Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Ciência de Dados

APRENDIZAGEM AUTOMÁTICA COMO SUPORTE ÀS CIÊNCIAS DA TERRA

TIAGO FILIPE RODRIGUES RIBEIRO

Leiria, 30 de setembro de 2023

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Ciência de Dados

APRENDIZAGEM AUTOMÁTICA COMO SUPORTE ÀS CIÊNCIAS DA TERRA

TIAGO FILIPE RODRIGUES RIBEIRO
Número: 2210785

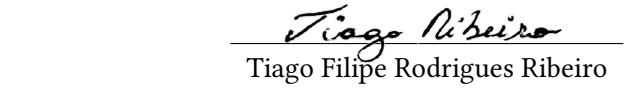
Dissertação realizada sob orientação do Professor Doutor Rogério Luís de Carvalho Costa (rogerio.l.costa@ipleiria.pt) e do Professor Doutor Fernando José Mateus da Silva (fernando.silva@ipleiria.pt).

Leiria, 30 de setembro de 2023

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"Aprendizagem Automática como Suporte às Ciências da Terra"*, é original e foi realizado por Tiago Filipe Rodrigues Ribeiro (número 2210785) sob orientação de Professor Doutor Rogério Luís de Carvalho Costa (rogerio.l.costa@ipleiria.pt) e do Professor Doutor Fernando José Mateus da Silva (fernando.silva@ipleiria.pt).

Leiria, 30 de setembro de 2023



Tiago Filipe Rodrigues Ribeiro

AGRADECIMENTOS

Esta dissertação foi parcialmente realizada no âmbito da bolsa de investigação "ESS Data Lab" (MIT-EXPL/ACC/0057/2021), financiada através da Fundação para a Ciência e a Tecnologia (FCT), no Centro de Investigação em Informática e Comunicações (CIIC) da Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria (IPL).

Começo por expressar os meus agradecimentos a todas as pessoas que contribuem para o bom funcionamento do CIIC, agradecendo pelos recursos alocados para esta dissertação.

Os meus sinceros agradecimentos à FCT pelo financiamento da bolsa de investigação, que me proporcionou a oportunidade de estudar um tema que me apaixona. Espero ter correspondido à responsabilidade.

Ao meu orientador, Professor Doutor Fernando José Mateus da Silva, desejo manifestar a minha gratidão pela sua acessibilidade, simpatia e encorajamento. Agradeço pelas ideias, acompanhamento, pelas contribuições nos artigos científicos, pelas sugestões e revisão desta dissertação.

Ao meu coorientador, Professor Doutor Rogério Luís de Carvalho Costa, a quem não basta este espaço destinado a agradecimentos, agradeço pela disponibilidade e trabalho árduo, pela simplicidade, energia positiva e paciência. Agradeço também pela troca de ideias pertinentes e pelas horas dedicadas, sem as quais não teria sido possível realizar os artigos e a documentação produzidos durante o projeto ESS Data Lab. A dinâmica de equipa imprimida no CIIC, de sua iniciativa, teve uma contribuição intangível na realização desta dissertação.

Ao Professor Doutor José Moreira, da Universidade de Aveiro, do Departamento de Eletrónica, Telecomunicações e Informática (DETI), agradeço pelas conversas produtivas e ideias, pela partilha do seu extenso conhecimento e pela oportunidade de colaboração.

Gostaria de expressar o meu agradecimento aos colegas de trabalho do CIIC, José Areias, Ivo Bispo, José Frade e Francisco Milícias, pela afabilidade e simpatia, contribuindo para o bom ambiente de trabalho.

À minha prima, Margarida Silva, agradeço pelo tempo dedicado e pela partilha de conhecimento acerca da hipsometria e caracterização da paisagem de Torre do Pinhão.

À minha irmã, Ana Ribeiro, agradeço pelo auxílio na organização estética e formal do póster publicado no âmbito deste trabalho.

RESUMO

Os incêndios florestais acarretam consequências de largo alcance, representando uma ameaça significativa para a vida humana, economia e o meio ambiente. A compreensão da dinâmica desses fogos florestais e dos seus impactos ambientais torna-se crucial, especialmente em regiões de elevada incidência.

Recentemente, modelos baseados em aprendizagem automática emergiram como promissoras ferramentas para facilitar o entendimento da complexa dinâmica dos incêndios florestais e de outros fenómenos naturais. Estas técnicas abrangem modelos visão computacional capazes de representar a geometria de objetos de interesse, e modelos capazes de simular a evolução de fenómenos espaçotemporais. No entanto, tipicamente carece-se de conjuntos de dados anotados de dimensões e qualidade significativas. No entanto, conjuntos de dados que capturam a evolução em tempo real de área ardida são escassos.

Esta dissertação propõe três contribuições principais: (**i**) um novo conjunto de dados de incêndios florestais para a segmentação semântica de áreas ardida; (**ii**) ferramentas para validação e teste de modelos de segmentação semântica automática de área ardida no contexto de incêndios florestais, (**iii**) um modelo Autocodificador para interpolação espaçotemporal capaz de representar fenómenos do mundo real, como a evolução de áreas ardida em incêndios florestais.

Descrevemos detalhadamente o processo de amostragem, anotação manual e validação de um novo conjunto de dados, proveniente de vídeos de fogo controlado capturados por *drone* no Norte de Portugal, o qual disponibilizamos num repositório de acesso livre. Adicionalmente, propomos métricas específicas para teste e validação de polígonos gerados por modelos automáticos de segmentação.

Com base no conjunto de dados BurnedAreaUAV, avaliamos modelos de segmentação automática utilizando a arquitetura totalmente convolucional U-Net, considerando métricas de similaridade geométrica e consistência temporal dos polígonos gerados.

Para a interpolação espaçotemporal dos polígonos de área ardida, propomos aplicar um modelo Autocodificador Variacional Condicional (C-VAE) e investigamos as suas capacidades para gerar representações contínuas da evolução espaçotemporal de regiões em movimento. Realizamos subamostragem das amostras do conjunto de dados e aplicamos o modelo C-VAE para gerar representações de regiões intermédias, comparando-o com outros algoritmos de interpolação da literatura. Avaliamos o desempenho desses

métodos comparando as suas interpolações com dados de referência do conjunto de dados BurnedAreaUAV e com regiões geradas por um modelo de segmentação automática de arquitetura U-Net. Aferimos a qualidade dos polígonos gerados considerando métricas de similaridade geométrica e de consistência temporal.

O conjunto de dados BurnedAreaUAV e as demais técnicas que propomos são ferramentas importantes que apoiam a avaliação comparativa de modelos de segmentação de área ardida em cenários de incêndios florestais. As técnicas baseadas em aprendizagem profunda que exploramos podem ser consideradas bases de referências

O conjunto de dados curado que criamos, denominado BurnedAreaUAV, preenche uma lacuna e constitui uma ferramenta válida para investigações futuras. O conjunto de dados BurnedAreaUAV e as demais técnicas que propomos são ferramentas importantes que apoiam a avaliação comparativa de modelos de segmentação de área ardida em cenários de incêndios florestais. As técnicas baseadas em aprendizagem profunda que exploramos podem ser consideradas bases de referências. No que respeita à abordagem baseada num C-VAE proposta para interpolação espaçotemporal, demonstramos que apresenta resultados competitivos em termos de métricas de similaridade geométrica, mas consistência temporal superior aos demais. As nossas experiências sugerem que os modelos C-VAE podem representar uma alternativa viável para modelar a evolução espaçotemporal de regiões móveis 2D.

O código, artigos, vídeos e documentação adicional relativos a esta dissertação podem ser consultados neste endereço: https://eesdatalab.ipleiria.pt/.

Palavras-chave: Aprendizagem profunda, C-VAE, Compressão, Incêndios florestais, Interpolação espaçotemporal, Segmentação de área ardida, U-Net

ABSTRACT

Forest fires have far-reaching consequences, posing a significant threat to human life, the economy, and the environment. Understanding the dynamics of wildfires and their environmental impacts is crucial, especially in regions with a high incidence.

Recently, machine learning-based models have emerged as promising tools to aid understanding of the complex dynamics of forest fires and other real-world phenomena.

These techniques include computer vision models capable of representing the geometry of objects of interest, and also models capable of simulating the evolution of spatiotemporal phenomena. However, they typically lack annotated datasets of significant size and quality. However, datasets capturing the real-time evolution of burnt area are scarce.

This dissertation proposes three main contributions: (**ii**) a new forest fire dataset for the semantic segmentation of burned area. (**ii**) tools for validating and testing automatic semantic segmentation models of burnt area in the context of forest fires. (**iii**) an Autoencoder model for spatiotemporal interpolation able to represent real-world phenomena, such as the evolution of burnt areas in forest fires.

We describe in detail the process of capturing, sampling, manually annotating and validating a new dataset from prescribed fire videos captured by drone in northern Portugal, which we have made available in an open-access repository. In addition, we propose specific metrics for testing and validating polygons generated by automatic segmentation models.

Drawing on the BurnedAreaUAV dataset, we evaluated automatic segmentation models using the fully convolutional U-Net architecture, considering geometric similarity and temporal consistency metrics of generated polygons.

We propose applying a Conditional Variational Autoencoder (C-VAE) model to perform the spatiotemporal interpolation of burnt area polygons and investigate its capabilities to generate continuous representations of the spatiotemporal evolution of moving regions.

We propose applying a Conditional Variational Autoencoder (C-VAE) model to perform the spatiotemporal interpolation of burnt area polygons and investigate its capabilities to generate continuous representations of the spatiotemporal evolution of moving regions. We subsampled the dataset and applied the C-VAE model to generate representations of intermediate regions, comparing it with other interpolation algorithms in the literature.

We evaluate the performance of these methods by comparing their interpolations with reference data from the BurnedAreaUAV dataset and with regions generated by an automatic segmentation model with U-Net architecture. The quality of the generated data considering geometric similarity and temporal consistency metrics.

This dataset fills a gap, providing a valuable tool for future research. We created tools that support the comparative evaluation of burned area segmentation models in forest fire scenarios, and explored deep learning-based segmentation techniques of which we established reference bases, with a Jaccard index value (IoU) greater than 95% in the BurnedAreaUAV test set. With regard to the C-VAE-based proposed method for spatiotemporal interpolation, we have shown that it presents competitive results in terms of geometric similarity metrics, yet superior temporal consistency to the others. Our experiments suggest that C-VAE models may represent a viable alternative for modeling the spatiotemporal evolution of 2D moving regions.

The code, articles, videos and additional documentation related to this dissertation can be found at the following link: https://eesdatalab.ipleiria.pt/.

Keywords: Burned area segmentation, C-VAE, Compression, Deep Learning, Forest fires, Spatiotemporal interpolation, U-Net

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÓNIMOS

AE	Autocodificador. 6, 9, 36–38, 41, 54, 99, 105
ANN	Rede Neuronal Artificial. 6, 7, 9, 11, 15, 17–20, 22, 27, 125–129
C-VAE	Autocodificador Variacional Condicional. 6, 7, 9, 38, 87, 88, 92, 93, 95, 96, 98, 99, 102, 103, 105
CCE	Entropia Cruzada Categórica. 19
CNN	Rede Neuronal Convolucional. 9, 24, 25, 27, 29, 31–33, 35, 41, 47, 128
CRF	Campos Aleatórios Condicionais. 46, 50
ELBO	Limite Inferior de Evidência. 37, 38
ELU	Unidade Linear Exponencial. 14
FCN	Rede Totalmente Convolucional. 35, 44, 46
FESB	Faculdade de Eng. Elétrica, Eng. Mecânica e Arquitetura Naval de Split. 56
FPN	Rede de Pirâmide de Características. 47, 48
GIGO	Lixo Entra, Lixo Sai. 101
GPU	Unidade de Processamento Gráfico. 64, 93, 129
HRNet	Rede de Alta Resolução. 46
IoU	Interseção sobre União. 67, 68, 72, 78, 80, 82, 102
ISBI	Simpósio Internacional sobre Imagens Biomédicas. 45
JSON	JavaScript Object Notation. 70
KML	Keyhole Markup Language. 64, 70

LeakyReLU	Unidade Linear Retificada com Vazamento. 14
LSTM	Memória Longa a Curto Prazo. 128
MSE	Erro Quadrático Médio. 18
PNG	Portable Network Graphics. 70
ReLU	Unidade Linear Retificada. 14
RGB	Vermelho, Verde e Azul. 29, 55–57, 61, 64, 72, 78
RIP	Problema da Interpolação de Regiões. 51
UNETR	Transformador U-Net. 48, 49
VAE	Autocodificador Variacional. 6, 9, 37, 38, 54, 88, 98, 99
WKT	Well-known text. 70, 93

1 INTRODUÇÃO

Alguns estudos apontam para a forte relação entre a incidência de incêndios florestais e as características climáticas e meteorológicas dos países europeus (Hoinka et al., 2009; Moreno et al., 2014). Porém, outras publicações relatem que as causas das ignições de incêndios florestais são principalmente de origem antropogénica (Gestão do Programa de Fogos Rurais - DAGFR, 2022; Parente et al., 2018). Independentemente das causas subjacentes, os incêndios florestais têm impactos do ponto de vista da perda de vidas humanas (Commission et al., 2022; Molina-Terrén et al., 2019), repercussões económicas e sociais (Paveglio et al., 2018), perturbações nos ecossistemas locais (Certini, 2005; Mansilha et al., 2019), além de contribuírem para a redução da biomassa e libertação de dióxido de carbono para a atmosfera (C3S, 2023). Compreender a dinâmica dos incêndios florestais e estabelecer os seus impactos torna-se indispensável, particularmente em territórios com uma incidência mais elevada.

A segmentação da região correspondente à área queimada e a definição de polígonos de segmentação para criar representações da evolução incêndios florestais ao longo do tempo é uns dos desafios que no qual estamos interessados. Estas representações permitem-nos apoiar estudos sobre emissões atmosféricas (Pessôa et al., 2020), assim como modelar o comportamento espaçotemporal dos incêndios florestais e de outros fenómenos naturais (Moreira, José Duarte et al., 2019), bem como realizar estudos de interpolação de regiões móveis deformáveis (R. L. C. Costa, Miranda, Dias et al., 2020a; J. Duarte et al., 2020).

Contudo, a segmentação da área queimada ao longo de um incêndio florestal apresenta desafios específicos. Ao contrário de outros objetos (por exemplo, um carro, uma pessoa ou um animal específico), a região que o fogo consome não tem partes identificáveis, conformação típica ou limites bem definidos. Pela natureza deste fenómeno, a área queimada é amorfa e muitas vezes parcialmente ocultada pelo fumo e pelo fogo produzidos (Figura 1.1). Além disso, os tipos e densidade de vegetação são frequentemente heterogéneos, e outros objetos não relacionados, como estradas, rochas, cercas ou lagos, podem estar na imagem, introduzem complexidade adicional à tarefa de segmentação, que exige algoritmos robustos capazes, capazes de compreender uma ampla gama de cenários. A captura de imagens com sensores diferentes em condições e posicionamentos variados, em diferentes

momentos do dia, em regiões e topografias diversas e com as restrições que envolvem a captura em tempo real de incêndios, como altas temperaturas, toxicidade do ar, por exemplo, adicionam-se à lista de desafios enumerados.

Porém, nos últimos anos, os avanços na área da aprendizagem profunda deram origem a uma nova geração de modelos de segmentação de imagens com melhorias significativas no desempenho, levando a uma mudança de paradigma (Minaee et al., 2021; Zhou et al., 2021). Modelos baseados em redes convolucionais profundas e, mais recentemente, na arquitetura Transformador (Vaswani et al., 2017), têm sido aplicados a vários problemas de segmentação. Tipicamente, os modelos de aprendizagem profunda são caracterizados pelo elevado número de parâmetros ajustáveis e pela capacidade de capturar padrões complexos. No entanto, para efetivamente capturar a variabilidade inerente aos fenómenos do mundo real, generalizar para exemplos não observados e evitar o problema do sobreajuste, estes modelos carecem conjuntos de dados abrangentes e anotações de alta qualidade.

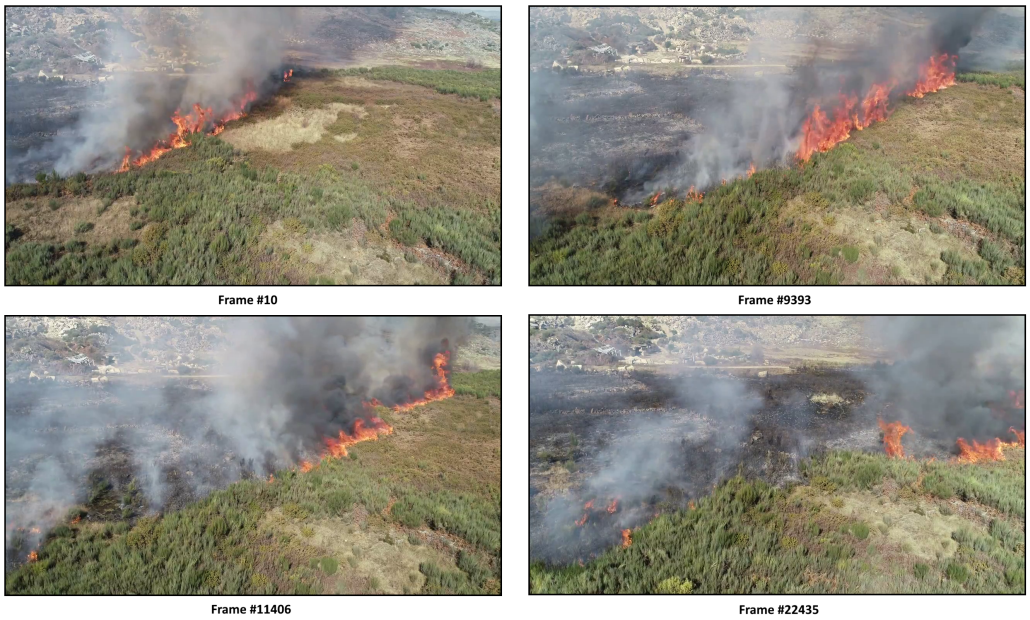


Figura 1.1: **Fotogramas problemáticos no vídeo capturado por *drone*** em Torre de Pinhão, Portugal. Em cada um dos fotogramas, observam-se áreas onde o fumo e as chamas ocluem os limites correspondentes à área queimada. Retirado de T. F. Ribeiro et al. (2023).

Obter dados de alta qualidade em quantidade suficiente para treinar modelos de aprendizagem profundo é um desafio enfrentado pela maioria das equipas que se dedicam a esta área de pesquisa. Em particular, os conjuntos de dados desempenham um papel crítico no treino e na avaliação de modelos de aprendizado profundo em aplicações de segmentação de imagens. No problema de segmentação semântica, um conjunto de dados consiste tipicamente numa coleção de imagens e das suas respetivas máscaras de segmentação. Ter um grande conjunto de dados permite ao modelo aprender as diversas características e

padrões presentes em imagens do mundo real e ajuda a garantir que o modelo generalize bem para dados novos. Além disso, a qualidade das anotações desempenha um papel crítico: anotações ruidosas ou incompletas podem degradar o desempenho do modelo (Yu et al., 2020).

Assumindo que a captura fenómeno espaçotemporal mediante técnicas de segmentação é feita de forma suficientemente precisa, dependendo da granularidade que se requeira, podem gerar-se problemas de desempenho e armazenamento dos dados (R. L. C. Costa, Miranda, Dias et al., 2020a). Paralelamente, sabemos que esses dados espaçotemporais são tipicamente armazenados usando amostras discretas, associando um *timestamp* denotando a data e hora a alguma representação da forma e posição da entidade. No entanto, algumas aplicações beneficiam de uma representação contínua da evolução espaçotemporal das entidades modeladas (Hamdi et al., 2022).

Esta representação contínua de dados espaçotemporais frequentemente depende de tipos de dados abstratos, como regiões móveis (*moving regions*), segmentos de reta móveis (*moving lines*) ou pontos móveis (*moving points*), e associa representações discretas das entidades modeladas a funções que representam a sua evolução (Mckennney e Frye, 2015; Tøssebro e R. H. Güting, 2001). Este paradigma tem algumas vantagens sobre o modelo discreto, como capacidade de compressão, mas também apresenta os seus próprios desafios, uma vez que a criação da representação contínua de uma entidade requer a especificação de um método para gerar a forma e a posição da entidade entre as representações armazenadas.

Por exemplo, na Figura 1.2, os primeiros e últimas amostras são imagens de um fenómeno do mundo real (área queimada de um incêndio florestal), e os polígonos intermediários recriam a evolução espaçotemporal da área queimada. Frequentemente, essa reconstituição emprega funções de interpolação de regiões, contudo, os métodos contemporâneos muitas vezes mostram-se incapazes de criar representações realistas da evolução de regiões 2D (José Duarte, B. Silva et al., 2019).

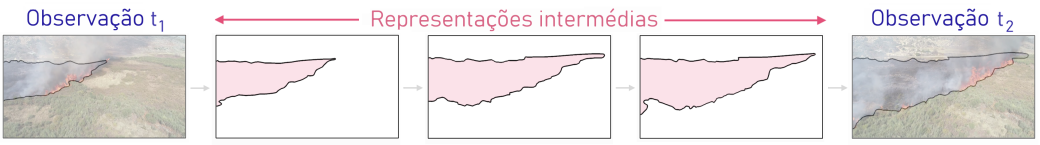


Figura 1.2: O modelo de representação contínua requer um método para recriar a evolução espaçotemporal de uma região, tal como a progressão da área ardida. Retirado de T. F. R. Ribeiro, F. Silva e C. Costa (2023)

Por outro lado, trabalhos recentes no campo da aprendizagem profunda têm demonstrado a capacidade desses modelos em aprender a interpolar fenómenos espaçotemporais contínuos a partir de características implícitas extraídas de amostras esparsas (Mi et al., 2021; Oring et al., 2021), o que levanta a possibilidade de poder expandir estes métodos

ao problema da evolução da área afetada por incêndios florestais, bem como a outras aplicações relativas a bases de dados espaçotemporais.

1.1 OBJETIVOS DA DISSERTAÇÃO

O âmbito desta dissertação, concentra-se no estudo de modelos de dados espaçotemporais e na aplicação de algoritmos de aprendizagem automática às ciências da Terra. Em particular, focámo-nos em conjuntos de dados relativos à evolução e dinâmica de fogos florestais. Especificamente, os objetivos delineados compreendem:

I verificação da qualidade dos conjuntos de representações, bem como a sua validação e melhoria, e geração de conjuntos de dados curados;

II desenvolvimento de métodos de segmentação de objetos de interesse em imagens e representação da sua geometria e localização em formato padronizado;

III simulação da evolução espaçotemporal dos objetos de interesse;

IV utilização de técnicas de compressão e simplificação das representações dos objetos de interesse obtidas.

Fazendo a reconstrução cronológica do trabalho realizado para completar os objetivos definidos, a primeira fase consistiu num levantamento e estudo de algoritmos de segmentação semântica clássicos, bem como na exploração dos mais recentes desenvolvimentos e aplicações de algoritmos de segmentação semântica baseados em ou aprendizagem automática. Esta pesquisa inicial, para além de ter sido essencial para providenciar a indispensável bagagem teórica, possibilitou a identificação de possíveis abordagens para o problema da segmentação da área ardida em fogos florestais, no qual nos focamos inicialmente. Para além dos aspetos teóricos e identificação das característica e valências dos algoritmos, esta primeira fase, serviu ainda para estudar as *frameworks*, bibliotecas e requisitos computacionais necessários para a implementação de modelos de aprendizagem automática e segmentação semântica de imagens, bem como métodos de desenho, implementação e avaliação do desempenho dos diversos modelos de segmentação.

Como caso de estudo, considerou-se um vídeo gravado por um *drone* em Torre de Pinhão, no norte de Portugal (2019), durante uma campanha de fogos controlados. Este vídeo captura a evolução da área ardida de um foco de incêndio, sendo composto por um total de 22.500 fotogramas. A filmagem é caracterizada por períodos em que o fumo e as chamas obstruem a área de interesse, o que torna a segmentação da área ardida mais difícil.

Uma vez estabelecidos os conhecimentos de base, os modelos de interesse e reconhecidos os desafios do problema, indo ao encontro do objetivo estabelecido no ponto I, iniciou-se o processo de criação de um conjunto de dados que para representar a evolução da área ardida e servir de base para o treino dos modelos de segmentação automática. Após considerar várias opções, decidiu-se anotar manualmente um número limitado de fotogramas do vídeo recorrendo uma ferramenta de anotação de imagens. Para garantir a coerência e correção do processo de anotação, foram estabelecidas regras e princípios considerando as dificuldades inerentes à anotação de imagens com oclusões, bem como as propriedades da evolução da área ardida. Os polígonos resultantes da anotação foram então sujeitos a uma validação iterativa tendo em conta critérios de consistência geométrica da área ardida ao longo da duração do vídeo. No total, resultaram 249 polígonos validados que representam a evolução da área ardida.

Deste processo resultou o conjunto de dados denominado BurnedAreaUAV, o qual foi publicado na plataforma Zenodo onde pode ser consultado e descarregado. O conjunto de dados compreende o vídeo original, os fotogramas e respetivas máscaras de segmentação em formato imagem *raster*, assim como em formato textual padronizado. A este conjunto de ficheiros, adicionou-se ainda ortofotografias de alta resolução da área do estudo, assim como os metadados referentes às especificações técnicas dos sensores e ao posicionamento do *drone* utilizado para captura o vídeo.

Foi então com base neste conjunto de dados curado, uma representação fidedigna do fenómeno que se pretende capturar, que se seguiu para a fase de implementação e avaliação experimental de modelos de segmentação semântica automática, como indicado no ponto II dos objetivos.

A arquitetura base que escolhemos foi um modelo totalmente convolucional codificador-descodificador denominado U-Net (Ronneberger et al., 2015). Partindo da arquitetura do modelo U-Net original, exploraram-se variantes com intuito de encontrar uma solução capaz de capturar as dependências espaçotemporais inerentes as sequências cronológicas de imagens. Assim, para além de modelos U-Net compostos por camadas convolucionais bidimensionais, experimentaram-se variantes com camadas convolucionais tridimensionais.

Estes modelos foram avaliados com uma abordagem de aprendizagem supervisionada para aferir o seu desempenho e para servir de referência para trabalhos futuros. Simultaneamente, sugerimos uma métrica de consistência temporal simples, mas específica para validar polígonos de área queimada não anotados gerados pelos modelos de segmentação nos fotogramas consecutivos de vídeos, e avaliamos cada um dos modelos utilizando esta métrica de consistência temporal em dados não anotados.

Terminado o que se estabelecido no objetivo II, iniciou-se o estudo de métodos de interpolação e compressão de representações de objetos de interesse. Tal como nos estudo dos métodos de segmentação automática, a fase inicial consistiu numa pesquisa de métodos clássicos da literatura utilizados para interpolação polígonos que descrevam a evolução de regiões de interesse.

Para além de nos propormos a avaliar o desempenho de modelos clássicos de bases de dados espaçotemporais (inteporlação Mckenney) (McKenney et al., 2016) e interpolação baseada na forma (Schenk et al., 2000), aproveitando os conhecimentos entretanto adquiridos para o desenvolvimento de modelos de segmentação de aprendizagem profunda, implementou-se um modelo baseados num Autocodificador Variacional Condicional (C-VAE) (Sohn et al., 2015), capaz de aprender representações espaçotemporais e simular a evolução de fenómenos naturais, tal como a evolução da área ardida de um foco de incêndio. A implementação do modelo de aprendizagem profunda, o C-VAE, foi feita de forma iterativa e empírica, até se encontrar uma solução com desempenho competitivo face às alternativas da literatura. Seguidamente, fez-se o teste e comparação dos algoritmos de interpolação de regiões referidos e, com o intuito de ir ao encontro do objetivo IV, testaram-se métodos de compressão da representação baseados na distância geométrica de polígonos para conjunto de dados.

1.2 ESTRUTURA DA DISSERTAÇÃO

O Capítulo 2, procura introduzir o leitor à Rede Neuronal Artificial, *Artificial Neural Network* em inglês (ANN), e a alguns dos conceitos fundamentais que servem de suporte para a restante dissertação. Iniciamos com uma breve introdução ao neurónio artificial e às redes neuronais do tipo *feedforward*. De seguida, abordamos as funções de custo, o algoritmo de retropropagação e o processo de treino das redes neurais. Continuamos com as redes neuronais do tipo convolucional e as suas particularidades. E por fim, introduzimos o modelo Autocodificador (AE) (G. E. Hinton e Zemel, 1993), com um foco especial no Autocodificador Variacional (VAE) (Kingma e Welling, 2014) e na sua variante condicionada, o C-VAE.

O Capítulo 3 é dedicado à análise da revisão de literatura que sustenta os dois principais temas abordados nesta dissertação: a Segmentação de Imagens, que desenvolvemos no Capítulo 4, e a Interpolação de Dados Espaçotemporais, que abordamos no Capítulo 5. Na Secção 3.1, dedicada à segmentação de imagens, começamos por definir o conceito de segmentação e exploramos diversas tarefas dentro deste domínio. De seguida, examinamos algoritmos de segmentação semântica da literatura, abrangendo as abordagens

clássicas e as técnicas mais recentes baseadas em aprendizagem profunda. Na Secção 3.2, centrada na interpolação de dados espaçotemporais, descrevemos os conceitos de bases de dados espaçotemporais, introduzimos a noção de regiões móveis, e continuamos com apresentação de abordagens distintas para a interpolação de amostras discretas de regiões bidimensionais. No final deste capítulo, apresentamos o levantamento de conjuntos de dados para segmentação e classificação de imagens e vídeos de incêndios florestais disponíveis publicamente, relevantes para o problemas em estudo.

No Capítulo 4, abordamos a segmentação semântica em incêndios florestais e apresentamos os métodos e modelos utilizados para enfrentar este desafio. Em particular, primeiro, introduzimos o conjunto de dados BurnedAreaUAV, criado especificamente para treinar e avaliar modelos de segmentação de vídeos de incêndios florestais. Em segundo lugar, avaliamos modelos de segmentação de imagens baseados em aprendizagem profundo e fornecemos um ponto de referência para futuras pesquisas. Terceiro, apresentamos uma métrica de consistência temporal para validar polígonos de área queimadas gerados pelos modelos nos fotogramas consecutivos do vídeo e a usamos para avaliar o desempenho dos modelos. Adicionalmente, apresentamos as métricas de avaliação, os modelos testados, resultados obtidos e os desafios futuros na segmentação semântica da área queimada em incêndios florestais.

No Capítulo 5, começamos por apresentar uma visão geral da interpolação de dados espaçotemporais, destacando a importância de representar a evolução espaçotemporal de entidades do mundo real a partir de amostras discretas. De seguida, exploramos a interpolação baseada num C-VAE, explicando como este modelo pode ser usado para interpolar diferentes representações codificadas de forma discreta ou contínua. Discutimos, diferentes estratégias de compressão de dados espaçotemporais, incluindo a amostragem periódica e a amostragem baseada na distância e apresentamos as métricas utilizadas para avaliar o desempenho dos algoritmos de interpolação testados. De seguida, descrevemos os detalhes da experiência realizada e, por fim, apresentamos os resultados da avaliação, no que concerne a métricas de similaridade e consistência temporal.

No Capítulo 6, procedemos à sumarização dos resultados relevantes, identificamos as limitações do nosso estudo, enumeramos as contribuições que este trabalho oferece à comunidade cientifica e delineamos os desafios potenciais que podem se manifestar em futuros desenvolvimentos.

Nos Anexos, dedicamos a Secção A a uma breve revisão histórica da Rede Neuronal Artificial (ANN), enquanto na Secção B disponibilizamos o póster resultante do trabalho desenvolvido nesta dissertação, o qual foi apresentado em conferência.

2 CONCEITOS

Neste capítulo procuramos introduzir o leitor a alguns dos conceitos essenciais para compreensão dos capítulos subsequentes desta dissertação.

Na Secção 2.1, começamos por explicar o funcionamento do neurónio artificial. De seguida, exploramos redes neuronais do tipo *feedforward*, abordando as funções de custo, o algoritmo de retropropagação do erro, assim como processo de treino tipico das ANN.

Depois, na Secção 2.2, mudamos o foco para a Rede Neuronal Convolucional (CNN), detalhamos as camadas convolucionais, camadas de *pooling*, as camadas de convolução transposta, bem como os modelos de segmentação semântica totalmente convolucionais.

Por fim, na Secção 2.3, introduzimos os Autocodificador (AE), e exploramos as variantes variacionais (VAE) e variacionais condicionais (C-VAE).

2.1 REDES NEURONAIS ARTIFICIAIS

As ANN são uma subcategoria de modelos computacionais da área da inteligência artificial, inspiradas pelas redes de neurónios presentes no cérebro dos animais. Estes modelos são atualmente empregados numa diversidade de tarefas, com desempenho progressivamente melhor. Tais tarefas incluem, mas não se limitam a reconhecimento de padrões em imagens, processamento de linguagem natural, análise de séries temporais, classificação e regressão em problemas complexos (Abiodun et al., 2018).

Estes modelos têm um história rica e extensa, feita de avanço incrementais e de trocas de conhecimento entre várias disciplinas. Ainda que seja objeto central deste trabalho, remetemos o leitor para o Anexo A, onde é feita uma breve resenha histórica das ANN.

2.1.1 *Neurónio Artificial*

Embora o neurónio biológico seja consideravelmente mais complexo do que o modelo simplificado proposto por Frank Rosenblatt (1958) no qual se baseia o neurónio artificial moderno, ambos processam a informação de sinais de entrada, interpretam estes estímulos

e geram sinais de saída. Enquanto nos neurónios biológicos, os sinais de entrada são captados através das dendrites, submetidos a processamento no corpo celular e as respostas transmitidas via axónio, nos neurónios artificiais, os sinais de entrada são recolhidos através dos nós de entrada, sujeitos a operações matemáticas e os resultados são transmitidos por meio dos nós de saída (C. Silva e B. Ribeiro, 2018).

Neste sentido, em termos simples, podemos conceber o neurónio artificial como uma unidade de processamento que recebe múltiplas entradas e produz um único sinal de saída. À sua entrada pode estar um conjunto de sinais vindos de sensores, valores relativos à intensidade dos pixeis de uma imagem, ou dados estatísticos históricos referentes ao um determinado fenómeno, por exemplo. Estas entradas $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, representadas a verde na Figura 2.1, são multiplicadas por parâmetros ajustáveis a que estão associadas denominados *pesos* $\boldsymbol{w} = (w_1, w_2, ..., w_n)$. De seguida, o neurónio realiza a soma de todos estes produtos (Σ) e adiciona ao resultante o valor do termo de polarização[1] (ou viés) b desse neurónio. Depois, ao valor obtido é aplicada uma função de ativação ϕ que determina saída do neurónio y, como esquematizado na Figura 2.1.

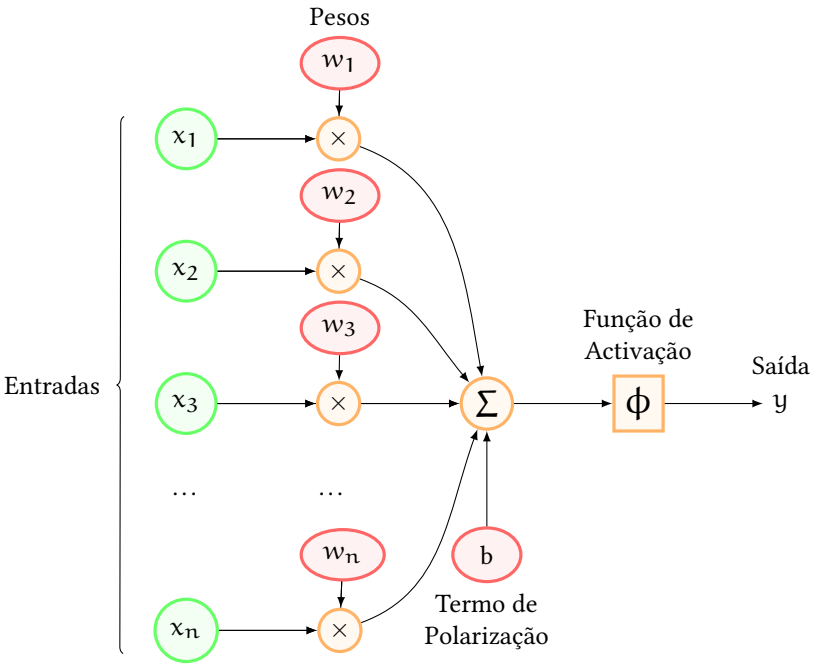


Figura 2.1: **Diagrama do neurónio artificial**. A verde, estão representadas as entradas $\boldsymbol{x} = (x_1, x_2, ..., x_n)$. A vermelho, os parâmetros ajustáveis ao longo do processo de otimização, nas quais de incluem os pesos $\boldsymbol{w} = (w_1, w_2, ..., w_n)$ e o termo de polarização b. A amarelo as operações matemáticas do neurónio artificial ($\times$, Σ e ϕ).

1 termo de polarização é um parâmetro ajustável que é adicionado à combinação linear das entradas antes de aplicar a função de ativação. Permite ajustar o ponto de ativação do neurónio, o que afeta a sensibilidade do neurónio a diferentes valores de entrada.

Mediante o ajuste dos pesos e do termo de polarização, estas unidades de processamento adquirem capacidades adaptativas ou de *aprendizagem*. Este processo pode ser conduzido pelo uso direto do método do gradiente ou outro algoritmo de otimização (Secção 2.1.4).

Em termos formais, um neurónio artificial é uma função matemática $f(\boldsymbol{x})$ cujo domínio é um vetor $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ que composto por n elementos, que é combinado linearmente com um vetor de n parâmetros ou pesos $\boldsymbol{w} = (w_1, w_2, ..., w_n)$ e respetivo termo de polarização b e, subsequentemente, transformado por uma função de ativação, ϕ na seguinte forma (Bishop e Nasrabadi, 2006):

$$y = f(x) = \phi\left(\sum_{i=1}^{n} \boldsymbol{w} \cdot \boldsymbol{x} + b\right) \tag{1}$$

Na prática, um neurónio artificial pode ser empregue tanto para tarefas de classificação binária, bem como em problemas de regressão linear, onde a sua saída representa um valor contínuo em vez de um valor de probabilidade de pertença a classe. Contudo, a capacidade de representação de um único neurónio artificial é limitada, sendo capaz apenas de *aprender* fronteiras de decisão lineares, e incapaz de capturar padrões ou interações complexas entre os dados de entrada (Minsky e Papert, 2017).

Estas limitações são ultrapassadas mediante a utilização de redes com múltiplas camadas de neurónios, que apresentaremos na secção seguinte.

2.1.2 *Redes Neuronais do tipo Feedforward*

A génese do campo das redes neuronais complexas remonta à arquitetura apresentada por F. Rosenblatt (1962), que estabeleceu as bases das ANN modernas e definiu os princípios de uma rede completamente conectada e de alimentação direta. Esse arranjo simples pode ser interpretado como um grafo direcionado acíclico, no qual a transferência de informações ocorre da esquerda para a direita, isto é, do sentido da entrada para a saída, conforme ilustrado na Figura 2.2.

As Redes Neuronais de Alimentação Direta, ou mais abreviadamente Redes *Feedforward* do inglês, são uma estrutura se compõe duma sequência de diversas camadas, em que cada neurónio duma camada está conectado a todos os neurónios da camada seguinte. Diz-se serem redes *densas* ou *totalmente conectadas*, pois todas as saídas de um neurónio na camada anterior são usadas como entradas para todos os neurónios na camada subsequente. Ademais, o sinal flui diretamente de camada para camada, sem pular camadas e,

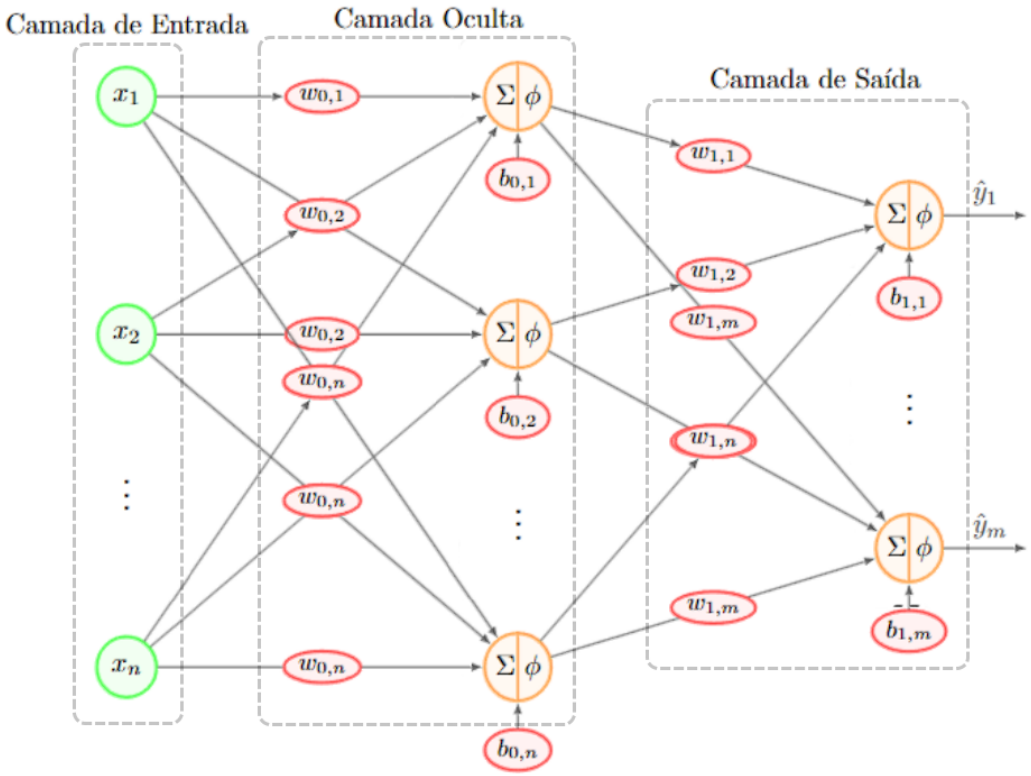


Figura 2.2: **Representação esquemática de uma Rede Neural *Feedforward* com uma camada oculta.** As conexões entre os neurónios são representadas por elipses vermelhas, que simbolizam os pesos sinápticos atribuídos a cada conexão. O valor do termo de polarização é indicado por uma pequena elipses vermelha abaixo de cada neurónio. Os detalhes das etapas de soma ponderada dos pesos e aplicação da função de ativação foram sintetizados num único bloco para maior clareza (circulo laranja).

crucialmente, essa conexão ocorre sem realimentação, isto é, as saídas de uma camada servem exclusivamente de entrada para a camada subsequente (Goodfellow et al., 2016).

A camada de entrada recebe os dados em bruto, com tantos nós quantas as dimensões do problema, e encaminha-os para a próxima camada. As camadas que se localizam entre as camadas de entrada e a camadas de saída são denominadas camadas *ocultas*, pois a suas atividades e valores não são diretamente observáveis. Cada neurónio numa camada oculta processa os sinais provenientes da camada anterior, realizando uma combinação ponderada das conexões de entrada e aplicando uma função de ativação (tipicamente) não linear, as quais exploraremos em maior detalhe na Secção 2.1.2. De relevo, a presença de múltiplas camadas ocultas permite que a rede capture e aprenda representações complexas a partir das variáveis de entrada, tornando-se capaz de identificar padrões e relações não lineares nos dados.

Finalmente, a camada de saída, é responsável por fornecer os resultados ou previsões do modelo. O número de neurónios nessa camada varia conforme o tipo de problema. Em

problemas de classificação, o número de neurónios corresponde ao número de classes possíveis, enquanto em problemas de regressão, o número de saídas é tipicamente equivalente ao número de variáveis que se pretendem estimar.

Funções de Ativação

Uma função de ativação (por vezes, denominada também como *função de transferência*) numa rede neural estabelece a forma pela qual a combinação ponderada das entradas é transformada num resultado nos nós de saída de uma determinada camada da rede (Raschka et al., 2016). A Figura 2.3 ilustra algumas das funções de ativação vulgarmente utilizadas em redes neuronais de que falaremos adiante, ainda que existam muitas outras.

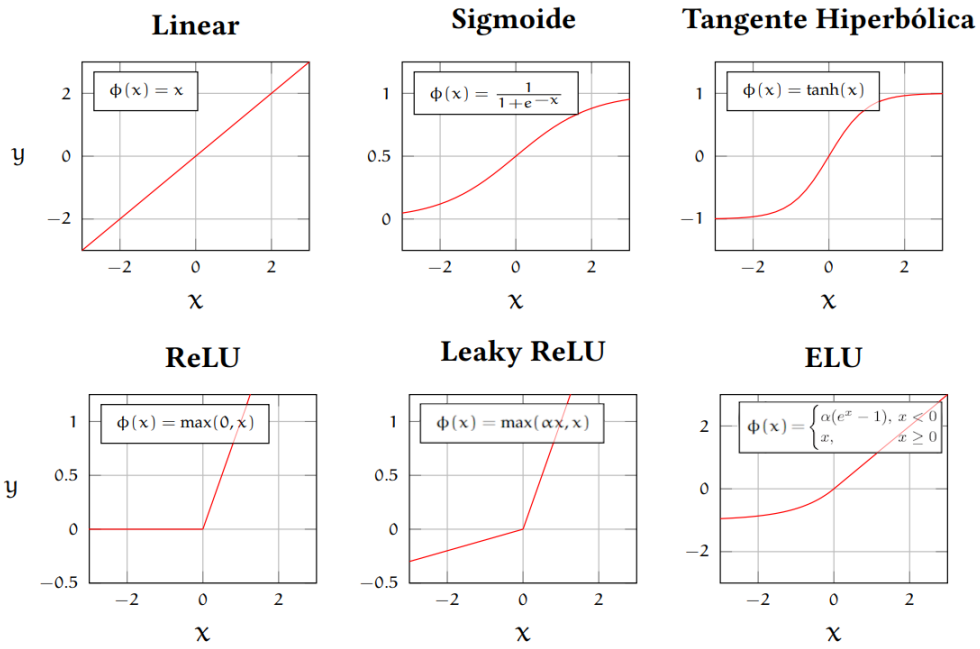


Figura 2.3: Algumas das principais funções de ativação usadas em redes *feedforward*. Cada função é representada por um gráfico que mostra a relação entre a entrada (eixo x) e a saída (eixo y) da função. As funções incluem a sigmoide logística, $\sigma(z)$ a tangente hiperbólica $\tanh(z)$, mais populares nos anos 1990, e a unidade linear retificada (ReLU), que ganhou destaque no início dos anos 2010 e contribuiu para a maior facilidade de treino de redes neuronais profundas.

As funções de ativação não-lineares conferem à rede a capacidade de aprenderem representações abstratas e hierárquicas dos dados, o que é crucial para o reconhecimento de padrões em dados de elevada dimensionalidade, como imagens ou texto. Adicionalmente, a não linearidade das funções de ativação permite que uma rede neural de múltiplas camadas aproxime qualquer função, independentemente da sua complexidade, quando adequadamente configurada e treinada (Hornik et al., 1989).

As funções de ativação também costumam ser diferenciáveis, o que implica a possibilidade de calcular as suas derivadas de primeira ordem em relação a uma entrada dada. Isso assume importância fundamental, visto que redes neuronais são frequentemente treinadas por meio do algoritmo de retropropagação de erro.

Em geral, todas as camadas ocultas aplicam a mesma função de ativação. Em contrapartida, a camada de saída se vale de uma função de ativação distinta das camadas ocultas, alinhando-se com a natureza das previsões requeridas pelo modelo.

Começando com as funções de ativação tipicamente empregadas nas camadas ocultas, historicamente, as funções de ativação sigmoide e tangente hiperbólica eram as mais utilizadas. Enquanto a função de ativação sigmoide mapeia a entrada para um valor entre 0 e 1, a função tangente hiperbólica mapeia a entrada para um valor entre -1 e 1. A tangente hiperbólica era comummente usada em redes neuronais antes da ascensão da Unidade Linear Retificada, em inglês *Rectified Linear Unit*, ReLU.

A ReLU, tornou-se popular em meados dos anos 2010, por ser simples e fácil de calcular: substitui os valores de entrada negativos por zero e mantém os valores positivos inalterados. Adicionalmente, acelera significativamente o treino devido à melhoria do fluxo dos gradientes (Krizhevsky et al., 2012a).

A LeakyReLU (B. Xu et al., 2015), traduzido livremente, Unidade Linear Retificada com Vazamento, é uma variação da função de ativação ReLU que aborda o problema da *morte* de neurónios. Este fenómeno geralmente sucede quando os pesos associados a esse neurónio são atualizados de tal forma que a soma ponderada das entradas (antes da aplicação da função ReLU) é sempre negativa. Nesse estado, o neurónio fica preso num estado perpetuamente inativo e, portanto, *morto*. Se muitos neurónios ficarem inativos, a capacidade do modelo diminui. Deste modo, a LeakyReLU atribui uma pequena inclinação α positiva para valores de x negativos. Este vazamento evita que os neurónios fiquem inativos durante o treino, pois estes ainda contribuem com informação para a rede, mesmo que numa escala menor.

Mais recentemente, foram introduzidas a Unidade Linear Exponencial (ELU), do inglês *Exponential Linear Unit* (Clevert et al., 2016) que representa uma variação mais complexa, mas muitas vezes com melhor desempenho que as ReLU e LeakyReLU.

A LeakyReLU apresenta semelhanças com a ReLU, porém tem uma saída suave para entradas negativas. Ao contrário da ReLU, a ELU pode produzir saídas negativas, conferindo-lhe maior flexibilidade. Adicionalmente, tem tendência para fazer convergir as redes mais rapidamente para valores da função de custo inferiores, o que se traduz em modelos com melhor desempenho que um modelo análogo utilizando ReLU (Ramachandran et al., 2018).

No que diz respeito às funções de ativação frequentemente empregadas na camada de saída, as escolhas variam conforme a natureza da tarefa, seja ela de classificação binária, classificação multiclasse ou regressão. Para tarefas de regressão, onde a previsão envolve a estimativa de valores contínuos, a função de ativação linear é frequentemente usada. Essa função não introduz não linearidades e permite que a rede gere saídas proporcionais às somas ponderadas das entradas.

Por outro lado, em cenários de classificação binária, a função de ativação sigmoide, já referida anteriormente, é uma escolha comum. Esta mapeia a saída para um intervalo entre 0 e 1, valore esse que representa uma estimativa da probabilidade de pertença de uma de duas classes possíveis.

Para problemas de classificação multiclasse, a função de ativação softmax é amplamente empregada. A softmax transforma as saídas em valores de probabilidade para várias classes, garantido que a soma das probabilidades de todas as classes seja igual a 1, o que a torna apropriada para selecionar a classe mais provável de um conjunto de opções (Bishop e Nasrabadi, 2006).

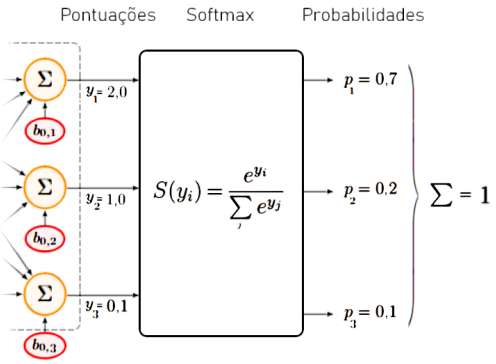


Figura 2.4: Função de ativação softmax na última camada de ANN para classificação multiclasse

Formulação Matemática duma ANN Feedfoward

Considerando uma ANN *feedforward* genérica, composta por L camadas ocultas, e dado um vetor de entrada $\boldsymbol{x}$ de dimensão n e um vetor alvo $\boldsymbol{y}$ de dimensão m, a primeira camada é a camada de entrada que recebe diretamente o vetor de entrada $\boldsymbol{x}$. A saída de cada neurónio na camada de entrada é simplesmente o componente correspondente do vetor de entrada.

As camadas subsequentes são as L camadas ocultas. Como referido anteriormente, cada neurónio nas camadas ocultas recebe como entrada as saídas dos neurónios da camada

anterior, e a saída de um neurónio nas camadas ocultas é calculada aplicando uma função de ativação ϕ à soma ponderada das suas entradas.

Assim, para o i-ésimo neurónio na l-ésima camada oculta, de forma análoga ao que do que se definiu na Equação 2 para um neurónio individual, a soma ponderada é calculada como:

$$z_i^{(l)} = \sum_{j=1}^{n^{(l-1)}} w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)} \tag{2}$$

em que $w_{ij}^{(l)}$ é o peso que liga o j-ésimo neurónio da $(l-1)$-ésima camada ao i-ésimo neurónio da l-ésima camada, $a_j^{(l-1)}$ é a saída do j-ésimo neurónio na camada $(l-1)$, e $b_i^{(l)}$ é o termo de polarização associado ao neurónio i da l-ésima camada.

A saída do mesmo neurónio é então obtida através da aplicação da função de ativação ϕ:

$$a_i^{(l)} = \phi(z_i^{(l)}) \tag{3}$$

A camada final ou a camada de saída, que produz o vetor de saída previsto $\hat{\boldsymbol{y}}$. O cálculo da camada de saída é semelhante ao das camadas ocultas, no entanto, a função de ativação utilizada pode diferir consoante a tarefa. Consideremos três cenários distintos: a tarefa de regressão, de classificação binária e de classificação multiclasse.

Na tarefa de regressão, a camada de saída é composta por um único neurónio que calcula uma saída numérica, representando uma estimativa ou previsão. Para a regressão multivariada, isto é, para problemas onde é necessário predizer várias variáveis simultaneamente, será preciso de um neurónio por variável. Em geral, para a regressão não é necessária uma função de ativação para os neurónios de saída, portanto estes ficam *livres* para gerar qualquer intervalo de valores e a saída $\hat{\boldsymbol{y}}$ do neurónio na camada de saída é obtida diretamente por:

$$\hat{\boldsymbol{y}} = \sum_{i=1}^{n^{(L)}} w_i^{(L)} a_i^{(L-1)} + b^{(L)} \tag{4}$$

onde $w_i^{(L)}$ são os pesos associados ao neurónio na camada de saída, $a_i^{(L-1)}$ é a saída do neurónio i-ésimo na última camada oculta, e $b^{(L)}$ é o termo de polarização do neurónio da camada de saída.

Para a classificação binária, a camada de saída possui um único neurónio que gera uma saída entre 0 e 1, representando a probabilidade de pertencer a uma das duas classes possíveis.

A saída $\hat{\boldsymbol{y}}$ é obtida através da aplicação de uma função de ativação sigmoide:

$$\hat{\boldsymbol{y}} = \sigma\left(\sum_{i=1}^{n^{(L)}} w_i^{(L)} a_i^{(L-1)} + b^{(L)}\right) \tag{5}$$

onde σ é a função sigmoide (ver Tabela 2.3).

Para o caso da classificação multiclasse, a camada de saída contém k neurónios, onde k é o número de classes possíveis. Como vimos antes, a função de ativação softmax garante todas as probabilidades estimadas estejam compreendidas entre 0 e 1 e que o somatório das probabilidades seja 1. Assim, cada neurónio na camada de saída gera uma probabilidade associada a uma classe específica e a saída $\hat{\boldsymbol{y}}_k$ do k-ésimo neurónio da camada de saída é obtida aplicando a função softmax, tal como definido na equação:

$$\hat{\boldsymbol{y}}_k = \frac{e^{z_k^{(L)}}}{\sum_{j=1}^{k} e^{z_j^{(L)}}} \tag{6}$$

onde $z_k^{(L)}$ é a soma ponderada das entradas para o k-ésimo neurónio da camada de saída, tal como definido na Equação 2.

2.1.3 *Funções de Custo*

A função de custo pode ser definida com uma função matemática que atribui a um evento ou aos valores de uma ou mais variáveis de entrada uma medida numérica que reflete o quão *caro* ou desvantajoso é o resultado desse evento ou conjunto de valores. Por outras palavras, no contexto da aprendizagem automática, função de custo (ou função de erro) é uma medida quantitativa que avalia o quão próximas estão as previsões do modelo dos resultados objetivo.

Crucialmente, para além de quantificar o erro dos modelos, o valor da função de custo é usado para guiar o processo de ajuste dos pesos e termos de polarização da ANN durante o treino. Isto é, tipicamente o ajuste dos parâmetros traduz-se na minimização duma função de custo $\mathcal{L}$ que reflete a desempenho da previsão num determinado conjunto de dados $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, onde cada exemplo é representado por um par (x_i, y_i), onde x_i é a entrada e y_i é a saída associada.

A escolha da função de custo depende do tipo de problema que se pretende resolver e da arquitetura empregada. De seguida, descrevemos algumas funções de custo mais comuns utilizadas para otimização de ANN em problemas de aprendizagem supervisionada[2].

Erro Quadrático Médio

O Erro Quadrático Médio, em inglês *Mean Squared Error* (MSE), é uma métrica comummente utilizada para avaliar a qualidade de um modelo em problemas de regressão. Esta mede a média dos quadrados das diferenças entre os valores previstos pelo modelo e os valores reais do conjunto de dados. No contexto do treino de modelos de aprendizagem automática e ANN, a função de custo é geralmente formulada como uma medida que se visa minimizar durante o processo de otimização. A formulação matemática da $\mathcal{L}_{MSE}$ para um conjunto de n exemplos de treino é dada por:

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7}$$

onde y_i é o valor real do i-ésimo exemplo e $\hat{y}_i$ é a previsão da ANN para o i-ésimo exemplo do conjunto de dados $\mathcal{D}$.

Para problemas de classificação, são utilizadas funções de custo como a Entropia Cruzada e a Entropia Cruzada Categórica, que quantificam a discrepância entre a distribuição de probabilidades dos dados observados e as probabilidades previstas pelo modelo.

Entropia Cruzada

A Função de custo Entropia Cruzada (também conhecida como função log-verosimilhança negativa de Bernoulli) é uma função de custo amplamente empregada em problemas de classificação binária. Esta mede o grau de dissimilaridade entre a distribuição de probabilidades reais das classes e as probabilidades estimadas pelo modelo. A fórmula geral para a função de custo de Entropia Cruzada é dada por:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{n}\left(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right) \tag{8}$$

onde n é o número de exemplos, y_i é o valor real da classe binária (0 ou 1) para o exemplo i, e $\hat{y}_i$ é a probabilidade prevista pelo modelo de que o exemplo i pertence à classe 1. A

2 A aprendizagem supervisionada é um tipo de aprendizagem automática onde um modelo é treinado usando um conjunto de dados que inclui exemplos anotados, ou seja, pares de entrada e saída desejados. Neste tipo de aprendizagem o objetivo é que o modelo aprenda a mapear as entradas para as saídas corretas, permitindo que este faça previsões precisas para novos dados não vistos com base no que foi aprendido durante o treino.

minimização Entropia Cruzada equivale a ajustar os parâmetros do modelo para que as probabilidades estimadas estejam o mais próximo possível das probabilidades reais.

Entropia Cruzada Categórica

Para problemas de classificação multiclasse, tipicamente utiliza-se uma extensão natural da função de custo de Entropia Cruzada, a Entropia Cruzada Categórica, em inglês *Categorical Cross Entropy* (CCE). A fórmula geral para a $\mathcal{L}_{CCE}$ é dada por:

$$\mathcal{L}_{CCE} = -\sum_{i=1}^{n}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{9}$$

onde C é o número de classes, y_{ij} é a indicação se o exemplo i pertence à classe j (1 se pertence, o caso contrário), e $\hat{y}_{ij}$ é a probabilidade prevista pelo modelo para que o exemplo i pertença à classe j. Análoga à congénere binária, a $\mathcal{L}_{CCE}$ avalia a discrepância entre a distribuição de probabilidades reais das classes e as probabilidades estimadas pelo modelo para todas as classes.

Existem outras alternativas além das mencionadas anteriormente, que podem ser utilizadas para no processo de otimização das ANN. O Erro Absoluto Médio é apropriado para problemas de regressão, e calcula a média das diferenças absolutas entre as previsões e os valores reais, servindo para reduzir a influência de valores discrepantes. A função de custo de Entropia Cruzada Esparsa é uma variação da $\mathcal{L}_{CCE}$ que adapta a sua fórmula no sentido de lidar com anotações codificados como números inteiros, ao invés de codificação *one-hot*. A função de custo de Entropia Cruzada Focal é uma variante da $\mathcal{L}_{CE}$ que aborda a questão dos desequilíbrios de classes em tarefas de classificação. Esta atribui maior peso aos exemplos das classes minoritárias, melhorando o desempenho em casos onde o conjunto de dados é desbalanceado.

A escolha da função de custo apropriada depende do tipo de problema e das características dos dados e contribui para a desempenho global do modelo treinado.

2.1.4 *Algoritmo de Retropropagação do Erro*

A otimização de ANN pelo algoritmo enxame de partículas (Gudise e Venayagamoorthy, 2003), pelo algoritmo forward-forward (G. Hinton, 2022) ou por algoritmos genéticos (Stanley e Miikkulainen, 2002), por exemplo, representam alternativas ao algoritmo de retropropagação do erro. Estas abordagens oferecem perspetivas distintas para o treino de ANN e podem ser consideradas alternativas viáveis ao algoritmo de retropropagação do

erro, dependendo do contexto do problema e dos objetivos do treino. Apesar disso, o algoritmo mais amplamente empregado para o problema de otimização de ANN contínua, indiscutivelmente, a ser o algoritmo de retropropagação do erro, também denominado em inglês simplesmente por *Backpropagation* (Géron, 2019). Examinamos, de seguida, em maior profundidade, o seu modo de operação.

Suponhamos que temos uma ANN do tipo *feedfoward* com parâmetros W, representando todos os pesos das camadas e b todos os termos de polarização das camadas, tal como demonstrado na Secção 2.1.2. A função de custo desta rede é denotada como $\mathcal{L}$ e depende da previsão da rede $\hat{y}_i$ e do valor real y_i para um exemplo de dados de treino específico x_i. O objetivo é calcular o gradiente $\nabla\mathcal{L}$ da função de custo em relação aos parâmetros W e b. Intuitivamente, o gradiente $\nabla\mathcal{L}$ pode ser visto como um vetor de derivadas de tantas dimensões quantos o número de parâmetros treináveis (W e b) que aponta na direção na qual a função de custo $\mathcal{L}$ aumenta mais rapidamente.

Estabelecido que o objetivo é a diferença entre o valor previsto $\hat{y}_i$ e o valor real y_i seja tão pequeno quanto possível, ajustam-se os parâmetros na direção na qual a função de custo diminui, ou seja, na direção oposta ao gradiente ou, como indica o nome do algoritmo, no sentido do gradiente descendente, com objetivo de minimizar a função de custo $\mathcal{L}$ (Figura 2.5).

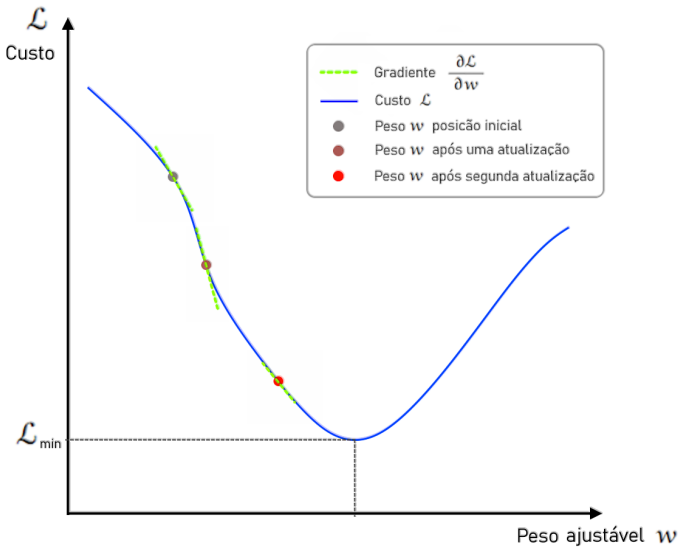


Figura 2.5: **Esquema simplificado do método do gradiente numa função de custo** $\mathcal{L}$. O peso w é iterativamente atualizado consoante a direção na qual o gradiente diminui visando atingir um mínimo global da função de csuto $\mathcal{L}$. A magnitude de cada passo de atualização é determinado pela taxa de aprendizagem α.

O cálculo do gradiente compreende dois passos: inicialmente, calcula-se o gradiente local em cada camada e, de seguida, propaga-se esse gradiente de volta para as camadas

anteriores. A regra da cadeia é usada para calcular o gradiente local de cada neurónio. Para um neurónio na camada $l \in \{1, 2, \ldots, L\}$, o gradiente local é dado por:

$$\delta_i^{(l)} = \frac{\partial \mathcal{L}}{\partial z_i^{(l)}} \tag{10}$$

onde $\mathcal{L}$ é a função de custo e $z_i^{(l)}$ é a soma ponderada das entradas para o neurónio, tal com definido na Equação 2.

O gradiente local é então propagado para as camadas anteriores usando a seguinte relação:

$$\delta_j^{(l-1)} = \sum_i w_{ji}^{(l)} \delta_i^{(l)} \cdot \phi'(z_j^{(l-1)}) \tag{11}$$

onde $w_{ji}^{(l)}$ é o peso da conexão entre o neurónio i na camada l e o neurónio j na camada $(l-1)$, $\delta_j^{(l-1)}$ é o gradiente local na camada $(l-1)$ e $\phi'(z_j^{(l-1)})$ é a derivada da função de ativação.

Após calcular o gradiente relativamente a cada peso e termo de polarização na rede, os parâmetros são atualizados mediante o método do gradiente. Os pesos e termos de polarização são ajustados na direção oposta ao gradiente, com uma taxa de aprendizagem α que controla a magnitude do passo.

A regra de atualização dos pesos e termos de polarização é dada por:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \alpha \delta_i^{(l)} a_j^{(l-1)} \tag{12}$$

$$b_i^{(l)} \leftarrow b_i^{(l)} - \alpha \delta_i^{(l)} \tag{13}$$

onde $\delta_i^{(l)}$ é o gradiente local do neurónio i na camada l e $a_j^{(l-1)}$ é a saída do neurónio j na camada $(l-1)$.

O algoritmo de retropropagação é aplicado iterativamente a um conjunto de dados de treino. A cada iteração, os pesos e termos de polarização são atualizados com base no gradiente calculado para um exemplo do conjunto de dados de treino. Tipicamente, este processo é repetido por várias épocas[3] até que a função de custo convirja para um mínimo ou alcance um ponto de convergência satisfatório.

Quando conjugado a uma seleção apropriada de função de custo, taxa de aprendizagem, configuração arquitetónica da rede, e demais hiperparâmetros, o algoritmo de retropro-

3 No contexto do treino de redes neuronais, uma *época* refere-se a uma passagem completa do conjunto de treino pela rede neuronal, abrangendo propagação direta, cálculo de gradientes e ajuste de pesos.

pagação, confere às ANN a capacidade de discernir padrões complexos e executar uma variedade de tarefas com eficácia.

2.1.5 *Treino das Redes Neuronais*

Na prática, para a otimização das ANN *feedfoward* mediante o algoritmo de retropropagação do erro é necessário considerar os aspetos relativos às métricas de avaliação de desempenho para o problema específico em causa, assim como a estipulação critérios de paragem. Sucintamente, enumeramos um processo genérico de treino de ANN como um procedimento iterativo que compreende várias etapas, conforme ilustrado na Figura 2.6.

O processo de treino de ANN começa com a inicialização dos pesos e dos termos de polarização. Uma inicialização adequada dos parâmetros aprendíveis pode acelerar a convergência do modelo. Geralmente, os pesos são inicializados aleatoriamente a partir de uma distribuição normal para evitar que a rede fique *presa* em mínimos locais. Outros métodos de inicialização, como a Inicialização Xavier/Glorot (Glorot e Bengio, 2010) ou a Inicialização He (He et al., 2015), também são amplamente utilizados. Estes métodos ajustam os pesos com base no número de entradas e saídas de cada neurónio, facilitando a convergência mais rápida da rede e evita problemas de saturação nos neurónios (Kumar, 2017; Wong et al., 2022).

Uma vez que os pesos e termos de polarização estão inicializados, a rede começa o processo normal de aprendizagem por retropropagação do erro, tal como discutido na Secção 2.1.4. Este processo prolonga-se tipicamente até todos os dados de treino terem sido processados, isto é, até terem contribuído para o ajuste dos parâmetros pelo algoritmo de retropropagação.

Usualmente, durante o treino ou no final de cada época, ausculta-se o desempenho da rede num conjunto de dados nunca *visto* pelo rede, o qual é geralmente designado de conjunto de validação. Este conjunto ajuda a monitorizar o progresso do treino e a evitar problemas como o sobreajuste, no qual a rede memoriza os dados de treino aos invés de generalizar a aprendizagem para novos dados.

A avaliação de desempenho envolve métricas específicas relacionadas à tarefa para a qual a rede está a ser treinada. Por exemplo, em tarefas de classificação, pode-se usar métricas como a Precisão, Revocação ou a Medida F1 (ver Capítulo 4, Secção 4.2) para avaliar a evolução do desempenho das previsões da rede no conjunto de dados de validação.

O treino de uma ANN é um processo iterativo que se pode prolongar indefinidamente, pelo que é necessário definir critérios de paragem que indiquem quando o treino deve

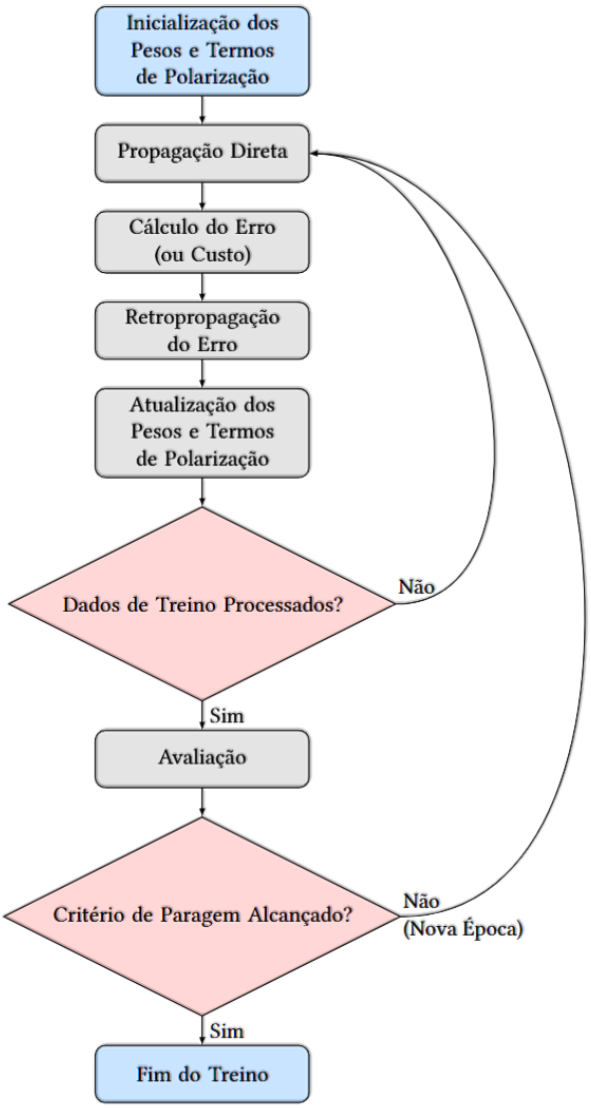


Figura 2.6: **Fluxo de Treino de ANN *Feedfoward***. O bloco de decisão inicial verifica se todos os dados de treino foram processados antes de prosseguir para as etapas de propagação, cálculo do erro, retropropagação e atualização dos pesos. O treino continua até que o critério de paragem seja alcançado, indicando o "Fim do Treino".

ser interrompido. Esses critérios podem incluir um número máximo de épocas ou iterações de treino, uma determinada taxa de variação de valor do erro sobre o conjunto de treino ou de validação, quando se verifica que valor de uma métrica de avaliação não apresenta melhorias ou uma combinação de vários critérios. Os critérios enumerados não são exaustivos, mas exemplificam situações de paragem possíveis.

Uma vez atingido o critério de paragem, dá-se por terminado treino e os valores dos parâmetros treináveis são *congelados* e salvos.

2.2 REDES CONVOLUCIONAIS

Nas últimas décadas, o campo da visão computacional tem testemunhado um avanço notável, impulsionado em grande parte pela capacidade das CNN em processar e *compreender* informações visuais de maneira análoga ao córtex visual primário humano (LeCun et al., 1998; Ulku e Akagündüz, 2022). A motivação subjacente a esse progresso reside na necessidade de extrair características de alto valor semântico de dados visuais complexos, tais como imagens e vídeos, com o fim de solucionar tarefas como a classificação, deteção ou segmentação de objetos, por exemplo.

O desafio intrínseco na análise de imagens reside na sua alta dimensionalidade e na complexidade das informações presentes. Características prontamente identificadas pelos seres humanos, como objetos, texturas, formas e contextos, são bastante desafiantes de serem formalmente definidas em termos de algoritmos tradicionais de processamento de digital de imagens. As abordagens clássicas, como a extração manual de características e algoritmos de classificação baseados em regras, mostraram-se, muitas vezes, limitadas diante da variabilidade e da natureza não-linear dos dados visuais (Alzubaidi et al., 2021).

Ao contrário das técnicas clássicas, as redes convolucionais mostraram ser capazes de solucionar alguns destes desafios. Inspiradas pela organização hierárquica do córtex visual humano, as CNN multicamada são projetadas para aprender automaticamente representações de características relevantes a partir dos dados brutos, abstraindo informações de valor semântico mais baixo, como bordas e texturas nas camadas iniciais e, progressivamente, representações de maior valor semântica, como partes ou a totalidade de objetos, bem como o contexto desses objetos ou classes semânticas, nas camadas finais (Figura 2.7).

Outra característica inerente às operações com camadas convolucionais é que permite a deteção de padrões espaciais independentemente da sua posição na figura, ao contrário do que é possível, por exemplo, com redes *feedfoward* totalmente conectadas. Adicionalmente,

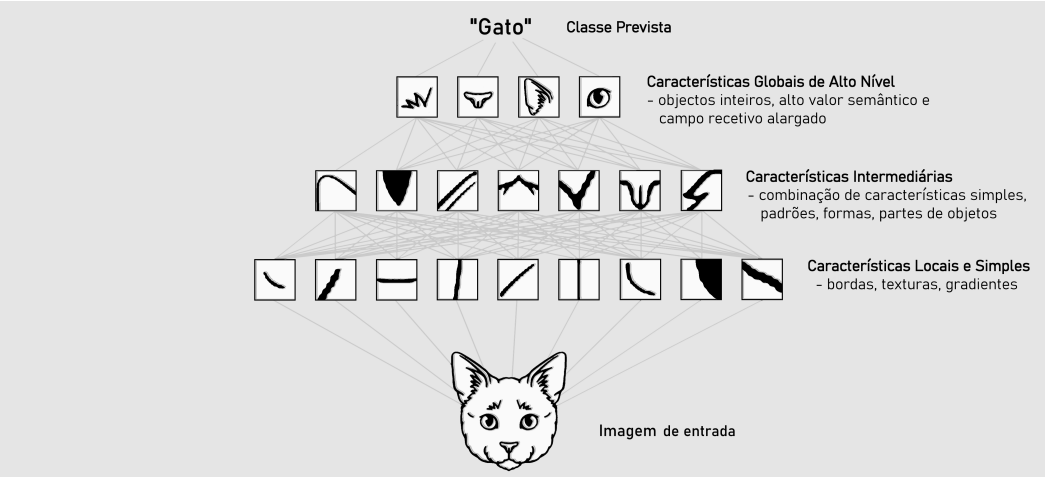


Figura 2.7: **Extração hierárquica de características numa CNN**. A informação visual é extraída mediante uma hierarquia espacial: linhas e texturas elementares convergem para formar objetos simples, como olhos ou orelhas, que por sua vez se combinam para criar conceitos mais complexos, como a representação de um gato.

as CNN empregam campos recetivos locais para operações de convolução, o que significa que cada unidade numa camada específica se conecta apenas a uma pequena região da entrada na camada anterior. Essa partilha de parâmetros nos campos recetivos locais permite que a rede aprenda a detetar a mesma característica, independentemente da sua posição na entrada, mantendo assim a invariância ao deslocamento.

Por outro lado, as redes *feedfoward* totalmente conectadas (tal como definido na Secção 2.1.2) estabelecem conexões entre cada unidade e todas as unidades na camada anterior, o que as torna menos eficazes para capturar padrões locais. A partilha de parâmetros reduz drasticamente a quantidade de parâmetros treináveis nas CNN, em comparação com redes totalmente conectadas. Ao contrário das redes totalmente conectadas, as CNN partilham um pequeno conjunto de parâmetros entre diferentes localizações, resultando num modelo mais compacto e eficiente. A partilha de parâmetros reduz a carga computacional tanto durante o treino quanto durante a inferência.

As redes convolucionais modernas incorporaram princípios oriundos de várias descobertas fundamentais que moldaram a compreensão do processamento visual no sistema nervoso. As experiências realizadas por Hubel e Wiesel, ao longo das suas carreiras, revelaram intuições importantes que expandiram o entendimento sobre a forma como o cérebro interpreta os estímulos visuais. Uma das descobertas mais impactantes decorreu das experiências que desvendaram a seletividade de orientação e a organização colunar no córtex visual. Ao explorarem o córtex visual primário de gatos e macacos (Hubel e Wiesel, 1959; Hubel e Wiesel, 1968) observaram neurónios que reagiam de maneira seletiva a estímulos visuais com orientações específicas. Esses neurónios agrupam-se em colunas

no córtex, cada uma respondendo a uma orientação distinta. A perceção de orientações específicas foi particularmente evidente nas chamadas células simples, que demonstraram reações localizadas a estímulos visuais orientados (Figura 2.8).

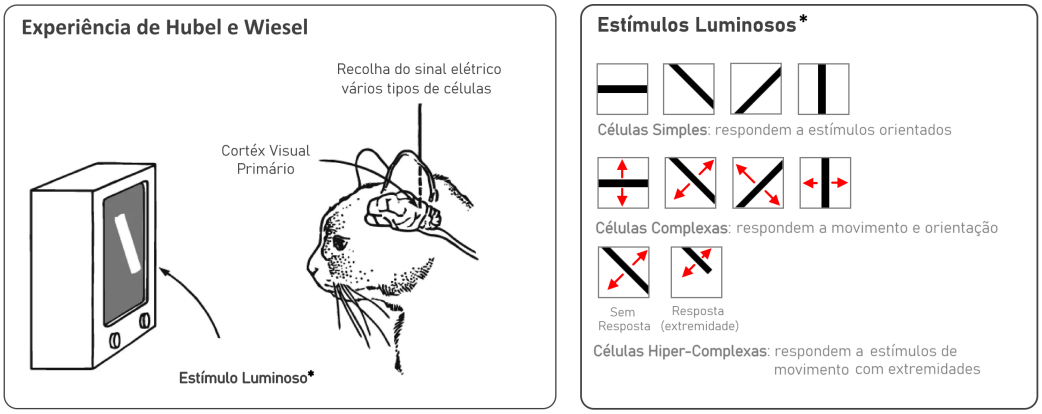


Figura 2.8: **Contribuições de Hubel e Wiesel para compreensão do córtex visual**. As experiências de Hubel e Wiesel revelaram a existência de conjunto de células distintas que disparam perante estímulos visuais de progressiva complexidade, implicando um processamento visual hierárquico e convergente. Adaptado de (Nguyen et al., 2019)

No entanto, a investigação não se deteve apenas na seletividade de orientação. Hubel e Wiesel identificaram um segundo tipo de células no córtex visual: as células complexas. Estas células complexas destacavam-se por responder a características visuais mais complexas, como o movimento. Ao contrário das células simples, as células complexas reagem a estímulos em qualquer posição do seu campo recetivo[4]. A interação das células complexas com um conjunto de células simples contribuí para a sua capacidade de formar campos recetivos mais amplos. Esta constatação levou à conceção de que o processamento visual é hierárquico e convergente, o que implica que o sistema visual gera representações complexas da informação visual a partir das características de estímulos mais elementares.

Inspirado pelo modelo hierárquico do córtex visual proposto por Hubel e Wiesel, Kunihiko Fukushima desenvolveu, cerca de duas décadas depois, o Neocognitron, uma rede neural artificial multicamada. O Neocognitron foi projetado para aprender a reconhecer padrões e identificar algarismos escritos à mão com base na semelhança geométrica das suas formas (Fukushima e Miyake, 1979). O Neocognitron é composto por camadas conectadas em cascata, com uma camada de entrada U_0 precedendo essas camadas. Além da camada de entrada, este modelo inclui dois tipos de células: células-C (U_C) e células-S (U_S), em analogia às células simples e complexas identificadas por Hubel e Wiesel (ver Figura 2.9). O Neocognitron é amplamente reconhecido como o ponto de partida das redes convolucionais modernas.

4 Campo recetivo é a porção do campo visual capaz de desencadear uma resposta neuronal na presença de estímulo

A sua arquitetura adota uma estrutura hierárquica com várias camadas de células, cada uma responsável por identificar padrões específicos com base nas saídas das camadas anteriores, em semelhança às redes convolucionais contemporâneas. Outro componente fundamental deste modelo são os filtros convolucionais que varrem as imagens em busca de características distintivas.

Outro traço distintivo do Neocognitron é a conceção de invariância de translação resultante do varrimento dos filtros. Como um mesmo filtro percorre as imagens, este processo permite a deteção de padrões independentemente das suas posições nos dados de entrada. Esta invariância de translação foi incorporada nas CNN, que compartilham a mesma abordagem.

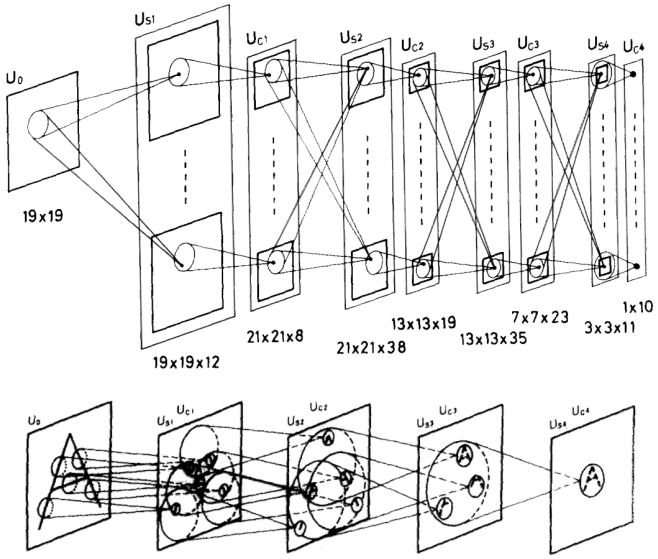


Figura 2.9: **Arquitetura do Neocognitron**. Este modelo é composta por camadas sucessivas conectadas em cascata, incluindo uma camada de entrada (U0), células-C e células-S capazes de extrair características progressivamente mais complexas e identificar caracteres ou outros padrões. Retirado de Fukushima (1988)

Em 1989, Le Cun et al. (1989) introduziram pela primeira vez uma ANN composta por camadas convolucionais, demonstrando a sua capacidade de classificar números de código postal manuscritos para o Serviço Postal dos Estados Unidos. Esta pesquisa para reconhecimento de caracteres escritos à mão em papel continuou nos anos seguintes, e em 1998, LeCun et al. (1998) publicaram um artigo seminal que detalhava a arquitetura da LeNet-5, esquematizada na Figura 2.10.

Este artigo recebeu grande interesse e apresentou os blocos básicos que compõem redes neuronais convolucionais atuais, incluindo as camadas de *pooling* (Secção 2.2.2) e as camadas convolucionais (Secção 2.2.1), além de discutir o treino da rede por meio do algoritmo de retropropagação. Os métodos descritos neste artigo serviram como ponto

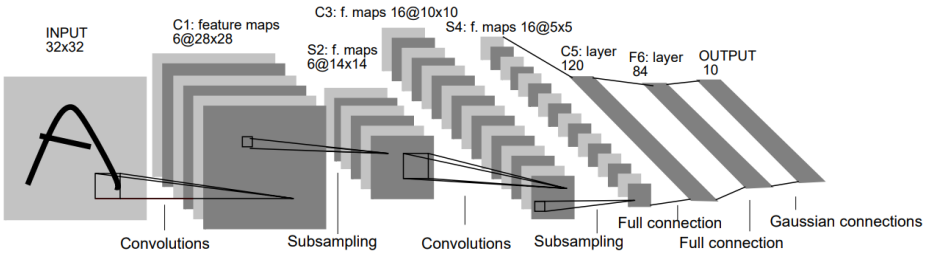


Figura 2.10: **Arquitetura da LeNet-5**. A LeNet-5 é uma rede neuronal convolucional projetada especialmente para o reconhecimento de caracteres escritos à mão. A sua arquitetura é composta por camadas convolucionais ara extração de características dos dados, seguidas por camadas de *pooling* (ou subamostragem) para reduzir a dimensão dos mapas de características. O estágio final da rede integra camadas totalmente conectadas, responsáveis por gerar a classificação das classes de interesse. Retirado de LeCun et al. (1998)

de partida essencial para a proliferação de arquiteturas e aplicações baseadas em redes convolucionais (S. Woo et al., 2023).

2.2.1 *Camadas Convolucionais*

A convolução é uma operação matemática que mede a sobreposição entre duas funções à medida que uma desliza sobre a outra. No contexto de sinais discretos e unidimensionais, um sinal pode ser representado como uma sequência de números reais $x[k]$, onde k representa o índice discreto do sinal no momento t_k. Uma característica importante dos sinais discretos é que eles podem ser processados por meio de filtragem.

Ao considerar um filtro (ou núcleo) definido por uma sequência finita de pesos $w = [w_1, w_2, \ldots, w_N]$, a operação de filtragem é realizada através da convolução, denotada como $y = x * w$, e definida como uma soma ponderada dos produtos:

$$y[n] = \sum_k x[k] \cdot w[n-k] \tag{14}$$

No caso específico de imagens monocromáticas, representadas como matrizes bidimensionais (2D) de píxeis, estendemos a operação de convolução discreta para duas dimensões. Suponhamos uma matriz bidimensional de entrada $x_{M\times N}$ e um filtro (ou núcleo) finito $w_{P\times Q}$. Aqui, M e N denotam as dimensões espaciais da matriz de entrada, enquanto P e Q denotam as dimensões do filtro. O sinal de saída $y_{(M-P+1)\times(N-Q+1)}$ é obtido da seguinte forma:

$$y[i,j] = \sum_{m=0}^{P-1} \sum_{n=0}^{Q-1} x[i+m, j+n] \cdot w[m,n] \tag{15}$$

Neste contexto bidimensional, o sinal resultante y é chamado de mapa de características (ou mapa de ativações) porque contém informações relacionadas aos padrões presentes no filtro w. Conforme mencionado anteriormente, a operação de convolução bidimensional apresenta a vantagem da invariância de translação, o que significa que a característica representada pelo filtro w será detetada em diferentes partes da imagem, independentemente da sua localização exata.

No que respeita a imagens RGB (Vermelho, Verde e Azul), tal como representado na Figura 2.11, as convoluções 2D são executadas sobre tensores tridimensionais que armazenam os valores de intensidade dos píxeis para cada um dos canais. Estes mapas de características de entrada possuem três eixos espaciais: altura, largura e um terceiro eixo de profundidade, que corresponde aos 3 canais RGB. A operação de convolução envolve o completo varrimento do mapa de características de entrada, onde *regiões* com as dimensões do filtro, são sucessivamente percorridas.

O filtro, também designado na indústria como *kernel*, assume igualmente a forma de um tensor tridimensional. As dimensões, tanto em altura como em largura (tipicamente 3×3 ou 5×5), podem ser adaptadas conforme as necessidades, enquanto a profundidade do filtro corresponde à quantidade de canais presentes na imagem de entrada. É importante notar que a profundidade do mapa de características de saída da operação de convolução é determinado pelo número de filtros aplicados em cada camada convolucional e cada filtro corresponde a uma característica particular que a rede extrairá. Esses filtros não retratam cores, como nos canais de entrada RGB, mas sim padrões relevantes para o problema em questão. Um filtro pode *aprender* a identificar bordas, determinadas texturas ou outras qualidades visuais mais complexas, permitindo que a CNN destile informações de alto valor semântico das imagens.

Padding e Stride

Em condições normais, quando a convolução 2D é aplicada conforme a equação 15, as dimensões do mapa de características de saída diferirão das do mapa de características de entrada, em particular para filtros de convolução com dimensões diferentes de 1×1. Esse fenómeno é influenciado por vários fatores, incluindo os efeitos de borda que podem surgir durante a convolução. Esses efeitos de borda são uma consequência da limitação espacial que ocorre à medida que o filtro percorre a imagem de entrada. Nas áreas próximas ao limite ou borda do mapa de características, a convolução torna-se parcial, uma vez que

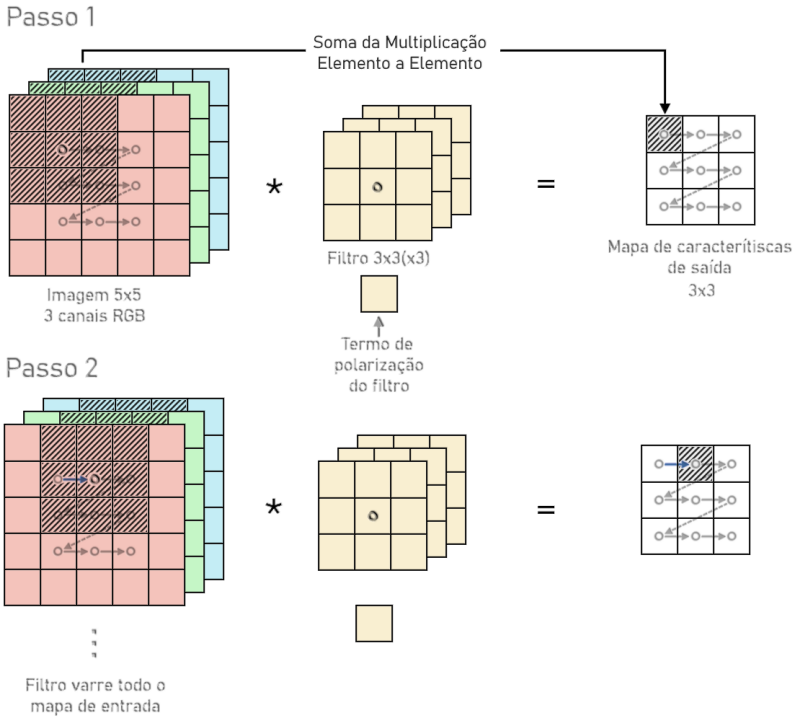


Figura 2.11: **Representação esquemática da convolução 2D**. O processo de convolução envolve o varrimento de um filtro sobre os mapas de características de entrada, multiplicando os valores dos píxeis pela correspondente ponderação no filtro e, em seguida, somando-os, juntamente com o termo de polarização. A magnitude dos valores nos mapas de saída é diretamente proporcional à sobreposição entre o filtro e o mapa de entrada, permitindo a deteção eficaz de características representadas pelo filtro.

não existe vizinhança de pixeis que permita que o filtro faca o varrimento total, resultando numa redução nas dimensões da saída, tal como esquematizado na Figura 2.12, à esquerda.

No entanto, é importante notar que os efeitos de borda podem ser solucionados por meio do preenchimento (ou *padding*) do mapa de características de entrada para adequar as dimensões do filtro às dimensões de saída desejados. Por outras palavras, ao adicionar píxeis de preenchimento ao redor da imagem de entrada, as dimensões do mapa de características de saída podem ser preservadas (Figura 2.12 à direita). Para ilustrar de forma mais intuitiva o conceito de *padding*, consideremos o mapa de características de tamanho 5×5 (25 elementos no total) representado na Figura 2.12, à esquerda. Existem apenas 9 elementos ao redor dos quais podemos centrar um filtro 3×3. Assim, o mapa de características resultante terá dimensões 3×3. Essencialmente, o mapa de saída fica com dimensão inferior.

No caso, se for objetivo manter as dimensões do mapa de saída, adição de *padding* ao redor do mapa de entrada garante que o mapa de saída preserve o tamanho 5×5 (Figura 2.12, à direita).

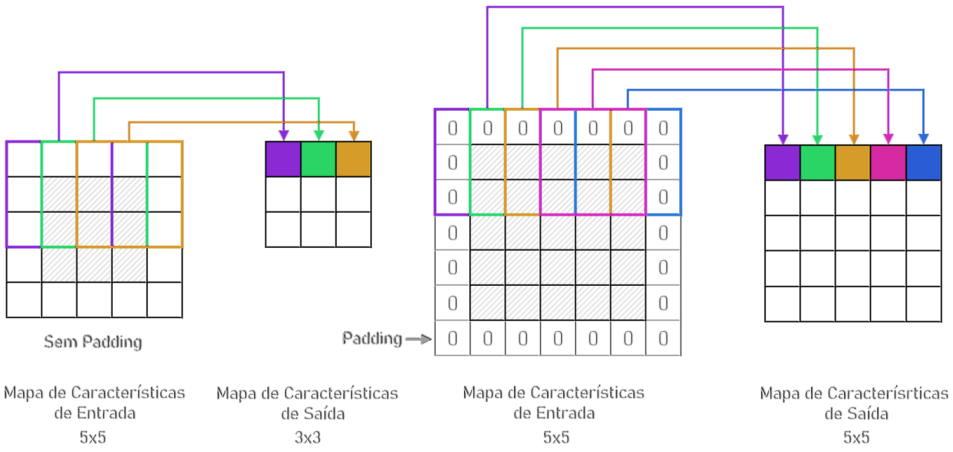


Figura 2.12: **Efeito da aplicação de *padding***. Mapa de características de 5×5 sem preenchimento, resultando num mapa de saída menor de 3×3 ao aplicar um filtro 3×3. À direita, ao aplicar *padding*, as dimensões do mapa de saída são preservadas, permitindo que o filtro varra todos os elementos do mapa de entrada.

Outro fator afeta as dimensões do mapa de características de saída é o parâmetro de *stride*. Este parâmetro é uma característica dos filtros das CNN que regula o deslocamento do filtro ao longo da imagem ou vídeo de entrada ao determinar o número de píxeis que o filtro avança durante cada aplicação. Quando o valor de *stride* é definido como 1, o filtro desloca-se píxel a píxel, mantendo uma sobreposição máxima entre as regiões afetadas pelo filtro (Figura 2.13 à esquerda).

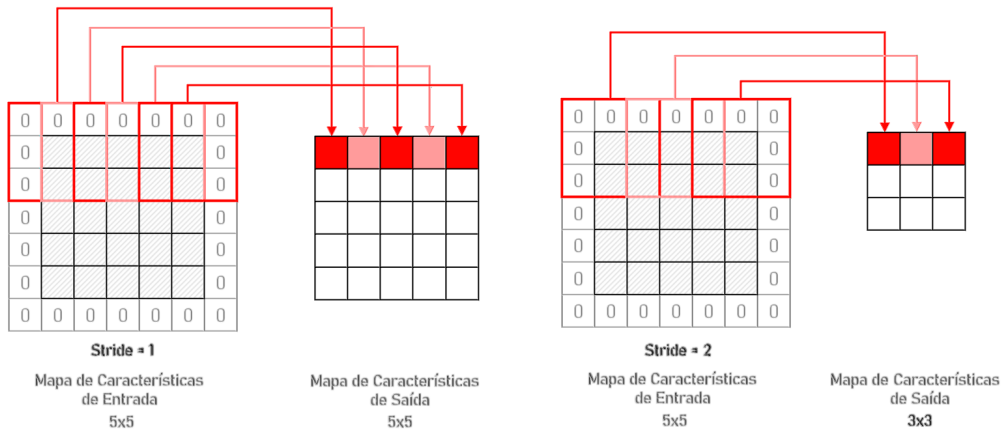


Figura 2.13: **Influência do parâmetro de *stride***. À esquerda, com *stride* igual a 1, o filtro avança píxel a píxel, mantendo sobreposição máxima. À direita, com *stride* = 2, ocorre uma redução nas dimensões do mapa de características de saída devido a uma amostragem mais esparsa.

Por outro lado, se o valor de *stride* for superior a 1, há uma redução correspondente nas dimensões espaciais do mapa de características de saída. Isso ocorre porque um *stride*

maior resulta numa amostragem mais esparsa da imagem de entrada durante a convolução (Figura 2.13 à direita).

Após examinarmos as intuições relacionadas ao *padding* e ao *stride*, avançamos para a aplicação do método de cálculo das dimensões (altura e largura) do mapa de características de saída. As dimensões obedecem à seguinte fórmula:

$$\mathrm{Dim}_{saída} = \frac{\mathrm{Dim}_{entrada} - \mathrm{Dim}_{filtro} + 2 \times padding}{stride} + 1 \tag{16}$$

onde $\mathrm{Dim}_{saída}$ corresponde a dimensão da altura ou largura do mapa de características resultante, $\mathrm{Dim}_{entrada}$ denota dimensão da imagem de entrada, Dim_{filtro} representa a dimensão do filtro de convolução aplicado, *padding* é o número de píxeis de preenchimento adicionados em torno do mapa de características de entrada e *stride* define o valor do deslocamento do filtro durante a operação de convolução.

2.2.2 *Camadas de Pooling*

Um pouco à semelhança do que é possível com o parâmetro de *stride*, o *pooling* é uma técnica utilizada com o propósito de reduzir as dimensões espaciais dos mapas de características e complexidade computacional, preservando, simultaneamente, as informações de maior relevância. A operação de *pooling* é útil quando existe a necessidade de diminuir a resolução dos mapas de características para otimizar a memória necessária no treino e na operação das CNN. Tipicamente, as camadas de *pooling* podem ser igualmente adicionadas após as camadas de convolução e preceder as camadas totalmente conectadas numa arquitetura de CNN clássica utilizada para classificação de imagens, tal como o exemplo da rede LeNet-5 representada na Figura 2.10,

A operação de *pooling* consiste no varrimento de um núcleo bidimensional sobre cada canal do mapa de características, seguido duma operação que calcula o valor médio ou máximo dos pixeis que compreendem as dimensões desse núcleo. Frequentemente, as dimensões do núcleo de *pooling* são 2×2 com um *stride* de 2, tal como exemplificado na Figura 2.14.

Na figura estão representados os dois tipos de *pooling* mais comuns. O *MaxPooling* devolve o valor máximo do valor de pixeis, o que resulta na preservação das características mais proeminentes dessa área, como seja uma borda fortemente prenunciada ou um padrão distinto. A outra variante, o *Average Pooling* calcula o valor médio dos pixeis correspondente ao núcleo. Ao passo que o Maxpooling se restringe à identificação das

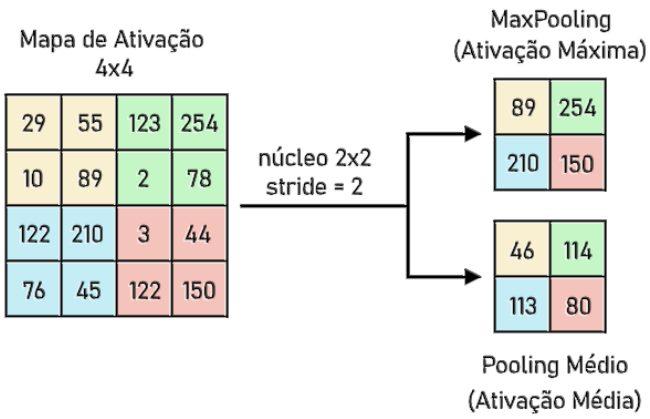


Figura 2.14: **Representação esquemática da operação de *pooling***. O *MaxPooling* preserva características proeminentes ao selecionar o valor máximo, enquanto o *MaxPooling* calcula a média dos valores, incentivando uma visão mais ampla do objeto.

características mais importante, o *Average Pooling* incentiva a rede a identificar toda a extensão do objeto e a preservação da estrutura global do mapa de características.

2.2.3 *Camadas de Convolução Transposta*

Nas arquiteturas convencionais de CNN utilizadas para classificação de imagens, é uma prática comum que a resolução espacial seja progressivamente reduzida à medida que se avança nas camadas. Essa estratégia é adotada com o propósito de diminuir a quantidade de parâmetros e a complexidade computacional da rede. Para a tarefa de classificação, a manutenção de uma alta resolução não é uma exigência crítica; em vez disso, é suficiente garantir uma representação adequada da posição relativa das características. Em situações em que é importante preservar uma resolução espacial elevada, como em tarefas de deteção ou segmentação semântica de objetos, onde é crucial identificar os limites com rigor, diversas estratégias podem ser empregadas para recuperar e sobreamostrar a resolução dos mapas de características.

Técnicas clássicas de redimensionamento de imagens digitais, como a interpolação bilinear e bicúbica, mostram-se eficazes na ampliação da resolução de imagens. Embora simples e computacionalmente eficientes, estes métodos podem resultar em imagens sem detalhe, perdendo informação de alta frequência. Em contraste, a convolução de transposta (Dumoulin e Visin, 2016; Long et al., 2015), permite a aprendizagem de filtros capazes de reproduzir características nítidas, tornando-a adequada para aumento de resolução com escalas arbitrárias e tarefas que necessitem a preservação de bordas e pequenos detalhes.

Esta abordagem pode ser interpretada como a aplicação de uma camada convolucional convencional de *stride* fracionário. De facto, enquanto o valor do *stride* controla a redução na resolução na convolução 2D convencional, aqui tem o efeito oposto. Em vez do filtro varrer o mapa de entrada e realizar a multiplicação e a soma dos elementos, na camada convolucional transposta cada elemento da entrada é multiplicado pelo filtro sucessivamente, e o resultado é somado ao mapa de características de saída que inicialmente é uma matriz de zeros, tal como esquematizado na Figura 2.15.

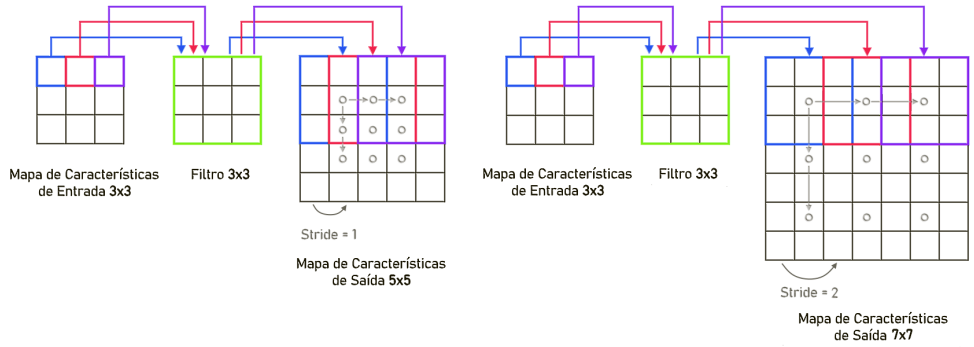


Figura 2.15: **Exemplo de Convolução Transposta 2D**. Cada elemento do mapa de características de entrada é multiplicado pela matriz que representa o filtro. O resultado dessa multiplicação é adicionado ao mapa de características sucessivamente.

Supondo que lidamos com a camada de convolução transposta de um mapa de características x com dimensões $m \times m$, onde o valor do *stride* é s tanto na direção horizontal quanto na vertical, não considerando *padding* e assumindo que usamos um filtro quadrado w de dimensões $n \times n$, podemos calcular a dimensão da saída (altura e largura), denotada como $\mathrm{Dim}_{saída}$, da seguinte forma:

$$\mathrm{Dim}_{saída} = (m - 1) \cdot s + n \tag{17}$$

O raciocínio aplicado à convolução 2D para múltiplos canais ou mapas de características de entrada que vimos na Secção 2.2.1, é também válido para a convolução transposta. A profundidade do mapa de características de saída na operação de convolução transposta é determinada pelo número de filtros aplicados e quanto maior o número de filtros, maior será a diversidade de características que a camada pode aprender a identificar.

2.2.4 *Modelos de Segmentação Totalmente Convolucionais*

A combinação das camadas de *pooling*, que têm o efeito de reduzir a resolução dos mapas de características, e as camadas de convolução transposta, que aumentam essa resolução,

permite às CNN capturar informações em diferentes escalas espaciais, tornando-as especialmente aptas para tarefas de segmentação de imagens, onde a localização precisa dos limites dos objetos de interesse desempenha um papel crucial.

Neste contexto, de seguida, apresentamos a Rede Totalmente Convolucional, em inglês *Fully Convolutional Network* (FCN). Em especial, abordaremos a arquitetura pioneira proposta por Long et al. (2015), que desencadeou o interesse e serviu de base para o desenvolvimento de uma nova classe de redes convolucionais com um desempenho assinalável.

Para melhor compreender o modelo FCN de Long et al. (2015), é essencial examinar a sua estrutura e as inovações introduzidas que a tornam um modelo de referência para a tarefa segmentação semântica, que discutiremos em maior detalhe no Capítulo 4. Sucintamente, a segmentação semântica é o processo de classificação de cada píxel de uma imagem numa classe ou categoria específica, com o objetivo de identificar e delinear as áreas correspondentes a diferentes objetos ou regiões de interesse com características semelhantes.

A estrutura da FCN começa com uma secção de extração de características baseada na arquitetura VGG16 (Simonyan e Zisserman, 2015), uma rede neural que consiste em 13 camadas convolucionais intercaladas com camadas de *pooling* e 3 camadas totalmente conectadas. Para adaptação para o modelo FCN, as camadas totalmente conectadas da VGG16 são substituidas por camadas de convolução transposta e de *MaxPooling*, de acordo com o diagrama da Figura 2.16.

Para contornar a gradual diminuição da resolução espacial nas camadas de convolução produto das operações de *pooling*, Long et al. (2015) propuseram a fusão de mapas de características provenientes de diferentes resoluções. A primeira versão, FCN-32s, não atingiu a definição de segmentação desejada, mas a FCN-16s representou um avanço ao incorporar a fusão de mapas de características da camada de *pool4*. A FCN-8s revelou-se a melhor variante, ao incluir igualmente a fusão de mapas de características da camada de *pool3* do VGG16. Em síntese, a incorporação de camadas convolucionais transpostas em conjunto com a fusão de mapas de características de diferentes camadas da VGG16 que as variantes FCN-16s e FCN-8s apresentam, constituem as inovações principais.

Apesar dos avanços, as FCN demonstram ter algumas limitações no que diz respeito à recuperação de informação de alta resolução (pouca definição), apresentam pouca capacidade para capturar o contexto global da imagem e não oferecem mecanismos adequados para processamento de classes em várias escalas (Géron, 2019; Nie et al., 2016) Estudos subsequentes tentaram colmatar estas lacunas propondo novas arquiteturas e métodos. Abordaremos algumas destas propostas no Capítulo 3, relativo ao Trabalho Relacionado.

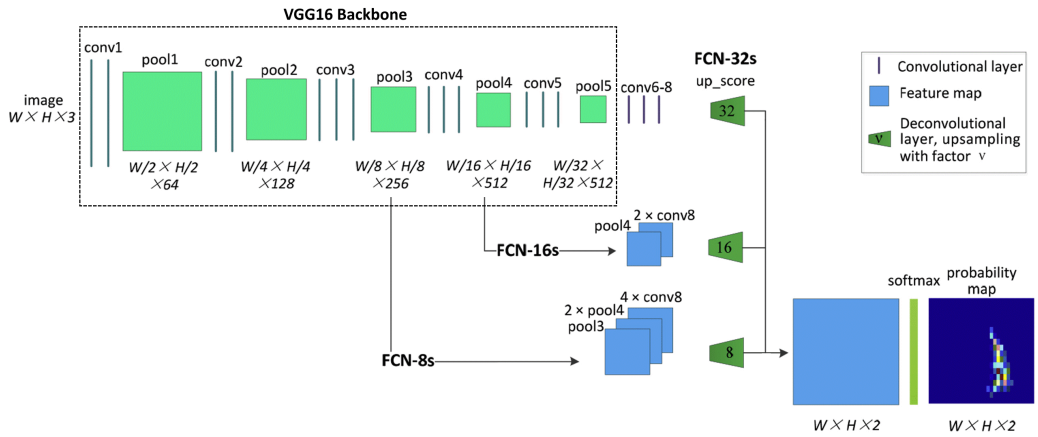


Figura 2.16: **Arquitetura da FCN**. A base da FCN é composta por uma VGG16 pré-treinada de onde se descartam as camadas densas usadas para a tarefa de classificação. Sob esta base assentam as três variantes mais importantes da FCN: FCN-32s, FCN-16s e FCN-8s. Na variante FCN-32s, a imagem é subamostrada por camadas *pooling* e sobreamostrada por convolução transposta 32 vezes para gerar um mapa de segmentação. A saída da rede é pouco definida devido à perda de informação de localização. Já a FCN-16s combina a saída da camada final com a camada *pool4* (VGG16), utilizando *stride* de 16, o que resulta em detalhes mais refinados em comparação com a FCN-32s. A FCN-8s adiciona a saída da camada pool3 com um *stride* de 8, proporcionando os melhores resultados das três variantes. Adaptado de Yang et al. (2018)

2.3 AUTOCODIFICADORES

Um Autocodificador, em inglês *Autoencoder* (AE) (G. E. Hinton e Zemel, 1993), é uma categoria de redes neuronais utilizada para aprendizagem não supervisionada[5] que recebe uma entrada de alta dimensionalidade $\mathbf{x} \in \mathbb{R}^D$, como uma imagem e, a transforma numa representação compacta e de baixa dimensão $\mathbf{z} \in \mathbb{R}^d$, designada por *espaço lantente* z, a qual é normalmente um vetor.

Esta representação comprimida é depois utilizada por um descodificador para reconstruir a entrada original. A arquitetura do AE pode ser decomposta em três componentes: o codificador $f_\phi(\cdot)$, que mapeia a entrada para o espaço latente, um descodificador $g_\theta(\cdot)$, que mapeia a representação latente de volta para o espaço de entrada, e um estrangulamento $\mathbf{z}$ que armazena os códigos comprimidos. O codificador e o descodificador são frequentemente implementados como redes neuronais com parâmetros aprendíveis ϕ e θ, respetivamente.

O treino de AE envolve minimizar a diferença entre a entrada original $\mathbf{x}$ e a saída reconstruída $\hat{\mathbf{x}}$. Por outras palavras, o objetivo é fazer com que o modelo consiga reconstruir

5 A aprendizagem não supervisionada é um paradigma de aprendizagem automática em que um algoritmo é treinado num conjunto de dados que não está anotado ou sem informação acerca de saída pretendida. O objetivo principal é encontrar estrutura subjacente ou representações úteis dos dados, sem que haja orientação externa, necessidade ou possibilidade de previsões específicas.

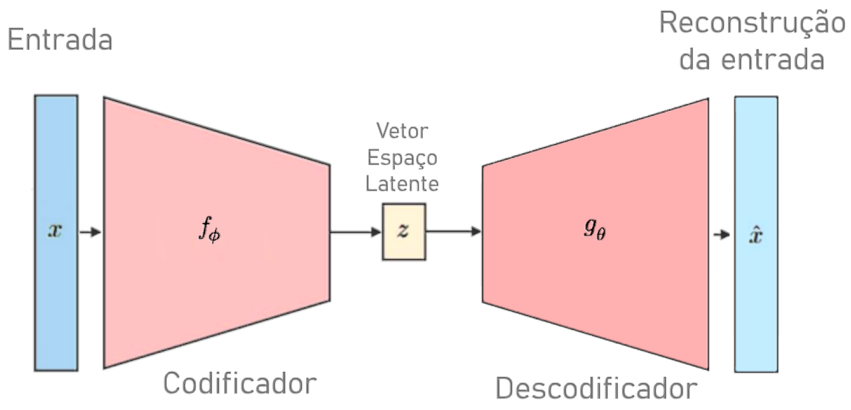


Figura 2.17: Arquitetura de um autocodificador (AE)

com precisão os dados de entrada a partir da sua representação latente comprimida **z**. Para alcançar isso, ajustam-se iterativamente os parâmetros ϕ e θ da rede para reduzir o erro de reconstrução.

Esta categoria de modelos têm diversas aplicações, incluindo compressão de imagens (Z. Cheng et al., 2018), deteção de anomalias (Z. Chen et al., 2018) e tarefas generativas, como a geração de novos dados semelhantes aos dados de treino (Demir et al., 2021). Estamos particularmente interessados no estudo destes modelos devido às suas notáveis capacidades de interpolação e reconstrução de dados espaçotemporais, o que os torna valiosos numa variedade de domínios.

2.3.1 *Autocodificadores Variacionais*

O Autocodificador Variacional (Kingma e Welling, 2014; Rezende et al., 2014), em inglês *Variational Autoencoder* (VAE), um desenvolvimento do AE, consiste num codificador $q_\phi(z|x)$ e num descodificador $p_\theta(x|z)$. Ao contrário dos AE padrão, os VAE aprendem uma distribuição probabilística do espaço latente em vez de um mapeamento determinístico.

Os VAE são treinados para minimizar o limite inferior de evidência, do inglês *Evidence Lower Bound* (ELBO), em $\log p(x)$, em que $p(x)$ é a distribuição geradora de dados. O ELBO pode ser expresso como:

$$\mathcal{L}(\theta, \phi, x) = \mathbb{E}q\phi(z|x)[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{18}$$

Aqui, $p(z)$ é uma distribuição a priori selecionada, como uma distribuição Gaussiana multivariada com média zero e matriz de covariância de identidade. O codificador prevê a

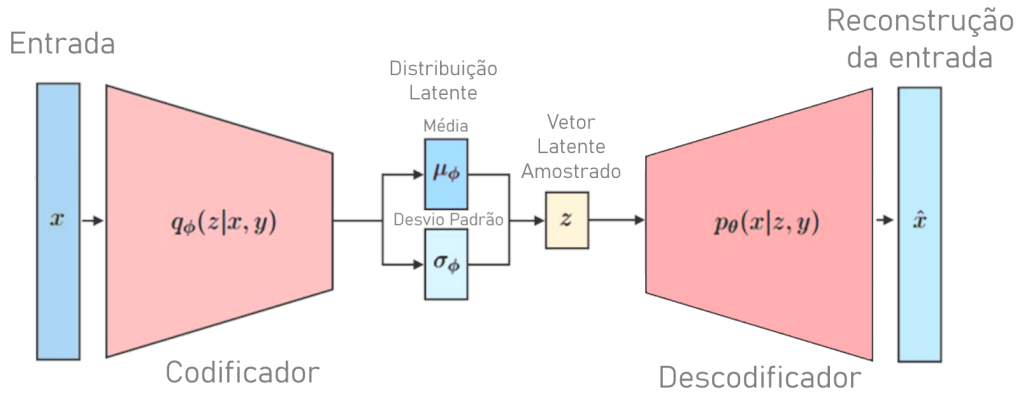


Figura 2.18: Arquitetura de um autocodificador variacional (VAE)

média $\mu_\phi(x)$ e o desvio padrão $\sigma_\phi(x)$ para uma determinada entrada x, e uma amostra latente $\hat{z}$ é extraída de $q_\phi(z|x)$ utilizando o truque de reparametrização: $\hat{z} = \mu_\phi(x) + \sigma_\phi(x) * \epsilon$, onde $\epsilon \sim \mathcal{N}(0,I)$. Ao escolher uma priori gaussiana multivariada, o termo de divergência Kullback-leibler (Shlens, 2014) pode ser calculado analiticamente. O primeiro termo na equação ELBO é normalmente aproximado calculando o erro de reconstrução entre muitas amostras de x e as suas reconstruções correspondentes $\hat{x} = D_\theta(E_\phi(x))$. Novas amostras, não presentes nos dados de treino, podem ser sintetizadas retirando primeiro amostras latentes da prévia, $z \sim p(z)$, e depois retirando amostras de dados de $p_\theta(x|z)$, o que equivale a passar as amostras latentes pelo descodificador, $D_\theta(z)$.

A arquitetura dos VAE permite uma melhor interpolação do que os AE tradicionais porque aprendem um espaço latente contínuo que pode ser facilmente amostrado para gerar novos dados, levando a transições *suaves* nas saídas geradas (Berthelot et al., 2018).

2.3.2 *Autocodificadores Variacionais Condicionais*

O Autocodificador Variacional Condicional (C-VAE) (Sohn et al., 2015) expande o VAE, visando a aprendizagem duma distribuição condicional $p_\theta(x|y)$ onde y representa alguma informação condicionante, como uma classe. Os C-VAE consistem num codificador $q_\phi(z|x,y)$ e num descodificador $p_\theta(x|z,y)$, ambos com a informação condicionante y. Os C-VAE também são treinados para minimizar o ELBO em $\log p_\theta(x|y)$.

A ELBO para C-VAE é semelhante à dos VAE, mas condicionada a y:

$$\mathcal{L}(\theta, \phi, x, y) = \mathbb{E}q\phi(z|x,y)[\log p_\theta(x|z,y)] - D_{KL}(q_\phi(z|x,y)||p(z|y)) \qquad (19)$$

em que $p(z)$ é uma distribuição prévia selecionada. O codificador prevê a média $\mu_\phi(x,y)$ e o desvio padrão $\sigma_\phi(x,y)$ para uma determinada entrada (x,y), e uma amostra do espaço latente $\hat{z}$ é obtida de $q_\phi(z|x,y)$ da seguinte forma: $\epsilon \sim \mathcal{N}(0,I)$ então $z = \mu_\phi(x,y) + \sigma_\phi(x,y) * \epsilon$. O primeiro termo na função de custo é normalmente aproximado calculando o erro de reconstrução, como o erro quadrático médio ou a função de custo de entropia cruzada binária, entre muitas amostras de x e $\hat{x} = D_\theta(E_\phi(x,y))$.

3 TRABALHO RELACIONADO

Este capítulo é dedicado à análise da revisão de literatura que sustenta os dois principais temas abordados nesta tese: a Segmentação de Imagens, que desenvolvemos no Capítulo 4, e a Interpolação de Dados Espaçotemporais, que abordamos adiante no Capítulo 5.

Na Secção 3.1, dedicada à segmentação de imagens, começamos por introduzir o leitor a noções essenciais relativas aos modelos de segmentação apresentados, começamos por descrever as diversas tarefas deste domínio, incluindo segmentação semântica, segmentação de instâncias e segmentação panótica. Nas subsecções seguintes, analisamos as abordagens clássicas de segmentação, modelos de segmentação semântica baseados em CNN, assim como modelos que incluem Transformadores. No fim desta subsecção, abordamos modelos especializados para segmentação semântica de vídeo.

Na Secção 3.2, centrada na interpolação de dados espaçotemporais, fazemos o enquadramento de conceitos de bases de dados espaçotemporais, introduzimos a noção de regiões móveis, e continuamos com a discussão de três abordagens distintas para a interpolação de amostras discretas de regiões bidimensionais: a interpolação McKenney, a interpolação Baseada na Forma e a interpolação com AE .

Na Secção 3.3, apresentamos conjuntos de dados pertinentes ao nosso problema, os quais foram empregados na tarefa de segmentação ou classificação de imagens, ou vídeos de incêndios florestais e outros fenómenos naturais de interesse.

Finalmente, na Secção 3.4 contextualizamos o trabalho dos próximos capítulos e as opções tomadas.

3.1 SEGMENTAÇÃO DE IMAGENS

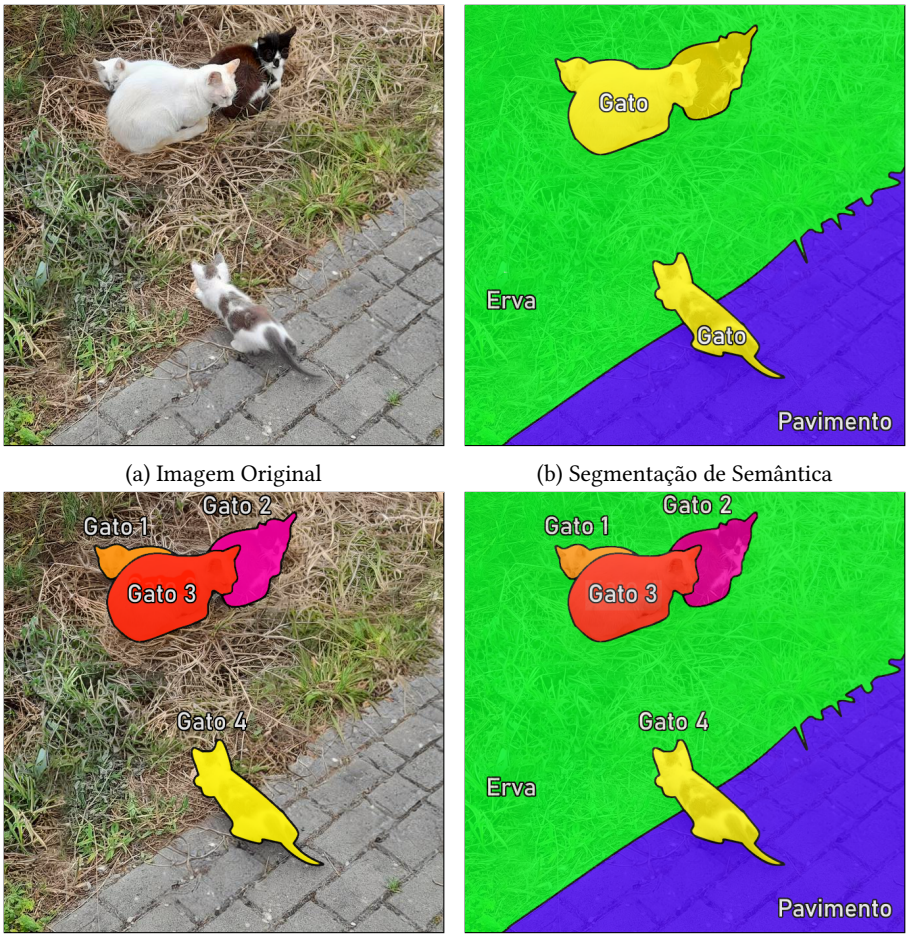
A segmentação de imagens pode ser definida como o processo de decomposição duma imagem em regiões com diferentes características, objetos ou categorias de interesse. Por outras palavras, a finalidade é dividir uma imagem em regiões coerentes que correspondam a estruturas significativas ou objetos de interesse (Gonzalez e Woods, 2008).

Por exemplo, na especialidade de imagiologia médica, as técnicas de segmentação de imagens podem ser usadas para identificar diferentes órgãos ou tecidos (Hesamian et al., 2019) ou anomalias como neoplasias (Kamnitsas et al., 2016). No campo da condução assistida e autónoma, a segmentação pode ser utilizada para identificar peões, veículos e outros objetos e contribuir para permitir uma navegação mais segura (Teichmann et al., 2018). De igual modo, a segmentação é também empregue em aplicações relacionadas com a interpretação de imagens de satélite (Hoeser e Kuenzer, 2020) ou em sistemas de videovigilância (Ojha e Sakhare, 2015), a título de exemplo.

As classes de interesse que se pretende identificar são, muitas vezes, abstrações, produto da compreensão que os humanos tem do mundo que os rodeia e do esforço que fazem para o catalogar e organizar (Rosch, 1978). Com o fim de melhor categorizar as diferentes entidades presentes nas imagens, alguns autores descriminam as partes das imagens em *things* e *stuff* (Caesar et al., 2016). *Things* refere-se a objetos ou instâncias contabilizáveis com partes identificáveis, conformação mais ou menos definida, tal como a categoria *automóvel*, *gato* ou *pessoa*. Em contraste, *stuff* refere-se a regiões amorfas, sem geometria ou partes definidas e não contabilizáveis, relativas ao contexto ou cenário, tal como o *céu*, *mar*, *erva* ou *fumo*.

No que respeita a tarefas de segmentação, dependendo dos requisitos das aplicações, podem ser organizada em três categorias distintas: *segmentação semântica*, *segmentação de instâncias* e *segmentação panóptica* (Kirillov, He et al., 2019), tal como procuramos ilustrar no exemplo da Figura 3.1.

A segmentação semântica consiste na classificação de cada píxel de uma imagem em diferentes classes pré-estabelecidas, como *pavimento*, *erva*, ou *pavimento* (Figura 3.1b), sem distinguir ou contabilizar cada um dos objetos (*things*) e contexto (*stuff*); na Segmentação de Instâncias, o objetivo é prever a classe e o formato da máscara e discriminar cada objeto presente na imagem, de entre o conjunto de classes pré-estabelecidas. Nesta segunda abordagem, cada objeto (*thing*) é identificado, segmentado individualmente e rotulado como sendo uma instância de uma classe particular, mas não se segmenta o contexto (*stuff*), como exemplificado na Figura 3.1c. Por fim, a segmentação panóptica consiste na combinação da segmentação semântica e da segmentação de instâncias. Isto é, para além de se prever a máscara de segmentação de cada objeto (*thing*), procura-se atribuir uma classe a cada um dos pixeis da imagem respeitantes ao contexto ou paisagem *stuff*. Esta abordagem busca fornecer uma visão mais completa e abrangente do cenário da imagem, combinando informações sobre as classes de contexto e assim como de cada instância encontrada (Figura 3.1d).



(a) Imagem Original

(b) Segmentação de Semântica

(c) Segmentação de Instâncias

(d) Segmentação Panóptica

Figura 3.1: **Tarefas de segmentação.** Para este exemplo, consideramos as categorias de objeto "gato" e de contexto "pavimento" e "erva"

3.1.1 *Algoritmos de Segmentação Clássicos*

Nesta secção, começamos por fazer uma breve referência a alguns algoritmos de segmentação *clássicos* e, de seguida, tomando em consideração os requisitos deste trabalho, concentramo-nos nos principais algoritmos utilizados na tarefa de segmentação semântica, destacando os desenvolvimentos mais recentes.

Desde os primórdios da disciplina de Visão Computacional que têm sido propostos métodos de segmentação das imagens (Hanson, 1978). Entre os métodos de segmentação *clássicos* estão os modelos baseados em regiões, que envolvem a divisão da imagem em regiões conectadas que compartilham características específicas, como a cor, a textura ou intensidade (Nameirakpam e Chanu, 2017), modelos baseados em bordas, que envolvem a deteção dos limites entre objetos na imagem através da identificação de alterações abruptas nos valores dos pixeis (Canny, 1986) ou métodos baseados em limiares, que funcionam

através da definição de um valor de limiar para a intensidade dos pixeis e, de seguida, classificam todos os pixeis com intensidade acima desse limiar como pertencentes ao objeto ou região de interesse (Bradley e Roth, 2007; Otsu, 1979), os quais exemplificamos com as imagens da Figura 3.2 comparando a imagem original com os resultado do processamento da imagem pelos algoritmos referidos.

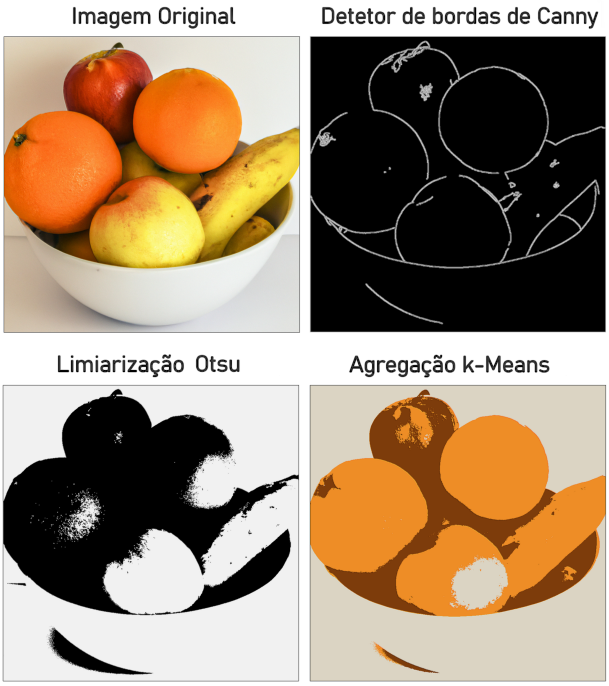


Figura 3.2: Exemplos de resultados de processamento por algoritmos clássicos de segmentação de imagens

3.1.2 *Segmentação Semântica com CNN*

Mais recentemente, os modelos de segmentação semântica baseados em aprendizagem profunda produziram ganhos de desempenho significativos (Ulku e Akagündüz, 2022). Geralmente, esta nova geração de modelos de segmentação baseia-se na arquitetura codificador-descodificador (Figura 3.3), em que o codificador faz a extração de características da imagem de entrada e codifica-a numa representação compacta de baixa resolução, enquanto o descodificador reconstrói os detalhes espaciais e gera os mapas de segmentação com a previsão de classe para cada um dos píxeis (Minaee et al., 2021), tal como esquematizado na Figura 3.3.

Tal como referenciado no Capítulo 2, quando comparado com modelos anteriores, o modelo FCN proposto por Long et al. (2015) alcançou uma melhoria significativa no

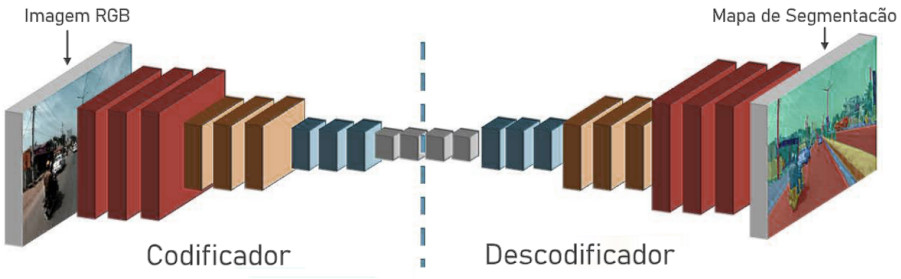


Figura 3.3: Representação simplificada de um modelo de segmentação semântica com arquitetura codificador-descodificador. Adaptado de Baheti et al. (2020)

desempenho e eficiência, no entanto, apresenta limitações, particularmente, no que respeita à resolução espacial das segmentações geradas.

Com o intuito de mitigar essas insuficiências, várias abordagens alternativas foram propostas. O modelo U-Net (Ronneberger et al., 2015), um codificador-descodificador totalmente convolucional, destacou-se ao vencer de forma expressiva uma competição no Simpósio Internacional sobre Imagens Biomédicas (ISBI) de 2015, dedicada à segmentação automatizada de cáries em radiografias. Este feito, aliado à elegância e simplicidade da solução apresentada, estabeleceu a arquitetura U-Net como um marco de referência em diversas áreas de pesquisa. Este modelo tornou-se uma escolha frequente tanto na comunidade biomédica (Çiçek et al., 2016; Zettler e Mastmeyer, 2021a) como numa variedade de outras aplicações no contexto da segmentação semântica (Oliveira et al., 2018; Shamsolmoali et al., 2019).

Muitas vezes os conjuntos de dados de imagens biomédicas têm uma dimensão limitada, o que dificulta o treino dos modelos de aprendizagem profunda. Especificamente, para o desafio de rastreamento de células do ISBI para o qual este modelo foi concebido, Ronneberger et al. (2015) adotaram uma estratégia de treino com aumento de dados[1] que consistiu em deformações elásticas das imagens utilizando interpolação bicúbica para providenciar invariância e robustez adicional ao modelo, gerando 20000 exemplos de treino a partir das poucas dezenas de imagens do conjunto de dados original (Wang et al., 2016).

No que respeita à arquitetura ilustrada na Figura 3.4, em linha com os modelos codificador-descodificador, a U-Net é composta por uma secção codificadora, que extrai e comprime sucessivamente a dimensão dos mapas de características e captura o contexto da imagem, e uma secção descodificadora simétrica cujo propósito é especificar a localização das classes

1 Aumento de dados refere-se ao processo de aplicar transformações variadas aos dados de treino, como rotações, espelhamento ou mudança de contraste, entre outras, visando expandir e diversificar o conjunto de dados disponível para treinar modelos de aprendizagem automática.

de segmentação. À semelhança do modelo FCN, o descodificador utiliza a convolução transposta para sobreamostrar os mapas de características. No entanto, a U-Net faz a ampliação dos mapas de características de forma mais progressiva em vários estágios, ao contrário da FCN.

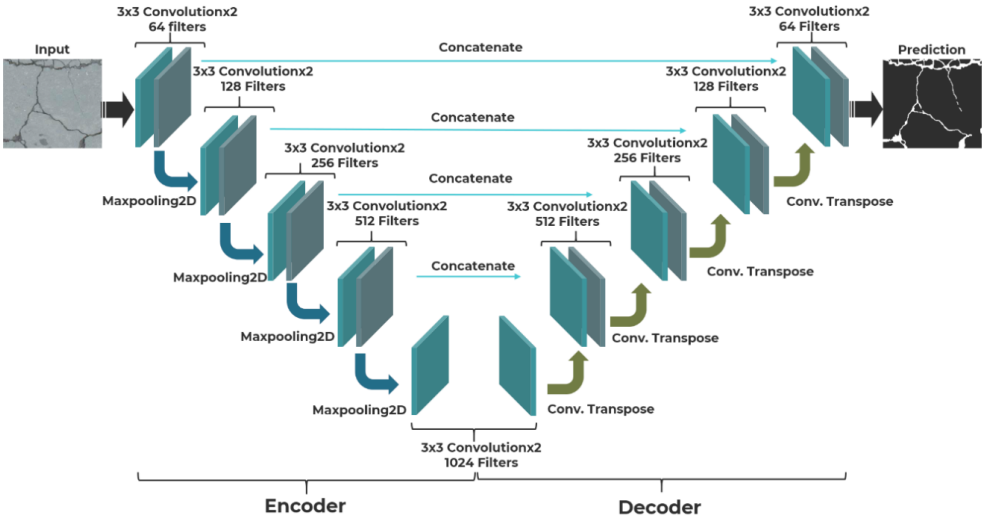


Figura 3.4: **Arquitetura U-Net original**. Este modelo codificador-descodificador totalmente convolucional se destaca por uma progressiva contração dos mapas de características no estágio codificador, através das camadas de *MaxPooling*, e uma expansão simétrica no estágio descodificador mediante operações de convolução transposta. As conexões *skip* permitem preservar detalhes e estruturas de interesse, resultando numa segmentação mais precisa. Adaptado de Eddy e Nagai (2021).

De relevo, a inovação principal introduzida pela U-Net é a cópia e concatenação dos mapas de características de maior resolução da etapa codificadora para a etapa de descodificação, visando preservar a informação de localização através do que se denominam conexões *skip*. Apesar da necessidade de maior alocação de memória devido à replicação dos mapas de características, a combinação das conexões *skip* e o aumento progressivo da resolução resulta em segmentações com limites mais definidos do que o modelo FCN de Long et al. (2015).

A inclusão dos Campos Aleatórios Condicionais, conhecidos como *Conditional Random Fields* (CRF) (Krähenbühl e Koltun, 2011), como uma etapa de pós-processamento, assim como a introdução dos módulos de Atenção Aditiva, originalmente denominados *Attention Gates* (Oktay et al., 2018), representaram outras contribuições significativas que contribuíram para a melhoria do desempenho das arquiteturas FCN.

Outra proposta de arquitetura denominada a Rede de Alta Resolução, em inglês *High-Resolution Net* (HRNet) (Sun et al., 2019), surgiu como outra solução para a manutenção da resolução dos mapas de segmentação. A HRNet destaca-se pela maneira como incorpora

fluxos de convolução de alta resolução a fluxos de resolução mais baixa, conectando esses fluxos multirresolução paralelamente, como ilustrado na Figura 3.5.

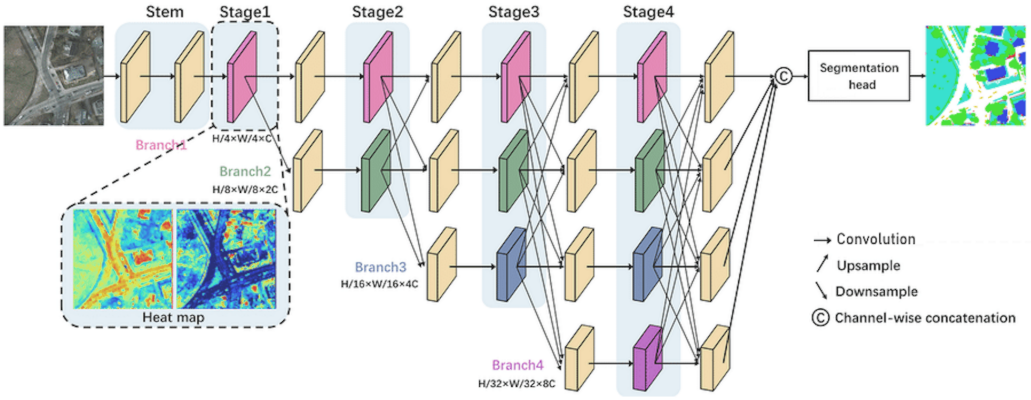


Figura 3.5: **Arquitetura da rede de alta resolução (HRNet).** Adaptada de Z. Xu et al. (2020).

Durante a sua operação, cada fluxo de processamento opera em resoluções distintas, visando a captura de características específicas associadas a cada escala. Simultaneamente, há a partilha dos mapas gerados em diferentes resoluções entre os fluxos de processamento de diversas escalas. Esta integração é possibilitada por meio de operações de convolução com um *stride* igual a 2, que efetuam a sobreamostragem dos mapas, e pela aplicação de interpolação bilinear para incrementar a resolução dos mesmos. Tais fusões garantem não apenas a preservação, mas também a incorporação dos detalhes de alta resolução na representação final da imagem. Esta propriedade torna-se especialmente crucial em tarefas como a segmentação semântica, onde a manutenção dos detalhes nas regiões de transição entre classes é essencial para atingir uma segmentação precisa.

A Rede de Pirâmide de Características, originalmente designada de *Feature Pyramid Network* (FPN) (T.-Y. Lin et al., 2017), foi projetada para a tarefa de deteção de objetos, mas revelou-se também uma solução eficaz para problemas de segmentação semântica.

O modelo FPN, o qual representamos na Figura 3.6, foi concebido com o intuito de para integrar características provenientes de diferentes níveis hierárquicos de uma CNN profunda, de modo a induzir a deteção de objetos e segmentação de objetos em escalas diferentes, isto é, promover a invariância de escala.

Com este fim, a FPN a gera mapas de características de dimensões progressivamente menores, utilizando do extrator de características (por vezes designado na indústria como *backbone*) de uma arquitetura ResNet (Jian et al., 2016). Este fluxo ascendente (*bottom-up*) corresponde à pirâmide do lado esquerdo da Figura 3.6. Por sua vez, a via descendente alucina características de resolução mais elevada mediante a sobreamostragem de mapas de características de menor resolução, mas semanticamente mais fortes, provenientes de

níveis superiores da pirâmide. As conexões laterais estabelecem ligações entre as camadas convolucionais do extrator de características e as camadas convolucionais de níveis mais altos, permitem que a informação de alta resolução percole para os níveis mais baixos da pirâmide. Por seu turno, as conexões descendentes (*top-down*) transmitem informações das camadas de resolução mais alta para as camadas de resolução mais baixa, incorporando detalhes em todas as escalas da pirâmide.

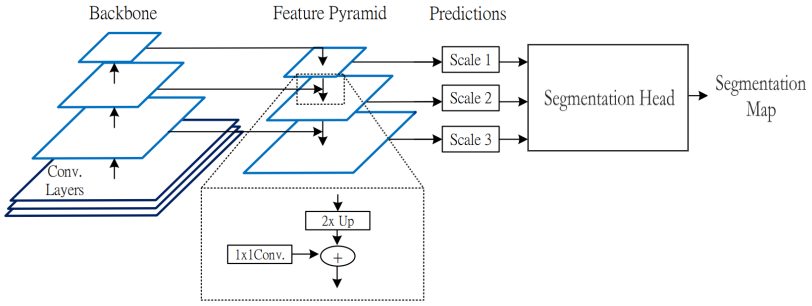


Figura 3.6: **Arquitetura da rede de pirâmide de características (FPN)**. Adaptado de P.-Y. Chen et al. (2019)

Embora tenha sido inicialmente concebido para deteção de objetos, este modelo pode ser adaptado para a tarefa de segmentação semântica, recorrendo ao que se designa de *cabeça de segmentação*. Esta cabeça de segmentação toma como entrada os mapas de diferentes resoluções que provêm da FPN e gera os mapas de segmentação por meio da sua combinação.

3.1.3 *Segmentação Semântica com Transformadores*

Embora os modelos de segmentação semântica que se baseiam em redes convolucionais mantenham a sua relevância, observa-se um aumento significativo no interesse e na proliferação de modelos baseados na arquitetura do Transformador (Vaswani et al., 2017).

Um exemplo representativo dessa variante, o *Segmenter* (Strudel et al., 2021) é um modelo baseado no *Vision Transformer* (ViT) (Dosovitskiy et al., 2020) e aplicado à segmentação semântica. Para o fazer, divide a imagem de entrada em secções, tal como ilustra a Figura 3.7 à esquerda, e processa os vetores de incorporação (ou *embeddings*) como *tokens* de entrada para a secção de codificação do Transformador. A sequência contextualizada de *tokens* produzida pelo codificador é depois sobreamostrada pelo descodificador e convertida em mapas de segmentação com classificações para cada um dos píxeis.

Outro exemplo notável da aplicação de arquiteturas baseadas em transformadores para a segmentação semântica é o Transformador U-Net (UNETR) (Hatamizadeh et al., 2022),

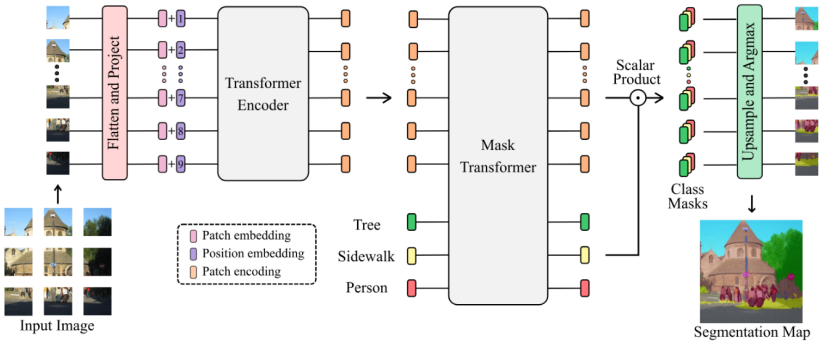


Figura 3.7: **Esquema da arquitetura do Segmenter.** No codificador, as secções da imagem são convertidos numa sequência de incorporações e, de seguida, processados pelo codificador baseado num transformador. A descodificação é feita num transformador máscara que utiliza a saída do codificador e as incorporações de classe como entrada para prever as máscaras de segmentação. Adaptado de Strudel et al. (2021).

focado especificamente na segmentação de imagens médicas em 3D. Inspirado na U-Net de Ronneberger et al. (2015), o UNETR aborda a tarefa de segmentação combinando o viés indutivo da convolução com as vantagens da auto-atenção e atenção cruzada dos Transformadores. Neste contexto, o UNETR utiliza um Transformador na etapa de codificação, permitindo a aprendizagem de representações sequenciais a partir do volume de entrada.

Este desenho permite o modelo a capturar informações globais em várias escalas, o que é essencial para a segmentação de imagens médicas 3D, onde as características podem variar significativamente em profundidade e detalhe. Em particular, módulo de auto-atenção do codificador, permitindo a aprendizagem de representações sequenciais a partir do volume de entrada.

Um dos aspetos distintivos do UNETR é a sua combinação de arquitetura em forma de "U" com um Transformador na etapa de codificação (Figura 3.8). O módulo de auto-atenção do codificador, permite a aprendizagem de representações sequenciais a partir do volume de entrada. Simultaneamente, a atenção cruzada é empregada nas conexões *skip*, e filtram características semânticas irrelevantes, possibilitando uma recuperação espacial precisa na fase de descodificação do UNETR.

3.1.4 *Segmentação Semântica de Vídeo*

As técnicas de segmentação semântica estendem-se ao domínio do vídeo. A estratégia mais simples consiste em aplicar um modelo de segmentação semântica de imagens individuais, fotograma a fotograma. No entanto, esta abordagem ignora a continuidade e a coerência temporal inerentes aos vídeos. Tendo em conta estes fatores, direcionaram-se esforços

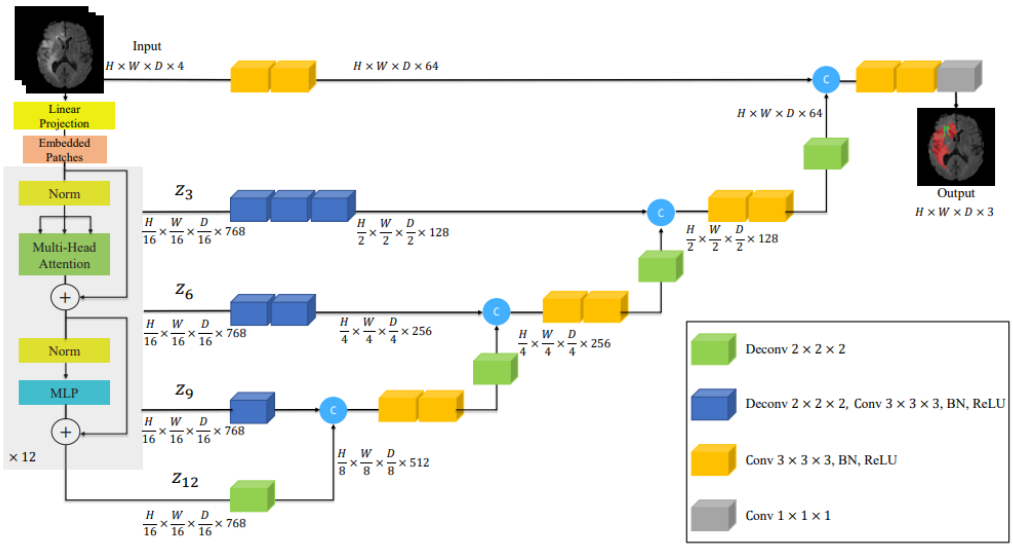


Figura 3.8: **Arquitetura do Transformador U-Net (UNETR).** A UNETR utiliza um esquema de contratação e expansão dos mapas de características análogo à U-Net, sendo a secção codificadora constituída por várias camadas de blocos transformadores, que estão ligados ao descodificador convolucional mediante conexões *skip*. Adaptado de Hatamizadeh et al. (2022)

para criar modelos com conceções alternativas. Os modelos de segmentação específicos para vídeo podem ser divididos, grosso modo, em dois grupos:

- (I) Modelos que privilegiam a precisão
- (II) Modelos que buscam segmentar de forma rápida e eficiente.

Quanto aos algoritmos que se centram na precisão, normalmente aplicam primeiro os algoritmos de segmentação de imagens fotograma a fotograma e depois acrescentam módulos adicionais, como a agregação de características guiada por fluxo ótico (Sistu. et al., 2019) ou por CRF 3D denso (Kundu et al., 2016). Outras abordagens utilizam unidades recorrentes para fundir os resultados de vários fotogramas durante a inferência para melhorar a precisão da segmentação (Fayyaz et al., 2017).

Quanto ao grupo de algoritmos que visam reduzir a sobrecarga computacional, as estratégias geralmente baseiam-se em aproximações das previsões computacionalmente intensivas em cada fotograma, reutilizando características de fotogramas adjacentes (Zhou et al., 2021). Por exemplo, o modelo Clockwork Convnet (Shelhamer et al., 2016) desativa determinadas camadas na rede convolucional e reutiliza características anteriores para diminuir a latência. Zhu et al. (2017) propõem aplicar o fluxo ótico para propagar as características de fotogramas-chave para fotogramas não-críticos. Numa outra abordagem, Liu et al. (2020) introduz um modelo que utiliza uma técnica de destilação de conhecimento baseada na consistência temporal para treinar uma rede compacta que é aplicada a todos

os fotogramas. Estas segundo conjunto de abordagens consegue acelerar o processo de inferência, embora em detrimento de uma menor precisão nas segmentações.

3.2 INTERPOLAÇÃO DE DADOS ESPAÇOTEMPORAIS

Os dados espaçotemporais são frequentemente codificados de forma discreta, associando atributos espaciais, como a localização e a forma, a instantes de tempo. Uma representação contínua típica dos dados de fenómenos espaçotemporais do mundo real é feita mediante regiões móveis (*moving regions*). Formalmente, uma região é um conjunto de segmentos de reta que não se intersectam e que ligam um conjunto distinto de pontos, formando um ciclo fechado que representa as faces externas de um polígono. Uma região pode conter orifícios, que também são delimitados por segmentos de reta num círculo fechado e que, crucialmente, não intersectam as faces externas (Tøssebro e R. H. Güting, 2001). Regiões móveis são tipos de dados abstratos utilizados para descrever no mundo real a evolução espaçotemporal de objetos de interesse, ou seja, como a sua forma e posição mudam ao longo do tempo (Forlizzi et al., 2000). As regiões móveis são descritas como uma série de regiões armazenadas sequencialmente, de modo a que uma região de intervalo (*interval regions*) represente o movimento de um objeto ao longo de um intervalo de tempo entre dois instantes definidos, denominados *slices* (Tøssebro e R. H. Güting, 2001).

3.2.1 *Representação da Evolução de Regiões Móveis*

Tøssebro e R. H. Güting (2001) propuseram uma estrutura na qual regiões móveis podem ser representadas a partir de observações armazenadas em bases de dados espaçotemporais. Este trabalho foi, desde então, expandido por diferentes autores usando os mesmos princípios (José Duarte, Dias et al., 2023; Heinz e R. H. Güting, 2020; Mckennney e Frye, 2015). O objetivo principal é produzir representações que mantenham continuamente a validade topológica, assegurando simultaneamente a coerência com os sistemas de bases de dados espaçotemporais subjacentes. O Problema da Interpolação de Regiões, em inglês *Region Interpolation Problem* (RIP), é o desafio de criar uma regiões móveis a partir de um conjunto de observações. Especificamente, considerando duas observações nos instantes t_1 e t_2, o objetivo é identificar uma função interpoladora f capaz de gerar uma representação válida do objeto em movimento, da sua posição e forma em qualquer ponto temporal entre t_1 e t_2 (José Duarte, B. Silva et al., 2019).

3.2.2 *Algoritmos de Interpolação Espaçotemporal*

Em linha com a *framewwork* de Tøssebro e R. H. Güting (2001), o algoritmo de McKenney (McKenney et al., 2016) cria regiões móveis através da conversão de representações estáticas dos polígonos correspondentes a t_1 a t_2 em listas de segmento de reta em ordem cíclica, designados de segmentos móveis.

O algoritmo mapeia os segmentos da região de origem (t_1) para pontos na região de destino e vice-versa (t_2), garantindo que não haja lacunas entre segmentos móveis adjacentes e que não se sobreponham, exceto nas extremidades de cada segmento onde há necessariamente sobreposição.

O foco de McKenney et al. (2016) está em garantir que, em todos os instantes do intervalo de t_1 a t_2, as regiões mantêm a validade topológica, evitando torções mediante a utilização ângulos de progresso, dividindo a região de origem e a de destino em regiões menores e detetando se concavidades que se intercetam usando um algoritmo baseado no OBBTree (Gottschalk et al., 1996). Uma vez que a região móvel é criada pelo algoritmo, o polígono que representa a região de um instante arbitrário t_i, $t_1 \leqslant t_i \leqslant t_2$ pode ser gerado mediante consulta.

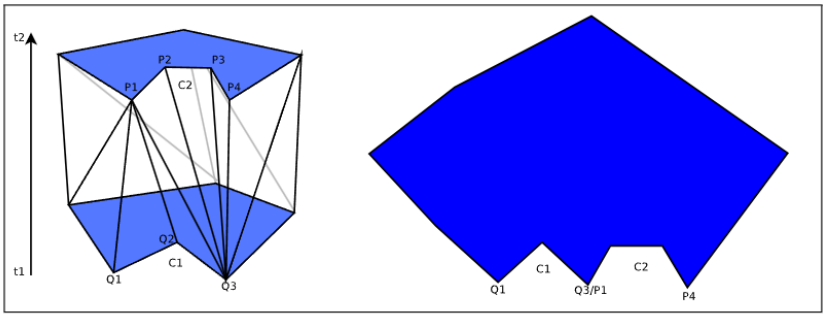


Figura 3.9: **Interpolação McKenney**. Exemplo de região móvel crida pelo algoritmo McKenney. O polígono do lado direito mostra o estado de interpolação a meio do intervalo de tempo $[t_1; t_2]$. Retirado de Heinz e R. Güting (2016)

De relevo, a biblioteca *pyspatiotemporalgeom* (McKenney et al., 2016) é uma implementação em Python que oferece suporte para processamento geométrico de regiões móveis e que permite executar o algoritmo de interpolação McKenney. Este recurso torna a exploração e experimentação mais acessível aos investigadores e programadores interessados no teste e análise deste método para interpolação de dados espaçotemporais.

Foram igualmente sugeridas outras abordagens para a interpolação. Por exemplo, se considerarmos uma região móvel como um poliedro, em que o tempo toma o lugar de uma terceira dimensão (altura) (Heinz e R. H. Güting, 2020), as técnicas utilizadas para interpolar polígonos que representam secções de um objeto volumétrico, como órgãos

humanos em imagens tomográficas, podem ser adaptadas para gerar regiões intermédias em dados espaçotemporais (Ehrhardt et al., 2007; Jiang Li e Narayanan, 2002)

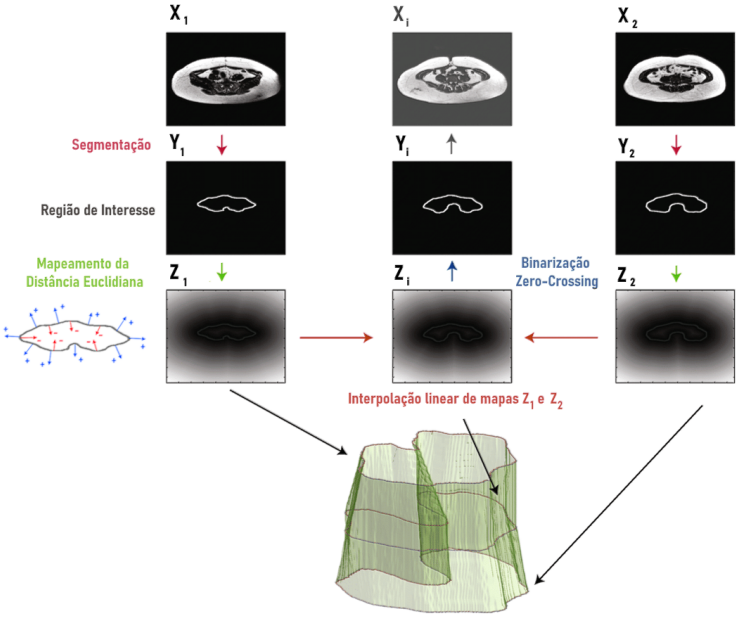


Figura 3.10: **Interpolação baseada na forma.** As imagens x_1 e x_2 são sujeitas a processo de segmentação para gerar as regiões de interesse y_1 e y_2, representando os contornos. Em seguida, é realizado o mapeamento da distância euclidiana aos contornos. Posteriormente, uma interpolação linear é aplicada entre os mapas de distância z_1 e z_2 e, por fim, a região interpolada é reconstruída por meio de um processo de binarização com deteção de pontos de cruzamento zero. Adaptado de Mendez et al. (2020).

A chamada interpolação baseada na forma (do inglês, *Shape-Based*) é um exemplo de um desses algoritmos. Ao contrário dos métodos acima referidos, este algoritmo funciona normalmente com dados rasterizados (Herman et al., 1992; Schenk et al., 2000). A Figura 3.10 representa o processo de criação de um polígono intermédio para um objeto tridimensional utilizando o método descrito por Schenk et al. (2000) e Herman et al. (1992), o qual pode ser adaptado para dados com componente temporal.

Em geral, o processo pode ser descrito numa sequência de passos, como se segue. Sejam x_1 e x_2 as representações 2D que contêm a forma da região nos instantes t_1 e t_2. Para cada imagem selecionada, é gerada uma imagem binária $y_k, k \in \{1,2\}$ através da segmentação da região de interesse. Em seguida, é gerado um mapa de distâncias de nível cinzento $z_k, k \in \{1,2\}$ para cada imagem binária y_k, mapeando a distância euclidiana para o limite da região. Os valores de distância dentro da região são definidos como positivos e os valores de distância fora da região são definidos como negativos. Os mapas z_k são depois reconstruídos mediante interpolação linear ao nível do píxel. A forma da região num determinado ponto arbitrário no tempo $t_i, t_1 < t_i < t_2$ é encontrada através

da identificação dos pontos de intersecção zero (*zero crossing*) dos mapas de distância interpolados. Finalmente, estes contornos geram a região de interesse em t_i.

Uma estratégia alternativa à interpolação de regiões anteriores, envolve a aplicação de algoritmos com a capacidade de aprender as representações dos fenómenos. Neste contexto, os modelos de aprendizagem profunda têm mostrado resultados promissores em várias aplicações de interpolação de imagens. Por exemplo, Cristovao et al. (2020) apresentam uma abordagem para a interpolação de imagens utilizando VAE para aprender representações latentes a partir de dados de duas imagens e gerar uma interpolação, ou seja, criar uma imagem intermediária entre as duas imagens de entrada.

O modelo proposto é composto por três VAE, cada um deles com codificador X e um descodificador X′, com z a corresponder ao espaço latente, como ilustra a Figura 3.11. Ademais, é adicionado o termo z' que representa a média do espaço latente das imagens de entrada, e o hiperparâmetro α, que pondera a importância dada às entradas médias e à representação latente intermediária e penaliza a rede se a imagem gerada se afastar da imagem intermediária real.

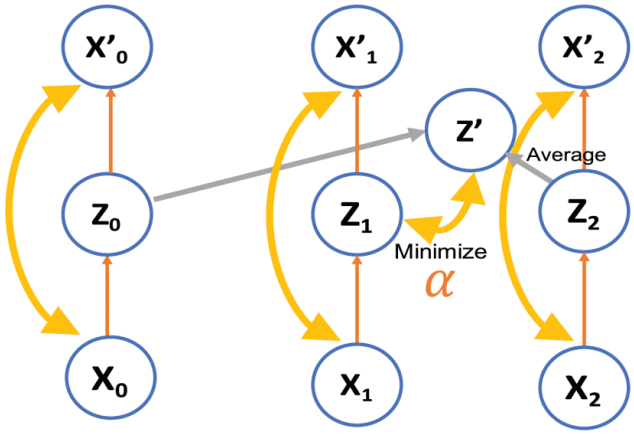


Figura 3.11: **Modelo para interpolação de imagens com VAE.** Retirado de Cristovao et al. (2020).

A rede é treinada com pares de imagens X_0 e X_2 e imagens intermédias X_1. Depois, na fase de inferência, para gerar uma imagem intermédia (interpolação), é calculada a média das representações latentes das redes adjacentes z' e da imagem intermédia real z_1.

Noutra proposta desta categoria, Oring et al. (2021) exploram técnicas de interpolação para imagens *raster* que representam poliedros em diferentes ângulos e outras representações geométricas com objetos deformáveis, interpolando o espaço latente de modelos AE. Da mesma forma, Mi et al. (2021) propõem métodos equivalentes para interpolar imagens contendo objetos tridimensionais em vários ângulos, bem como fotogramas representando

objetos em movimento, usando a interpolação do espaço latente de vários modelos de Variáveis Latentes Gerativas.

Da mesma forma, Mi et al. (2021) propõem métodos equivalentes para interpolar imagens *raster* de objetos tridimensionais em vários ângulos, bem como de amostras representando objetos em movimento, usando a interpolação do espaço latente de vários modelos gerativos.

3.3 CONJUNTOS DE DADOS RELACIONADOS

Foi realizada uma pesquisa bibliográfica focada na identificação de conjuntos de dados relevantes para a segmentação e classificação de imagens e vídeos de incêndios florestais, assim como outros conjuntos de dados relacionados a incêndios e fumo em diferentes contextos. Após este levantamento foi-nos possível compreender e identificar as características e diversidade dos conjuntos de dados de acesso público empregados nessas tarefas. De seguida, apresentamos alguns dos conjuntos de dados que consideramos mais relevantes.

O **FLAME** (Shamsoshoara et al., 2021) é um conjunto de dados recolhido com *drones* durante um fogo controlado num pinhal no Arizona, Estados Unidos. Este conjunto de dados é composto por diferentes repositórios. Um dos repositórios contém 47992 fotogramas RGB anotados (39375 pertencentes ao conjunto de treino e validação e 8617 ao de teste) com uma resolução de 254×254 para o problema de classificar a presença ou ausência de fogo na imagem. O outro subconjunto de dados está relacionado com o problema de segmentação semântica do fogo (Figura 3.12). É composto por 2003 fotogramas e as respetivas máscaras (que indicam a presença de pilhas de madeira a arder) com uma resolução de 3480×2160, igualmente divididos em conjuntos de treino, validação e teste. Para além dos conjuntos anotados, estão também disponíveis dados brutos de vídeo capturados pelos *drones*, os quais foram o produto da recolha feita para os conjuntos de dados acima referidos, bem como três vídeos capturados com câmaras termográficas montadas nos *drones*.

Figura 3.12: **Amostras do conjunto de dados FLAME**. Retirado de (Shamsoshoara et al., 2021)

O **Corsigan Fire** (Toulouse et al., 2017) é um conjunto de dados constituído por 500 imagens RGB de incêndios florestais recolhidas em várias regiões do mundo, 100 imagens multimodais (banda do espetro visível em conjunção com a banda de infravermelho próximo) e cinco sequências de cerca de 30 fotogramas cada de fogos controlados capturados no exterior, igualmente multimodais (espetro visível e infravermelho próximo). Cada imagem está associada à correspondente máscara de segmentação binária que indica a presença ou ausência de fogo. Adicionalmente, cada imagem deste conjunto de dados é anotada utilizando descritores gerais de configuração indicando o modelo da câmara, espetro e afinação, região, data, dados de posicionamento global, entre outros (Figura 3.13).

Figura 3.13: **Amostras do conjunto de dados Corsigan Fire**. Retirado de (Toulouse et al., 2017)

O **FESB MLID** (Braovic et al., 2017) consiste num conjunto de dados acessível ao público fornecido pela Faculdade de Eng. Elétrica, Eng. Mecânica e Arquitetura Naval de Split (FESB), Croácia, que contém 400 imagens RGB de resolução variável (200 para treino e 200 para teste) da paisagem natural mediterrânica, juntamente com as respetivas mascaras de segmentação de referência anotadas manualmente. As segmentações têm 12 categorias diferentes, que incluem smoke, clouds and fog, sun and light effects e outras classes relacionadas com elementos da paisagem captada. Para além deste conjunto de dados, o Centro de Investigação sobre Incêndios Florestais da FESB também disponibiliza no seu sítio eletrónico (Jakovcevic e Krstinic, 2010) dois conjuntos de dados adicionais: um conjunto de dados para segmentação semântica do fumo constituído por 218 imagens RGB com uma resolução de 720×576, contidas em 4 sequências de vídeo, e um segundo conjunto de dados também com 115 imagens RGB contendo fumo e respetivas máscaras de segmentação anotadas em 3 classes definidas como smoke, maybe smoke e no smoke (Figura 3.14).

O **DeepFire** (Khan et al., 2022) é outro conjunto de dados para classificação binária que consiste numa compilação de imagens de incêndios florestais recolhidas da Internet, em que a classe fire contém imagens de florestas e montanhas com chamas visíveis ou colunas de fumo resultantes de fogo, e a classe no-fire contém imagens de florestas e montanhas verdes em diferentes ângulos sem a presença de fogo. O conjunto de dados DeepFire tem

Figura 3.14: **Amostra do conjunto de dados FESB MLID**. À esquerda uma imagem com fumo, à direita, a segmentação correspondente. Diferentes tons de cinzento do mapa de segmentação correspondem a diferentes classes. Retirado de (Braovic et al., 2017)

um total de 1900 imagens RGB de resolução 250×250, em que 950 imagens são da classe fire e as restantes 950 imagens pertencem à categoria no-fire.

Fora do universo dos incêndios exclusivamente florestais, o conjunto de dados **BoWFire** (Chino et al., 2015) é constituído por imagens de incêndios em situações de emergência, como edifícios em chamas, incêndios industriais, acidentes de viação e incêndios provocados por motins. Com 226 imagens RGB com resoluções variáveis (119 imagens que contêm fogo e as restantes imagens que consistem em emergências sem fogo visível e outras com propriedades semelhantes ao fogo, como o pôr do sol e objetos vermelhos ou amarelos) recolhidas manualmente por peritos humanos, este conjunto de dados foi reunido para o problema de classificação de imagens. O BowFire contém também um segundo conjunto de dados constituído por 240 imagens com uma resolução de 50×50 pixeis: 80 imagens classificadas como fire, 80 como non-fire e 80 como sendo da categoria smoke. O BoWFire foi incluído numa compilação de conjuntos de dados de emergências envolvendo incêndios e fumo que se designou de FiSmo (Mirela T Cazzolato et al., 2017). Para além do BoWFire, esta compilação é constituída por 5 outros conjuntos de dados: 3 deles para classificação de imagens e 2 para classificação de vídeo. Um destes conjuntos de dados, o **SmokeBlock** (Mirela T. Cazzolato et al., 2016), é composto por 832 imagens anotadas como contendo fumo (classe smoke) e 834 sem fumo (classe non-smoke), todas elas extraídas do Flickr. Além disso, relativamente aos conjuntos de dados de imagens, o FiSmo é ainda composto pelo conjunto de dados **Flickr-FireSmoke**, com 5556 anotados com as classes fire and smoke, only fire, only smoke e none. Quanto aos dados de vídeo, um outro conjunto de dados que compõe o FisMo, o **FireVid** (Avalhais et al., 2016), foi recolhido a partir do YouTube, sendo constituído por 27 vídeos de diferentes resoluções (320×240 a 600×336 pixeis) e um total de 83675 fotogramas anotados com as categorias fire, not-fire e ignore (fogo pouco percetível ou conteúdo não útil). Este trabalho utiliza

também 61 vídeos (com 29895 fotogramas anotados) produzidos no âmbito do projeto RESCUER (Villela et al., 2018) com uma resolução que varia entre 320×240 e 1920×1080 pixeis, anotados manualmente seguindo o mesmo protocolo.

Uma abordagem muito utilizada para rastrear a evolução de áreas queimadas em incêndios florestais e estudar os seus efeitos é por meio do uso de imagens hiperespectrais recolhidas por satélite. Neste âmbito, muitos trabalhos recentes extraem e processam dados publicamente disponíveis para identificar a área ardida (Chuvieco et al., 2020; Al-Dabbagh e Ilyas, 2022; Gaveau et al., 2021; Luca et al., 2022), no entanto, a taxa de amostragem e a resolução das imagens captadas são de uma ordem de grandeza diferente do objeto de estudo deste trabalho. Outros autores utilizam imagens captadas por satélite para rastrear e segmentar de objetos deformáveis como icebergs (Barbat et al., 2019; Koo, 2021), monitorizar outros fenómenos naturais como erupções vulcânicas e a dispersão das nuvens de cinzas (Guerrero Tello et al., 2022; Wilkes et al., 2022) ou para fazer a segmentação de nuvens para determinação da nebulosidade (Xie et al., 2020), por exemplo.

3.4 CONSIDERAÇÕES FINAIS

O vídeo do fogo florestal, capturado por *drone*, que referenciamos na introdução (ver Figura 1.1) e no qual nos debruçamos nos capítulos seguintes, foi objeto em trabalhos anteriores. Nesses estudos, R. L. C. Costa, Miranda e Moreira (2020) desenvolveram-se uma abrangente *framework* para visualização, refinamento, operação com diversos algoritmos de interpolação de regiões móveis 2D, assim como a disponibilização de várias métricas de medição do desempenho.

Fazendo proveito desse extenso trabalho, o ponto de partida e desafio inicial desta dissertação, foi um conjunto de dados que inclui o vídeo da progressão do fogo associado a polígonos representando a área ardida. Porém, após análise e experimentação, e mesmo após tentativas infrutíferas de limpeza e filtragem, estas representações revelaram-se demasiado ruidosas e pouco representativas da evolução da área ardida. Esta limitação levou-nos a considerar outra via e a descartar este o conjunto de dados inicial. Estabelecemos que seria necessário um novo conjunto de dados curado e validado, tão representativo da evolução da área ardida quanto possível.

Este novo conjunto dados permitiu-nos, mais tarde, validar e treinar modelos de segmentação baseados em aprendizagem profunda reconhecidamente eficientes, e abriu a possibilidade de explorar novos modelos de interpolação 2D baseados em aprendizagem automática.

Neste contexto, no Capítulo 4, que se segue, descrevemos detalhadamente a construção de um novo conjunto de dados e desenvolvemos de modelos convolucionais para segmentação semântica automática. Depois, no Capítulo 5, exploramos vários métodos de interpolação de regiões móveis 2D, incluindo métodos de aprendizagem automática.

4 SEGMENTAÇÃO SEMÂNTICA

Neste capítulo descrevemos a segmentação semântica em fogos florestais e expomos os métodos e modelos utilizados para abordar este problema. Especificamente, trabalhamos no sentido de fornecer ferramentas para testar e validar modelos de segmentação de área ardida no contexto de incêndios florestais.

Desta forma, (**I**) apresentamos o **BurnedAreaUAV**, um novo conjunto de dados anotado manualmente com o propósito de segmentar áreas ardida em fogos florestais. Este conjunto de dados é destinado ao treino e avaliação de modelos de segmentação em vídeos de incêndios, e toma como ponto de partida trabalhos anteriores (R. L. C. Costa, Miranda, Dias et al., 2020b; R. L. C. Costa, Miranda, Dias et al., 2021; R. L. C. Costa, Miranda e Moreira, 2020; R. L. d. C. Costa e Moreira, 2022) e expande-os. Este conjunto de dados baseia-se num vídeo de um incêndio controlado capturado por um *drone* com um sensor RGB no norte de Portugal. A filmagem é caracterizada por períodos em que o fumo e as chamas obstruem a área de interesse, o que torna a segmentação da área ardida mais complexa. Ademais, (**II**) avaliamos modelos de segmentação de imagens baseados em aprendizagem profunda, com uma abordagem de aprendizagem supervisionada, e determinamos o desempenho de modo a servir de referência para trabalhos futuros. Simultaneamente, (**III**) sugerimos uma métrica de consistência temporal simples, mas específica, para validar polígonos de áreas queimadas não anotadas gerados pelos modelos de segmentação nos fotogramas consecutivos do vídeo, e avaliamos cada um dos modelos utilizando esta métrica de consistência temporal nos dados não anotados.

De seguida, na Secção 4.1, descrevemos a criação e validação do conjunto de dados BurnedAreaUAV. Na Secção 4.2, explicamos as métricas utilizadas para avaliar os modelos de segmentação automática. Na Secção 4.3 apresentamos os modelos testados, o fluxo de trabalho da experiência. Na Secção 4.5, apresentamos os resultados da aplicação dos modelos de segmentação automática nos conjuntos de dados, e produzimos a análise do desempenho e da qualidade das segmentações obtidas. Finalmente, na Secção 4.6 discutimos os resultados obtidos e concluímos o capítulo apontando os desafios em aberto e o trabalho futuro no que respeita à segmentação semântica da área ardida em fogos florestais.

4.1 CONJUNTO DE DADOS BURNEDAREAUAV

Tendo-se identificado a falta de conjuntos de dados de vídeos capturados por *drone* que retratem a evolução da área ardida e a dinâmica de um foco de incêndio, propomos criar as ferramentas de apoio à avaliação comparativa para testar e validar modelos de segmentação neste contexto. Para o efeito, criamos um conjunto de dados para a segmentação semântica da área ardida, constituído por fotogramas de vídeo e máscaras de segmentação, propomos as métricas de avaliação e obtemos resultados de modelos de aprendizagem profunda para servir como referência para trabalhos futuros. A recolha de dados, a anotação e a organização dos ficheiros resultantes são igualmente descritas nesta secção.

4.1.1 *Coleção de Dados*

O conjunto de dados BurnedAreaUAV deriva de um vídeo capturado na Torre do Pinhão, concelho de Sabrosa, distrito de Vila Real no norte de Portugal, nas coordenadas de latitude 41° 23' 37,56"e longitude -7° 37' 0,32"e altitude aproximada de 730 metros acima do nível médio do mar (ver Figura 4.1 à esquerda). A zona de estudo encontra-se condicionada pela orografia local e solos rochosos de origem granítica, com um estrato arbóreo esparso (sobretudo pinheiro-bravo e carvalho) e um predomínio de vegetação arbustiva rasteira (Direção-Geral do Território, 2019), composta por giestais e urze, caraterística da região em causa (Abreu et al., 2004).

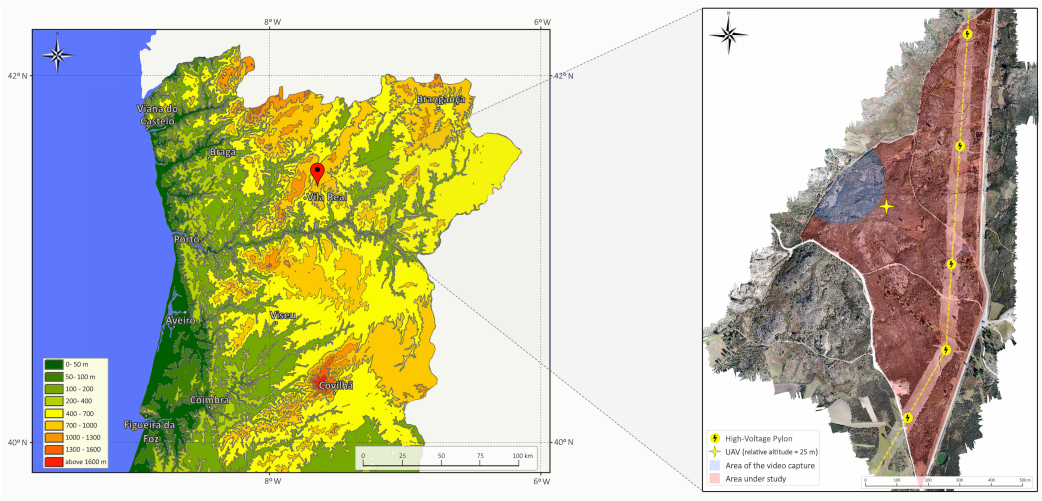


Figura 4.1: **Localização da área de estudo**. À esquerda está o mapa hipsométrico do norte de Portugal, indicando a posição relativa da área de estudo. À direita está a ortofoto da área de estudo, destacando a região específica onde foram efetuados os fogos controlados. Retirado de T. F. Ribeiro et al. (2023)

Trabalhos anteriores utilizaram este vídeo como base para estudos de simulação da evolução espaçotemporal de fenómenos (R. L. C. Costa, Miranda, Dias et al., 2020b; R. L. C. Costa, Miranda, Dias et al., 2021; R. L. d. C. Costa e Moreira, 2022). No entanto, a anotação, processamento, bem como utilização de modelos de aprendizagem profunda para a segmentação semântica destes dados descritos de seguida, constituem trabalho original.

4.1.2 *Segurança e Orientações durante a Coleção de Dados*

Com efeito, a captação destas imagens inseriu-se numa campanha de fogos controlados cujo objetivo foi limpar o terreno, evitar a acumulação excessiva de matéria orgânica combustível e reduzir o risco de incêndios florestais. O vídeo selecionado corresponde à propagação de uma das frentes de fogo.

Quanto aos critérios que orientaram a seleção deste local, optou-se por uma localização relativamente plana para garantir uma visão clara do terreno, em detrimento de outros fogos controlados na mesma região onde a orografia do terreno era menos favorável. A acessibilidade também foi um fator importante, uma vez que era essencial ter uma área adequada disponível para as frequentes e imprescindíveis mudanças de bateria do *drone*.

Uma vez no local, foram estabelecidas diretrizes gerais para o posicionamento do *drone*. O primeiro objetivo passou por evitar grandes oclusões causadas pelo fumo. Embora a direção predominante do vento ditasse direção de propagação do fumo, mudanças ocasionais exigiram o ajuste da posição do *drone*. Não obstante, é importante notar que, para o vídeo utilizado na criação do conjunto de dados, o *drone* foi mantido numa posição quase estacionária durante a recolha de dados. Além disso, para garantir um controlo preciso, um operador experiente manobrou manualmente o *drone* e operou as câmaras. A segunda prioridade foi captar imagens da frente de fogo, assegurando que nenhuma parte da área de interesse fosse quase totalmente obstruída pela altura das chamas. Outro fator tido em consideração foi o fumo proveniente de áreas circundantes recentemente queimadas ou ainda em chamas. Graças ao posicionamento estratégico do *drone*, foi possível aterrar em segurança sem problemas de maior causados pela má visibilidade ou quaisquer outras dificuldades decorrentes da toxicidade. Por fim, foram tomadas precauções para evitar quaisquer riscos potenciais colocados por linhas de alta tensão nas proximidades (Figura 4.1, à direita).

O fogo controlado foi iniciado em três locais dentro da área designada (aproximadamente 430.000 m^2), definindo um triângulo. A sua propagação foi controlada por três equipas de bombeiros encarregadas de fazer convergir as três frentes para o centro do triângulo. Assim,

a propagação do incêndio foi determinada, em pequena medida, por condições naturais como a topografia, a temperatura ou o vento. Localizada a cerca de 2,5 km a sul-sudeste do local, a estação meteorológica da Torre do Pinhão registou uma velocidade média horária do vento de 2 m/s com uma direção de 158 graus (12h a 13h) e sem precipitação ao longo do dia. A estação meteorológica da Mina de Lajes, localizada a 8 km a norte do local, registou uma humidade relativa horária de 54 % e uma temperatura média horária de 14,6 ℃ (12h a 13h) (SNIRH, 2019).

4.1.3 *Características e Metadados do Vídeo Capturado*

O vídeo, com aproximadamente 15 minutos, foi capturado durante esta operação no dia 1 de março de 2019, com início por volta das 12 horas e 20 minutos. No início da sequência de vídeo, uma parte significativa do campo de visão do sensor do *drone* já está queimada, e a área ardida expande-se com o passar do tempo, como representado na Figura 4.2.

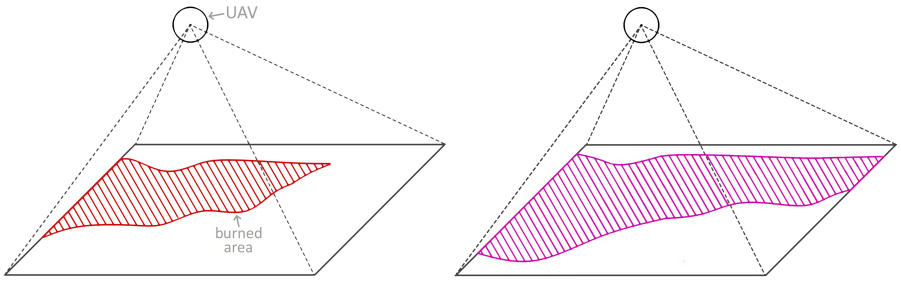


Figura 4.2: **Representação simplificada do campo de visão do *drone* ao longo do vídeo**. À esquerda, uma representação da área queimada no início do vídeo. À direita, uma representação da área queimada no final do video

Foi utilizado um *drone* DJI Phantom 4 PRO equipado com uma câmara FC6310S RGB. Concomitantemente, foram recolhidos metadados relativos à posição aproximada do *drone*, incluindo a altitude e as coordenadas GPU, a orientação do *drone* e da câmara, entre outros, extraindo a informação armazenada em fotografias tiradas no decorrer do ensaio. Os dados geoespaciais relacionados com a localização do fogo controlado são armazenados num ficheiro Keyhole Markup Language (KML), que pode ser lido por *software* como o Google Earth e outras ferramentas geoespaciais. Também fornecemos duas ortofotos de alta resolução da área de interesse antes e depois da queima.

A taxa de amostragem do vídeo é de 25 fotogramas por segundo, totalizando 22500 imagens. O vídeo original foi armazenado em H.264 (ou MPEG-4 Parte 10) com uma resolução de 720×1280. Não foi recolhido qualquer sinal áudio. No total, o ficheiro ocupa 222 megabytes.

Esta gravação foi a base para a anotação manual da área queimada que descrevemos na secção seguinte. Para além dos dados processados e segmentados, disponibilizamos também são os dados em bruto da versão original do vídeo num repositório de Zenodo de acesso livre: https://zenodo.org/record/7944963.

4.1.4 *Processo de Anotação de Dados*

Nesta secção, descrevemos o processo de anotação e validação do conjunto de dados, conforme representado a um nível elevado pela Figura 4.3. Antes da anotação, extraímos os fotogramas do vídeo usando as funções disponíveis na biblioteca OpenCV (Bradski, 2000), mantendo a resolução original. De seguida, efetuámos a segmentação semântica utilizando a ferramenta de anotação de imagens Labelme (Russell et al., 2008), que permite desenhar intuitivamente polígonos de segmentação com um rato de computador.

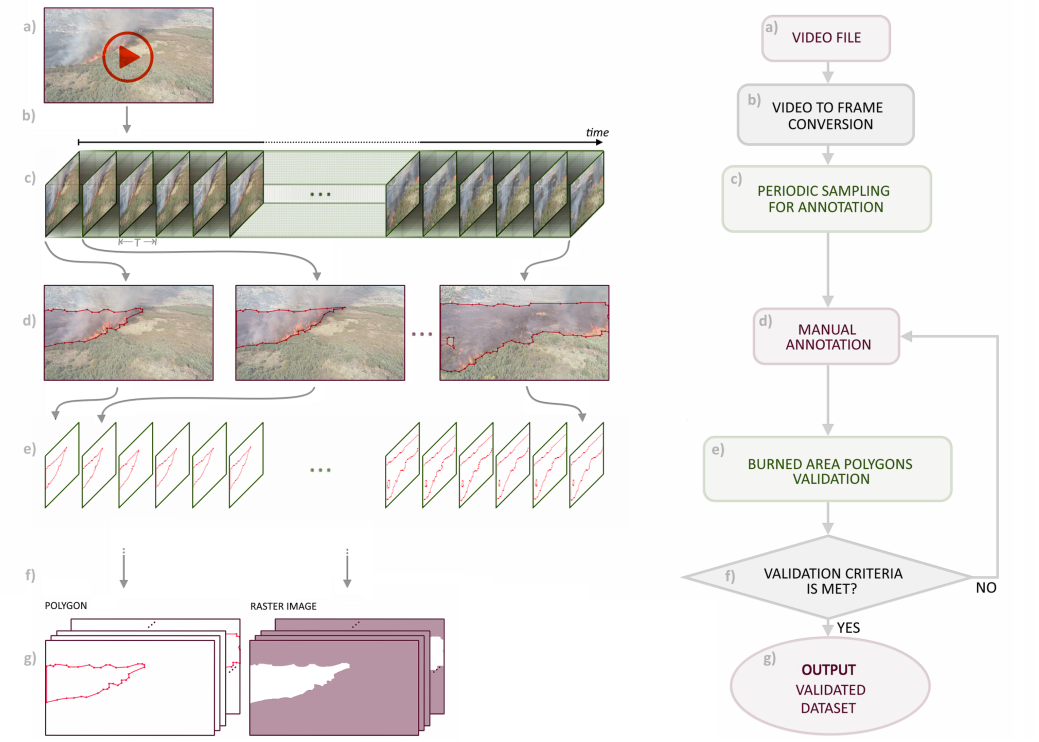


Figura 4.3: **Processo de anotação e validação do conjunto de dados**. O processo começa com a extração de fotogramas individuais do ficheiro de vídeo (*a* a *b*). Posteriormente, é efetuada uma amostragem periódica, descartando as imagens intermédias (*c*). A anotação manual é então cuidadosamente realizada, identificando a área queimada (*d*). As anotações passam por um processo de validação (*f*) e, se cumprirem os critérios de validação, são guardadas em ficheiros WKT, JSON e PNG (*g*). Retirado de T. F. Ribeiro et al. (2023)

Anotações de Referência

Para este problema, consideramos duas classes: burned_area e unburned_area. Procurámos definir a segmentação de forma a eliminar, tanto quanto possível, a componente subjetiva. Nesse sentido, estabelecemos regras para a anotação. Começámos por definir (**I**) a classe burned_area como a totalidade da área onde ardeu o fogo, independentemente do grau de carbonização ou do tempo de atividade do fogo numa determinada área e não considerando eventuais projeções do fogo, o fumo, o volume de vegetação ou combustível consumido pelas chamas, mas apenas a superfície do terreno ardido. O unburned_area é toda a área que não cumpre este critério. Em alguns casos, o fumo e as chamas podem introduzir oclusões que colocam desafios à segmentação precisa de determinados fotogramas amostrados do vídeo. Para remediar esta dificuldade, definimos que (**II**) se não for possível ter a certeza de que uma determinada área foi consumida pelo fogo, esta não é considerada ardida. Adicionalmente, como medida de último recurso, (**III**) utilizamos fotogramas anteriores do vídeo para estabelecer os limites da área ardida, sempre de forma conservadora, ou seja, não consideramos uma área como ardida, exceto se tenha a certeza disso.

Esta anotação manual foi feita com uma periodicidade de 4 segundos (ou frequência de 0,25 Hz), ou que corresponde a anotar a cada 100 fotogramas, como representado na Figura 4.4.

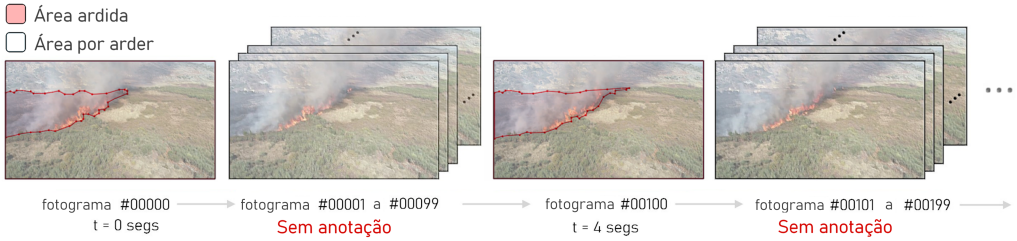


Figura 4.4: **Representação esquemática da anotação periódica dos fotogramas do vídeo.** A anotação foi efetuada a cada 100 fotogramas, o que corresponde a um período de amostragem de 4 segundos. São consideradas duas classes: área ardida (burned_area) e área por arder (unburned_area). Esta anotação foi efetuada para toda a duração do vídeo.

4.1.5 *Validação do Conjunto de Dados*

Pese embora a nossa diligência, o processo de anotação não está isento de erros. Em alguns casos, não é possível definir com precisão o limite da segmentação devido a oclusões mais ou menos graves de fumo e fogo. Como tal, é necessário validar a segmentação manual e garantir que os fotogramas sucessivos são consistentes. Independentemente da dinâmica

complexa da propagação dos incêndios florestais, algumas propriedades simples podem ser definidas para identificar polígonos de segmentação pouco prováveis.

Em trabalhos anteriores, R. L. C. Costa, Miranda, Dias et al. (2021) propõem estabelecer regras de consistência para remover *outliers* espaçotemporais em polígonos de segmentação de área ardida. Recorremos a essas propostas para validar os polígonos de segmentação resultantes da nossa anotação manual. Sintetizamos as regras da seguinte forma:

I. Cada $\text{polígono}_t, t \in \{1,2,...,T\}$, produto da segmentação da sequência de T fotogramas amostrados, cuja área é menor do que a área do polígono inicial da área queimada $A_{\text{polígono}_{t=1}}$ ou maior do que a área do polígono do último instante do vídeo $A_{\text{polígono}_{t=T}}$ é considerado um *outlier* (Figura 4.5).

Esta regra decorre do pressuposto de que, num incêndio, a área ardida é monotónica crescente.

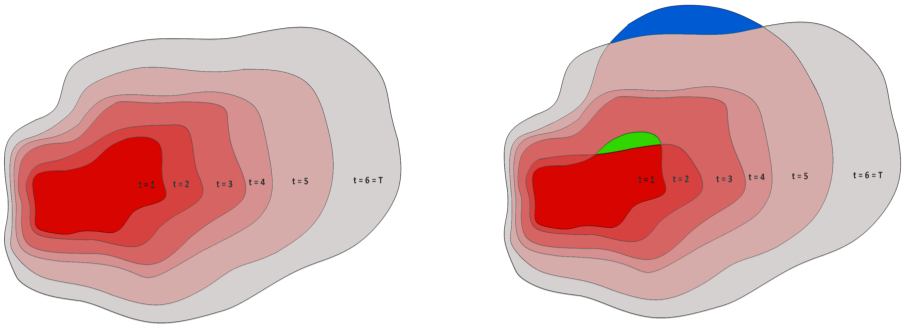


Figura 4.5: **Representação simplificada da evolução da área ardida de um foco de incêndio e exemplos de polígonos de segmentação inconsistentes**. À esquerda, uma representação consistente da evolução da área ardida: a área está a aumentar e os polígonos anteriores estão contidos nos polígonos seguintes. À direita, o mesmo diagrama com polígonos inconsistentes: a verde, uma instância na qual o polígono subsequente ($t = 2$) não contém o primeiro polígono ($t = 1$); a azul, uma instância na qual o polígono intermédio ($t = 5$) excede os limites do último polígono ($t = 6 = T$). Retirada de T. F. Ribeiro et al. (2023)

Ademais, como segunda regra define-se:

II. Qualquer $\text{polígono}_t, t \in \{2,3,\ldots,T-1\}$ cujo Interseção sobre União (IoU) entre o primeiro polígono de segmentação $\text{polígono}_{t=1}$ ou o último polígono de segmentação $\text{polígono}_{t=T}$ seja inferior ao IoU entre o último polígono $\text{polígono}_{t=T}$ e o primeiro $\text{polígono}_{t=1}$ constitui um *outlier*.

em que IoU representa a Intersecção sobre a União, conforme descrito na secção 4.2. Isto significa que o IoU será mínimo entre os polígonos de segmentação inicial e final

e qualquer polígono cujo IoU seja inferior a este limiar é descartado, assumindo que o primeiro e o último polígono de segmentação são representações fidedignas da área ardida.

III. Qualquer $\texttt{polígono}_t, t \in \{2,3,\ldots,T\}$ que suceda ao $\texttt{polígono}_{t=1}$ inicial tem de conter o polígono inicial na sua geometria. Da mesma forma, todos os $\texttt{polígono}_t, t \in \{1,2,\ldots,T-1\}$ que precedem o polígono final $\texttt{poligono}_{t=T}$, têm de estar contidos neste. Se uma destas condições não for satisfeita, este polígono intermédio é um *outlier* (Figura 4.5).

Decorrente da regra III, os indicadores valor de Inconsistência Relativa à Representação Inicial $IR_{inicial}$ e valor de Inconsistência Relativa à Representação Final IR_{final} são definidos como diferenças geométricas (R. L. C. Costa, Miranda, Dias et al., 2021):

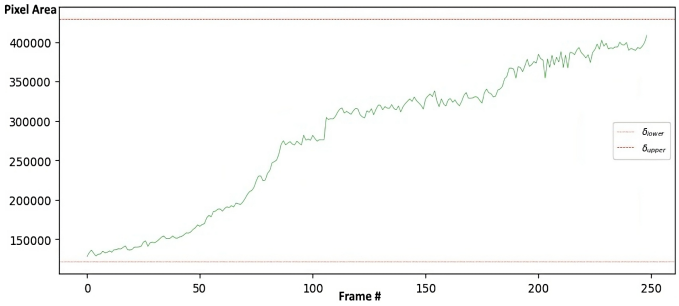
$$IR_{inicial} = \frac{A_{\texttt{polígono}_{t=1}} - A_{\texttt{polígono}_t}}{A_{\texttt{polígono}_{t=1}}}, \forall t \in \{1,2,...,T\} \tag{20}$$

$$IR_{final} = \frac{A_{\texttt{polígono}_t} - A_{\texttt{polígono}_{t=T}}}{A_{\texttt{polígono}_{t=T}}}, \forall t \in \{1,2,...,T\} \tag{21}$$

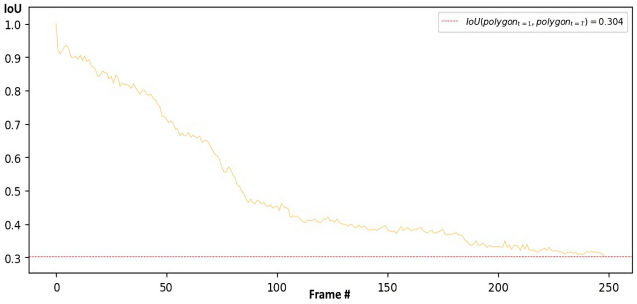
onde $IR_{inicial}$ e IR_{final} medem as variações relativas da área dos polígonos de segmentação em relação ao polígono inicial e final, respetivamente.

Estas regras de consistência forneceram a base para a validação dos polígonos de segmentação anotados manualmente. Para verificar se os polígonos são consistentes, traçámos os gráficos que correspondem à área ardida (Figura 4.6a), calculámos o IoU de cada polígono relativamente ao primeiro (Figura 4.6b) e último polígono de segmentação (Figura 4.6c), bem como o $IR_{inicial}$ (Figura 4.6d) e IR_{final} (Figura 4.6e). Uma vez que estamos a manipular dados de um fenómeno inerentemente ruidoso, são considerados níveis de tolerância para os limiares de rejeição de polígonos. Para a área do polígono, é definida uma tolerância de 5% δ_{area} para os limiares inferior e superior. Da mesma forma, é definida uma tolerância de δ_{IoU} de 5% para os IoU. Finalmente, $\delta_{rv} = 10\%$ é definido para os indicadores $IR_{inicial}$ e IR_{final}. Os valores de tolerância são estipulados empiricamente, em associação com o que se entende ser adequado para gerar polígonos relativamente consistentes.

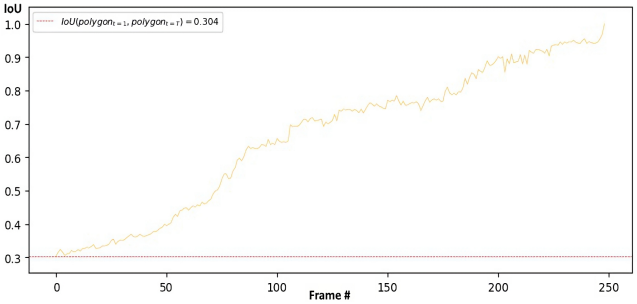
Todos os polígonos que não obedecem a estes limiares são considerados de qualidade insuficiente e, portanto, rejeitados. Nesta circunstância, o anotador volta a analisar a segmentação produzida na ferramenta de anotação manual (LabelMe) e verifica se é possível fazer ajustes que estejam em conformidade com os princípios estabelecidos nesta secção. Uma vez feitos os ajustes, os polígonos de segmentação são novamente avaliados. Este processo iterativo de ajuste é repetido até que todas as condições de validação sejam



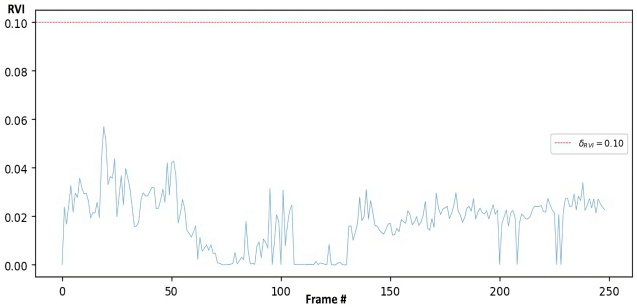
(a) Área dos polígonos da área ardida.



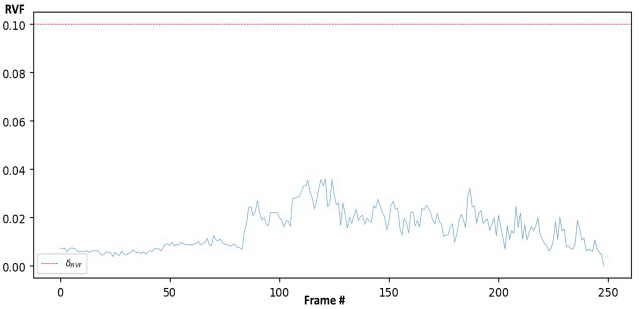
(b) IoU dos polígonos intermediários em relação ao polígono inicial.



(c) IoU dos polígonos intermediários em relação ao polígono final.



(d) Valor de Inconsistência Relativa à Representação Inicial ($\mathrm{IR}_{inicial}$).



(e) Valor de Inconsistência Relativa à Representação Final (IR_{final}).

Figura 4.6: **Gráficos das regras de consistência das anotações produzidas.**

satisfeitas. Durante a construção deste conjunto de dados, foi necessário fazer ajustes apenas em alguns polígonos para atender a todos os critérios de validação.

4.1.6 *Divisão de Dados*

O conjunto de dados é composto por um total de 249 fotogramas e máscaras de segmentação correspondentes. O subconjunto considerado para treino e validação contém 226 pares fotograma-máscara (≈ 90%), enquanto o subconjunto de teste contém 23 pares (≈ 10%). Cada um dos fotogramas de treino e validação foi gerado através da amostragem de um fotograma a cada 4 segundos (corresponde a uma periodicidade de 100 fotogramas), começando no fotograma inicial e terminando no fotograma número 22500. Os fotogramas e as anotações do conjunto de teste têm a mesma taxa de amostragem, mas são deslocados dos 50 fotogramas (2 segundos) dos fotogramas de treino e validação para evitar sobreposições. No caso dos fotogramas de teste, começam no fotograma número 20250 e terminam no fotograma número 22450, que corresponde à parte final do vídeo.

4.1.7 *Organização dos Ficheiros do Conjunto de Dados*

Cada fotograma segmentado resultou num ficheiro JavaScript Object Notation (JSON) com vários campos, como ilustrado na Figura 4.7, de aproximadamente 1,6 megabytes cada. Este ficheiro contém a imagem na resolução original (720×1280) e os pontos do polígono de segmentação que corresponde à área queimada. Tudo o que não está contido no polígono da área ardida corresponde à área não ardida.

Os polígonos de segmentação e as imagens correspondentes contidas no ficheiro JSON foram posteriormente convertidos em imagens *raster* Portable Network Graphics (PNG) de canal único e em polígonos guardados no formato Well-known text (WKT). Os ficheiros PNG estão organizados em duas pastas, uma para as imagens e outra para as máscaras de segmentação. As imagens têm o formato `frame_<frame_num>.png` e as máscaras `mask_<frame_num>.png`. Como exemplo, o par fotograma-máscara da segunda segmentação produzida corresponde aos ficheiros `frame_000100.png` e `mask_000100.png`, produzidos a partir do fotograma número 100 do vídeo original. Quanto aos polígonos WKT, são armazenados num ficheiro em que cada linha corresponde a uma amostra da sequência, como mostra a Figura 4.7. Para além dos fotogramas e polígonos de segmentação, fornecemos também ortofotos de alta resolução (antes e depois dos incêndios), bem como os ficheiros KML e fotografias adicionais. O conjunto de dados pode ser consultado e descarregado aqui https://zenodo.org/record/7944963.

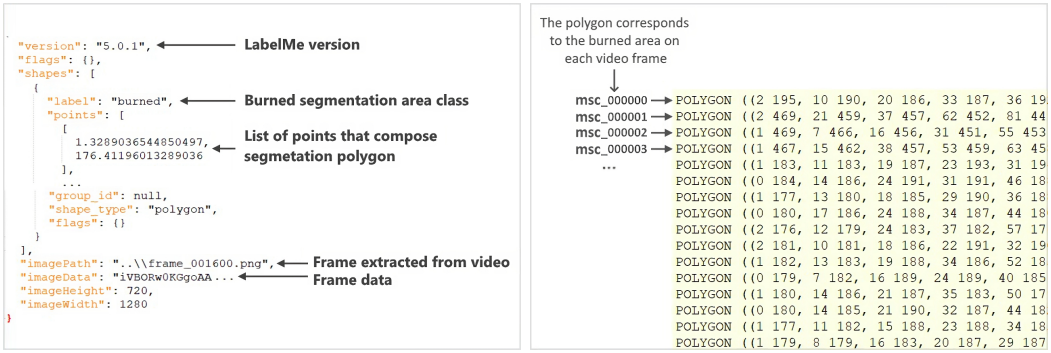


Figura 4.7: **Estrutura dos ficheiros JSON e WKT gerados**. À esquerda, o ficheiro JSON contém os pontos do polígono de segmentação na chave points e os dados da imagem a partir da qual a segmentação foi gerada no campo ImageData. À direita, o arquivo WKT armazena os pontos do polígono de segmentação. Cada linha de corrida corresponde a um fotograma. Adaptado de T. F. Ribeiro et al. (2023)

Após a construção do conjunto de dados, seguiu-se a fase experimental. Nesta secção, descrevemos os métodos de experimentação, os modelos testados e os resultados obtidos.

4.2 MÉTRICAS DE AVALIAÇÃO

Para este estudo, o desempenho da segmentação de áreas ardidas foi analisada tendo em conta as máscaras de segmentação de referência produzidas manualmente. Neste contexto, este estudo utiliza algumas das métricas clássicas para medir a precisão dos problemas de segmentação de imagens.

4.2.1 *Métricas Clássicas de Classificação*

Dado que lidamos com um problema de classificação binária de píxeis, a Precisão e a Recuperação podem ser definidas como:

$$\text{Precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{22}$$

$$\text{Revocação} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

em que TP se refere ao número de Verdadeiros Positivos, FP de Falsos Positivos e FN ao número de Falsos Negativos. Geralmente, as medidas de Precisão e Recuperação não são consideradas isoladamente. A combinação destas duas medidas pode ser representada pela média harmónica de Precisão e Recuperação, designada de Medida F1 e definida por:

$$\text{Medida F1} = \frac{2 \cdot \text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{24}$$

4.2.2 *Índice de Jaccard*

A Interseção sobre União (IoU) ou Índice de Jaccard é uma das métricas mais comuns na segmentação semântica para medir a similaridade entre diferentes amostras. O IoU é definido como a intersecção entre o mapa de segmentação previsto e o mapa de segmentação de referência, dividido pela união entre a segmentação prevista e o mapa de segmentação de referência.

$$\mathrm{IoU} = J(A,B) = \frac{A \cap B}{A \cup B} \tag{25}$$

em que A representa o mapa de referência e B simboliza o mapa de segmentação previsto (Figura 4.8).

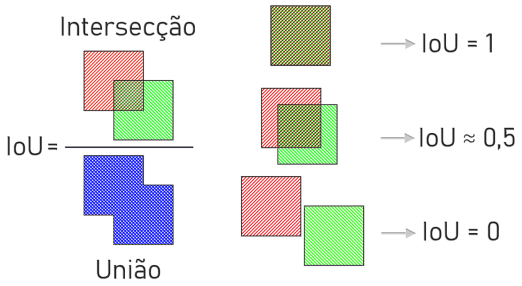


Figura 4.8: **Representação esquemática do Índice de Jaccard.** Quando há total coincidência entre polígonos, IoU é máximo (=1). Na pior das hipóteses, se polígonos forem totalmente disjuntos, o IoU é 0.

Um modelo de segmentação semântica de vídeo deve ser capaz de produzir previsões consistentes em fotogramas consecutivos, dado que numa sequência de vídeo, dependendo da taxa de amostragem, haverá uma variação limitada de fotograma para fotograma, especialmente se se tratar de um fenómeno com uma progressão relativamente lenta, tal como um incêndio florestal. A geometria que pretendemos captar é a área ardida, pelo que podemos estabelecer alguns pressupostos simples inerentes: à semelhança do que foi discutido na Secção 4.1.5, sabemos que para um mesmo foco de incêndio, uma zona que se estabelece como ardida não pode deixar de o ser num momento posterior. Da mesma forma, sabemos que a área ardida nunca diminui.

Com estas considerações em mente, começamos por afirmar que F é o modelo de segmentação semântica cuja consistência temporal queremos avaliar e denotamos um fotograma individual do vídeo por $x \in I^{H \times W \times C}$, onde x uma imagem com altura H, largura W, e $C = 3$ canais (RGB), e $I = 1,2,\ldots,255$ representa a intensidade do píxel de cada canal. O nosso objetivo é processar uma sequência de fotogramas $x_1^T = (x_1,\ldots,x_T)$ de tamanho T, em que x_t representa um fotograma da sequência no tempo $t \in 1,2,\ldots,T$. $\hat{y}_t = F(x_t)$ denota a previsão do modelo, em que $\hat{y}_t \in S^{H \times W}$ e $S \in [0,1]$ é o espaço de

resultados num problema de segmentação binária com classes que significam burned_area = 1 e unburned_area = 0.

Definimos com a métrica de consistência temporal TC entre dois fotogramas contíguos contendo polígonos representativos da área ardida como:

$$TC = \frac{\text{polígono}_{\hat{y}_t} - \text{polígono}_{\hat{y}_{t+1}}}{\text{poligono}_{\hat{y}_{t+1}}}, \forall t \in \{1, 2, ..., T\} \tag{26}$$

onde $\text{polígono}_{\hat{y}_t}$ e $\text{polígono}_{\hat{y}_{t+1}}$ representam os polígonos previstos pelo modelo em instantes consecutivos. Quando $TC = 0$, as previsões do modelo para fotogramas sucessivos apresentam uma consistência temporal perfeita, de acordo com a métrica. Pelo contrário, um valor mais elevado de TC indica uma menor consistência temporal entre os dois fotogramas.

Da mesma forma, se for desejada uma métrica única para toda a sequência, a média da consistńcia temporal entre todos os fotogramas consecutivos da sequńcia pode ser definida por:

$$mTC = \frac{1}{1 - T} \sum_{t=2}^{T} TC \tag{27}$$

em que T é o número de fotogramas do vídeo.

4.3 MODELOS DE APRENDIZAGEM PROFUNDA AVALIADOS

A arquitetura U-Net (Ronneberger et al., 2015) é amplamente conhecida e compreendida pela comunidade de visão computacional, em geral, e provou ser altamente eficaz numa série de aplicações de segmentação semântica, tal como em aplicações de imagiologia médica, teledeteção (Shamsolmoali et al., 2019), deteção de defeitos de fabrico (L. Cheng et al., 2023) ou segmentação do *layout* de documentos (Markewich et al., 2022). Por esse motivo, como linha de base, optámos por testar modelos de segmentação semântica baseados na U-Net e na U-Net 3D (Çiçek et al., 2016).

Como primeira variante, seguindo quase na íntegra a arquitetura originalmente proposta por Ronneberger et al. (2015), delineámos uma rede U-Net como mostra a Figura 4.9. O número de filtros por camada é idêntico ao da rede original, assim como a parametrização das camadas convolucionais e as suas funções de ativação. Não adicionamos nenhuma camada de *Dropout* (Srivastava et al., 2014) ou Regularização em Lote (Ioffe e Szegedy, 2015) para treino. Designamos esta variante como **U-Net Base**.

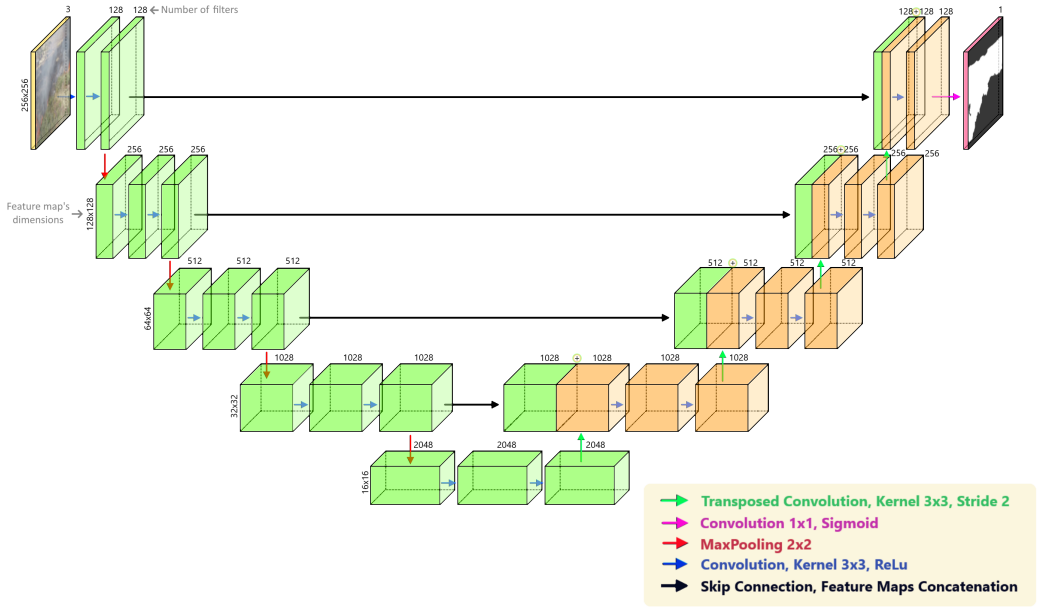


Figura 4.9: **Arquitetura U-Net utilizada.** O modelo é composto por um codificador, que extrai características espaciais da imagem, seguido de um descodificador que constrói o mapa de segmentação a partir das características codificadas. A secção codificadora é formada por uma sequência de camadas convolucionais consecutivas (2 ou 3) com filtros de dimensões 3×3, seguidas de uma camada Maxpooling de dimensões 2×2 e stride 2. Este bloco de convoluções e Maxpooling é repetido 4 vezes, reduzindo a resolução do mapa de características e aumentando o número de filtros à medida que a resolução diminui. Uma sequência de 2 camadas convolucionais seguida de uma camada convolucional transposta (stride 2) faz a transição da secção de codificação para a secção de descodificação e inicia a expansão dos mapas de características. Por outro lado, o descodificador é composto por camadas convolucionais com filtros de dimensões 3×3 e camadas convolucionais transpostas que duplicam sucessivamente as dimensões dos mapas de características e reduzem para metade o número de filtros. Este bloco é repetido 4 vezes. Finalmente, os mapas de segmentação são produzidos por uma camada de convolução com um filtro de dimensão unitária 1×1 e função de ativação sigmoide, por oposição às restantes camadas convolucionais com a função de ativação ReLU (Unidade Linear Retificada). Finalmente, o mapa de probabilidade gerado pela função sigmoide é binarizado nas categorias burned_area = 1 unburned_area = 0. Retirado de T. F. Ribeiro et al. (2023)

Em segundo lugar, propomos um modelo idêntico à variante U-Net Base, com a diferença de que apenas recebe como entrada o canal vermelho das imagens. A lógica subjacente a esta variação, a que chamamos **U-Net RED**, é a hipótese de que, nas zonas das imagens onde há atividade de fogo, a intensidade do valor do canal vermelho é superior à dos canais azul e verde (Kim et al., 2014; Mouelhi et al., 2020). Uma análise empírica das imagens do conjunto de dados corrobora esta afirmação, como mostra o exemplo da Figura 4.10. Com este exercício, pretendemos avaliar se o desempenho tem quebras significativas e se faz sentido considerar a hipótese de descartar os canais azul e verde como uma forma simples de compressão de dados em aplicações como a segmentação da área ardida. Este modelo, a que chamamos **U-Net RED**, segue a mesma arquitetura da Figura 4.9, à exceção do número de canais de entrada, com um limiar de decisão de 0,5.

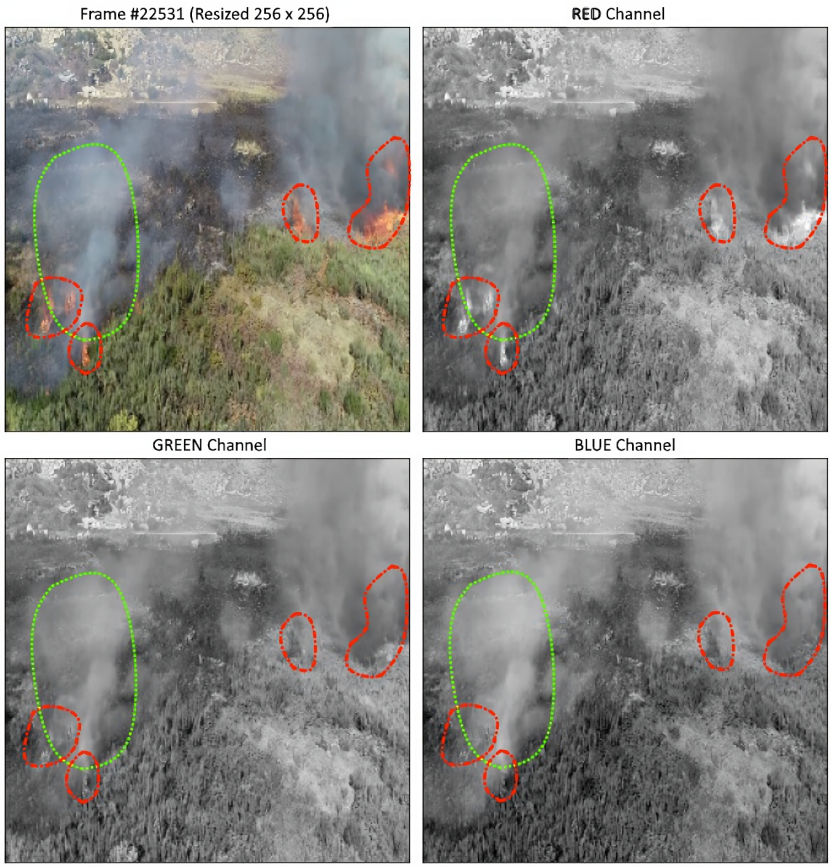


Figura 4.10: **Imagem do vídeo e de cada um dos canais RGB**. Uma coluna de fumo é destacada a verde e as áreas com chamas a vermelho. Empiricamente, observamos que o canal vermelho parece preservar a informação relativa à parte luminosa das chamas e, simultaneamente, as zonas de fumo parecem ser menos visíveis. Retirado de T. F. Ribeiro et al. (2023)

Alguns autores exploram a utilização de convoluções 3D, incorporando-as em modelos com convoluções 2D (Carreira e Zisserman, 2017; Mahadevan et al., 2020). A intuição é que as convoluções 3D são capazes de extrair características espaçotemporais capturando uma

representação de múltiplos fotogramas adjacentes e propagando-a através das camadas do modelo (Hou et al., 2019). Especificamente, uma camada convolucional 3D padrão é semelhante à sua contraparte bidimensional, mas recebe uma dimensão adicional N como entrada, que no nosso caso representa o número de fotogramas consecutivos. O mapa de características de entrada F da camada convolucional 3D tem dimensões de $H \times W \times N \times C$, representando a altura H, a largura W, o número de fotogramas N e os canais de entrada C. Gera um mapa de características de saída G com dimensões de $H \times W \times N \times O$, em que O representa o número de canais de saída. Em comparação com as convoluções 2D padrão, os filtros convolucionais 3D introduzem uma dimensão adicional no filtro K, com dimensões de $K_h \times K_w \times K_t$. Aqui, K_h, K_w e K_t correspondem às dimensões espacial e temporal do filtro.

Seguindo essa linha de pensamento, também avaliamos uma variante da U-Net que usa convoluções 3D, com base na arquitetura proposta em Çiçek et al. (2016), que denominamos **U-Net 3D**. Originalmente utilizada para segmentação de dados volumétricos, a U-Net 3D tem uma arquitetura semelhante à da U-Net, exceto que tem uma entrada tridimensional e processa estas entradas com operações convolucionais 3D, Maxpooling 3D e camadas de convolução transpostas 3D. Para o nosso cenário, adaptamos o modelo para uma entrada sequencial, convertendo o eixo z no eixo do tempo, ou seja, uma amostra (fotograma) corresponde a uma fatia do volume de um exame de ressonância magnética, por exemplo. A Figura 4.11 mostra os detalhes da arquitetura implementada. Tal como no modelo 2D U-Net, não adicionámos quaisquer camadas para regularização durante o treino.

Os modelos que incorporam convoluções 3D são frequentemente dispendiosos do ponto de vista computacional devido às operações de convolução ao longo do eixo temporal num grande número de fotogramas empilhados. As restrições de memória limitam o número de fotogramas N a um máximo de 16. O número de filtros em cada nível do codificador e do descodificador também teve de ser substancialmente reduzido, quando comparado com o modelo U-Net da Fig. 4.9. O nosso objetivo é encontrar uma solução de compromisso entre as capacidades de extração de características e a capacidade de captar a representação temporal.

Para treinar o modelo, dividimos o vídeo em sequências de 16 pares contíguos de máscara de fotograma. Criámos amostras com valores de sobreposição de 2, 4, 8 e 16 e treinámos modelos diferentes para cada um dos quatro valores de sobreposição (ver Figura 4.12). De todas as sobreposições, a que obteve o melhor desempenho médio nas métricas de avaliação (Secção 4.2) foi a que tinha sobreposição = 8. Consequentemente, consideramos esta configuração para comparação de modelos.

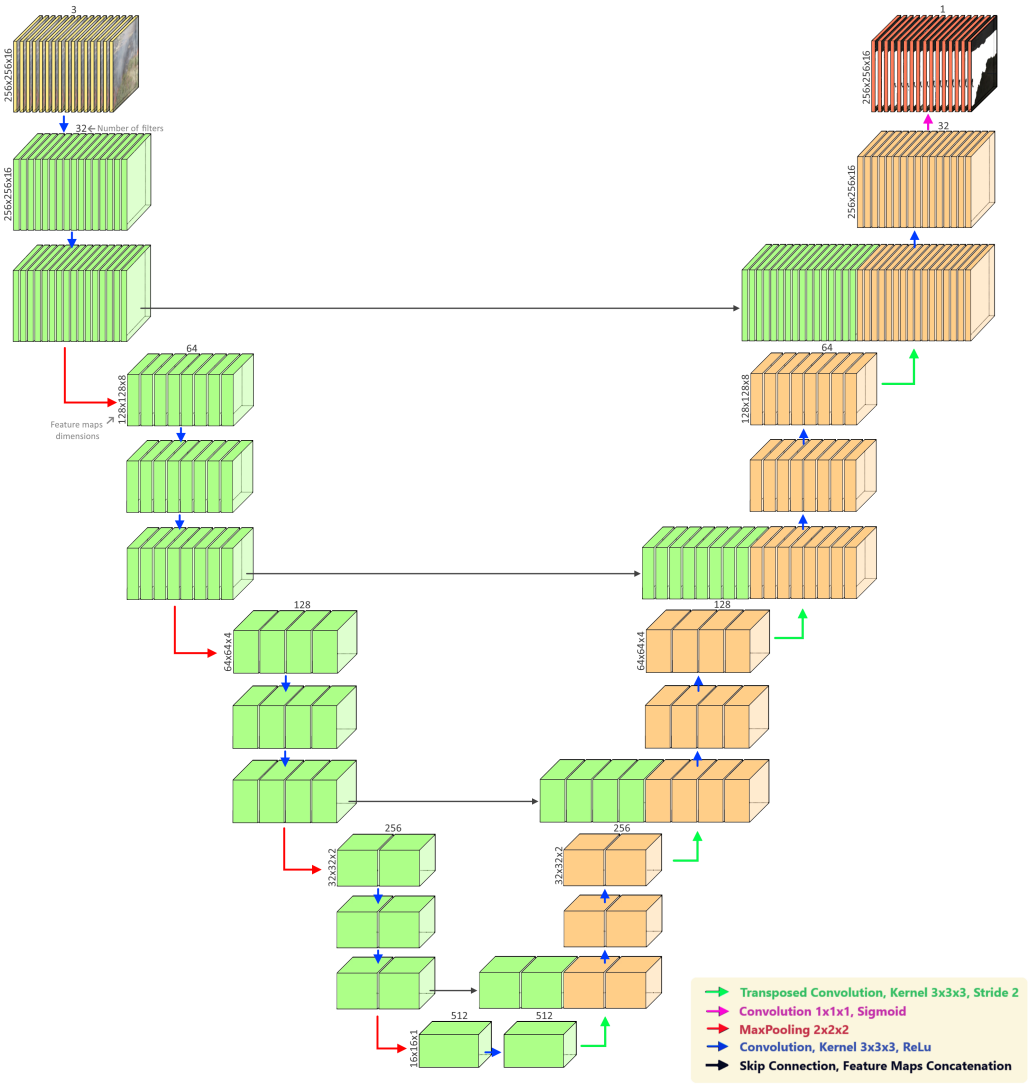


Figura 4.11: **Arquitetura da U-Net 3D utilizada**. Semelhante ao modelo U-Net 2D, este modelo é um codificador-descodificador. A principal diferença está no número de dimensões de entrada e na aplicação de camadas convolucionais 3D, Maxpooling 3D e camadas convolucionais transpostas 3D. A U-Net 3D recebe como entrada um bloco de 16 imagens RGB consecutivas de dimensão 256×256. A secção codificadora é formada por sequências (2 ou 3) de camadas convolucionais com filtros de dimensões $3 \times 3 \times 3$ seguidas de Maxpooling 3D $2 \times 2 \times 2$ e stride 2. Tal como na U-Net 2D, este bloco de convoluções e Maxpooling é repetido 4 vezes e culmina em duas camadas convolucionais que fazem a transição para a secção do descodificador. O descodificador espelha o codificador, mas utiliza camadas convolucionais transpostas com filtros $3 \times 3 \times 3$ e stride 2 para aumentar progressivamente a resolução do mapa de características. A camada final é uma convolução de $1 \times 1 \times 1$ com uma função de ativação sigmoide, gerando o mapa de probabilidade de segmentação. As setas pretas representam a cópia e a concatenação de mapas de características entre o codificador e o descodificador. Retirado de T. F. Ribeiro et al. (2023)

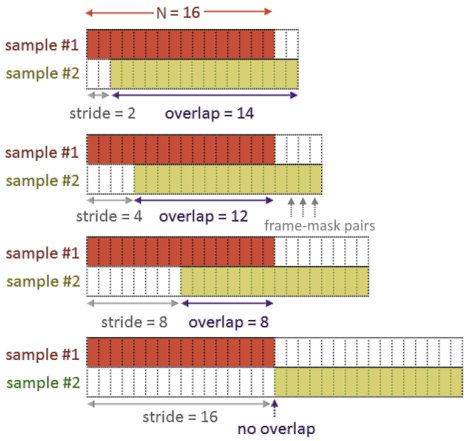


Figura 4.12: **Representação das várias strides temporais e sobreposições testados para o modelo U-Net 3D**. A linha superior representa um caso com uma sobreposição maior (14), enquanto a linha inferior mostra um exemplo sem sobreposição. A combinação ótima de sobreposição e passo que produziu os melhores resultados foi determinada como sendo $\mathtt{sobreposição} = \mathtt{stride} = 8$. Para simplificar, exibe-se apenas as duas primeiras amostras. Retirado de T. F. Ribeiro et al. (2023)

4.4 EXPERIÊNCIA

As experiências foram realizadas num computador com o sistema operativo Windows 10, com 32 GB de RAM, um processador Intel i7-10700K e uma placa gráfica Nvidia GeForce RTX 3090. A *framework* utilizado para desenvolver e testar os modelos de segmentação foi o Keras-TensorFlow, e o código foi desenvolvido em Jupyter Notebooks na linguagem de Python.

Antes do treino e da validação, foi necessário pré-processar os dados para torná-los compatíveis com os modelos. Tanto os fotogramas selecionados como as máscaras correspondentes foram redimensionados para uma resolução de 256×256 e as imagens foram normalizadas dividindo os valores de píxeis por 255, resultando num mapa de píxeis com valores entre 0 e 1. Esta operação de normalização é um passo de pré-processamento comum no treino de modelos de redes convolucionais. Ela garante que cada entrada tenha uma distribuição de dados semelhante, o que ajuda na convergência do modelo (Huang et al., 2023). Além disso, para o modelo U-Net RED, os canais azul e verde das imagens RGB foram descartados, resultando numa imagem de canal único (Fig.4.13 A.1). As mesmas etapas de pré-processamento foram aplicadas durante a fase de inferência para todos os fotogramas do vídeo (Fig.4.13 A.1). Para avaliar o desempenho dos modelos, utilizámos o conjunto de dados BurnedAreaUAV, bem como a totalidade do vídeo capturado pelo *drone*. Primeiramente, aplicámos o método de validação cruzada com 3 partições nos dados BurnedAreaUAV. Para cada partição, registámos e calculámos a média das métricas Revocação, Precisão, Medida F1 e IoU (Fig.4.13 B.1). Após a avaliação inicial, cada modelo

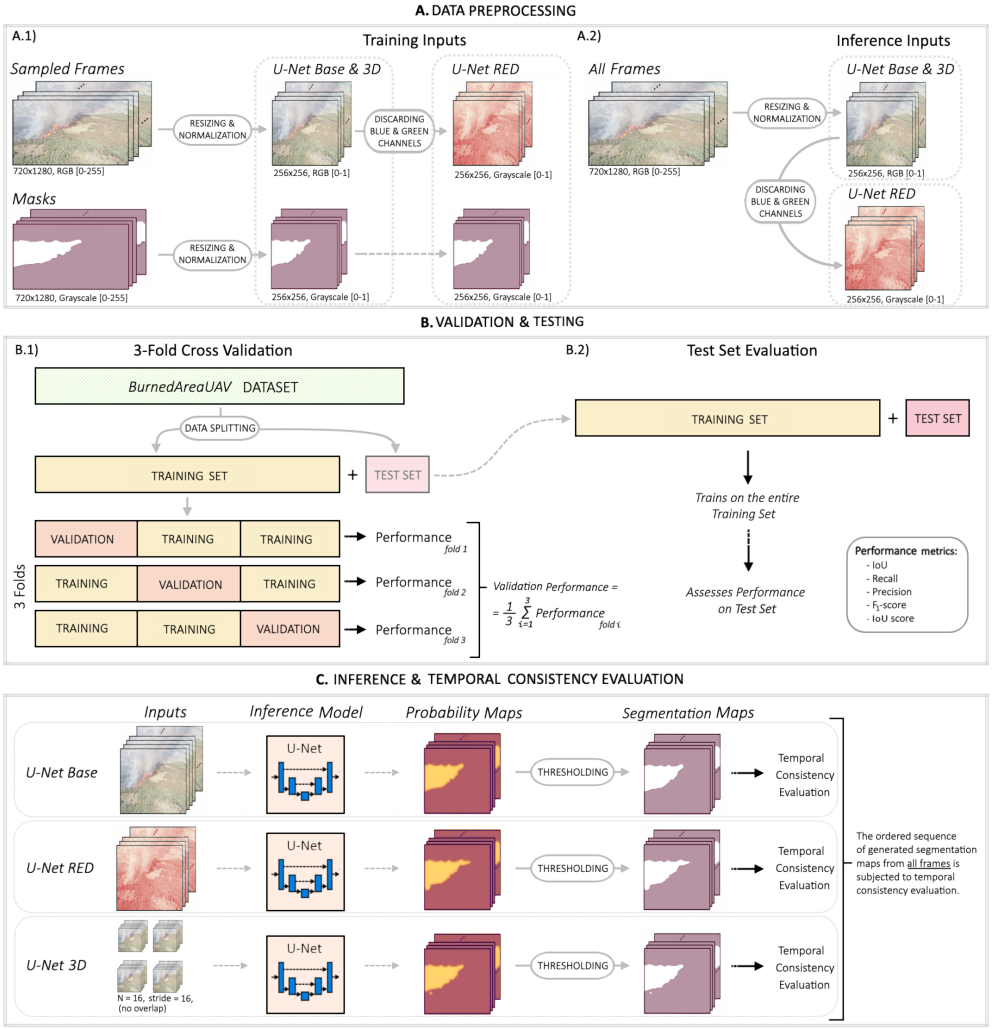


Figura 4.13: **Fluxo de trabalho da experiência.** No topo, a fase *A.* fase de Pré-processamento de dados, *B.* os detalhes de Validação cruzada 3 vezes e Teste e, em baixo, *C.* a fase de Inferência e Avaliação de consistência temporal. Retirado de T. F. Ribeiro et al. (2023)

foi treinado com os dados de treino completos e o seu desempenho no conjunto de teste foi avaliado utilizando as mesmas métricas. Os resultados foram então comparados para determinar o modelo com melhor desempenho (Fig.4.13 B.2).

O treino dos modelos foi feito do zero, com os pesos do modelo inicializados com o método Kaiming He. O otimizador escolhido foi o Adam com os parâmetros predefinidos do Keras (taxa de aprendizagem =0,001, β_1=0,9, β_2=0,999, ϵ=1e-08, decaimento=0,0) em conjunto com uma estratégia de programa de decaimento da taxa de aprendizagem por etapas (Krizhevsky et al., 2012a), em que se reduz para metade a taxa após 10 épocas de treino sem que o valor do custo (ou do erro) de validação diminua. A função de custo escolhida foi a entropia cruzada binária e não foi empregue qualquer aumento de dados. Todos os modelos foram treinados um total de 250 épocas. Os modelos U-Net Base e RED

foram treinados com um tamanho de lote de 2, enquanto o modelo U-Net 3D foi treinado com um tamanho de lote unitário.

Após avaliar os modelos no conjunto de dados BurnedAreaUAV, foram geradas máscaras de segmentação para todos os fotogramas do vídeo, e a consistência temporal dos dados segmentados foi avaliada usando os modelos treinados nos dados BurnedAreaUAV, conforme mostrado na Figura 4.13 C.

4.5 RESULTADOS

A Tabela 4.1 apresenta os resultados em termos de IoU, Revocação, Precisão e Medida F1 obtidos nos conjuntos de treino e validação. Ela mostra que, em média, o modelo U-Net Base alcança os melhores resultados para todas as métricas avaliadas. Num segundo patamar surge o modelo U-Net RED, seguido do modelo U-Net 3D.

Quanto aos resultados relativos ao conjunto de teste (Tabela 4.2), o modelo U-Net Base destaca-se novamente com um IoU de 95,31 %. Com um desempenho pior do que o modelo U-Net Base, a U-Net 3D obteve um IoU de 94,01 % e a U-Net RED 92,74 %. No entanto, o modelo U-Net RED obteve o maior valor de Precisão, o que indica que essa variante exibe uma baixa taxa de falsos positivos.

	Conjunto de validação			
	IoU [%]	Revocação [%]	Precisão [%]	Medida F1 [%]
U-Net Base	**93,62 ± 2,00**	**97,01 ± 0,85**	**96,62 ± 1,11**	**96,81 ± 0,96**
U-Net RED	88,18 ± 1,27	93,33 ± 0,92	94,23 ± 0,56	93,78 ± 0,70
U-Net 3D	86,00 ± 8,60	92,88 ± 5,99	92,12 ± 4,14	92,88 ± 5,99

Tabela 4.1: **Métricas para validação cruzada de 3 partições.**

	Conjunto de testes			
	IoU [%]	Revocação [%]	Precisão [%]	Medida F1 [%]
U-Net Base	**95,31**	**98,30**	96,92	**97,61**
U-Net RED	92,74	95,34	**97,15**	96,24
U-Net 3D	94,01	98,21	95,67	96,92

Tabela 4.2: **Métricas de desempenho no conjunto de teste.**

Também avaliamos a aplicação dos modelos treinados nos dados do BurnedAreaUAV para segmentar o vídeo completo (composto por mais de 22.500 fotogramas). Neste cenário, não temos dados anotados para validar os resultados da segmentação. Por isso, aplicámos a métrica de consistência temporal (TC) para avaliar a coerência dos polígonos resultantes.

A tabela 4.3 resume os indicadores TC, que medem a consistência temporal de segmentos de fotogramas sucessivos para todo o vídeo. Os seus valores estão discretizados nos gráficos da Figura 4.14. O modelo U-Net Base produziu a segmentação menos inconsistente, enquanto os modelos U-Net 3D e U-Net RED apresentam maior inconsistência, com particular destaque para a primeira metade da sequência. Neste aspeto, o modelo U-Net 3D foi o que obteve os piores resultados.

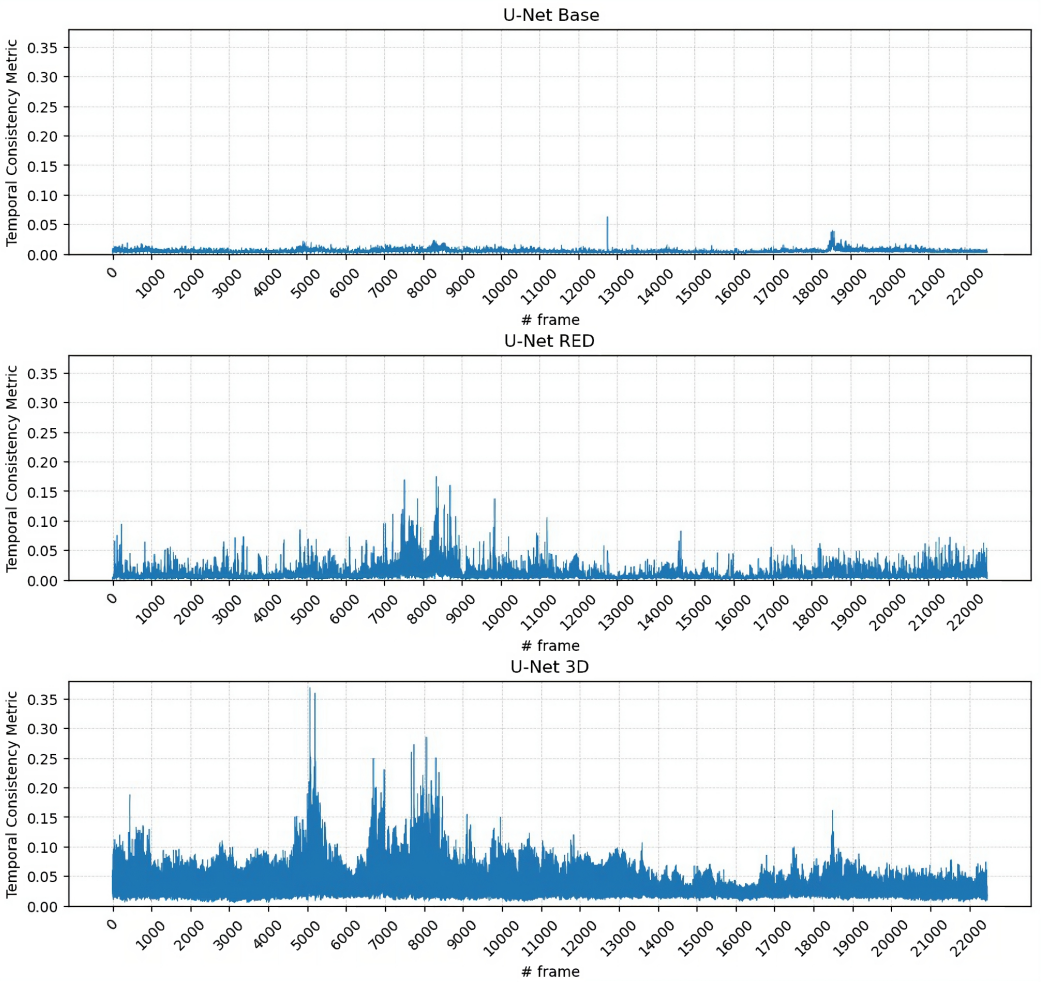


Figura 4.14: **Consistência temporal de fotogramas sucessivos para toda a sequência de vídeo**. Retirado de T. F. Ribeiro et al. (2023)

A análise da Figura 4.14 relativamente à consistência temporal para cada segmentação ao longo do vídeo, corrobora o valor da Tabela 4.3. Notavelmente, o modelo U-Net Base só mostra valores de TC superiores a 0,05 numa instância.

	mTC
U-Net Base	$\mathbf{4{,}40 \times 10^{-03}}$
U-Net RED	$9{,}11 \times 10^{-03}$
U-Net 3D	$2{,}64 \times 10^{-02}$

Tabela 4.3: **Consistência temporal média**. O modelo U-Net Base, em média, tem o valor TC menor, o que indica que tem uma maior consistência temporal ao longo do vídeo.

4.6 DISCUSSÃO E LIMITAÇÕES

Algumas amostras das segmentações geradas pertencentes ao conjunto de teste são mostradas na Figura 4.15. Em geral, os três modelos parecem ser capazes de capturar a maioria dos limites da área queimada. Globalmente, é evidente que a U-Net Base produz segmentações mais próximas do pretendido quando a anotação foi efetuada, preservando os limites da área queimada através do vídeo, mesmo nos fotogramas nos quais as oclusões causadas pelo fumo são significativas.

Curiosamente, o modelo U-Net RED demonstrou a métrica de precisão mais elevada no conjunto de teste. No entanto, é observada uma queda significativa no desempenho em termos da métrica IoU em particular, nos conjuntos de validação como de teste, indicando que esta estratégia simples tem um impacto não negligenciável neste conjunto de dados. A exclusão dos canais azul e verde pode não permitir que a rede aceda a características importantes para mapear a área ardida e afeta a robustez e estabilidade da classificação, como corroborado nos resultados da consistência temporal (Tabela 4.3).

A aplicação direta de convoluções 3D à arquitetura existente do modelo 2D U-Net conduz a um aumento significativo dos requisitos de memória, tanto em termos de contagem de parâmetros como de tamanho de entrada de amostras. Para resolver as limitações de *hardware* subjacentes à nossa configuração, o número de filtros utilizados nas camadas convolucionais em cada fase do modelo teve de ser substancialmente reduzido (ver pormenores nos diagramas da Figura 4.9 e 4.11). A redução do número de filtros pode ter afetado a capacidade do modelo (Mou e Jun Li, 2020). O compromisso entre a utilização de memória e a representação de características realça a necessidade de uma ponderação cuidadosa ao adaptar as arquiteturas do modelo de convoluções 2D para 3D. Adicionalmente, utilizámos a mesma função de perda, otimizador e programador de taxa de aprendizagem para treinar as 3 variações de U-Net. Dada a diferença substancial na arquitetura 3D da U-Net, esta parametrização também pode ter tido um impacto nos resultados demonstrados, mas é necessária mais experimentação para validar estas hipóteses.

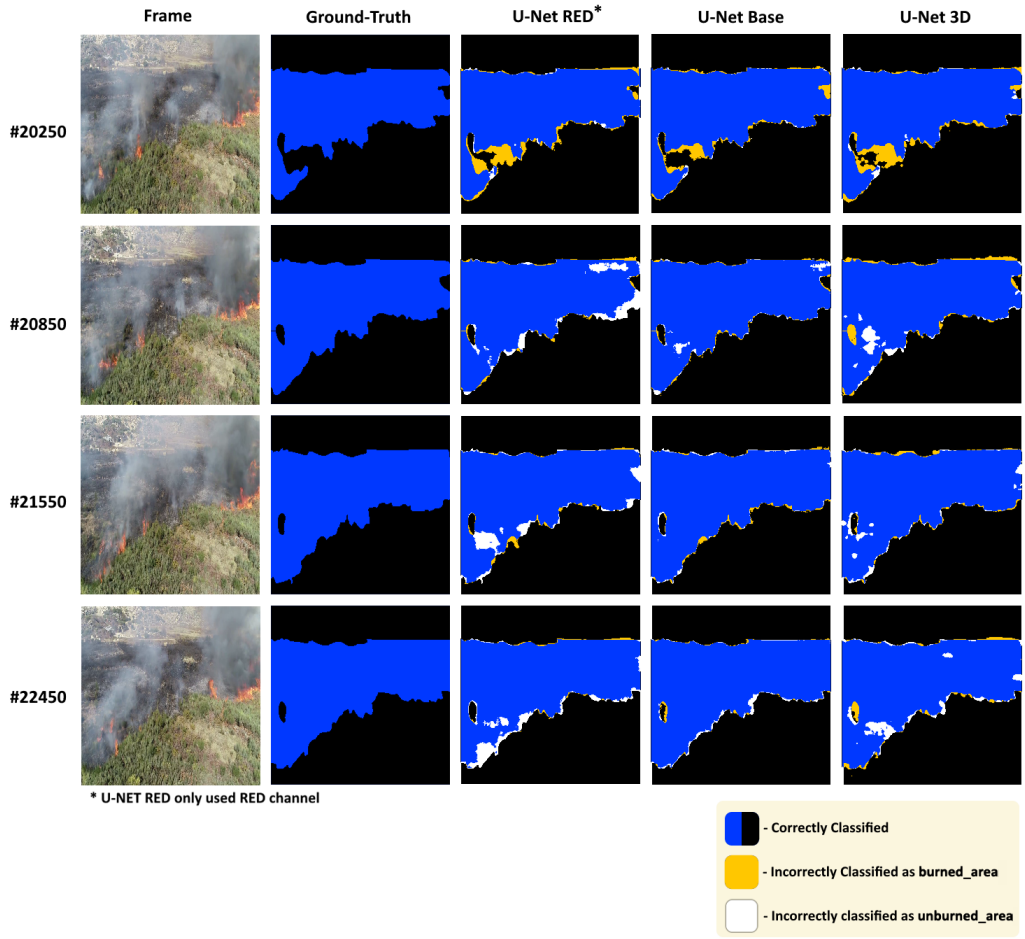


Figura 4.15: **Resultados da segmentação da área queimada para algumas amostras do conjunto de teste**. As duas primeiras colunas mostram o fotograma de vídeo e a respetiva segmentação de referência do conjunto de teste, enquanto as outras três colunas apresentam as segmentações geradas pelos modelos. Tanto as entradas como as saídas têm uma resolução de 256×256. Retirado de T. F. Ribeiro et al. (2023)

Ademais, inspecionámos qual o par de fotogramas com o valor TC mais elevado para cada um dos modelos e produzimos a Figura 4.16. A presença de áreas classificadas como queimadas num fotograma e depois como não queimadas no fotograma seguinte é particularmente notória nas segmentações produzidas pelo modelo U-Net 3D e pelo modelo U-Net RED, enquanto é menos proeminente no modelo Base, o qual é claramente o modelo com melhor desempenho no indicador de consistência temporal. Apesar de o modelo U-Net 3D ter acesso a 16 amostras sucessivas do conjunto de dados em simultâneo durante as fases de formação e inferência, a nossa implementação teve o pior desempenho em termos de consistência temporal.

Em estudos anteriores realizados em diferentes contextos, observou-se que o desempenho das U-net baseadas em convolução 3D pode não ultrapassar necessariamente o das suas congéneres 2D (Nemoto et al., 2020; Zettler e Mastmeyer, 2021b). De relevo,

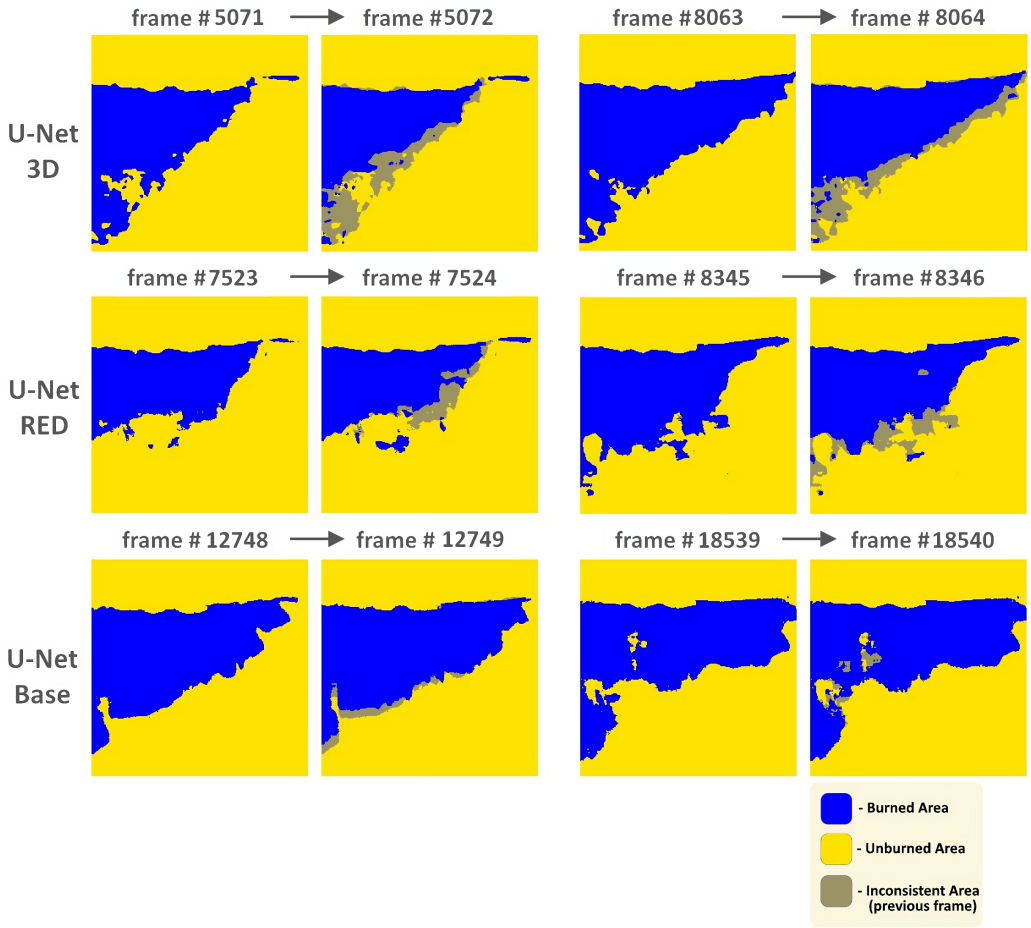


Figura 4.16: **Segmentações com maior inconsistência temporal para cada um dos modelos**. É possível observar que nos últimos fotogramas existem zonas que no fotograma anterior tinham sido consideradas como queimadas. Estas inconsistências são particularmente visíveis nos modelos U-Net 3D e U-Net RED, em menor grau. Retirado de T. F. Ribeiro et al. (2023)

para além do aumento dos requisitos de memória, estes modelos 3D também envolvem etapas de pré-processamento de dados mais complexas (B. Woo e Myungeun Lee, 2021). No contexto do nosso estudo, a introdução desta complexidade adicional não produziu um desempenho adicional, levando-nos a considerá-la como um fator a ter em conta na seleção da arquitetura adequada.

De facto, no cômputo geral, a abordagem de aprendizagem supervisionada utilizada em conjunto com os modelos totalmente convolucionais avaliados foi capaz de produzir máscaras de segmentação próximas das desejáveis, segmentando fotogramas onde existem oclusões parciais e áreas que, mesmo sob a análise cuidadosa de um anotador humano, não são triviais de classificar porque nem sempre é evidente se um determinada zona foi consumido pelo fogo com recurso apenas a um fotograma isolado. Os resultados demonstram que esta abordagem é uma opção válida para segmentar a área queimada em vídeos com anotações esparsas como ponto de partida. Ficou também demonstrado que, para este caso específico, o modelo U-Net Base tem um melhor desempenho em todas as métricas. A variante 3D não se revelou vantajosa do ponto de vista da consistência temporal, contrariamente às intuições iniciais.

É importante referir que, a fim de aplicar estes modelos em vídeos distintos, é essencial que os modelos sejam previamente treinados em anotações esparsas desses novos dados ou, em alternativa, que os modelos sejam treinados num conjunto tão vasto e diverso quanto possível de vídeos e imagens de incêndios florestais.

Com a finalidade de criar um conjunto de dados mais diverso, no futuro, planeamos expandir o BurnedAreaUAV adicionando outros vídeos capturados por *drones* em locais distintos e em condições diferentes. Adicionalmente, planeamos realizar avaliações noutros conjuntos de dados publicamente disponíveis. Consideramos ainda a possibilidade de explorar novas técnicas de anotação automática, bem como métodos de segmentação fracamente supervisionados, em que há menos necessidade de dados anotados.

4.7 CONSIDERAÇÕES FINAIS

Neste capitulo, apresentamos um novo conjunto de dados para a segmentação de áreas queimadas em incêndios florestais capturados com um *drone*, delineamos o processo de anotação e validação dos polígonos de segmentação, e descrevemos as dificuldades e características inerentes à segmentação destes fenómenos. Descrevemos métricas de desempenho e testámos três modelos profundos de segmentação semântica de imagens utilizando uma abordagem de aprendizagem supervisionada, avaliando o desempenho de classificação de píxeis de cada modelo, bem como a consistência temporal dos polígonos

de segmentação gerados para todo o vídeo. Os resultados mostram que a segmentação semântica de fenómenos naturais, como a identificação da área queimada de incêndios florestais, baseada em modelos totalmente convolucionais é promissora. De facto, os resultados da U-Net podem ser utilizados como base de comparação em futuras avaliações de segmentação de imagens sobre o conjunto de dados proposto BurnedAreaUAV.

Em trabalhos futuros, pretendemos aumentar este conjunto de dados adicionando outros vídeos capturados por *drones* em locais distintos e em condições diferentes. Consideramos a possibilidade de explorar novas técnicas de anotação automática, bem como métodos de segmentação fracamente supervisionados, em que há menos necessidade de dados anotados.

Além disso, no que diz respeito à avaliação dos modelos de segmentação e para facilitar o benchmarking, planeamos realizar avaliações entre conjuntos de dados, incorporando conjuntos de dados de queimadas publicamente disponíveis. Ao comparar o desempenho dos modelos propostos com outras abordagens, poderemos avaliar a robustez do modelo e as capacidades de generalização em diferentes cenários.

5 INTERPOLAÇÃO DE DADOS ESPAÇOTEMPORAIS

Mediante a obtenção de uma sequência de fotografias, vídeo ou outro método de aquisição que retrate a evolução da geometria de entidades ou objetos de interesse, tal como a propagação de um incêndio florestal ou o rastreamento de icebergues, geramos uma amostra que reflete a evolução espaçotemporal dessas entidades. Esta amostra discreta conduz necessariamente a uma representação comprimida, mas também acarreta um certo grau de perda de informação. Assim, para a reconstrução desse fenómeno espaçotemporal é necessário um método capaz de representar a evolução do objeto de interesse de forma contínua a partir de amostras esparsas (Mckenney e Frye, 2015).

Assumindo que existe um método de reconstrução, uma variedade de fatores extrínsecos podem afetar a qualidade da reconstrução. Este processo é suscetível à introdução de ruído ou distorção no momento da captura das amostras. Concomitantemente, uma taxa de amostragem baixa pode resultar na representação incompleta do fenómeno, e uma taxa demasiado elevada redunda em elevados requisitos de processamento e armazenamento. Ademais, fenómenos complexos podem dificultar a interpolação. Outro fator a ter em consideração que está de algum modo relacionada com as necessidades de computacionais e otimização de recursos, o método de compressão do utilizado após a captura, tem influência na qualidade da reconstrução (R. L. Costa et al., 2020).

Tendo estes elementos em consideração, com o objetivo de explorar o potencial dos modelos recentes de aprendizagem profunda para realizar interpolação espaçotemporal contínua e gerar representações intermediárias de regiões móveis 2D, propomos e avaliamos um modelo C-VAE para esta tarefa. Além disso, comparamos o seu desempenho com outros dois métodos de interpolação amplamente referenciados na literatura (McKenney et al., 2016; Schenk et al., 2000), expostos na Secção 3.2.2. Para esta avaliação, empregamos polígonos de segmentação provenientes do conjunto de dados BurnedAreaUAV, cujas características são relatadas no Capítulo 4. Considerando vários de cenários e dois métodos de compressão, utilizamos métricas de similaridade geométrica para quantificar a concordância entre os polígonos gerados e os dados de referência do conjunto de dados BurnedAreaUAV assim como de polígonos que resultaram de segmentação semântica automática da área ardida do vídeo capturado por *drone*. Adicionalmente, aplicamos métricas de consistência temporal para avaliar a congruência dos polígonos gerados.

Quanto à organização deste capitulo, iniciamos o capítulo com a Secção 5.1 ao apresentar o modelo de interpolação espaçotemporal baseado em C-VAE que concebemos. Subsequentemente, na Secção 5.2, que se dedica aos métodos de compressão, exploramos a Amostragem Periódica e a Amostragem Baseada na Distância. A Secção 5.3 é reservada à análise das métricas de avaliação, enquanto na Secção 5.4, detalhamos o método experimental e as ferramentas empregadas na avaliação dos algoritmos de interpolação. Os resultados obtidos são apresentados na Secção 5.5, seguidos pela Secção 5.6, na qual discutimos as descobertas e as limitações identificadas. Por último, na Secção 5.7, expomos as conclusões decorrentes desta investigação e delineamos potenciais direções para trabalhos futuros.

5.1 INTERPOLAÇÃO BASEADA EM C-VAE

Na Secção 2.3.2, apresentamos o modelo C-VAE e explicamos que este expande as capacidades do VAE ao incorporar a aprendizagem de uma distribuição condicional, onde y representa a variável condicionante. Yan et al. (2016) mostram que para além de gerar novas amostras condicionadas por y, os C-VAE têm a capacidade de realizar a edição condicional de imagens no espaço latente. Dadas duas entradas condicionadas diferentes, x_1 e x_2, é possível interpolar no espaço latente entre os códigos latentes correspondentes z_1 e z_2 para gerar novas imagens que combinam *suave* e coerente as duas imagens. Esta propriedade torna as C-VAE adequadas para interpolar diferentes representações codificadas de forma discreta ou contínua.

A nossa abordagem consiste em treinar um modelo C-VAE convolucional no conjunto de amostras a interpolar, condicionado pelo registo temporal (ou *timestamp*) de cada amostra. Para aplicações que operam com variáveis discretas, *i.e.* um número limitado de classes, é comum codificar as classes com codificação *one-hot* e depois concatenar a variável condicionante codificada tanto ao vetor à entrada x como ao espaço latente z.

Uma vez que lidamos com um fenómeno contínuo, nomeadamente a evolução da área ardida, e a variável de condicionamento (`label`) que representa o instante de interpolação também é contínua, optamos por utilizar simplesmente o número de fotogramas do vídeo original, normalizá-lo para um intervalo entre 0 e 1 e, de seguida, concatenar o valor resultante y ao espaço latente z e à entrada x, como se ilustra na parte superior da Figura 5.1.

Após o treino, durante a fase de inferência (Figura 5.1 b)), recolhemos amostras da variável de espaço latente z e da variável de condicionamento y para gerar novas amostras da distribuição condicional aprendida $p_\theta(x|y)$. Mais especificamente, primeiro amostramos

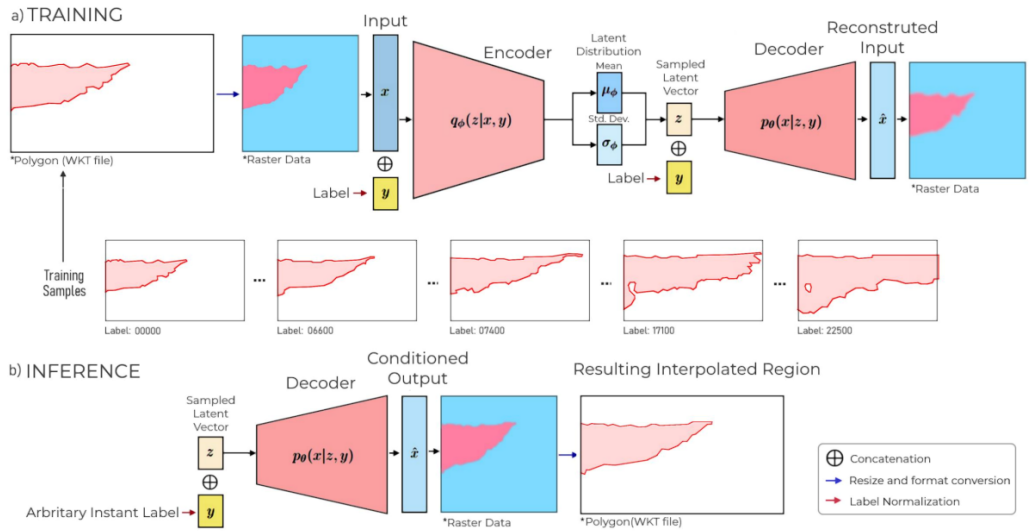


Figura 5.1: **Arquitetura C-VAE utilizada.** a) Treino: cada região armazenada no formato WKT é convertida em imagem matricial (ou *raster*) para ser processada pelo modelo b) Inferência: é gerada uma nova imagem condicionada por *timestamp* arbitrário e convertida para o formato WKT. Retirado de T. F. Ribeiro (2023)

um vetor aleatório ϵ a partir de uma distribuição normal padrão e, de seguida, utilizamo-lo para calcular uma amostra z a partir da distribuição posterior aproximada aprendida $q_\phi(z|x,y)$ utilizando o truque de reparametrização (Secção 2.3.1). Subsequentemente, definimos uma variável condicionante específica y_i, que representa um instante arbitrário da duração do vídeo, e concatenamo-la com a amostra z da rede de descodificação $p_\theta(x|z,y)$ para gerar uma imagem reconstruída $\hat{x}_i$ que representa o polígono estimado do fenómeno para o instante t_i.

5.2 MÉTODOS DE COMPRESSÃO

As diferentes estratégias de (sub)amostragem reduzem a quantidade de dados espaço-temporais armazenados, mas também determinam as representações utilizadas para a interpolação, influenciando assim o desempenho dos métodos de reconstrução (R. L. Costa et al., 2020).

5.2.1 *Amostragem Periódica*

Como primeira abordagem de compressão, consideramos as regiões correspondentes à área queimada como uma sequência de observações $x_t = \{x_1, x_2, ..., x_n\}$, em que n é o comprimento da sequência, ordenada no tempo com uma determinada frequência de

amostragem f_s, correspondente à taxa de fotogramas do vídeo. Em seguida, amostramos a sequência periodicamente utilizando um fator de decimação de $d \in \mathbb{N}$. Isto resulta numa nova sequência de observações $w_t = \{w_1, w_2, ..., w_m\}$, em que $m = \lfloor n/d \rfloor$ é o comprimento da sequência reduzida. Cada observação w_i corresponde à observação original x_{id}, em que i é o índice da sequência com amostra reduzida e d é o fator de amostragem reduzida. Embora esta abordagem simples reduza efetivamente o tamanho da sequência por um fator de d, no entanto, este método pode descartar amostras relevantes.

5.2.2 *Amostragem Baseada na Distância.*

Como segunda abordagem, seguimos a estratégia sugerida por R. L. Costa et al. (2020). Este método seleciona amostras de uma sequência de observações selecionando são as que considerados representativos pela sua dissimilaridade das que lhe precedem. Toma como entrada um conjunto ordenado de observações esparsas e uma função de distância que calcula a dissimilaridade entre duas observações. O algoritmo inicializa a sequência amostrada com a primeira observação e adiciona iterativamente observações subsequentes à sequência se e só se forem atingirem um valor limiar α pré-definido de dissimilaridade em relação à última observação selecionada. Este processo continua até que todas as observações tenham sido consideradas ou o tamanho da sequência subasmostrada atinja o comprimento desejado. A função de distância pode ser qualquer métrica que calcule a dissimilaridade entre polígonos, como o índice de Jaccard (descrito na Secção 4.2), a distância de Hausdorff, ou uma combinação de várias métricas.

5.3 MÉTRICAS DE AVALIAÇÃO

Para avaliar o desempenho de cada algoritmo de interpolação espaçotemporal e quantificar a semelhança entre os polígonos resultantes e os polígonos de referência, recorremos à utilização das métricas do índice de Jaccard (conforme descrito na Secção 4.2) e da distância de Hausdorff. Além disso, introduzimos uma variação e refinamento da métrica de consistência temporal apresentada anteriormente no Capítulo 4, que foi originalmente concebida para avaliar a congruência nas sequências de segmentação geradas pelos modelos U-Net. Neste contexto, essa métrica é aplicada para medir a coerência das sequências de interpolações geradas.

5.3.1 *Distância de Hausdorff*

Em termos simples, a distância de Hausdorff (d_H) mede o grau de dissemelhança entre dois conjuntos, encontrando a distância máxima entre um ponto de um conjunto e o seu ponto mais próximo no outro conjunto (Huttenlocher et al., 1993).

Considerando dois conjuntos de pontos não vazios, finitos e fechados $X = \{x_1, x_2, x_3, \ldots\}$ e $Y = \{y_1, y_2, y_3, \ldots\}$ num espaço métrico, a distância de Hausdorff direcionada de X para Y, denotada como $d_H(X, Y)$, é definida por:

$$d_H(X, Y) = \inf\{\varepsilon \geqslant 0 \,|\, X \subseteq Y^\varepsilon \text{ e } Y \subseteq X^\varepsilon\} \tag{28}$$

onde X^ε e Y^ε representam os ε-vizinhanças de X e Y, respetivamente.

Por outras palavras, $d_H(X, Y)$ é o menor valor de ε tal que X está completamente contido na ε-vizinhança de X e Y está completamente contido na ε-vizinhança de X. Essa medida representa a mínima quantidade de expansão ou contração necessária para fazer X estar totalmente contido em X e Y estar totalmente contido em X.

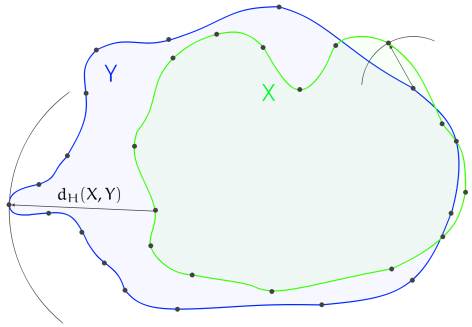


Figura 5.2: Representação esquemática da distância de Hausdorff entre os conjuntos de pontos X e Y

O ínfimo (maior limite inferior) é tomado sobre todos os valores possíveis de ε, garantindo que X e Y estejam perfeitamente alinhados entre si. Adicionalmente, a distância de Hausdorff direcionada $d_H(X, Y)$ satisfaz a propriedades de (I) Não-negatividade: $d_H(X, Y) \geqslant 0$ para quaisquer conjuntos X e Y; de (II) Identidade dos indiscerníveis: $d_H(X, X) = 0$ se e somente se X é idêntico a Y; e de (III) Simetria: $d_H(X, Y) = d_H(Y, X)$ se X e Y são simétricos relativamente ao deslocamento.

Em suma, A distância de Hausdorff é uma métrica útil em diversas aplicações, tais como correspondência de formas, alinhamento de imagens e reconhecimento de objetos, que quantifica a dissimilaridade entre dois conjuntos, considerando a sua disposição espacial e alinhamento.

5.3.2 *Consistência Temporal*

Sabemos que para um mesmo foco de incêndio, uma área estabelecida como ardida não pode deixar de o ser numa fase posterior. Da mesma forma, sabemos que a área ardida nunca diminui. Com estas considerações em mente, definimos a consistência temporal TC como um complemento de uma diferença geométrica:

$$TC_{stride} = 1 - \frac{A_t - A_{t+stride}}{A_{t+stride}}, \forall t \in \{1, 2, ..., T - stride\} \tag{29}$$

onde A_t e $A_{t+stride}$ representam a região da área ardida separada por $stride$ amostras, onde $stride$ representa o intervalo entre duas amostras consecutivas consideradas. Para avaliar amostras separadas por diferentes escalas de temporais, consideramos vários valores de $stride$ a partir de uma progressão geométrica $stride_n = ar^{n-1}, \forall n \in \{1, 2, ..., N\}$, com N menor que o número total de polígonos na sequência, onde a é o coeficiente de cada termo e r é o rácio comum entre termos adjacentes. Depois, para cada um dos valores em $stride_n$, calculamos a média de TC_{stride}, que representa a coerência temporal do algoritmo para amostras separadas de $stride$. Finalmente, também podemos estimar a consistência temporal global calculando a média de todas as médias de TC_{stride}.

5.4 EXPERIÊNCIA

Nesta experiência, avaliámos três algoritmos diferentes: (**I**) o método de interpolação de Mckenney (McKenney et al., 2016), (**II**) a interpolação baseada na forma a qual denominamos de (do inglês *Shape Based*) (Bouazizi et al., 2021; Schenk et al., 2000), descritos nas Secções 3.2.2, e (**III**) o método baseado em C-VAE.

Para cada algoritmo, geramos amostras intermédias correspondentes aos *timestamps* dos fotogramas do vídeo original, utilizando as 226 amostras resultantes da amostragem periódica, bem como o subconjunto de 13 amostras baseadas na distância. Comparamos os polígonos gerados pelos algoritmos com os gerados pela segmentação automática (*Amostras U-Net*) e validados usando o subconjunto de teste do conjunto de dados BurnedAreaUAV e calculamos as métricas de similaridade Jaccard e Hausdorff. Ademais, avaliamos a qualidade dos polígonos gerados em termos do indicador de Consistência Temporal formulado na Secção 5.3.2. Para calcular o IoU e o d_H, descartámos as amostras que suportavam o cálculo das regiões intermédias, tanto para a amostragem periódica como para a amostragem baseada na distância. Ou seja, de um universo de 22.500 observações correspondentes aos fotogramas do vídeo, consideramos 22.274 regiões intermediárias

para a Amostragem Periódica e 22.487 para a Amostragem Baseada em Distância. Todas as métricas foram calculadas considerando a resolução da filmagem original (1280×720).

Configuração experimental

As experiências foram realizadas num computador com Windows 10 e equipado com um processador Intel i7-10700K, uma Unidade de Processamento Gráfico (GPU) Nvidia GeForce RTX 3090 e 32 GB de RAM. O código foi desenvolvido em Python, quase inteiramente em Jupyter Notebooks. O nosso modelo C-VAE é construído sobre uma implementação típica de rede neuronal convolucional para o codificador e descodificador, praticamente sem ajuste de hiperparâmetros. O código está disponível em https://github.com/CIIC-C-T-Polytechnic-of-Leiria/Reconstr_CVAE_paper.

O subconjunto de treino do conjunto de dados BurnedAreaUAV é considerado uma representação fidedigna dos polígonos da área ardida e serve de base para a interpolação com Amostragem Periódica. Subamostramos depois o conjunto de 226 fotogramas aplicando a amostragem baseada na distância. Neste processo, utilizamos a índice de Jaccard como medida de dissimilaridade, com um limiar de tolerância fixado em $\alpha = 0{,}15$, resultando na compressão do número de amostras para 13.

Adicionalmente, o modelo de segmentação U-Net Base que resultou do treino no conjunto de dados BurnedAreaUAV, produziu as máscaras de segmentação para todos os fotogramas de vídeo, que foram depois convertidas em polígonos compatíveis com o formato WKT. Como os polígonos produzidos pelo modelo U-Net Base obtiveram um valor global de índice Jaccard superior a 0,95 no conjunto de teste do conjunto de dados BurnedAreaUAV, são considerados boas aproximações da progressão real da área ardida. Designámos este conjunto de 22.500 polígonos como *Amostras U-Net*.

5.5 RESULTADOS

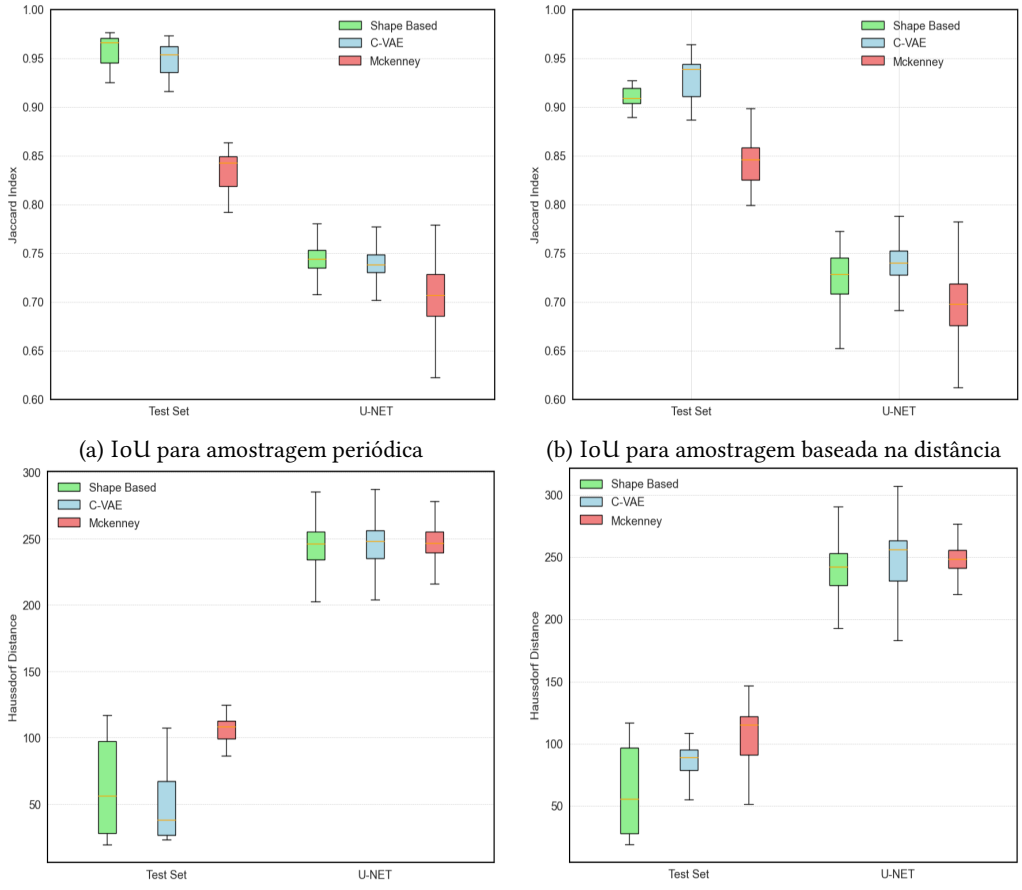
A Tabela 5.1 apresenta os valores obtidos para as métricas de similaridade e a Tabela 5.2 resume os resultados em termos da avaliação da consistência temporal.

As interpolações *Shape-Based* e C-VAE superaram o método de interpolação Mckenney tanto na amostragem periódica como na amostragem baseada na distância (como se ilustra na Tabela 5.1 e nas Figura 5.3 (a) e (c)). Isto é particularmente notável no conjunto de teste BurnedAreaUAV, no qual o algoritmo *Shape-Based* e o algoritmo C-VAE apresentam um desempenho relativamente próximo, com uma pequena vantagem para o primeiro. O

Tabela 5.1: **Avaliação da similaridade**. Comparação de IoU e d_H para as amostras U-Net e conjunto de teste *BurnedAreaUAV* utilizando amostragem periódica e baseada na distância.

Amostragem Periódica							
Dados	Algoritmo	Distância de Jaccard			Distância de Hausdorff		
		Média	DP*	Min-Máx	Média	DP*	Min-Máx
Amostras U-Net	*Shape-Based*	**0.958**	**0.011**	**0.870-0.982**	42.460	37.503	9.849-243.994
	C-VAE	0.951	0.011	0.852-0.975	**41.866**	**26.045**	**9.849-242.745**
	Mckenney	0,892	0,048	0,519-0,982	72,195	44,284	9,659-364,660
Conjunto de teste *BurnedAreaUAV*	*Shape-Based*	**0.959**	**0.016**	**0.925-0.977**	**48.382**	**33.312**	**19.444-117.000**
	C-VAE	0.949	0.017	0.916-0.974	60.815	24.926	23.000-107.201
	Mckenney	0,703	0,073	0,493-0,864	113,161	33,832	86,279-266,303
Amostragem baseada na distância							
Dados	Algoritmo	Distância de Jaccard			Distância de Hausdorff		
		Média	DP*	Min-Máx	Média	DP*	Min-Máx
Amostras U-Net	*Shape-Based*	**0.928**	**0.020**	**0.845-0.982**	**68.315**	**38.443**	**10.296-306.026**
	C-VAE	0.905	0.026	0.825-0.987	76.464	52.362	16.000-338.095
	Mckenney	0.876	0.040	0.763-0.978	85.426	38.922	10.471-269.194
Conjunto de teste *BurnedAreaUAV*	*Shape-Based*	0.910	0.021	0.887-0.964	**60.815**	**33.312**	**19.444-117.000**
	C-VAE	**0.930**	**0.011**	**0.889-0.928**	85.220	14.827	52.773-108.853
	Mckenney	0.850	0.038	0.799-0.960	103.068	30.744	23.014-146.521

* Desvio padrão



(a) IoU para amostragem periódica

(b) IoU para amostragem baseada na distância

(c) d_H para amostragem periódica

(d) d_H para amostragem baseada na distância

Figura 5.3: **Diagrama de Quartis para as métricas de desempenho**. Retirado de T. F. R. Ribeiro, F. Silva e C. Costa (2023)

Tabela 5.2: **Comparação da consistência temporal**. Consistência temporal média para os diferentes algoritmos com amostragem periódica e baseada na distância

Consistência Temporal Média						
	Amostragem Periódica			Amostragem baseada na Distância		
Algoritmo	Média	DP*	Min-Máx	Média	DP*	Min-Máx
Shape-Based	0,986	0,011	0,971-1,000	0,994	0,006	0,985-1,000
C-VAE	**0,993**	**0,007**	**0,982-0,998**	**0,999**	**0,001**	**0,997-1,000**
Mckenney	0,970	0,019	0,951-0,995	0,983	0,018	0,948-0,998

* Desvio padrão

algoritmo *Shape-Based* em ambos os conjuntos de dados obteve o melhor desempenho em termos da métrica da distância de Hausdorff.

As interpolações do tipo *Shape-Based* e o algoritmo C-VAE demonstraram um desempenho superior em comparação com o método de interpolação Mckenney, tanto na amostragem periódica como na amostragem baseada na distância, conforme evidenciado na Tabela 5.1 e nas Figuras 5.3 (a) e (c). Este desempenho é especialmente notável quando aplicado ao conjunto de teste BurnedAreaUAV, no qual o algoritmo *Shape-Based* e o algoritmo C-VAE apresentam resultados semelhantes, com uma ligeira vantagem para o primeiro.

É importante destacar que o algoritmo *Shape-Based* obteve o melhor desempenho em ambas as bases de dados, conforme avaliado pela métrica de distância de Hausdorff.

Também podemos observar que a redução do número de amostras de suporte para interpolação não teve um impacto muito pronunciado nos valores do índice Jaccard e de distância de Hausdorff, o que apoia a validade do algoritmo de compressão baseado na distância (de 226 para 13 amostras) para este conjunto de dados especificamente.

Os resultados na Tabela 5.2 indicam a Consistência Temporal média para todos os valores de *stride* temporal considerados (1, 10, 100, 1.000 e 10.000) e mostram que o modelo C-VAE consegue produzir polígonos com maior consistência em ambos os conjuntos de dados. A Figura 5.4 representa a área (ou número de pixeis) dos polígonos ao gerados e corrobora esta noção ao mostrar a monotonicidade superior e a evolução mais *suave* das representações geradas pelo modelo C-VAE.

A análise da Figura 5.5 indica que tanto o algoritmo C-VAE como *Shape-Based*, apresentam consistência temporal superior para valores de *stride* temporal até 10, para a amostragem periódica e baseada em distância. No entanto, observa-se que o desempenho do algoritmo *Shape-Based* sofre uma quebra para *stride* maior ou igual a 100, enquanto o

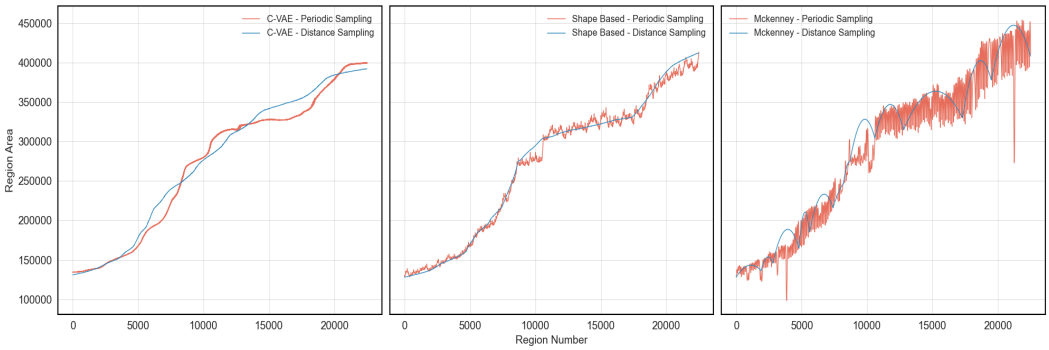


Figura 5.4: Representação da evolução da área (número de pixeis) dos polígonos de área ardida. Retirado de T. F. R. Ribeiro, F. Silva e C. Costa (2023)

.

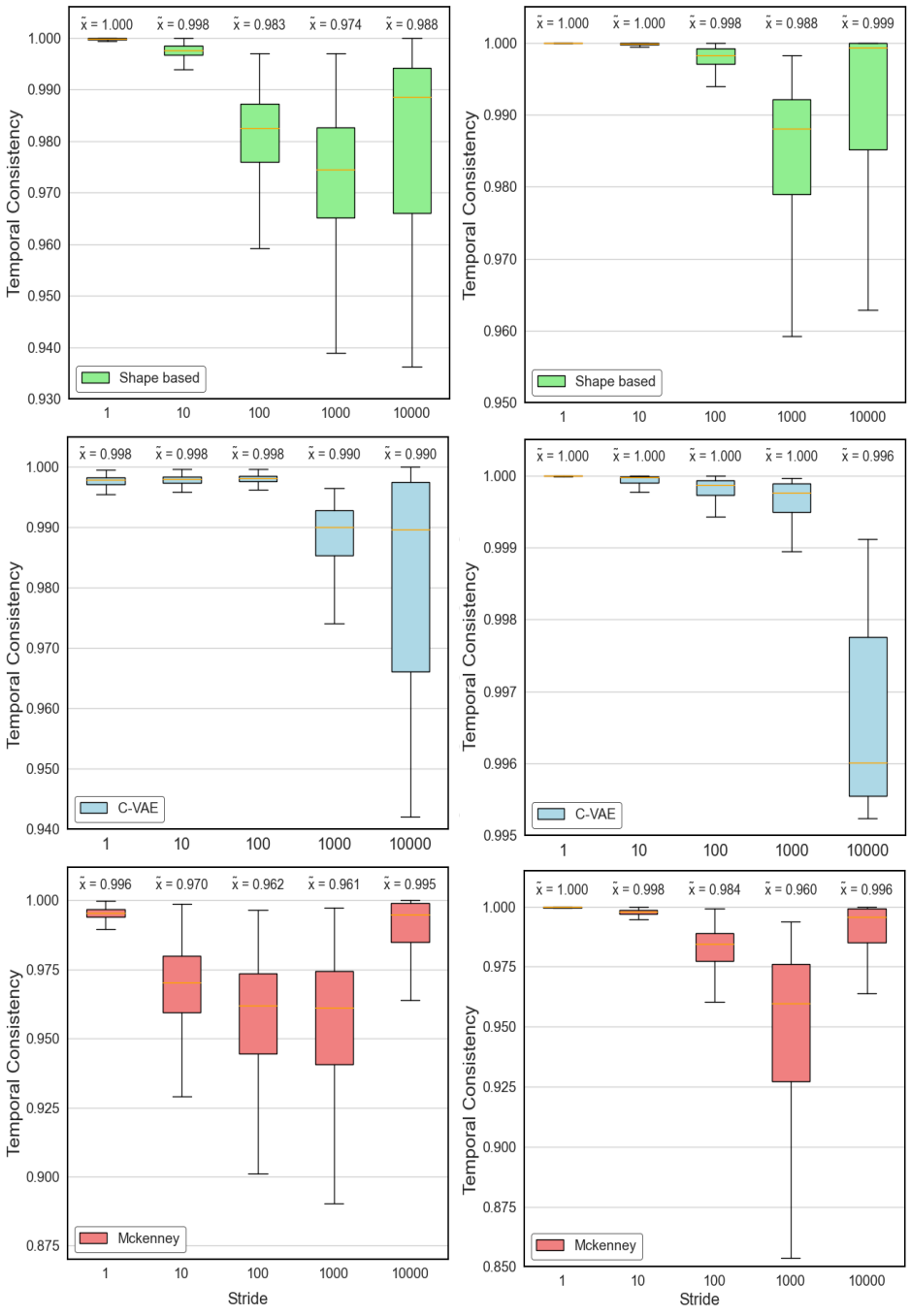
C-VAE se mantém relativamente estável até *stride* de 10.000 fotogramas, o parece indicar uma menor capacidade de manter a consistência numa janela de temporal mais longa. De facto, observa-se uma que para *stride* de 10000 (o valor máximo testado), o C-VAE tem pior desempenho. No que respeita ao algoritmo Mckenney, apresenta uma menor Consistência temporal para todos os valores de *stride*, mas é bastante competitivo para o valor de *stride* temporal máximo.

5.6 DISCUSSÃO E LIMITAÇÕES

Geralmente, uma interpolação de alta qualidade deve apresentar duas características principais: em primeiro lugar, os pontos intermediários devem se assemelhar de perto aos dados reais; em segundo lugar, os pontos intermediários devem permitir uma transição suave e coerente entre os pontos de suporte.

Como o algoritmo McKenney se concentra na criação de interpolações com topologia válida, ou seja, sem segmentos que se auto-interceptam, as sequências de regiões geradas num conjunto de dados real e ruidoso como o BurnedAreaUAV, revelaram deformações e incoerências. Moreira, Dias et al. (2016) e José Duarte, Dias et al. (2018) descrevem as limitações do algoritmo Mckenney, as quais vemos refletidas nos resultados desta experiência. Essas inconsistências tornam-se manifestas ao visualizar as interpolações produzidas por meio de um vídeo (Figura 5.6).

Embora o algoritmo *Shape-Based* tenha obtido resultados relativamente positivos nas métricas de similaridade, tende a produzir artefactos e parece mostrar-se menos capaz quando os polígonos a interpolar tem topologias significativamente diferentes (T.-Y. Lee e C.-H. Lin, 2002). O desempenho inferior observado para interpolações utilizando a



(a) Amostragem Periódica (b) Amostragem baseada na distância

Figura 5.5: Resultados da amostragem periódica e da amostragem baseada na distância, e diferentes *stride* temporais. São utilizadas escalas diferentes do eixo y para uma melhor visibilidade. Retirado de T. F. R. Ribeiro, F. Silva e C. Costa (2023)

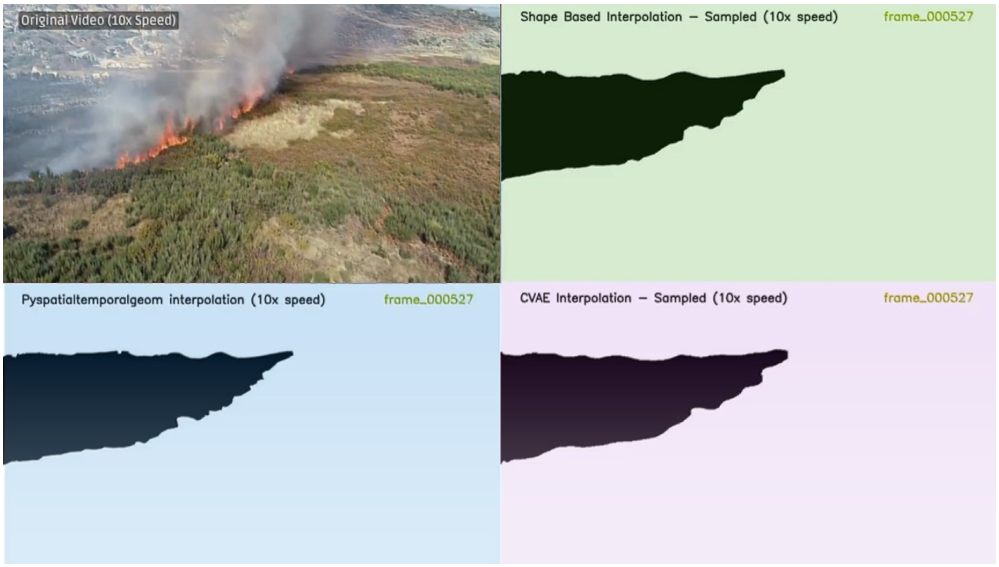


Figura 5.6: Fotograma do vídeo do que resultou da interpolação espaçotemporal dos três modelos testados.

amostragem baseada na distância no subconjunto de teste BurnedAreaUAV pode refletir esta limitação, mas são necessárias experiências adicionais para confirmar esta hipótese.

Os C-VAE podem aprender representações contínuas e suaves de dados complexos de elevada dimensionalidade, otimizando simultaneamente duas funções de custo: a função de custo de reconstrução e a de divergência de Kullback-Leibler. O modelo é encorajado descrever o estado latente para uma observação com distribuições próximas da anterior, mas desviar-se quando necessário para descrever as características salientes da entrada.

A minimização da divergência Kullback-Leibler força as codificações a aproximarem-se umas das outras, permitindo uma interpolação suave e a geração de novas amostras. Por outras palavras, o equilíbrio alcançado pela natureza de formação de *clusters* da perda de reconstrução e a natureza de aglomeramento denso da divergência Kullback-Leibler provoca a formação de *clusters* distintos, permitindo ao descodificador interpolar suavemente e evitando lacunas súbitas entre *clusters*. Este mecanismo providencia uma explicação para a superioridade da consistência temporal obtida pela solução proposta baseada em C-VAE.

No entanto, a C-VAE também tem desvantagens. Em primeiro lugar, ao contrário dos modelos clássicos, tem de ser treinado antes da interpolação, o que pode ser moroso. Em segundo lugar, os VAE padrão tendem a gerar resultados desfocados, o que se traduz numa má definição dos limites da região. Finalmente, quanto à codificação do *timestamp* (`label`) referida na Secção 5.1, poder-se-ia considerar outros métodos, no entanto, para esta aplicação específica, a nossa abordagem provou ser suficiente, mesmo considerando

a introdução do erro de quantização relacionado com a resolução finita dos números de vírgula flutuante utilizados para armazenar os valores do espaço latente.

5.7 CONSIDERAÇÕES FINAIS

A representação contínua de dados espaçotemporais requer métodos para gerar uma representação intermediária para observações entre dois pontos. Para implementar a abstração de regiões móveis, são geralmente utilizados algoritmos de interpolação de regiões. No entanto, os avanços recentes em modelos baseados em aprendizagem profunda mostram que podem ser uma possível alternativa para esse problema. Neste capitulo, comparamos o desempenho de um modelo C-VAE com algoritmos clássicos de interpolação de regiões. Utilizamos dois conjuntos de dados obtidos a partir do capturado por *drones* de um fogo controlado e avaliamos o desempenho das soluções usando métricas de similaridade geométrica e consistência temporal.

O algoritmo C-VAE teve um desempenho competitivo em comparação com o algoritmo de melhor desempenho (*Shape-Based*) em termos de métricas de similaridade e alcançou superiorizou-se em termos de consistência temporal. Os resultados sugerem que modelos baseados em VAE são opções viáveis para interpolação espaçotemporal e motivam-nos a explorar diferentes variantes de AE para abordar as limitações identificadas. O modelo C-VAE gerou uma representação relativamente realista e suave da evolução da área, um desafio enfrentado pelos métodos de interpolação de regiões.

No futuro, planeamos testar as capacidades de outros modelos baseados em AE para gerar a evolução espaçotemporal de uma gama mais vasta de fenómenos do mundo real.

6 CONCLUSÃO

Iniciamos este capítulo final com uma análise dos resultados relevantes obtidos no decorrer deste estudo, fazendo referência aos objetivos iniciais e indicando a importância deste trabalho para área de pesquisa. De seguida, discutimos as limitações do trabalho e sugerimos possíveis soluções. Na Secção 6.3, enumeramos a produção cientifica, documentação e dados disponibilizados. Depois, na Secção 6.4, apresentamos algumas propostas para trabalho futuro.

6.1 SUMÁRIO DOS RESULTADOS

Garbage in, Garbage Out (GIGO), numa tradução livre Lixo Entra, Lixo Sai (Kilkenny e Robinson, 2018), é um termo coloquial que existe em ciências da computação para descrever que a saída de um processo ou algoritmo será tão bom, ou tão mau quanto a qualidade das suas entradas. Mais especificamente, GIGO é a ideia e que por muito vasto que seja um conjunto de dados de treino, se for de má qualidade, enviesado e incompleto e resultados serão necessariamente pobres. No subtexto dessa máxima reside uma recomendação tácita: cientistas de dados, engenheiros de aprendizagem automática e investigadores devem ser diligentes na colheita, análise, limpeza e preparação dos dados e dedicar o tempo necessário à criação de um conjunto de dados de qualidade, antes de iniciar qualquer modelação de dados.

Com o propósito de gerar um conjunto de dados curados de qualidade para treino e avaliação de modelos de segmentação semântica de área ardida em fogos florestais, descrevemos detalhadamente o fluxo de trabalho que compreende a captura, amostragem, anotação manual e a método de validação desta segmentação e providenciamos o conjunto de dados, tanto em formato de imagem *raster*, como poligonal padronizado, assim como o vídeo original. Até onde alcança o nosso conhecimento, não existe um conjunto de dados com as características descritas, pelo que esta contribuição poderá servir como uma ferramenta útil para futuros trabalhos.

Com base no conjunto de dados BurnedAreaUAV que acabamos de descrever, testámos modelos de segmentação semântica totalmente convolucionais baseados na arquitetura

U-Net, os quais demonstraram representar de forma relativamente fidedigna a evolução da área ardida. A escolha da arquitetura U-Net foi motivada pelo seu amplo reconhecimento na área de Visão Computacional e pelo seu desempenho bem estabelecido como referência. Não obstante, a melhor variante testada (U-Net Base) obteve IoU superior a 95% no conjunto de teste, o que indica boa capacidade de representação para este caso de estudo.

Além da avaliação do ponto de vista da similaridade geométrica, os modelos de segmentação automática foram avaliados em termos de consistência temporal, com uma métrica especialmente concebida para avaliação da congruência dos polígonos da evolução de área ardida. Esta métrica de implementação simples acrescenta-se a outros métodos de avaliação de consistência temporal da literatura, mas tem a particularidade de ser adaptada a evolução de área de focos de fogo em ambiente florestal.

No que diz respeito aos modelos de interpolação espaçotemporais, propomos a utilização de um C-VAE para aprender a representação da evolução da área ardida e gerar amostras em momentos temporais arbitrários. Embora soluções semelhantes tenham sido propostas por outros autores em diferentes domínios, o trabalho desenvolvido valida a capacidade da nossa solução como uma alternativa eficaz aos modelos clássicos.

De relevo, o modelo C-VAE convolucional com hiperparametrização mínima, demonstrou ser competitivo ao nível das métricas de similaridade geométricas, e mostrou se superior em termos de consistência temporal, pelo que fica em aberto a possibilidade de um estudo futuro.

Adicionalmente, testamos um método de subsmostragen baseado na distância geométrica das amostras de suporte que, na experiência efetuada, reduz o número de amostras em aproximadamente 17 vezes (de 226 para 13 amostras). Observamos que o desempenho dos modelos de interpolação após a amostragem é circunscrito, o que valida este método no universo dos conjuntos de dados testados.

6.2 LIMITAÇÕES

Na nossa perspetiva, a principal limitação deste estudo prender-se com o universo relativamente reduzido que está representado no conjunto de dados BurnedAreaUAV. Por um lado, não nos é possível faz considerações acerca da capacidade de generalização do modelo U-Net. Por outro lado, devido à falta de variabilidade neste conjunto de dados, os resultados obtidos devem ser interpretados num contexto mais circunscrito. O trabalho foi desenvolvido tendo consciência deste constrangimento, mas apontamos soluções possíveis para esta limitação na Secção relativa aos Desafios e Trabalho Futuro.

No Capítulo relativo à Interpolação de Dados Espaçotemporais identificamos uma desvantagem do modelo C-VAE face aos modelos clássicos. Ao contrário dos outros modelos testados, o C-VAE tem de ser treinado no conjunto de dados representativo antes de se proceder à geração de representações intermédias. Ainda que não nos pareça um entrave à sua implementação em aplicações de bases de dados espaçotemporais, é um aspeto a ter em consideração.

6.3 CONTRIBUIÇÕES DA DISSERTAÇÃO

Os principais contribuições desta dissertação são:

I um novo conjunto de dados anotado manualmente para treino e avaliação de modelos de segmentação semântica de vídeos de área ardida em fogos florestais, assim como o método detalhado da criação e validação deste mesmo conjunto de dados;

II avaliação de modelos de segmentação semântica convolucionais com uma abordagem de aprendizagem supervisionada para aferir a sua capacidade e criar uma base de referência para trabalhos futuros;

III introdução de uma métrica específica de consistência temporal para avaliar a congruência dos polígonos gerados por modelos de segmentação em problemas relativos à evolução da área ardida;

IV um método de compressão de dados espaçotemporais baseados na distância geométrica de amostras;

V validação de método interpolação de polígonos que representam a evolução de área ardida num fogo florestal a partir de um número limitado de amostras baseado num modelo C-VAE;

VI avaliação sistemática do modelo C-VAE, comparando com alternativas da literatura, considerando métodos de compressão distintos, métricas de similaridade geométrica e de consistência temporal.

Como resultado do trabalho desenvolvido durante na dissertação, foi produzido um artigo científico publicado num jornal do primeiro quartil do indicador *SCImago Journal Rank*, na área de aplicações de ciências de computação, e um segundo artigo publicado em conferência internacional:

- Tiago F.R. Ribeiro et al. (2023). «Burned area semantic segmentation: A novel dataset and evaluation using convolutional networks». Em: *ISPRS Journal of Photogrammetry*

and Remote Sensing 202, pp. 565–580. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2023.07.002

- Tiago F. R. Ribeiro, Fernando Silva e Rogério Luís de C. Costa (2023). «Reconstructing Spatiotemporal Data with C-VAEs». Em: *Advances in Databases and Information Systems*. Ed. por Alberto Abelló et al. Cham: Springer Nature Switzerland, pp. 59–73. ISBN: 978-3-031-42914-9. DOI: 10.1007/978-3-031-42914-9_5

Adicionalmente, disponibilizamos à comunidade o conjunto de dados BurnedAreadUAV, acessível num repositório Zenodo de acesso livre:

- Tiago F. R. Ribeiro, Fernando Silva, José Moreira et al. (mai. de 2023). *BurnedAreaUAV Dataset (v1.1)*. Versão 1.1. DOI: 10.5281/zenodo.7944963

Foi ainda produzido um póster de divulgação do projeto ESSDataLab (MIT-EXPL-ACC-0057-2021) numa conferência nacional, o qual foi distinguido na categoria *Student Poster* na área de pesquisa *Climate Science & Climate Change* e que disponibilizamos igualmente no Anexo B:

- Tiago F.R. Ribeiro (2023). *From Fire to Data: Capturing Wildfire Dynamics with Semantic Segmentation & Spatiotemporal Reconstruction*. Best Poster in Climate Science & Climate Change category. Braga, Portugal. DOI: 10.13140/RG.2.2.28400.64002. URL: https://mitportugal.mit.edu/poster-gallery/2023/student-posters

Agregamos as informações relevantes deste trabalho, incluindo hiperligações para os artigos científicos, o conjunto de dados, código-fonte, vídeos explicativos e diversos outros recursos, no sítio do projeto "ESSDataLab — Modelos de dados espaçotemporais e algoritmos para as ciências da Terra", acessível através do endereço https://eesdatalab.ipleiria.pt/.

6.4 DESAFIOS E TRABALHO FUTURO

No que respeita ao trabalho descrito no Capítulo 4, entendemos ser importante a expansão e enriquecimento do conjunto de dados BurnedAreadUAV através da inclusão de vídeos e anotações de capturas em locais e condições distintos, com o intuito de avaliar a robustez e as capacidades de generalização dos modelos.

Adicionalmente, consideramos de interesse futuro a exploração de novas técnicas de anotação automática de dados, bem como de métodos de segmentação semântica fracamente supervisionados, para os quais existe menos necessidade de dados anotados. A disponibilização de modelos generalistas pré-treinados de segmentação semântica de

grande escala (Kirillov, Mintun et al., 2023), vem abrir outra via possível de trabalho futuro que importa explorar.

No que concerne ao trabalho desenvolvido no Capítulo 5, achamos de grande interesse o ensaio com outras variantes de modelos AE, que façam face às limitações identificadas com o C-VAE. Do mesmo modo, para futuro parece-nos importante testar as capacidades dos modelos baseados em AE para gerar a evolução espaçotemporal numa gama mais vasta de fenómenos do mundo real. Finalmente, entendemos que a desafiante integração de dados multimodais para descrever fenómenos reais com modelos de aprendizagem profunda, afigura-se uma interessante área de estudo.

6.5 NOTAS FINAIS

Termino com uma breve nota pessoal que reflete o período no qual este trabalho foi desenvolvido.

Não me é possível quantificar a extensão dos conceitos, livros e artigos que consultei, o código que assimilei ou as conversas interessantes que mantive com os colegas de trabalho. As condições, o tempo — sobretudo o tempo —, a oportunidade e o privilégio que me foram concedidos estão, à partida, injustamente fora do alcance de uma parte significativa da população.

Por essa razão, reservo esta secção final para expressar o meu agradecimento a todas as pessoas que tornam possível a investigação, a educação, a ciência, a curiosidade e a aprendizagem.

BIBLIOGRAFIA

Abiodun, Oludare et al. (nov. de 2018). «State-of-the-art in artificial neural network applications: A survey». Em: *Heliyon* 4, e00938. DOI: 10.1016/j.heliyon.2018.e00938.

Abreu Alexandre Cancela dánd Correia, Teresa Pinto, Rosário Oliveira e Inês Magro (2004). «Contributos para a identificação e caracterização da paisagem em Portugal». Em: vol. II. Direcção Geral do Ordenamento do Território e Desenvolvimento Urbano. Cap. Trás-os-Montes, pp. 129–131.

Alzubaidi, Laith et al. (2021). «Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions». Em: *Journal of big Data* 8, pp. 1–74.

Avalhais, Letricia P.S., Jose Rodrigues e Agma J.M. Traina (2016). «Fire Detection on Unconstrained Videos Using Color-Aware Spatial Modeling and Motion Flow». Em: *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 913–920. DOI: 10.1109/ICTAI.2016.0141.

Baheti, Bhakti et al. (2020). «Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment». Em: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1473–1481. DOI: 10.1109/CVPRW50498.2020.00187.

Barbat, Mauro M. et al. (2019). «Three Years of Near-Coastal Antarctic Iceberg Distribution From a Machine Learning Approach Applied to SAR Imagery». Em: *Journal of Geophysical Research: Oceans* 124.9, pp. 6658–6672. DOI: https://doi.org/10.1029/2019JC015205.

Berthelot, David et al. (2018). «Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer». Em: *CoRR* abs/1807.07543. arXiv: 1807.07543.

Bishop, Christopher M e Nasser M Nasrabadi (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer.

Bouazizi, Khaoula et al. (2021). «Abdominal adipose tissue components quantification in MRI as a relevant biomarker of metabolic profile». Em: *Magnetic Resonance Imaging* 80, pp. 14–20.

Bradley, Derek e Gerhard Roth (jan. de 2007). «Adaptive Thresholding using the Integral Image». Em: *Journal of Graphics Tools* 12 (2), pp. 13–21. ISSN: 1086-7651. DOI: 10.1080/2151237X.2007.10129236.

Bradski, G. (2000). «The OpenCV Library». Em: *Dr. Dobb's Journal of Software Tools*.

Braovic, M., D. Stipanicev e D. Krstinic (2017). «Cogent Confabulation based Expert System for Segmentation and Classification of Natural Landscape Images». Em: *Advances in*

Electrical and Computer Engineering 17 (2), pp. 85–94. ISSN: 1582-7445. DOI: 10.4316/AECE.2017.02012.

C3S, Copernicus Climate Change Service (2023). *2022 was a year of climate extremes, with record high temperatures and rising concentrations of greenhouse gases*. [Accessed 08-Feb-2023].

Caesar, Holger, Jasper R. R. Uijlings e Vittorio Ferrari (2016). «COCO-Stuff: Thing and Stuff Classes in Context». Em: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1209–1218.

Cajal, Santiago Ramón y (1911). *Histologie du système nerveux de l'homme & des vertébrés: Cervelet, cerveau moyen, rétine, couche optique, corps strié, écorce cérébrale générale & régionale, grand sympathique*. Vol. 2. A. Maloine.

– (1894). *Y, Les Nouvelles Idées sur la fine anatomie des centres nerveux*.

Canny, John (1986). «A Computational Approach to Edge Detection». Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6, pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

Carreira, João e Andrew Zisserman (2017). «Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset». Em: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 4724–4733. DOI: 10.1109/CVPR.2017.502.

Cauchy, Augustin (1847). «Méthode générale pour la résolution des systemes d'équations simultanées». Em: *Comp. Rend. Sci. Paris* 25.1847, pp. 536–538.

Cazzolato, Mirela T et al. (2017). «FiSmo: a compilation of datasets from emergency situations for fire and smoke analysis». Em: SBC.

Cazzolato, Mirela T. et al. (2016). «Unveiling smoke in social images with the SmokeBlock approach». Em: *Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016*, pp. 49–54. DOI: 10.1145/2851613.2851634.

Certini, Giacomo (mar. de 2005). «Effects of fire on properties of forest soils: a review». Em: *Oecologia* 143 (1), pp. 1–10. ISSN: 0029-8549. DOI: 10.1007/s00442-004-1788-8.

Chen, Ping-Yang et al. (set. de 2019). «Smaller Object Detection for Real-Time Embedded Traffic Flow Estimation Using Fish-Eye Cameras». Em: pp. 2956–2960. DOI: 10.1109/ICIP.2019.8803719.

Chen, Zhaomin et al. (2018). «Autoencoder-based network anomaly detection». Em: *2018 Wireless telecommunications symposium (WTS)*. IEEE, pp. 1–5.

Cheng, Le et al. (2023). «Fabric defect detection based on separate convolutional UNet». Em: *Multimedia Tools and Applications* 82.2, pp. 3101–3122.

Cheng, Zhengxue et al. (2018). «Deep convolutional autoencoder-based lossy image compression». Em: *2018 Picture Coding Symposium (PCS)*. IEEE, pp. 253–257.

Child, J.M. (1920). *The Early Mathematical Manuscripts of Leibniz*. The Open Court Publishing Company.

Chino, Daniel Y. T. et al. (2015). «BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis». Em: *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 95–102. DOI: 10.1109/SIBGRAPI.2015.19.

Chuvieco, Emilio et al. (abr. de 2020). *ESA fire climate change initiative (FIRE CCI): Modis Fire cci burned area grid product, version 5.1*.

Çiçek, Özgün et al. (2016). «3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation». Em: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II*. Athens, Greece: Springer-Verlag, pp. 424–432. ISBN: 978-3-319-46722-1. DOI: 10.1007/978-3-319-46723-8_49.

Cireşan, Dan, Ueli Meier e Jürgen Schmidhuber (2012). «Multi-column deep neural networks for image classification». Em: *2012 IEEE conference on computer vision and pattern recognition*. IEEE, pp. 3642–3649.

Clevert, Djork-Arné, Thomas Unterthiner e Sepp Hochreiter (2016). «Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)». Em: arXiv: 1511.07289 [cs.LG].

Commission, European et al. (2022). *Forest Fires in Europe, Middle East and North Africa 2021*. Publications Office of the European Union. DOI: doi/10.2760/34094.

Costa, Rogério Luís et al. (2020). «Sampling strategies to create moving regions from real world observations». Em: *Proceedings of the 35th Annual ACM Symposium on Applied Computing (ACM SAC)*, pp. 609–616.

Costa, Rogério Luís C., Enrico Miranda, Paulo Dias et al. (jun. de 2020a). «Evaluating Preprocessing and Interpolation Strategies to Create Moving Regions from Real-World Observations». Em: *SIGAPP Appl. Comput. Rev.* 20.2, pp. 46–58. ISSN: 1559-6915. DOI: 10.1145/3412816.3412820.

– (2020b). «Sampling Strategies to Create Moving Regions from Real World Observations». Em: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. SAC '20. Brno, Czech Republic: Association for Computing Machinery, pp. 609–616. ISBN: 9781450368667. DOI: 10.1145/3341105.3374019.

– (jan. de 2021). «Experience: Quality Assessment and Improvement on a Forest Fire Dataset». Em: *J. Data and Information Quality* 13.1. ISSN: 1936-1955. DOI: 10.1145/3428155.

Costa, Rogério Luís C., Enrico Miranda e José Moreira (2020). «Towards the Automatic Selection of Moving Regions Representation Methods». Em: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation*. GeoSim '20. Seattle,

Washington: Association for Computing Machinery, pp. 60–63. ISBN: 9781450381611. DOI: 10.1145/3423335.3428170.

Costa, Rogério Luís de C. e José Moreira (2022). «Automatic Quality Improvement of Data on the Evolution of 2D Regions». Em: *Advanced Data Mining and Applications*. Ed. por Bohan Li et al. Cham: Springer International Publishing, pp. 288–300. ISBN: 978-3-030-95408-6.

Cristovao, Paulino et al. (2020). «Generating In-Between Images Through Learned Latent Space Representation Using Variational Autoencoders». Em: *IEEE Access* 8, pp. 149456–149467.

Al-Dabbagh, Ali e Muhammad Ilyas (2022). *Deep Learning and Remote Sensing Dataset For Turkey's Wildfire 2021 Multispectral Sentinel-2 Satellite Imagery*. DOI: 10.17632/hgctmx9y6c.1.

Dean, Jeffrey (2020). «The Deep Learning Revolution and Its Implications for Computer Architecture and Chip Design». Em: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, pp. 8–14.

Demir, Sumeyra et al. (2021). «Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting». Em: *Applied Energy* 304, p. 117695.

Direção-Geral do Território, DGT (nov. de 2019). *Carta de Uso e Ocupação do Solo, COS2018*.

Dosovitskiy, Alexey et al. (2020). «An image is worth 16x16 words: Transformers for image recognition at scale». Em: *arXiv preprint arXiv:2010.11929*.

Duarte, J., P. Dias e J. Moreira (2020). «On the Evaluation and Comparison of Region Interpolation Methods». Em: *AGILE: GIScience Series* 1, p. 3. DOI: 10.5194/agile-giss-1-3-2020.

Duarte, José, Paulo Dias e José Moreira (nov. de 2018). «An Evaluation of Smoothing and Remeshing Techniques to Represent the Evolution of Real-World Phenomena: 13th International Symposium, ISVC 2018, Las Vegas, NV, USA, November 19 – 21, 2018, Proceedings». Em: pp. 57–67. ISBN: 978-3-030-03800-7.

– (2023). «Approximating the evolution of rotating moving regions using Bezier curves». Em: *International Journal of Geographical Information Science* 37.4, pp. 839–863.

Duarte, José, Bruno Silva et al. (2019). «Towards a Qualitative Analysis of Interpolation Methods for Deformable Moving Regions». Em: SIGSPATIAL '19, pp. 592–595.

Dumoulin, Vincent e Francesco Visin (2016). «A guide to convolution arithmetic for deep learning». Em: *arXiv preprint arXiv:1603.07285*.

Eddy, Liyanto e Kohei Nagai (set. de 2021). «Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural Network: A Comparison Study Between U-Net and DeepLabV3+». Em: *Journal of the Civil Engineering Forum* 7, pp. 323–334. DOI: 10.22146/jcef.65288.

Ehrhardt, Jan, D Säring e H Handels (2007). «Structure-preserving interpolation of temporal and spatial image sequences using an optical flow-based method». Em: *Methods of Information in Medicine* 46.03, pp. 300–307.

Fayyaz, Mohsen et al. (2017). «STFCN: Spatio-Temporal Fully Convolutional Neural Network for Semantic Segmentation of Street Scenes». Em: *Computer Vision – ACCV 2016 Workshops*. Ed. por Chu-Song Chen, Jiwen Lu e Kai-Kuang Ma. Cham: Springer International Publishing, pp. 493–509.

Forlizzi, Luca et al. (2000). «A Data Model and Data Structures for Moving Objects Databases». Em: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 319–330.

Fukushima, Kunihiko (1988). «Neocognitron: A hierarchical neural network capable of visual pattern recognition». Em: *Neural networks* 1.2, pp. 119–130.

Fukushima, Kunihiko e Sei Miyake (1979). «A Self-Organizing Neural Network with a Function of Associative Memory–Feedback-Type Cognitron». Em: *NHK STRL* 13, p46–53.

Gaveau, David L. A. et al. (nov. de 2021). «Refined burned-area mapping protocol using Sentinel-2 data increases estimate of 2019 Indonesian burning». Em: *Earth System Science Data* 13 (11), pp. 5353–5368. ISSN: 1866-3516. DOI: 10.5194/essd-13-5353-2021.

Géron, Aurélien (2019). *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books.

Gestão do Programa de Fogos Rurais - DAGFR, Divisão de (set. de 2022). *5.º Relatório Provisório de Incêndios Rurais – 2022*.

Glorot, Xavier e Yoshua Bengio (2010). «Understanding the difficulty of training deep feedforward neural networks». Em: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop e Conference Proceedings, pp. 249–256.

Golgi, C (1873). «Sulla struttura della sostanza grigia del cervelo. Gazzetta Medica Italiana». Em: *Lombardia* 33, p. 244.

Gonzalez, Rafael C. e Richard E. Woods (2008). *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall.

Goodfellow, Ian, Yoshua Bengio e Aaron Courville (2016). *Deep learning*. MIT press.

Gottschalk, Stefan, Ming C Lin e Dinesh Manocha (1996). «OBBTree: A hierarchical structure for rapid interference detection». Em: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 171–180.

Gudise, Venu G e Ganesh K Venayagamoorthy (2003). «Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks». Em: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*. IEEE, pp. 110–117.

Guerrero Tello, José Francisco et al. (2022). «Convolutional Neural Network Algorithms for Semantic Segmentation of Volcanic Ash Plumes Using Visible Camer Imagery». Em: *Remote Sensing* 14.18. ISSN: 2072-4292. DOI: 10.3390/rs14184477.

Hadamard, Jacques (1908). «Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées». Em: vol. 33. Imprimerie nationale.

Hamdi, Ali et al. (2022). «Spatiotemporal data mining: a survey on challenges and open problems». Em: *Artificial Intelligence Review*, pp. 1–48.

Hanson, Allen (1978). *Computer vision systems*. Elsevier.

Hatamizadeh, Ali et al. (jan. de 2022). «UNETR: Transformers for 3D Medical Image Segmentation». Em: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 574–584.

He, Kaiming et al. (2015). «Delving deep into rectifiers: Surpassing human-level performance on imagenet classification». Em: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hebb, Donald (1949). «The Organization of Behavior – A Neuropsychological Theory». Em: John Wiley & Sons, Inc.

Heinz, Florian e Ralf Güting (jul. de 2016). «Robust high-quality interpolation of regions to moving regions». Em: *GeoInformatica* 20.

Heinz, Florian e Ralf Hartmut Güting (2020). «A polyhedra-based model for moving regions in databases». Em: *International Journal of Geographical Information Science* 34.1, pp. 41–73.

Hellström, Ian (2020). *A Brief History of Machine Learning Platforms — databaseline.tech*. https://databaseline.tech/a-brief-history-of-ml-platforms/. [Accessed 07-08-2023].

Herman, G.T., J. Zheng e C.A. Bucholtz (1992). «Shape-based interpolation». Em: *IEEE Computer Graphics and Applications* 12.3, pp. 69–79.

Hesamian, Mohammad Hesam et al. (ago. de 2019). «Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges». Em: *Journal of Digital Imaging* 32 (4), pp. 582–596. ISSN: 0897-1889. DOI: 10.1007/s10278-019-00227-x.

Hinton, Geoffrey (2022). «The Forward-Forward Algorithm: Some Preliminary Investigations». Em: arXiv: 2212.13345 [cs.LG].

Hinton, Geoffrey E. e Richard S. Zemel (1993). «Autoencoders, Minimum Description Length and Helmholtz Free Energy». Em: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. NIPS'93. Denver, Colorado: Morgan Kaufmann Publishers Inc., pp. 3–10.

Hochreiter, Sepp e Jürgen Schmidhuber (nov. de 1997). «Long Short-Term Memory». Em: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

Hoeser, Thorsten e Claudia Kuenzer (2020). «Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends». Em: *Remote Sensing* 12.10, p. 1667.

Hoinka, Klaus P., A. Carvalho e Ana Isabel Miranda (2009). «Regional-scale weather patterns and wildland fires in central Portugal». Em: *International Journal of Wildland Fire* 18, pp. 36–49.

Hopfield, Jonh (1982). «Neural networks and physical systems with emergent collective computational abilities.» Em: *Proceedings of the National Academy of Sciences* 79.8, pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554.

Hornik, Kurt, Maxwell Stinchcombe e Halbert White (1989). «Multilayer feedforward networks are universal approximators». Em: *Neural networks* 2.5, pp. 359–366.

Hou, Rui et al. (2019). «An Efficient 3D CNN for Action/Object Segmentation in Video». Em: *ArXiv* abs/1907.08895.

Huang, Lei et al. (fev. de 2023). «Normalization Techniques in Training DNNs: Methodology, Analysis and Application». Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, pp. 1–20. DOI: 10.1109/TPAMI.2023.3250241.

Hubel, David H e Torsten N Wiesel (1959). «Receptive fields of single neurones in the cat's striate cortex». Em: *The Journal of physiology* 148.3, p. 574.

– (1968). «Receptive fields and functional architecture of monkey striate cortex». Em: *The Journal of physiology* 195.1, pp. 215–243.

Huttenlocher, D.P., G.A. Klanderman e W.J. Rucklidge (1993). «Comparing images using the Hausdorff distance». Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9, pp. 850–863.

Ioffe, Sergey e Christian Szegedy (2015). «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift». Em: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 448–456.

Jakovcevic, Toni e Damir Krstinic (nov. de 2010). *Image and video databases*.

Jian, S et al. (2016). «Deep residual learning for image recognition». Em: *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778.

Kamnitsas, Konstantinos et al. (2016). «DeepMedic for Brain Tumor Segmentation». Em: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. por Alessandro Crimi et al. Cham: Springer International Publishing, pp. 138–149. ISBN: 978-3-319-55524-9.

Khan, Ali et al. (abr. de 2022). «DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection». Em: *Mobile Information Systems* 2022, pp. 1–14. ISSN: 1875-905X. DOI: 10.1155/2022/5358359.

Kilkenny, Monique F e Kerin M Robinson (2018). «Data quality:"Garbage in–garbage out"». Em: *Health Information Management Journal* 47.3, pp. 103–105.

Kim, Yoon-Ho, Alla Kim e Hwa-Young Jeong (abr. de 2014). «RGB Color Model Based the Fire Detection Algorithm in Video Sequences on Wireless Sensor Network». Em: *International Journal of Distributed Sensor Networks* 2014, pp. 1–10. DOI: 10.1155/2014/923609.

Kingma, Diederik P. e Max Welling (2014). «Auto-Encoding Variational Bayes». Em: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. por Yoshua Bengio e Yann LeCun. URL: http://arxiv.org/abs/1312.6114.

Kirillov, Alexander, Kaiming He et al. (2019). «Panoptic Segmentation». Em: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9396–9405. DOI: 10.1109/CVPR.2019.00963.

Kirillov, Alexander, Eric Mintun et al. (2023). *Segment Anything*. arXiv: 2304.02643 [cs.CV].

Koo, YoungHyun (out. de 2021). *YoungHyunKoo/GEE_iceberg_tracking: GEE-based tracking of iceberg B43*. Versão v1.0.0. DOI: 10.5281/zenodo.5550530.

Krähenbühl, Philipp e Vladlen Koltun (2011). «Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials». Em: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., pp. 109–117. ISBN: 9781618395993.

Krizhevsky, Alex, Ilya Sutskever e Geoffrey E Hinton (2012a). «ImageNet Classification with Deep Convolutional Neural Networks». Em: *Advances in Neural Information Processing Systems*. Ed. por F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

– (2012b). «ImageNet Classification with Deep Convolutional Neural Networks». Em: *Advances in Neural Information Processing Systems*. Ed. por F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kumar, Siddharth Krishna (2017). «On weight initialization in deep neural networks». Em: arXiv: 1704.08863 [cs.LG].

Kundu, Abhijit, Vibhav Vineet e Vladlen Koltun (2016). «Feature Space Optimization for Semantic Video Segmentation». Em: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3168–3175. DOI: 10.1109/CVPR.2016.345.

L'Hospital, Guillaume François Antoine Marquis de (1696). «Analyse des infiniment petits, pour l'intelligence des lignes courbes». Em: François Montalant.

Lagrange, J.L. et al. (1797). «Œuvres de Lagrange: Théorie des fonctions analytiques, contenant les principes du calcul différentiel, dégagés de toute considération d'infiniment petits ou d'évanouissans, de limites ou de fluxions et réduits á l'analyse algébrique des quantités finies.» Em: Imprimerie de la République.

Le Cun, Yann et al. (1989). «Handwritten digit recognition: Applications of neural network chips and automatic learning». Em: *IEEE Communications Magazine* 27.11, pp. 41–46.

LeCun, Yann et al. (1998). «Gradient-based learning applied to document recognition». Em: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Tong-Yee e Chao-Hung Lin (2002). «Feature-guided shape-based image interpolation». Em: *IEEE Transactions on Medical Imaging* 21.12, pp. 1479–1489.

Li, Jiang e R.A. Narayanan (2002). «Shape-based change detection and information mining in remote sensing». Em: *IEEE International Geoscience and Remote Sensing Symposium.* Vol. 2, 1035–1037 vol.2. DOI: 10.1109/IGARSS.2002.1025767.

Lin, Tsung-Yi et al. (2017). «Feature Pyramid Networks for Object Detection». Em: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

Linnainmaa, Seppo (jun. de 1970). «Algoritmin kumulatiivinen pyöritysvirhe yksittäisten pyöritysvirheiden Taylor-kehityksenä». Available at https://people.idsia.ch/~juergen/linnainmaa1970thesis.pdf. Tese de mestrado. University of Helsinki.

Liu, Yifan et al. (2020). «Efficient Semantic Video Segmentation with Per-Frame Inference». Em: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*. Glasgow, United Kingdom: Springer-Verlag, pp. 352–368. ISBN: 978-3-030-58606-5. DOI: 10.1007/978-3-030-58607-2_21.

Long, Jonathan, Evan Shelhamer e Trevor Darrell (2015). «Fully convolutional networks for semantic segmentation». Em: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

Luca, Colomba et al. (mai. de 2022). *Satellite Burned Area Dataset*. DOI: 10.5281/zenodo.6597139.

Mahadevan, Sabarinath et al. (2020). «Making a Case for 3D Convolutions for Object Segmentation in Videos». Em: *ArXiv* abs/2008.11516.

Mansilha, Catarina et al. (mar. de 2019). «Impact of wildfire on water quality in Caramulo Mountain ridge (Central Portugal)». Em: *Sustainable Water Resources Management* 5 (1), pp. 319–331. ISSN: 2363-5037. DOI: 10.1007/s40899-017-0171-y.

Markewich, Logan et al. (2022). «Segmentation for document layout analysis: not dead yet». Em: *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–11.

McCulloch, W. S. e W. Pitts (1943). «A Logical Calculus of Ideas Immanent in Nervous Activity». Em: *Bulletin of Mathematical Biophysics* 5.

Mckenney, Mark e Roger Frye (jul. de 2015). «Generating Moving Regions from Snapshots of Complex Regions». Em: *ACM Trans. Spatial Algorithms Syst.* 1.1.

McKenney, Mark et al. (2016). «Pyspatiotemporalgeom: A Python Library for Spatiotemporal Types and Operations». Em: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPACIAL '16. ISBN: 9781450345897.

Mckennney, Mark e Roger Frye (2015). «Generating Moving Regions from Snapshots of Complex Regions». Em: 1.1. ISSN: 2374-0353.

Mendez, Martin O et al. (2020). «Assisted quantification of abdominal adipose tissue based on magnetic resonance images». Em: *Multimedia Tools and Applications* 79, pp. 1519–1534.

Mi, Lu et al. (2021). «Revisiting Latent-Space Interpolation via a Quantitative Evaluation Framework». Em: *ArXiv* abs/2110.06421.

Minaee, Shervin et al. (2021). «Image Segmentation Using Deep Learning: A Survey». Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2021.3059968.

Minsky, Marvin e Seymour A Papert (2017). *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. MIT press.

Molina-Terrén, Domingo M. et al. (2019). «Analysis of forest fire fatalities in Southern Europe: Spain, Portugal, Greece and Sardinia (Italy)». Em: *International Journal of Wildland Fire* 28 (2), p. 85. ISSN: 1049-8001. DOI: 10.1071/WF18004.

Moreira, José, Paulo Dias e Pedro Amaral (2016). «Representation of continuously changing data over time and space: Modeling the shape of spatiotemporal phenomena». Em: *IEEE 12th International Conference on e-Science (e-Science)*, pp. 111–119.

Moreira, José, José Duarte e Paulo Dias (2019). «Modeling and Representing Real-World Spatio-Temporal Data in Databases (Vision Paper)». Em: *14th International Conference on Spatial Information Theory (COSIT 2019)*. Ed. por Sabine Timpf et al. Vol. 142. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 6:1–6:14. ISBN: 978-3-95977-115-3. DOI: 10.4230/LIPIcs.COSIT.2019.6.

Moreno, M. Vanesa et al. (2014). «Fire regime changes and major driving forces in Spain from 1968 to 2010». Em: *Environmental Science & Policy* 37, pp. 11–22. ISSN: 1462-9011. DOI: https://doi.org/10.1016/j.envsci.2013.08.005.

Mou, James e Jun Li (2020). «Effects of Number of Filters of Convolutional Layers on Speech Recognition Model Accuracy». Em: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 971–978. DOI: 10.1109/ICMLA51294.2020.00158.

Mouelhi, A. et al. (2020). «Fire Tracking in Video Sequences Using Geometric Active Contours Controlled by Artificial Neural Network». Em: *2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 338–343.

Nameirakpam, Dhanachandra e Yambem Chanu (jan. de 2017). «A Survey on Image Segmentation Methods using Clustering Techniques». Em: *European Journal of Engineering Research and Science* 2, p. 15. DOI: 10.24018/ejers.2017.2.1.237.

Neal, Radford M (1992). «Bayesian Learning via Stochastic Dynamics». Em: *Advances in Neural Information Processing Systems*. Ed. por S. Hanson, J. Cowan e C. Giles. Vol. 5. Morgan-Kaufmann.

– (mar. de 1995). «Bayesian Learning for Neural Networks, Thesis (Ph.D.)., University of Toronto». Tese de doutoramento.

Nemoto, Takafumi et al. (fev. de 2020). «Efficacy evaluation of 2D, 3D U-Net semantic segmentation and atlas-based segmentation of normal lungs excluding the trachea and main bronchi». Em: *Journal of Radiation Research* 61.2, pp. 257–264. ISSN: 1349-9157. DOI: 10.1093/jrr/rrz086.

Nguyen, Anh, Jason Yosinski e Jeff Clune (2019). «Understanding neural networks via feature visualization: A survey». Em: *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 55–76.

Nie, Dong et al. (2016). «Fully convolutional networks for multi-modality isointense infant brain image segmentation». Em: *2016 IEEE 13Th international symposium on biomedical imaging (ISBI)*. IEEE, pp. 1342–1345.

Ojha, Shipra e Sachin Sakhare (2015). «Image processing techniques for object tracking in video surveillance- A survey». Em: *2015 International Conference on Pervasive Computing (ICPC)*, pp. 1–6. DOI: 10.1109/PERVASIVE.2015.7087180.

Oktay, Ozan et al. (2018). «Attention u-net: Learning where to look for the pancreas». Em: *arXiv preprint arXiv:1804.03999*.

Oliveira, Sofia Ares, Benoit Seguin e Frederic Kaplan (2018). «dhSegment: A generic deep-learning approach for document segmentation». Em: *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, pp. 7–12.

Oring, Alon, Zohar Yakhini e Yacov Hel-Or (2021). «Autoencoder Image Interpolation by Shaping the Latent Space». Em: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139, pp. 8281–8290.

Otsu, Nobuyuki (1979). «A Threshold Selection Method from Gray-Level Histograms». Em: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1, pp. 62–66. DOI: 10.1109/TSMC.1979.4310076.

Parente, J. et al. (mai. de 2018). «Negligent and intentional fires in Portugal: Spatial distribution characterization». Em: *Science of The Total Environment* 624, pp. 424–437. ISSN: 00489697. DOI: 10.1016/j.scitotenv.2017.12.013.

Paveglio, Travis B., Catrin M. Edgeley e Amanda M. Stasiewicz (2018). «Assessing influences on social vulnerability to wildfire using surveys, spatial data and wildfire simulations». Em: *Journal of Environmental Management* 213, pp. 425–439. ISSN: 0301-4797. DOI: https://doi.org/10.1016/j.jenvman.2018.02.068.

Pessôa, Ana Carolina M. et al. (2020). «Intercomparison of Burned Area Products and Its Implication for Carbon Emission Estimations in the Amazon». Em: *Remote Sensing* 12.23. ISSN: 2072-4292. DOI: 10.3390/rs12233864.

Ramachandran, Prajit, Barret Zoph e Quoc V. Le (2018). «Searching for Activation Functions». Em: *ArXiv* abs/1710.05941. URL: https://api.semanticscholar.org/CorpusID:10919244.

Raschka, Sebastian, David Julian e John Hearty (2016). *Python: deeper insights into machine learning*. Packt Publishing Ltd.

Rezende, Danilo Jimenez, Shakir Mohamed e Daan Wierstra (2014). «Stochastic Backpropagation and Approximate Inference in Deep Generative Models». Em: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, II–1278–II–1286.

Ribeiro, Tiago F. R., Fernando Silva e Rogério Luís de C. Costa (2023). «Reconstructing Spatiotemporal Data with C-VAEs». Em: *Advances in Databases and Information Systems*. Ed. por Alberto Abelló et al. Cham: Springer Nature Switzerland, pp. 59–73. ISBN: 978-3-031-42914-9. DOI: 10.1007/978-3-031-42914-9_5.

Ribeiro, Tiago F. R., Fernando Silva, José Moreira et al. (mai. de 2023). *BurnedAreaUAV Dataset (v1.1)*. Versão 1.1. DOI: 10.5281/zenodo.7944963.

Ribeiro, Tiago F.R. (2023). *From Fire to Data: Capturing Wildfire Dynamics with Semantic Segmentation & Spatiotemporal Reconstruction*. Best Poster in Climate Science & Climate Change category. Braga, Portugal. DOI: 10.13140/RG.2.2.28400.64002. URL: https://mitportugal.mit.edu/poster-gallery/2023/student-posters.

Ribeiro, Tiago F.R. et al. (2023). «Burned area semantic segmentation: A novel dataset and evaluation using convolutional networks». Em: *ISPRS Journal of Photogrammetry and Remote Sensing* 202, pp. 565–580. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2023.07.002.

Ronneberger, Olaf, Philipp Fischer e Thomas Brox (2015). «U-Net: Convolutional Networks for Biomedical Image Segmentation». Em: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. por Nassir Navab et al. Cham: Springer International Publishing, pp. 234–241. ISBN: 978-3-319-24574-4.

Rosch, Eleanor (1978). «Principles of Categorization». Em: ed. por Eleanor Rosch e B. B. Lloyd, pp. 27–48.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Cornell Aeronautical Laboratory. Report no. VG-1196-G-8. Spartan Books. URL: https://books.google.pt/books?id=7FhRAAAAMAAJ.

Rosenblatt, Frank (1958). «The perceptron: a probabilistic model for information storage and organization in the brain.» Em: *Psychological review* 65.6, p. 386.

Rumelhart, David E., Geoffrey E. Hinton e Ronald J. Williams (1986). «Learning representations by back-propagating errors». Em: *Nature* 323, pp. 533–536.

Russell, Bryan C. et al. (mai. de 2008). «LabelMe: A Database and Web-Based Tool for Image Annotation». Em: *International Journal of Computer Vision* 77 (1-3), pp. 157–173. ISSN: 0920-5691. DOI: 10.1007/s11263-007-0090-8.

Schenk, Andrea, Guido Prause e Heinz-Otto Peitgen (2000). «Efficient Semiautomatic Segmentation of 3D Objects in Medical Images». Em: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2000*, pp. 186–195.

Shamsolmoali, Pourya et al. (2019). «A novel deep structure U-Net for sea-land segmentation in remote sensing images». Em: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12.9, pp. 3219–3232.

Shamsoshoara, Alireza et al. (2021). «Aerial imagery pile burn detection using deep learning: The FLAME dataset». Em: *Computer Networks* 193, p. 108001. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2021.108001.

Shelhamer, Evan et al. (2016). «Clockwork Convnets for Video Semantic Segmentation». Em: *Computer Vision – ECCV 2016 Workshops*. Ed. por Gang Hua e Hervé Jégou. Cham: Springer International Publishing, pp. 852–868. ISBN: 978-3-319-49409-8.

Shlens, Jonathon (2014). «Notes on kullback-leibler divergence and likelihood». Em: *arXiv preprint arXiv:1404.2000*.

Silva, Catarina e Bernardete Ribeiro (2018). *Aprendizagem Computacional em Engenharia*. Imprensa da Universidade de Coimbra/Coimbra University Press.

Simonyan, Karen e Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV].

Sistu., Ganesh, Sumanth Chennupati. e Senthil Yogamani. (2019). «Multi-stream CNN based Video Semantic Segmentation for Automated Driving». Em: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019) - Volume 5: VISAPP*. INSTICC. SciTePress, pp. 173–180. ISBN: 978-989-758-354-4. DOI: 10.5220/0007248401730180.

SNIRH (2019). *MINAS DE JALES (05L/02C), Sistema Nacional de Informação de Recursos Hídricos (SNIRH): dados de base*. [Accessed 19-may-2023]. URL: https://snirh.apambiente.pt/index.php?idRef=MTIyMw==&FILTRA_BACIA=12&FILTRA_COVER=920123704&FILTRA_SITE=920685484.

Sohn, Kihyuk, Honglak Lee e Xinchen Yan (2015). «Learning Structured Output Representation using Deep Conditional Generative Models». Em: *Advances in Neural Information Processing Systems*. Ed. por C. Cortes et al. Vol. 28. Curran Associates, Inc.

Srivastava, Nitish et al. (jan. de 2014). «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». Em: *J. Mach. Learn. Res.* 15.1, pp. 1929–1958. ISSN: 1532-4435.

Stanley, Kenneth O e Risto Miikkulainen (2002). «Evolving neural networks through augmenting topologies». Em: *Evolutionary computation* 10.2, pp. 99–127.

Strudel, Robin et al. (2021). «Segmenter: Transformer for Semantic Segmentation». Em: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7242–7252. DOI: 10.1109/ICCV48922.2021.00717.

Sun, Ke et al. (2019). «Deep High-Resolution Representation Learning for Human Pose Estimation». Em: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5686–5696. DOI: 10.1109/CVPR.2019.00584.

Teichmann, Marvin et al. (2018). «MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving». Em: *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1013–1020. DOI: 10.1109/IVS.2018.8500504.

Tøssebro, Erlend e Ralf Hartmut Güting (2001). «Creating Representations for Continuously Moving Regions from Observations». Em: *Advances in Spatial and Temporal Databases*, pp. 321–344.

Toulouse, Tom et al. (set. de 2017). «Computer vision for wildfire research: An evolving image dataset for processing and analysis». Em: *Fire Safety Journal* 92, pp. 188–194. ISSN: 03797112. DOI: 10.1016/j.firesaf.2017.06.012.

Ulku, Irem e Erdem Akagündüz (2022). «A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images». Em: *Applied Artificial Intelligence*. ISSN: 10876545. DOI: 10.1080/08839514.2022.2032924.

Vaswani, Ashish et al. (2017). «Attention is all you need». Em: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Villela, Karina et al. (2018). «Reliable and Smart Decision Support System for Emergency Management Based on Crowdsourcing Information». Em: *Exploring Intelligent Decision Support Systems: Current State and New Trends*. Cham: Springer International Publishing. Cap. Exploring Intelligent Decision Support Systems, pp. 177–198. ISBN: 978-3-319-74002-7. DOI: 10.1007/978-3-319-74002-7_9.

Wang, Ching-Wei et al. (2016). «A benchmark for comparison of dental radiography analysis algorithms». Em: *Medical Image Analysis* 31, pp. 63–76. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2016.02.004.

Werbos, Paul (jan. de 1974). «Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph.D.). Appl. Math. Harvard University». Tese de doutoramento.

Widrow, Bernard e Marcian E. Hoff (1960). «Adaptive Switching Circuits». Em: *1960 IRE WESCON Convention Record, Part 4*. New York: IRE, pp. 96–104.

Wiener, Norbert (1948). *Cybernetics: or Control and Communication in the Animal and the Machine*. 2ª ed. Cambridge, MA: MIT Press.

Wilkes, T.C., T.D. Pering e A.J.S. McGonigle (2022). «Semantic segmentation of explosive volcanic plumes through deep learning». Em: *Computers & Geosciences* 168, p. 105216. ISSN: 0098-3004. DOI: https://doi.org/10.1016/j.cageo.2022.105216.

Wong, Kit, Rolf Dornberger e Thomas Hanne (nov. de 2022). «An analysis of weight initialization methods in connection with different activation functions for feedforward neural networks». Em: *Evolutionary Intelligence*. DOI: 10.1007/s12065-022-00795-y.

Woo, Boyeong e Myungeun Lee (2021). «Comparison of tissue segmentation performance between 2D U-Net and 3D U-Net on brain MR Images». Em: *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4. DOI: 10.1109/ICEIC51217.2021.9369797.

Woo, Sanghyun et al. (2023). «Convnext v2: Co-designing and scaling convnets with masked autoencoders». Em: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16133–16142.

Xie, W. et al. (2020). «SegCloud: a novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation». Em: *Atmospheric Measurement Techniques* 13.4, pp. 1953–1961. DOI: 10.5194/amt-13-1953-2020.

Xu, Bing et al. (2015). «Empirical evaluation of rectified activations in convolutional network». Em: *arXiv preprint arXiv:1505.00853*.

Xu, Zhiyong et al. (dez. de 2020). «HRCNet: High-Resolution Context Extraction Network for Semantic Segmentation of Remote Sensing Images». Em: *Remote Sensing* 13. DOI: 10.3390/rs13010071.

Yan, Xinchen et al. (2016). «Attribute2Image: Conditional Image Generation from Visual Attributes». Em: arXiv: 1512.00570 [cs.LG].

Yang, Aqing et al. (2018). «High-accuracy image segmentation for lactating sows using a fully convolutional network». Em: *Biosystems Engineering* 176, pp. 36–47. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2018.10.005.

Yu, Shaode et al. (ago. de 2020). «Robustness study of noisy annotation in deep learning based medical image segmentation». Em: *Physics in Medicine & Biology* 65.17, p. 175007. DOI: 10.1088/1361-6560/ab99e5.

Zettler, Nico e Andre Mastmeyer (2021a). «Comparison of 2D vs. 3D U-Net Organ Segmentation in abdominal 3D CT images». Em: *arXiv preprint arXiv:2107.04062*.

– (2021b). «Comparison of 2D vs. 3D U-Net Organ Segmentation in abdominal 3D CT images». Em: *arXiv preprint arXiv:2107.04062*.

Zhou, Tianfei et al. (2021). *A Survey on Deep Learning Technique for Video Segmentation.* DOI: 10.48550/ARXIV.2107.01153.

Zhu, Xizhou et al. (2017). «Deep Feature Flow for Video Recognition». Em: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4141–4150. DOI: 10.1109/CVPR.2017.441.

ANEXOS

A

RESENHA HISTÓRICA DAS REDES NEURONAIS ARTIFICIAIS

A busca pela panaceia da inteligência artificial, baseada na emulação dos processos do sistema nervoso humano, não é recente. Os poderosos modelos que vão permeando muitos aspetos do quotidiano estão assentes numa longa história de avanços incrementais. À semelhança de muitos outros campos do conhecimento, não é fácil nem talvez seja justo identificar um único momento fundacional. Por outro lado, tentar reconstruir o trajeto percorrido até hoje, será inevitavelmente um exercício incompleto e redutor. Reconhecendo estas limitações, fazemos de seguida um levantamento de algumas contribuições importantes, o qual está sintetizado no cronograma da Figura A.1.

Os fundamentos teóricos das ANN remontam talvez ao século XVII e XVIII, com os trabalho desenvolvido por Leibniz (Child, 1920), L'Hospital (1696) ou Lagrange et al. (1797) no desenvolvimento do cálculo diferencial, bem como a descrição da regra da cadeia, fundamental para o treino da grande maioria das ANN atuais.

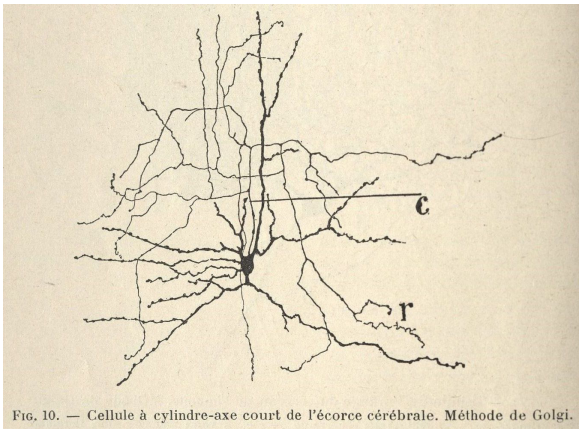


Fig. 10. — Cellule à cylindre-axe court de l'écorce cérébrale. Méthode de Golgi.

Figura A.2: "Um neurónio com um axónio curto no córtex cerebral. Método de Golgi". Figura do livro "Histologie du système nerveux de l'homme & des vertébrés" de S. R. y. Cajal (1911).

A invenção do método do gradiente, outro elemento importante para a otimização das redes neuronais atuais (que descrevemos na Secção 2.1.4), é atribuída a Augustin Cauchy (1847) em meados do século XIX, e, mais tarde, a Jacques Hadamard (1908). Em paralelo, no final do século XIX e início do século XX, avanços na compreensão da estrutura dos neurónios e da plasticidade neuronal, resultantes da observação histológica de células nervosas do cérebro feitas por Camillo Golgi (1873) e Santiago Ramón y S. R. y. Cajal (1894), lançaram as bases e inspiraram autores a criar modelos neuronais artificiais (Figura A.2).

Já no segundo quarto do século XX, McCulloch e Pitts (1943) publicaram o artigo *A Logical Calculus of the Ideas Immanent in Nervous Activity*, no qual eles introduziram o conceito de neurónio lógico. Neste trabalho é proposto um modelo matemático simplificado

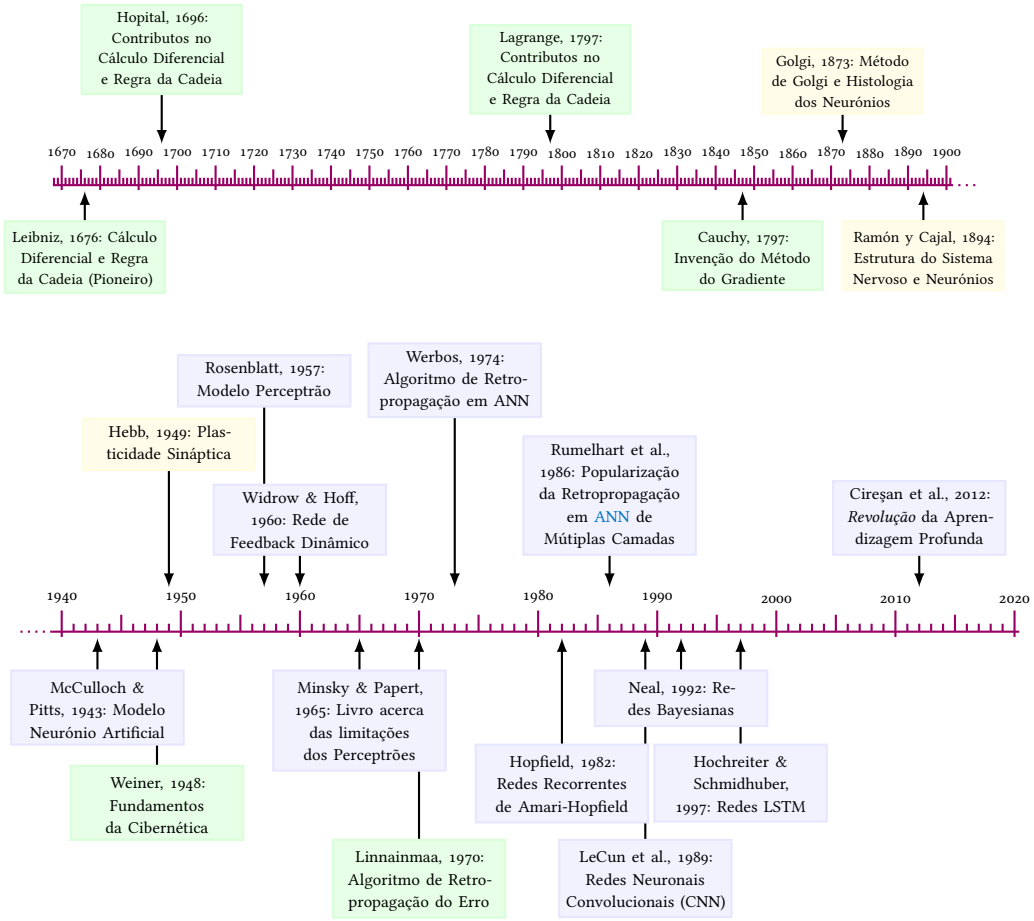


Figura A.1: **Cronograma de contribuições para o desenvolvimento das ANN**. As caixas verdes representam desenvolvimentos de natureza matemático relacionados com as ANN. As caixas amarelas representam avanços na compreensão do sistema nervoso e do cérebro biológico. As caixas azuis representam inovações na modelação de ANN, especificamente.

do neurónio biológico capaz de implementar funções lógicas, que serviu de base para o desenvolvimento de modelos mais sofisticados de neurónios artificiais.

No final da década de 40, Wiener (1948) publicou o livro *Cybernetics: Or Control and Communication in the Animal and the Machine* que estabeleceu os fundamentos teóricos da Cibernética, o estudo de sistemas de controlo e comunicação em seres vivos e máquinas. As abordagens propostas por Wiener influenciaram o desenvolvimento das ANN ao introduzir conceitos de retroalimentação e controlo que foram posteriormente aplicados para projetar modelos adaptativos de aprendizagem automática.

Na mesma década, Donald Hebb (1949) descreve um mecanismo fundamental de plasticidade sináptica nas redes neuronais biológicas, no qual descreve o processo de ativação simultânea das células quer leva ao fortalecimento das conexões sinápticas. A teoria de Hebb é amplamente utilizada para explicar a aprendizagem associativa e a formação de

memórias nas redes neuronais biológicas e inspirou o desenvolvimento de modelos de ANN capazes de aprender e armazenar informações.

No decorrer da década de 1950, baseando também no trabalho McCulloch e Pitts e aproveitando os desenvolvimentos dos primeiros computadores comerciais, Frank Rosenblatt (1958) desenvolveu o modelo Perceptrão de camada única, um tipo de rede neural que funciona como um classificador binário capaz de ajustar os pesos de ligações entre dos neurónios e a reconhecer padrões (Figura A.4).

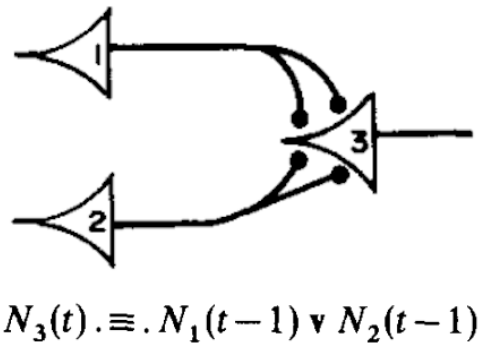


Figura A.3: **Função lógica OU representada por três neurónios**. O neurónio 3 dispara se pelo menos um dos neurónios 1 e 2 estiver ativo. Adaptado de McCulloch e Pitts (1943)

Durante a década de 1960 surgiu a Rede de Feedback Dinâmico de Widrow e Hoff (1960) que se caracteriza por ser uma rede de camada única que ajusta os seus pesos pela Regra Delta Generalizada (LMS), um caso especial do método do gradiente.

Porém, foi somente na década seguinte que Linnainmaa (1970) descreveu o algoritmo de Retropropagação do Erro em redes conectadas na sua tese de mestrado. Decorridos alguns anos, Paul Werbos (1974) descreve pela primeira vez o processo de treino de ANN por meio do algoritmo de Retropropagação do Erro na sua tese de doutoramento. Este algoritmo permitiu o treino eficiente das redes neuronais de múltiplas camadas e abriu caminho para o desenvolvimento de redes neuronais profundas[1].

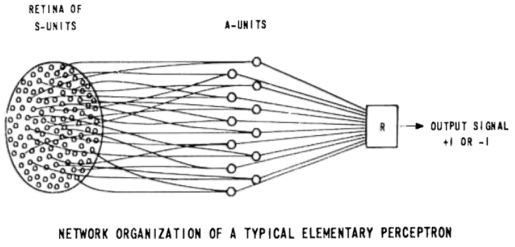


Figura A.4: **Perceptrão de Rosenblatt**. Neste modelo de rede neural as *S-units* representam as unidades sensoriais que recebem padrões de iluminação duma 'retina' simulada. As *A-units* são as unidades de associação, responsáveis por processar as informações das *S-units* e transmitir os sinais para a *R-unit*. Por sua vez, a R-unit produz a saída do Perceptrão. Adaptado de F. Rosenblatt (1962)

Na década subsequente, Hopfield (1982) populariza a rede Amari-Hopfield, um modelo de ANN recorrente capaz de armazenar e recuperar padrões. As redes recorrentes possuem conexões retroalimentadas entre as camadas que permitem que a informação flua de estados anteriores para os próximos e assim manter uma memória implícita dos dados de entrada precedentes.

1 Redes neuronais profundas são redes compostas por várias camadas sucessivas de neurónios interconectados. O termo "profundo" refere-se às camadas ocultas entre a entrada e a saída. Essa estrutura em camadas permite que as redes neuronais aprendam representações hierárquicas e abstratas dos dados.

Em 1986, o termo *Backpropagation* (ou, em português, algoritmo de Retropropagacão do Erro) e a sua utilização geral em ANN de múltiplas camadas foi popularizado por Rumelhart et al. (1986). Desde então, o uso cada vez mais generalizado deste método de otimização tem contribuído para o avanço das ANN, impulsionando significativamente o treino de redes neuronais profundas e encontrando cada vez mais aplicações práticas.

Já nos anos 90, são propostas as Redes Bayesianas (Neal, 1992; Neal, 1995), uma nova classe de modelos gráficos probabilísticos que usam grafos acíclicos direcionados para representar relações de dependência entre variáveis aleatórias. Cada nó representa uma variável aleatória, que pode ser uma observação ou um evento, enquanto as arestas denotam relações probabilísticas diretas entre essas variáveis. Essas relações são estabelecidas por meio da aplicação da Teorema de Bayes, que permite quantificar a probabilidade condicional, ou seja, a probabilidade de uma variável ocorrer dado o conhecimento das outras variáveis. Este modo mostrou-se fundamental, pois permitir expressar e explorar incertezas e relações de causa e efeito entre as variáveis num contexto probabilístico.

Em 1997, Hochreiter e Schmidhuber (1997) publicaram um relatório técnico que descreve a rede Memória Longa a Curto Prazo (LSTM) como uma solução para o problema do desvanecimento ou dissipação do gradiente[2]. Esta arquitetura de ANN recorrente foi um avanço importante neste campo, por permitir que as ANN aprendam e retenham informações relevantes de sequências longas de dados. Isso tornou as LSTM especialmente adequadas para tarefas que envolvem séries temporais e processamento de texto, por exemplo.

Em 1998, LeCun et al. (1998) publicaram um estudo pioneiro que demonstrou o potencial das Redes Neuronais Convolucionais (CNN), projetadas especialmente para lidar com a variabilidade de formas bidimensionais, ao superar outras técnicas disponíveis na época na tarefa de reconhecimento de caracteres manuscritos.

No que respeita à arquitetura, as CNN são um tipo de rede neuronal *feedforward* que aprende a modelar características dos dados por si própria mediante a otimização dos filtros das camadas convolucionais. Este artigo abriu caminho para a ampla adoção das CNN em diversas áreas. São particularmente úteis para tarefas de visão computacional (classificação e segmentação de imagens ou deteção de objetos), de processamento de linguagem natural (classificação de texto, a análise de sentimentos, a deteção de *spam* e a

2 O problema da dissipação do gradiente ocorre durante o treino de ANN com métodos de aprendizagem baseados em gradiente, como o método de retropropagação do erro. Nestes métodos, os pesos da rede são atualizados em cada iteração de treino, com base nas derivadas parciais da função de custo em relação aos pesos atuais. A dissipação surge quando os gradientes calculados pelas camadas anteriores da rede se tornam muito pequenos, aproximando-se de zero, à medida que se retropropagam na rede. Como tal, os pesos das camadas mais próximas da entrada não são atualizados de forma eficaz, o que redunda num treino lento ou inoperante.

classificação de tópicos) e na análise e previsão de séries temporais (previsão de séries temporais financeiras, deteção de anomalias).

Em 2012, Cireşan et al. (2012) publicaram o artigo *Multi-Column Deep Neural Network for Traffic Sign*, que impulsionou a utilização de GPU (*Graphics Processing Unit*, ou Unidade de Processamento Gráfico) para o treino de redes neuronais profundas. Nesse trabalho, é demonstrada a eficácia das redes neuronais convolucionais profundas no reconhecimento de placas de trânsito em imagens urbanas. Este e outros trabalhos que demonstraram o desempenho superior das redes neuronais convolucionais profundas em tarefas de classificação de imagens (Krizhevsky et al., 2012b), levaram à adoção generalizada de GPU para treino e implementação de modelos de redes neuronais profundas, impulsionando o progresso em várias áreas da inteligência artificial.

Desde então verificou-se uma explosão cambriana de arquiteturas de ANN, a qual não está alheia a aparecimento de computadores pessoais de maior capacidade de processamento matricial (Dean, 2020), bem como de *frameworks* que facilitam a experimentação com ANN (Hellström, 2020).

From Fire to Data: Capturing Wildfire Dynamics with Semantic Segmentation & Spatiotemporal Reconstruction

Tiago F. R. Ribeiro[1]
tiago.f.ribeiro@ipleiria.pt

Rogério Luís de C. Costa[2], José Moreira[3]

1. CIIC, ESTG, Polytechnic of Leiria
2. CIIC, Polytechnic of Leiria
3. IEETA, LASI, DETI, University of Aveiro

MIT Portugal 2023 Annual Conference

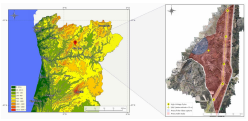
Introduction and Motivation

Wildfires have far-reaching consequences, threatening lives, economies, and the environment. Understanding their dynamics and environmental impacts is crucial, especially in high-incidence regions. Recently, machine learning-based models have emerged as promising solutions for comprehending the dynamics of wildfires.
In that vein, this research proposes three different elements:

I. We introduce a wildfire dataset for semantic segmentation of burned area
II. Provide tools to benchmark testing and validating semantic segmentation models in the context of wildfires
III. Present an autoencoder-based continuous spatiotemporal interpolation models to represent real-world phenomena such as wildfire burned area evolution.

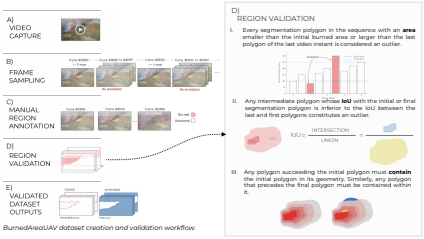
I. A novel burned area dataset: construction and validation

Datasets play a crucial role in training and validating ML models. However, collecting UAV videos in harsh conditions of high temperatures and toxicity further complicates data collection. Furthermore, segmenting burned areas presents unique challenges due to frequent occlusion by fire and flames and the and the amorphous nature of the burned area. Recognizing the scarcity of datasets with these specific characteristics and the ongoing challenges in semantic segmentation of burned areas, we propose a new dataset.



Hypsometric map indicating the relative position of the video capture (left) and orthophoto highlighting the region where the prescribed fire were conducted (right).

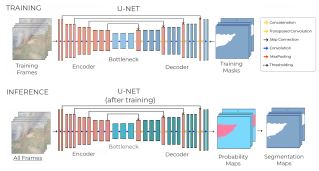
This novel dataset is based on a UAV-captured video of a prescribed fire at Torre de Pinhão in northern Portugal. Below, we present the workflow of creation of semantic segmentation dataset.



BurnedAreaUAV dataset creation and validation workflow.

II. U-Net as a baseline to capture frame-wise burned area

We employ the U-Net, a fully convolutional deep learning architecture widely utilized for semantic image segmentation tasks, to perform the semantic segmentation of the burned area.



Employed U-Net Architecture: Training and Segmentation Maps generation (U-Net Samples).

We evaluate three U-Net variants: U-Net Base, U-Net RED, and U-Net 3D. We train the models in the *BurnedAreaUAV* dataset and perform semantic segmentation for the entirety of the video.

Results

Model	IoU [%]	Recall [%]	Precision [%]	F1-Score [%]
U-Net Base	**95,31**	**98,30**	**96,92**	**97,61**
U-Net RED	92,74	95,34	97,15	96,24
U-Net 3D	94,01	98,21	95,67	96,92

U-Net classification metrics for BurnedAreaUAV testing set.

Model	Temp. Inconsistency
U-Net Base	**4,40E-03**
U-Net RED	9,11E-03
U-Net 3D	2,64E-02

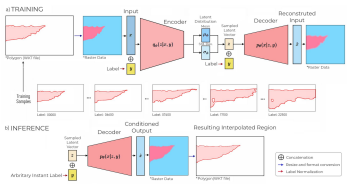
Average Temporal Inconsistency.

Key takeaways

- U-Net models perform well on the *BurnedAreaUAV* dataset.
- U-Net *Base* shows superior classification and temporal consistency.
- Generalization to other wildfires footage remains untested.

III. Representing burned area fire evolution with C-VAEs

C-VAEs are deep learning models that extend VAEs by learning a conditional distribution. They are capable of learning a conditional distribution, allowing to generate in-between representations by *smoothly* interpolate the latent space. We compare the performance of the C-VAE with alternative models from the literature. We evaluate these models on the *BurnedAreaUAV* dataset and compare the generated interpolations with the U-Net segmentations in the entirety of the wildfire video.



Employed C-VAE Architecture. a) each region stored in WKT format is converted to raster image to be processed by the model b) a new image is generated conditioned by a label and converted to WKT format.

Results

Data	Algorithm	IoU	Hausdorff Dist.
Periodic Sampling			
U-Net Samples	*Shape-Based*	**0,96**	42,5
	Mackenney	0,89	72,2
	C-VAE	0,95	**41,9**
BurnedAreaUAV Test Set	*Shape-Based*	**0,95**	**48,4**
	Mackenney	0,82	113,2
	C-VAE	0,90	60,8
Distance Based Sampling			
U-Net Samples	*Shape-Based*	**0,93**	**68,3**
	Mackenney	0,88	86,4
	C-VAE	0,91	76,5
BurnedAreaUAV Test Set	*Shape-Based*	0,91	**60,8**
	Mackenney	0,85	103,1
	C-VAE	**0,93**	85,2

Similarity metrics for the tested models.

Algorithm	Temp. Consistency
Periodic Sampling	
Shape-Based	0,980
Mackenney	0,970
C-VAE	**0,993**
Distance Based Sampling	
Shape-Based	0,994
Mackenney	0,983
C-VAE	**0,999**

Average Temporal Consistency.

Key findings

- C-VAE shows competitive results in terms of Similarity metrics.
- C-VAE outperforms in terms of Temporal Consistency.
- C-VAE show promising results and may be an alternative for applied earth sciences continuous time polygon interpolation applications.

Conclusion and Future Work

In this project, we have developed novel methodologies for the creation of high-quality burned area datasets. These dataset enabled us to effectively utilize U-Net networks to capture representation of the evolution of the burned area. Additionally, we propose a straightforward Autoenconder-based model that performs competitively against classical models.

In the future, we intend to augment BurnedAreaUAV by incorporating additional drone-captured videos from diverse locations and varying conditions. We aim to continue testing AE-based models' capabilities to generate continuous spatiotemporal interpolations. Furthermore, we plan to apply these approaches to study other phenomena, such as artificial reef monitoring, iceberg tracking, cloud segmentation, and other spatiotemporal database applications.

Papers, Code, Demo Videos

Co-funded by:

POLITÉCNICO DE LEIRIA | CIIC COMPUTER SCIENCE AND COMMUNICATION RESEARCH CENTRE | fct Fundação para a Ciência e a Tecnologia | MIT Portugal

under the Exploratory Project: MIT-EXPL-ACC-0057-2021 | MIT Portugal Climate Science & Climate Change