arquitetura. Outras famílias de wavelets, como as curvelets, poderão também revelar-se úteis neste tipo de problema.

Por último, a segmentação automática pode ser vista como um passo intermédio para a criação de sistemas de rastreio. Sistemas automáticos para extrair as características morfológicas da árvore vascular da retina podem abrir portas a um nova forma de diagnosticar e monitorizar várias doenças.

Bibliografia

[1] Michael D Abràmoff, Mona K Garvin, and Milan Sonka. Retinal imaging and image analysis. *IEEE reviews in biomedical engineering*, 3:169–208, 2010.

[2] TJ MacGillivray, E Trucco, JR Cameron, B Dhillon, JG Houston, and EJR Van Beek. Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions. *The British journal of radiology*, 87(1040):20130832, 2014.

[3] Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. Blood vessel segmentation methodologies in retinal images–a survey. *Computer methods and programs in biomedicine*, 108(1):407–433, 2012.

[4] Américo Oliveira, Sérgio Pereira, and Carlos A Silva. Augmenting data when training a cnn for retinal vessel segmentation: How to warp? In *Bioengineering (ENBENG), 2017 IEEE 5th Portuguese Meeting on*, pages 1–4. IEEE, 2017.

[5] Clyde W Oyster. *The human eye: structure and function*. Sinauer Associates, 1999.

[6] David H Hubel, Janice Wensveen, and Bruce Wick. *Eye, brain, and vision*. Scientific American Library New York, 1995.

[7] Charles Molnar and Jane Gair. *Concepts of Biology*. BC Campus, 2015.

[8] Steven M Lavalle. Virtual reality. 2015.

[9] Pearse A Keane and Srinivas R Sadda. Retinal imaging in the twenty-first century: state of the art and future directions. *Ophthalmology*, 121(12):2489–2500, 2014.

[10] Philipp Roberts, Sebastian M Waldstein, and Ursula Schmidt-Erfurth. A focus on the imaging of the retina. *Expert Review of Ophthalmology*, 10(6):595–611, 2015.

[11] Florian Alten, Peter Heiduschka, Christoph R Clemens, and Nicole Eter. Multifocal electroretinography in eyes with reticular pseudodrusenmferg in reticular pseudodrusen. *Investigative ophthalmology & visual science*, 53(10):6263–6270, 2012.

[12] Klara Landau and Malaika Kurz-Levin. Retinal disorders. *Handb Clin Neurol*, 102:97–116, 2011.

[13] Rama D Jager, William F Mieler, and Joan W Miller. Age-related macular degeneration. *New England Journal of Medicine*, 358(24):2606–2617, 2008.

[14] Robert N Weinreb, Tin Aung, and Felipe A Medeiros. The pathophysiology and treatment of glaucoma: a review. *Jama*, 311(18):1901–1911, 2014.

[15] Alan W Stitt, Timothy M Curtis, Mei Chen, Reinhold J Medina, Gareth J McKay, Alicia Jenkins, Thomas A Gardiner, Timothy J Lyons, Hans-Peter Hammes, Rafael Simó, et al. The progress in understanding and treatment of diabetic retinopathy. *Progress in retinal and eye research*, 51:156–186, 2016.

[16] Tien Wong and Paul Mitchell. The eye in hypertension. *The Lancet*, 369(9559):425–435, 2007.

[17] M Bhargava, MK Ikram, and TY Wong. How does hypertension affect your eyes? *Journal of human hypertension*, 26(2):71, 2012.

[18] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: a cancer journal for clinicians*, 66(1):7–30, 2016.

[19] John R Heckenlively and Henry A Ferreyra. Autoimmune retinopathy: a review and summary. In *Seminars in immunopathology*, volume 30, pages 127–134. Springer, 2008.

[20] Monica Gandhi and Suneeta Dubey. Evaluation of the optic nerve head in glaucoma. *Journal of current glaucoma practice*, 7(3):106, 2013.

[21] Tomi Kauppi et al. *Eye fundus image analysis for automatic detection of diabetic retinopathy*. Lappeenranta University of Technology, 2010.

[22] Jayant Venkatramani, Justin L Gottlieb, Thom S Thomassen, and Adamarie Multari. Bilateral serous retinal detachment due to protein-losing enteropathy. *Archives of Ophthalmology*, 122(7):1067–1070, 2004.

[23] Jacqueline MS Winterkorn. Transient monocular visual loss due to primary vasospasm: A case study. 2008.

[24] Dong Yoon Kim. Non-small-cell lung cancer choroidal meta os fundus, 2013.

[25] Chetan L Srinidhi, P Aparna, and Jeny Rajan. Recent advancements in retinal vessel segmentation. *Journal of medical systems*, 41(4):70, 2017.

[26] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[27] Marco Foracchia, Enrico Grisan, and Alfredo Ruggeri. Luminosity and contrast normalization in retinal images. *Medical Image Analysis*, 9(3):179–190, 2005.

[28] MM Fraz, Paolo Remagnino, Andreas Hoppe, Alicja R Rudnicka, Christopher G Owen, PH Whincup, and SA Barman. Quantification of blood vessel calibre in retinal images of multi-ethnic school children using a model based approach. *Computerized Medical Imaging and Graphics*, 37(1):48–60, 2013.

[29] Yu Qian Zhao, Xiao Hong Wang, Xiao Fang Wang, and Frank Y Shih. Retinal vessels segmentation based on level set and region growing. *Pattern Recognition*, 47(7):2437–2446, 2014.

[30] George Azzopardi, Nicola Strisciuglio, Mario Vento, and Nicolai Petkov. Trainable cosfire filters for vessel delineation with application to retinal images. *Medical image analysis*, 19(1):46–57, 2015.

[31] Benjun Yin, Huating Li, Bin Sheng, Xuhong Hou, Yan Chen, Wen Wu, Ping Li, Ruimin Shen, Yuqian Bao, and Weiping Jia. Vessel extraction from non-fluorescein fundus images using orientation-aware detector. *Medical image analysis*, 26(1):232–242, 2015.

[32] Yitian Zhao, Yonghuai Liu, Xiangqian Wu, Simon P Harding, and Yalin Zheng. Retinal vessel segmentation: An efficient graph cut approach with retinex and local phase. *PloS one*, 10(4):e0122332, 2015.

[33] Yitian Zhao, Lavdie Rada, Ke Chen, Simon P Harding, and Yalin Zheng. Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images. *IEEE transactions on medical imaging*, 34(9):1797–1807, 2015.

[34] Subhasis Chaudhuri, Shankar Chatterjee, Norman Katz, Mark Nelson, and Michael Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on medical imaging*, 8(3):263–269, 1989.

[35] AD Hoover, Valentina Kouznetsova, and Michael Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical imaging*, 19(3):203–210, 2000.

[36] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 130–137. Springer, 1998.

[37] Uyen TV Nguyen, Alauddin Bhuiyan, Laurence AF Park, and Kotagiri Ramamohanarao. An effective retinal blood vessel segmentation method using multi-scale line detection. *Pattern recognition*, 46(3):703–715, 2013.

[38] Luo Gang, Opas Chutatape, and Shankar M Krishnan. Detection and measurement of retinal vessels in fundus images using amplitude modified second-order gaussian filter. *IEEE transactions on Biomedical Engineering*, 49(2):168–172, 2002.

[39] Abdolhossein Fathi and Ahmad Reza Naghsh-Nilchi. Automatic wavelet-based retinal blood vessels segmentation and vessel diameter estimation. *Biomedical Signal Processing and Control*, 8(1):71–80, 2013.

[40] João VB Soares, Jorge JG Leandro, Roberto M Cesar, Herbert F Jelinek, and Michael J Cree. Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification. *IEEE Transactions on medical Imaging*, 25(9):1214–1222, 2006.

[41] George Azzopardi and Nicolai Petkov. Automatic detection of vascular bifurcations in segmented retinal images using trainable cosfire filters. *Pattern Recognition Letters*, 34(8):922–933, 2013.

[42] Ana Maria Mendonca and Aurelio Campilho. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE transactions on medical imaging*, 25(9):1200–1213, 2006.

[43] Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. An ensemble classification-based approach applied to retinal blood vessel segmentation. *IEEE Transactions on Biomedical Engineering*, 59(9):2538–2548, 2012.

[44] Yangfan Wang, Guangrong Ji, Ping Lin, and Emanuele Trucco. Retinal vessel segmentation using multiwavelet kernels and multiscale hierarchical decomposition. *Pattern Recognition*, 46(8):2117–2133, 2013.

[45] Eddie YK Ng, U Rajendra Acharya, Jasjit S Suri, and Aurelio Campilho. *Image Analysis and Modeling in Ophthalmology*. CRC Press, 2014.

[46] György Kovács and András Hajdu. A self-calibrating approach for the segmentation of retinal vessels by template matching and contour reconstruction. *Medical image analysis*, 29:24–46, 2016.

[47] Jiong Zhang, Behdad Dashtbozorg, Erik Bekkers, Josien PW Pluim, Remco Duits, and Bart M ter Haar Romeny. Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores. *IEEE transactions on medical imaging*, 35(12):2631–2644, 2016.

[48] Frederic Zana and J-C Klein. Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE transactions on image processing*, 10(7):1010–1019, 2001.

[49] Mohammad Saleh Miri and Ali Mahloojifar. Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction. *IEEE Transactions on Biomedical Engineering*, 58(5):1183–1192, 2011.

[50] Elaheh Imani, Malihe Javidi, and Hamid-Reza Pourreza. Improvement of retinal blood vessel detection using morphological component analysis. *Computer methods and programs in biomedicine*, 118(3):263–279, 2015.

[51] Iching Liu and Ying Sun. Recursive tracking of vascular networks in angiograms based on the detection-deletion scheme. *IEEE Transactions on Medical Imaging*, 12(2):334–341, 1993.

[52] Yi Yin, Mouloud Adel, and Salah Bourennane. Retinal vessel segmentation using a probabilistic tracking method. *Pattern Recognition*, 45(4):1235–1244, 2012.

[53] Yi Yin, Mouloud Adel, and Salah Bourennane. Automatic segmentation and measurement of vasculature in retinal fundus images using probabilistic formulation. *Computational and mathematical methods in medicine*, 2013, 2013.

[54] Erik Bekkers, Remco Duits, Tos Berendschot, and Bart ter Haar Romeny. A multi-orientation analysis approach to retinal vessel tracking. *Journal of Mathematical Imaging and Vision*, 49(3):583–610, 2014.

[55] Elisa Ricci and Renzo Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE transactions on medical imaging*, 26(10):1357–1365, 2007.

[56] Koen A Vermeer, Frans M Vos, Hans G Lemij, and Albert M Vossepoel. A model based method for retinal blood vessel detection. *Computers in Biology and Medicine*, 34(3):209–219, 2004.

[57] Benson Shu Yan Lam and Hong Yan. A novel vessel segmentation algorithm for pathological retina images based on the divergence of vector fields. *IEEE Transactions on Medical Imaging*, 27(2):237–246, 2008.

[58] Benson SY Lam, Yongsheng Gao, and Alan Wee-Chung Liew. General retinal vessel segmentation using regularization-based multiconcavity modeling. *IEEE Transactions on Medical Imaging*, 29(7):1369–1381, 2010.

[59] Bashir Al-Diri, Andrew Hunter, and David Steel. An active contour model for segmenting and measuring retinal vessels. *IEEE Transactions on Medical imaging*, 28(9):1488–1497, 2009.

[60] Zhiyong Xiao, Mouloud Adel, and Salah Bourennane. Bayesian method with spatial constraint for retinal vessel segmentation. *Computational and mathematical methods in medicine*, 2013, 2013.

[61] Meindert Niemeijer, Joes Staal, Bram van Ginneken, Marco Loog, Michael D Abramoff, et al. Comparative study of retinal vessel segmentation methods on a new publicly available database. In *SPIE medical imaging*, volume 5370, pages 648–656. SPIE, 2004.

[62] Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4):501–509, 2004.

[63] Lili Xu and Shuqian Luo. A novel method for blood vessel detection from retinal images. *Biomedical engineering online*, 9(1):14, 2010.

[64] Xinge You, Qinmu Peng, Yuan Yuan, Yiu-ming Cheung, and Jiajia Lei. Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. *Pattern Recognition*, 44(10):2314–2324, 2011.

[65] Diego Marín, Arturo Aquino, Manuel Emilio Gegúndez-Arias, and José Manuel Bravo. A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. *IEEE Transactions on medical imaging*, 30(1):146–158, 2011.

[66] Sohini Roychowdhury, Dara D Koozekanani, and Keshab K Parhi. Blood vessel segmentation of fundus images by major vessel extraction and subimage classification. *IEEE journal of biomedical and health informatics*, 19(3):1118–1128, 2015.

[67] Jiong Zhang, Yuan Chen, Erik Bekkers, Meili Wang, Behdad Dashtbozorg, and Bart M ter Haar Romeny. Retinal vessel delineation using a brain-inspired wavelet transform and random forest. *Pattern Recognition*, 69:107–123, 2017.

[68] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[69] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[70] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[71] Qiaoliang Li, Bowei Feng, LinPei Xie, Ping Liang, Huisheng Zhang, and Tianfu Wang. A cross-modality learning approach for vessel segmentation in retinal images. *IEEE transactions on medical imaging*, 35(1):109–118, 2016.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[73] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[74] Martina Melinščak, Pavle Prentašić, and Sven Lončarić. Retinal vessel segmentation using deep neural networks. In *VISAPP 2015 (10th International Conference on Computer Vision Theory and Applications)*, 2015.

[75] Paweł Liskowski and Krzysztof Krawiec. Segmenting retinal blood vessels with deep neural networks. *IEEE transactions on medical imaging*, 35(11):2369–2380, 2016.

[76] Huazhu Fu, Yanwu Xu, Stephen Lin, Damon Wing Kee Wong, and Jiang Liu. Deepvessel: Retinal vessel segmentation via deep learning and conditional random field. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 132–139. Springer, 2016.

[77] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[78] AL Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[79] Tom M Mitchell et al. Machine learning. wcb, 1997.

[80] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[81] Arden Dertat. Applied deep learning - part 1: Artificial neural networks. https://medium.com/towards-data-science/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6, 2017. [*Online* (acedido em Setembro de 2017)].

[82] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[83] Suzana Herculano-Houzel and Roberto Lent. Isotropic fractionator: a simple, rapid method for the quantification of total cell and neuron numbers in the brain. *Journal of Neuroscience*, 25(10):2518–2521, 2005.

[84] Anna Gummeson. Prostate cancer classification using convolutional neural networks. *Master's Theses in Mathematical Sciences*, 2016.

[85] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[86] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[87] Matej Uličn. Methods for increasing robustness of deep convolutional neural networks, 2015.

[88] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[89] YT Zhou and R Chellappa. Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks*, volume 1998, pages 71–78, 1988.

[90] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.

[91] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[92] Simon S Haykin. *Neural networks: a comprehensive foundation*. Tsinghua University Press, 2001.

[93] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[94] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[95] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[96] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[97] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[98] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[99] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[100] Gilbert Strang and Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996.

[101] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

[102] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990.

[103] Guido Van Rossum et al. Python programming language. In *USENIX Annual Technical Conference*, volume 41, page 36, 2007.

[104] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[105] Eric Jones, Travis Oliphant, and Pearu Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[106] Wes McKinney. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, pages 1–9, 2011.

[107] John D Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[108] Plotly Technologies Inc. Collaborative data science, 2015.

[109] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[110] F Wasilewski. The pywavelets python wavelet package, 2010.

[111] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[112] François Chollet et al. Keras, 2015.

[113] Wikipedia. Adaptive histogram equalization. https://en.wikipedia.org/wiki/Adaptive_histogram_equalization, 2017. [*Online* (acedido em Outubro de 2017)].

[114] László G Nyúl, Jayaram K Udupa, et al. On standardizing the mr image intensity scale. *image*, 1081, 1999.

[115] José Adriano Azevedo da Silva Pinto et al. *Segmentação do tecido tumoral cerebral em imagens de ressonância magnética*. PhD thesis, 2015.

[116] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5):1240–1251, 2016.

[117] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[118] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.

[119] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962. Citeseer, 2003.

[120] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[121] Leonardo Araújo Santos. Convolutional neural networks. https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/convolutional_neural_networks.html, 2017. [*Online* (acedido em Outubro de 2017)].

[122] Aleksandra Popovic, Matías De la Fuente, Martin Engelhardt, and Klaus Radermacher. Statistical validation metric for accuracy assessment in medical image segmentation. *International Journal of Computer Assisted Radiology and Surgery*, 2(3):169–181, 2007.

[123] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[124] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.