



# **Deteção automática da espécie *Acacia dealbata* em imagens recolhidas com drones**

Mestrado em Ciência de Dados

João Pedro Azenha Romão

Leiria, setembro de 2025



# **Deteção automática da espécie *Acacia dealbata* em imagens recolhidas com drones**

Mestrado em Ciência de Dados

João Pedro Azenha Romão

Trabalho de Projeto realizado sob a orientação do Professor Doutor António Pereira, do  
Professor Doutor Carlos Grilo e do Professor Doutor Luís Correia

Leiria, setembro de 2025

# **Originalidade e Direitos de Autor**

O presente relatório de projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Curso de Mestrado em Ciência de Dados, no ano letivo 2024/2025, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

# Agradecimentos

Gostaria de expressar a minha mais profunda gratidão a todos aqueles que, de diferentes formas, contribuíram para a realização desta dissertação.

Em primeiro lugar, agradeço aos meus pais, pelo apoio incondicional, pela paciência e pela motivação constante ao longo de todo este percurso acadêmico e pessoal. À minha namorada, pela compreensão, carinho e incentivo nos momentos mais desafiantes, o meu sincero obrigado.

Agradeço também aos meus orientadores, pela orientação científica, disponibilidade e dedicação, que foram fundamentais para a concretização deste trabalho.

Um agradecimento especial à Escola Superior de Tecnologia e Gestão (ESTG) por me ter proporcionado os recursos necessários ao desenvolvimento desta dissertação, em particular o acesso a um servidor equipado com uma placa gráfica NVIDIA A100. Este recurso teve um papel crucial na redução do tempo computacional dos resultados.

A todos os que, de forma direta ou indireta, estiveram presentes nesta caminhada, deixo o meu mais sincero reconhecimento e gratidão.

# Resumo

Esta dissertação tem como objetivo principal o desenvolvimento e validação de uma arquitetura baseada em Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*) para o reconhecimento automático da espécie invasora *Acacia dealbata* em ambientes florestais, recorrendo a imagens capturadas por Veículos Aéreos Não Tripulados (VANTs). A *Acacia dealbata* é uma árvore que possui um crescimento rápido, originária do sudeste da Austrália, que pode atingir cerca de 30 metros de altura e distingue-se pelas suas flores amarelas. Para tal, foram definidos objetivos específicos, como a automação do pré-processamento de dados através de *scripts* dedicados à extração, segmentação e rotulagem de imagens de vídeos de VANTs.

Para o processo de desenvolvimento da CNN, foram exploradas diversas arquiteturas. Os testes iniciais com VGG16 revelaram limitações, especialmente devido ao elevado tamanho e quantidade de imagens semelhantes do *dataset*, que resultava em valores de *Accuracy* e *F1-Score* baixos. A transição para a arquitetura ResNet50V2 trouxe algumas melhorias com um desempenho geral superior. No entanto, o avanço mais significativo foi alcançado com o desenvolvimento de uma CNN criada de raiz, aliado a um *dataset* mais otimizado.

A avaliação do desempenho do modelo demonstrou a sua eficácia e robustez. O modelo final, obtido dos testes com a CNN de raiz e o segundo *dataset*, apresentou uma capacidade de generalização na deteção da *Acacia dealbata* que foi confirmada pelos resultados da *Accuracy* de 0.8178 e *F1-Score* de 0.8459. Tais conclusões validam o potencial das CNNs na classificação de espécies em ambientes florestais e são uma contribuição significativa para a gestão ambiental.

No seguimento do estudo efetuado pode proceder-se à expansão e diversificação do *dataset* com novas gravações e em diferentes localizações geográficas e condições atmosféricas, visando também uma maior variedade de espécies. Poderá ainda ser explorada a adaptação do modelo para uma classificação multi-classe, permitindo a identificação de múltiplas espécies.

**Palavras-chave:** Redes Neurais Convolucionais, Classificação Binária, *Acacia dealbata*, VANTs, *Machine Learning*, Pré-processamento de Dados.

# Abstract

The main objective of this dissertation is the development and validation of an architecture based on Convolutional Neural Networks (CNNs) for the automatic recognition of the invasive species *Acacia dealbata* in forest environments, using images captured by Unmanned Aerial Vehicles (UAVs). The *Acacia dealbata* is a fast-growing tree, native to southeastern Australia, which can reach up to 30 meters in height and is distinguished by its yellow flowers. To this end, specific goals were defined, such as automating data preprocessing through scripts dedicated to the extraction, segmentation, and labeling of images from UAV videos.

For the CNN development process, several architectures were explored. Initial tests with VGG16 revealed limitations, particularly due to the large size and number of similar images in the dataset, which resulted in low Accuracy and F1-Score values. The transition to the ResNet50V2 architecture brought some improvements, with overall better performance. However, the most significant progress was achieved with the development of a CNN built from scratch, combined with a more optimized dataset.

The performance evaluation of the model demonstrated its effectiveness and robustness. The final model, obtained from the tests with the custom CNN and the second dataset, presented a strong generalization ability in detecting *Acacia dealbata*, confirmed by an Accuracy of 0.8178 and an F1-Score of 0.8459. These results validate the potential of CNNs in species classification in forest environments and represent a significant contribution to environmental management.

Following this study, future work may include expanding and diversifying the dataset with new recordings in different geographic locations and under various atmospheric conditions, also aiming at a wider variety of species. Moreover, adapting the model for multi-class classification could be explored, enabling the identification of multiple species.

**Keywords:** Convolutional Neural Networks, Binary Classification, *Acacia dealbata*, UAVs, Machine Learning, Data Preprocessing.

# Índice

Originalidade e Direitos de Autor .....	iii
Agradecimentos .....	iv
Resumo .....	v
Abstract .....	vi
Lista de figuras .....	ix
Lista de tabelas .....	x
Lista de siglas e acrónimos.....	xi
1. Introdução .....	1
1.1. Objetivos do trabalho .....	1
1.2. Metodologia .....	2
1.3. Estrutura do documento .....	3
2. Conceitos fundamentais de <i>Deep Learning</i> .....	5
3. Revisão da literatura .....	11
3.1. O Papel Estratégico da Detecção Remota e dos VANTs.....	12
3.2. Abordagens Tradicionais de Aprendizagem Automática .....	13
3.3. Classificação e Detecção de Objetos em Imagens Aéreas .....	15
3.4. Detecção e Mapeamento de Espécies Invasoras da Flora com Aprendizagem Profunda e VANTs .....	16
3.5. Reconhecimento Automático de Espécies Invasoras a partir de Imagens Aéreas.....	17
3.6. Classificação de Espécies de Florestas Secundárias com Imagens Hiperespectrais de VANT e Dados LiDAR.....	19
4. Recolha e pré-processamento de dados .....	21
4.1. Extração e redimensionamento de <i>frames</i> .....	24
4.2. Agrupamento de <i>frames</i> e classificação .....	25
4.3. Balanceamento de classes e criação do <i>dataset</i> .....	29
5. Modelação .....	35

<b>5.1. Modelação para o <i>dataset_1</i> .....</b>	<b>35</b>
<b>5.2. Modelação para o <i>dataset_2</i> .....</b>	<b>39</b>
<b>6. Avaliação de resultados .....</b>	<b>45</b>
<b>7. Conclusão e trabalho futuro .....</b>	<b>53</b>
<b>Referências Bibliográficas .....</b>	<b>55</b>

# Lista de figuras

Figura 1 - Passos da metodologia seguida.....	3
Figura 2 - Relação hierárquica entre AI, ML, DL e CNN [6] .....	5
Figura 3 - Exemplo da arquitetura de uma CNN [8] .....	7
Figura 4 - Localizações das gravações efetuadas .....	22
Figura 5 - Drone Dji Mavic 2 enterprise advanced [44].....	22
Figura 6 - Extração e redimensionamento das <i>frames</i> .....	25
Figura 7 - Agrupamento de imagens semelhantes .....	26
Figura 8 - Aplicação web - lista de diretorias.....	27
Figura 9 - Aplicação web - classificação de imagens.....	28
Figura 10 - Estratégia de divisão do <i>dataset</i> .....	31
Figura 11 - Comparação visual entre <i>dataset_1</i> e <i>dataset_2</i> .....	33

# Lista de tabelas

Tabela 1 - Vídeos, tempos e imagens respectivas.....	23
Tabela 2 - Resultados do agrupamento de <i>frames</i> em <i>chunks</i> para cada vídeo.....	27
Tabela 3 - Resultados da classificação de cada vídeo em quantidades por classe.....	29
Tabela 4 - Quantidades de <i>chunks</i> do reagrupamento para cada vídeo.....	30
Tabela 5 - Resultados de contagem de imagens após amostragem sequencial com saltos.....	31
Tabela 6 - Resultados de contagem de imagens do <i>dataset_1</i> balanceado.....	32
Tabela 7 - Resultados de contagem de imagens do <i>dataset_2</i> não balanceado.....	32
Tabela 8 - Resultados de contagem de imagens do <i>dataset_2</i> balanceado.....	32
Tabela 9 - Características das CNNs com arquitetura VGG16 para o <i>dataset_1</i> .....	36
Tabela 10 - Características das CNNs com arquitetura ResNet50V2 para o <i>dataset_1</i> .....	37
Tabela 11 - Características das CNNs criadas de raiz para o <i>dataset_1</i> .....	38
Tabela 12 - Características das CNNs criadas de raiz para o <i>dataset_2</i> .....	40
Tabela 13 - Características das CNNs com arquitetura VGG16 para o <i>dataset_2</i> .....	42
Tabela 14 - Características das CNNs com arquitetura ResNet50V2 para o <i>dataset_2</i> .....	43
Tabela 15 - Resultados de testes com arquitetura VGG16 com o <i>dataset_1</i> .....	46
Tabela 16 - Resultados de testes com arquitetura ResNet50V2 com o <i>dataset_1</i> .....	47
Tabela 17 - Resultados de testes com uma nova CNN com o <i>dataset_1</i> .....	48
Tabela 18 - Resultados de testes com uma nova CNN com o <i>dataset_2</i> .....	49
Tabela 19 - Resultados de testes com arquitetura VGG16 com o <i>dataset_2</i> .....	50
Tabela 20 - Resultados de testes com arquitetura ResNet50V2 com o <i>dataset_2</i> .....	51

## Lista de siglas e acrónimos

ANN	Artificial Neural Networks
AUC	Area Under the Curve
CBAM	Convolutional Block Attention Module
CNN	Convolutional Neural Network
CRISP-DM	Cross Industry Standard Process for Data Mining
DL	Deep Learning
ESTG	Escola Superior de Tecnologia e Gestão
HOG	Histogram of Oriented Gradients
HSI	Hyperspectral Imaging
ITCD	Individual Tree Crown Delineation
LiDAR	Light Detection and Ranging
ML	Machine Learning
ReLU	Rectified Linear Unit
RGB	Red, Green and Blue
SVM	Support Vector Machine
VANT	Veículo Aéreo Não Tripulado



# 1. Introdução

A crescente prevalência de espécies invasoras, como a *Acacia dealbata* em Portugal, representa desafios significativos para os ecossistemas nativos, causando perdas económicas e impactando os serviços ecossistémicos. Esta árvore de crescimento rápido, originária do sudeste da Austrália, distingue-se pelas suas flores amarelas e pela capacidade de formar densas florestas de uma só espécie, competindo com a flora nativa e alterando a química do solo. A ineficácia e os elevados custos dos métodos de controlo convencionais reforçam a necessidade de estratégias inovadoras. A deteção e monitorização precisas e atempadas da espécie são essenciais para otimizar as ações de controlo e reduzir a sua propagação [1].

O surgimento de tecnologias como os Veículos Aéreos Não Tripulados (VANTs) e o avanço no campo de *Machine Learning*, especificamente as redes neuronais convolucionais, oferecem uma grande oportunidade para revolucionar a forma como abordamos esta problemática. A automação do reconhecimento de espécies invasoras através de sistemas baseados em inteligência artificial não só pode reduzir drasticamente os custos operacionais, mas também aumentar a precisão e a frequência da monitorização, permitindo uma resposta mais rápida e eficaz contra as espécies invasoras. Torna-se necessária a procura de uma solução tecnológica que contribua significativamente para a mitigação dos impactos negativos das espécies invasoras.

## 1.1. Objetivos do trabalho

Esta dissertação tem como objetivo principal desenvolver e validar uma arquitetura baseada em redes neuronais convolucionais para o reconhecimento automático da espécie invasora *Acacia dealbata* em ambientes florestais, utilizando imagens capturadas por VANTs.

A *Acacia dealbata*, vulgarmente designada por acácia-mimosa, é uma espécie arbórea originária do sudeste da Austrália. Apresenta um crescimento rápido, podendo atingir cerca de 30 metros de altura. Possui casca lisa, de tonalidade cinzento-esverdeada, e folhas compostas, com coloração que varia entre o verde-acinzentado e o prateado.

O período de floração decorre, habitualmente, entre os meses de janeiro e abril, produzindo pequenas flores redondas de cor amarelo-vivo e aroma intenso e característico. Os seus frutos são vagens que parecem achatadas, de cor castanho-avermelhada [2].

Para alcançar este objetivo principal foram identificados os seguintes objetivos específicos:

- **Automatizar o Pré-processamento de Dados:** Desenvolver um conjunto de *scripts* que facilitem a extração, segmentação e rotulagem de imagens de vídeos de VANTs, otimizando o processo de preparação dos dados para o treino da rede neuronal;
- **Desenvolver uma Rede Neuronal Convolutacional Binária:** Conceber e otimizar uma CNN capaz de classificar imagens como contendo ou não a espécie *Acacia dealbata*;
- **Avaliar o Desempenho do Modelo:** Testar e analisar a *accuracy*, *loss*, *precision* e *recall*, demonstrando a sua eficácia no reconhecimento da espécie invasora.

## 1.2. Metodologia

A metodologia adotada neste trabalho segue uma abordagem estruturada, inspirada no modelo *Cross Industry Standard Process for Data Mining* (CRISP-DM), adaptada para o contexto do desenvolvimento de uma solução de *Machine Learning* para reconhecimento de espécies invasoras, representado na Figura 1 [3]. Este processo iterativo e faseado garante uma progressão lógica e sistemática do projeto:

1. **Recolha de Dados (*Data Collecting*):** Envolveu a exploração e caracterização das fontes de dados disponíveis, nomeadamente as filmagens realizadas por VANTs em áreas florestais. Foi crucial entender a natureza das imagens (resolução, condições de iluminação, perspetiva, etc.) e identificar os atributos relevantes para o reconhecimento da espécie-alvo;
2. **Pré-processamento de Dados (*Data Preprocessing*):** Esta fase foi crucial para a criação do *dataset* de treino, validação e teste. Incluiu a extração de imagens relevantes a partir das filmagens de VANTs e a classificação dessas imagens, identificando a presença ou ausência de *Acacia dealbata*;
3. **Modelação (*Modeling*):** Nesta etapa, foi desenvolvida a arquitetura da rede neuronal convolutacional binária. Foram exploradas diferentes configurações e parâmetros para a CNN, e o modelo foi treinado utilizando o conjunto de dados preparado. O processo

de modelação incluiu a seleção da arquitetura mais adequada e a otimização dos hiperparâmetros;

4. **Avaliação (*Evaluation*):** O desempenho do modelo treinado foi avaliado utilizando o conjunto de dados de teste. Foram calculadas métricas de desempenho como a *precision*, *recall*, *f1-score* e *accuracy*, para verificar a eficácia do modelo no reconhecimento da *Acacia dealbata*.



Figura 1 - Passos da metodologia seguida

### 1.3. Estrutura do documento

Esta dissertação está organizada em capítulos que refletem as diferentes etapas da investigação, desde os fundamentos teóricos até à implementação prática e análise de resultados. A estrutura foi delineada para garantir uma progressão lógica, permitindo uma compreensão clara do problema, das metodologias adotadas e das conclusões obtidas.

O Capítulo 2 explora os fundamentos das Redes Neurais Convolucionais, a sua arquitetura e aplicações no processamento de imagens. O Capítulo 3 analisa abordagens existentes para deteção de espécies invasoras, incluindo métodos tradicionais e baseados em *Machine Learning*. O Capítulo 4 descreve a metodologia de aquisição, tratamento, balanceamento dos dados e criação do *dataset* para o treino do modelo. O Capítulo 5 detalha a arquitetura da CNN desenvolvida, hiperparâmetros ajustados e estratégias de otimização. O Capítulo 6 descreve os resultados obtidos em diversos testes e explica a escolha do modelo final com base nas várias métricas utilizadas. Por fim, o Capítulo 7 conclui os objetivos da dissertação realizados e descreve os adicionais trabalhos futuros a realizar.

Esta página foi intencionalmente deixada em branco

## 2. Conceitos fundamentais de *Deep Learning*

A Aprendizagem Profunda (DL, do inglês *Deep Learning*), uma subcategoria de *Machine Learning* (ML), Figura 2, destaca-se pelo uso de redes neuronais com múltiplas camadas, capazes de processar e classificar diversos tipos de informação, incluindo imagens [4]. No contexto do processamento e análise de imagens, as Redes Neuronais Convolucionais representam um modelo de DL de particular relevância [5]. A distinção fundamental das CNNs em relação às redes neuronais clássicas reside na sua arquitetura de interligação: nas CNNs, cada unidade recebe apenas uma fração do resultado da saída da unidade anterior, determinada pela dimensão de um "filtro" [5]. Este contraste com as redes clássicas, onde os neurónios de cada camada se interligam com todos os neurónios da camada seguinte, confere também às CNNs uma maior eficiência computacional para o processamento de imagens.

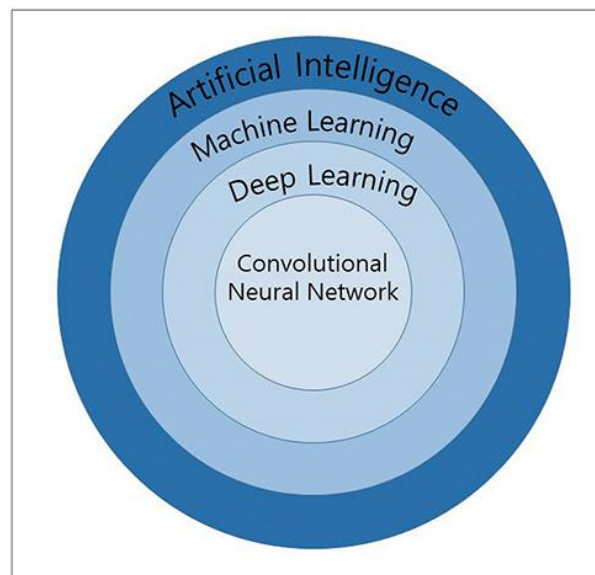


Figura 2 - Relação hierárquica entre AI, ML, DL e CNN [6]

A principal vantagem das CNNs é a sua capacidade de realizar a extração automática de características, que representam a informação recebida pelos modelos. Este atributo elimina a necessidade do pré-processamento manual de seleção de características, uma limitação inerente às abordagens tradicionais [4]. Os padrões detetados tornam-se progressivamente mais específicos e abstratos à medida que se avança pelas camadas da rede [5].

A arquitetura das CNNs (Figura 3) é composta por várias camadas com funções específicas:

- **Camadas Convolucionais:** Responsáveis pelas operações de convolução entre a imagem de entrada (geralmente representada por matrizes RGB, do inglês *Red, Green and Blue*) e os filtros de convolução, que detetam padrões na imagem. A operação de convolução calcula o produto interno através de uma "janela deslizante". O *stride* (número de pixéis que o filtro transita) influencia a rapidez do processo. Para evitar a perda de informação nos limites da imagem e preservar a dimensão, é comum adicionar pixéis com valor zero (*zero-padding*) [5];
- **Camadas de Sub-amostragem (*Pooling Layers*):** Têm como objetivo reduzir o número de pixéis da matriz resultante da convolução e, conseqüentemente, remover a informação redundante, reduzindo ainda a quantidade de processamento e memória necessária. A subamostragem pode ser realizada selecionando o valor máximo (*max pooling*) ou a média (*average pooling*) dos pixéis abrangidos pelo filtro de subamostragem;
- **Funções de Ativação:** Após as operações de convolução e subamostragem, os valores resultantes sofrem uma transformação não linear através das funções de ativação. Isso é crucial para garantir que a rede não seja excessivamente sensível a pequenas variações e para permitir a aprendizagem de padrões não lineares. A função ReLU (do inglês *Rectified Linear Unit*) é amplamente utilizada nas camadas convolucionais, convertendo valores negativos para zero e mantendo os positivos inalterados [7], o que pode diminuir o tempo de treino;
- **Camadas Completamente Conectadas (*Fully Connected Layers*):** Nas últimas camadas da rede, as informações recebidas são combinadas num único vetor unidimensional para a classificação final. A função de ativação *softmax* é comum ser utilizada para previsões de duas ou mais classes, retornando um vetor com as probabilidades de o objeto pertencer a cada classe.

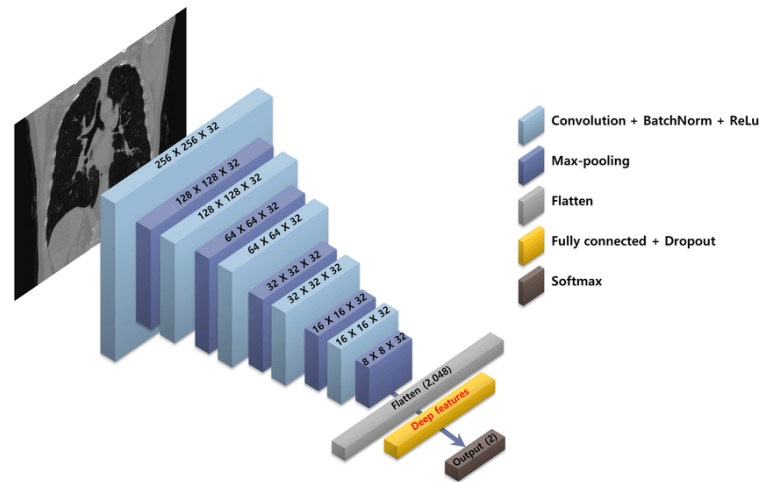


Figura 3 - Exemplo da arquitetura de uma CNN [8]

O processo de aprendizagem das CNNs inicia-se com a receção de uma imagem (matrizes RGB ou outras dimensões de cores) na entrada. Essa informação atravessa a rede, sofrendo operações de convolução, subamostragem e transformações não lineares, até chegar às últimas camadas para a classificação. Nessas camadas, é computada a função de perda (*Loss*), que quantifica a distância entre a classe prevista pela rede e a verdadeira classe. A entropia cruzada categórica (do inglês, *Categorical Cross Entropy*) é frequentemente utilizada para classificação de múltiplas classes. Com base no resultado da função de perda, a rede atualiza os pesos dos filtros de convolução por meio de uma função de otimização para minimizar o erro. Este processo de aprendizagem é realizado através do algoritmo de retro propagação (*backpropagation*) [9], que calcula as derivadas em cadeia da função de perda e as utiliza para atualizar os pesos até que o erro seja minimizado. A atualização dos pesos pode ser feita por amostra (Gradiente Descendente Estocástico), por lotes (subconjuntos de amostras, mais eficiente) ou por algoritmos adaptativos como *RMSprop* e *Adam* [10], [9]. O algoritmo *Adam* é particularmente eficaz, pois combina as vantagens do *RMSprop* e do *momentum*, adaptando a taxa de aprendizagem com base nos pesos e acelerando a convergência para o mínimo global. A taxa de aprendizagem é um hiperparâmetro crucial que controla a magnitude das atualizações de peso.

Para controlar o desempenho da CNN e evitar o *overfitting* (sobreajuste aos dados de treino), o conjunto de dados é tipicamente dividido em amostras de treino, validação e teste. O *overfitting* ocorre quando a rede aprende a classificar os dados de treino muito bem, mas falha em generalizar para novas imagens. Podem ser aplicadas técnicas de regularização para atenuar este problema, entre as quais:

- **Geração de Dados Sintéticos (*Data Augmentation*):** Permite reduzir a variância do modelo, aplicando transformações (rotações, ampliações, inversões horizontais e verticais) às imagens durante o treino. É particularmente útil para *datasets* de pequenas dimensões;
- **Desativação Aleatória de Neurónios (*Dropout*):** Consiste em "desligar" aleatoriamente um certo número de unidades de uma camada em cada iteração, forçando a rede a aprender novamente e prevenindo a adaptação excessiva aos dados de treino [11];
- **Paragem Antecipada (*Early Stopping*):** Interrompe o treino da rede quando o valor da *loss* nos dados de validação deixa de diminuir, enquanto o de treino continua a diminuir;
- **Normalização do Valor de Entrada (*Batch Normalization*):** Uma técnica que normaliza os valores resultantes das camadas de ativação anteriores, impedindo flutuações extremas dos pesos da rede e reduzindo o tempo de treino [7];
- **Regularizador do *kernel* (*Kernel Regularizer*):** É um hiperparâmetro que define a intensidade da penalização que é adicionada à função de *loss* do modelo e é proporcional ao quadrado da magnitude dos pesos da camada.

A principal inovação das CNNs reside na sua capacidade de transitar de um paradigma de engenharia de características para aprendizagem de características [4]. Isso significa que a rede aprende automaticamente as representações mais relevantes dos dados, permitindo a descoberta de padrões hierárquicos e abstratos que seriam impossíveis ou impraticáveis de serem definidos manualmente. Este avanço é fundamental para lidar com a complexidade visual das imagens aéreas e a variabilidade das espécies biológicas. Embora as CNNs ofereçam um desempenho superior, a sua aplicação introduz um novo conjunto de desafios relacionados com a necessidade de grandes volumes de dados rotulados e um elevado custo computacional para o treino. A discussão sobre *fine-tuning* e *data augmentation* não são meras opções, mas sim estratégias cruciais para mitigar estes desafios, tornando a aprendizagem profunda viável em contextos onde os recursos são limitados. As técnicas de regularização (e.g., *data augmentation*, *dropout*, *batch normalization*) não são apenas ajustes finos para otimizar o desempenho, mas componentes essenciais e intrínsecos para garantir a robustez e a capacidade de generalização das CNNs. Sem elas, o elevado poder de modelação das CNNs levaria rapidamente ao sobreajuste (*overfitting*), tornando-as

ineficazes para dados não observados durante o treino. A sua aplicação sistemática é tão importante como a própria arquitetura da rede.

Esta página foi intencionalmente deixada em branco

### 3. Revisão da literatura

As espécies invasoras são reconhecidas pela sua capacidade de rápida disseminação, um fenómeno que acarreta impactos negativos substanciais nos ecossistemas, manifestando-se, em particular, na perda de biodiversidade das zonas invadidas. Além das consequências ecológicas, estas espécies geram prejuízos económicos consideráveis, especialmente quando invadem campos agrícolas e áreas florestais, onde competem agressivamente pela aquisição de recursos, resultando na perda de espécies nativas [12].

As acácias, conhecidas pelo seu nome científico *Acacia*, são um género botânico vasto e diversificado que abrange milhares de espécies. A sua presença é notável em todo o mundo, mas a sua maior diversidade encontra-se na Austrália. Estas possuem uma grande variedade de formas, desde arbustos a árvores de grande porte, e são amplamente valorizadas tanto pela sua beleza ornamental como pelas suas propriedades ecológicas e utilitárias [13].

No contexto da ecologia de invasões biológicas, as acácias são amplamente reconhecidas como um grupo de espécies com um elevado potencial invasor [14]. Embora sejam valorizadas pelas suas propriedades ornamentais e utilitárias, a sua introdução fora da sua área de distribuição nativa tem gerado impactos ecológicos significativos. O comportamento invasor das acácias é impulsionado por uma combinação de características biológicas vantajosas.

Em primeiro lugar, estas árvores e arbustos exibem um crescimento rápido e uma elevada produção de sementes, que são capazes de permanecer viáveis no solo por longos períodos, formando um banco de sementes resiliente [15]. Além disso, a sua germinação é frequentemente estimulada por distúrbios, como os incêndios, o que lhes permite colonizar rapidamente áreas perturbadas. A capacidade de rebentar a partir de raízes também dificulta a sua erradicação após o corte.

Ecologicamente, as acácias transformam os ecossistemas que invadem. Ao formarem densas florestas de uma só espécie (monoculturas), competem eficazmente com a flora nativa por recursos como a luz e a água, levando ao seu declínio e, conseqüentemente, à perda de biodiversidade. Adicionalmente, a sua capacidade de fixar azoto atmosférico altera a química do solo, o que pode favorecer outras espécies invasoras e prejudicar a flora adaptada

a solos com baixo teor de nutrientes. Desta forma, o estudo da gestão e do controlo das acácias é crucial para a conservação dos ecossistemas locais.

A *Acacia dealbata*, especificamente, serve como um exemplo paradigmático desta problemática [15], [1]. Conhecida pela sua floração e folhas prateadas, foi introduzida em várias partes do mundo, incluindo Portugal, com fins ornamentais e de estabilização de encostas. No entanto, a sua facilidade de propagação e a sua adaptabilidade a diferentes tipos de solo e climas fizeram com que se naturalizasse rapidamente e passasse a dominar paisagens, especialmente após incêndios florestais [1]. A capacidade da *Acacia dealbata* de formar densas cortinas arbóreas e a sua resistência a condições adversas de solo e clima ilustram perfeitamente o sucesso invasor deste género botânico.

As técnicas convencionais empregadas para o controlo destas espécies são frequentemente dispendiosas e nem sempre garantem a erradicação completa do problema. Por exemplo, a aplicação de agentes de controlo biológico exige uma monitorização contínua das áreas afetadas, tanto durante como após a intervenção, para avaliar a evolução das espécies. A natureza multifacetada do impacto das espécies invasoras, abrangendo tanto a esfera ecológica quanto a económica, aliada à ineficácia e ao custo elevado das técnicas de controlo tradicionais, sublinha a necessidade de desenvolver ferramentas de monitorização mais eficientes e sustentáveis.

A *Acacia dealbata* serve como um caso de estudo exemplar desta exigência, justificando a procura por soluções inovadoras baseadas em tecnologias avançadas.

### **3.1.O Papel Estratégico da Deteção Remota e dos VANTs**

Historicamente, a deteção remota, através do uso de imagens de satélite, tem sido uma técnica estabelecida para mapear a vegetação e controlar a ocupação do solo. Quando combinada com métodos de aprendizagem automática, como as *Support Vector Machines* (SVM) e as Redes Neurais Artificiais (ANN, do inglês *Artificial Neural Networks*), esta abordagem permite a classificação automática de diversas espécies da flora. A imagiologia de deteção remota constitui um componente vital em levantamentos florestais, possibilitando a avaliação atempada do estado e distribuição dos recursos florestais, bem como a realização de observações dinâmicas e investigação para o desenvolvimento sustentável florestal [16].

Com o avanço tecnológico, surgiram os VANTs, que se distinguem na captura de imagens aéreas de alta resolução. Estes dispositivos permitem a criação de mapas de vegetação e fornecem informações detalhadas sobre a distribuição de espécies numa determinada área. A integração de VANTs com a deteção remota representa uma evolução qualitativa na granularidade e agilidade da monitorização ambiental. Enquanto os satélites oferecem uma visão macro, os VANTs proporcionam detalhes micro, essenciais para a deteção de espécies individuais e a gestão localizada de invasoras. Esta sinergia permite uma observação dinâmica e adaptável, crucial para a natureza em constante mudança dos ecossistemas invadidos.

As vantagens dos VANTs em relação aos satélites são notáveis: apresentam um menor custo de aquisição e de utilização. Por exemplo, o Phantom 4 (DJI), utilizado em estudos de identificação de espécies arbóreas, tem um custo inferior a 3000 USD [17]. Adicionalmente, permitem uma obtenção rápida de imagens da área em estudo. A capacidade de obter imagens de alta resolução é uma vantagem significativa, facilitando a identificação de espécies arbóreas com base em características de textura, como as formas das folhas e os padrões de ramificação [17]. Outro benefício é o potencial de robustez: modelos treinados num local podem ser aplicados noutros dias ou locais, uma vez que o sistema não depende exclusivamente da informação de refletância espectral, mas sim das características visuais detalhadas [17]. A combinação de VANTs com técnicas de aprendizagem automática permite, assim, o estudo da ocupação do território e o mapeamento e deteção de espécies da flora. Esta complementaridade otimiza a recolha de dados, permitindo uma monitorização detalhada e em tempo real, o que é fundamental para a gestão proativa de espécies invasoras que exigem intervenções localizadas e rápidas.

### **3.2. Abordagens Tradicionais de Aprendizagem Automática**

A combinação da deteção remota com algoritmos de aprendizagem automática clássicos tem sido um pilar na classificação automática de imagens de grande escala. Estas abordagens encontraram aplicações no estudo da ocupação do território, na distinção de zonas geográficas e na monitorização e mapeamento da vegetação.

Diversos classificadores e as suas aplicações ilustram o panorama das técnicas tradicionais. Martins [18] comparou classificadores de Máxima Verossimilhança, *Support Vector Machines* e Redes Neurais Artificiais para identificar a *Acácia Dealbata* em imagens de

satélite multiespectral, com o classificador de Máxima Verossimilhança a demonstrar os melhores resultados, alcançando uma *precision* global de 73% e um valor *Kappa* de 0.7. No que respeita à classificação da ocupação do território, Mendes e Dal Poz [19] empregaram ANN em imagens aéreas para seis classes distintas, evidenciando que a inclusão de informações adicionais, como dados radiométricos e um Modelo Digital de Superfície Normalizado, melhorava o desempenho do modelo, atingindo um *Kappa* de 0.86. Tudorache et al. [20] desenvolveram um método baseado em ANN para classificar áreas como vegetação ou zonas de inundação, utilizando características de textura e Histograma de Gradientes Orientados (HOG, do inglês *Histogram of Oriented Gradients*). Os autores concluíram que um conjunto de dados mais elaborado seria necessário para reduzir a taxa de classes incorretamente identificadas, embora isso pudesse implicar um aumento do tempo computacional e da taxa de falsos positivos. Na monitorização e identificação de espécies invasoras, Lopatin et al. [21] mapearam a *Acácia dealbata*, *ulex europaeus* e *pinus radiata* em imagens aéreas de VANTs utilizando o classificador Máxima Entropia, observando que a informação da sombra afetava o desempenho do modelo. De forma complementar, aplicaram o classificador *Random Forest* para criar um mapa da floração da Acácia longifólia, alcançando um desempenho de identificação superior a 96% [22].

A principal limitação destas técnicas reside na necessidade de uma preparação prévia dos dados, que implica a extração manual e a seleção criteriosa de características relevantes para o treino dos classificadores. Para aprimorar o desempenho, é frequentemente imperativo incorporar informações adicionais. Este pré-processamento é um processo intensivo e pode restringir a escalabilidade e a adaptabilidade dos modelos. A dependência da engenharia manual de características nas abordagens tradicionais constitui uma barreira significativa à automação completa e à escalabilidade. Este processo não só exige conhecimento especializado do domínio e consome tempo considerável, mas também pode falhar em capturar padrões complexos e não lineares nos dados, limitando a capacidade do modelo de generalizar para cenários diversos e subtis, como as variações morfológicas das espécies invasoras. A necessidade de "informações adicionais" e a sensibilidade a "elementos com texturas semelhantes" nas abordagens tradicionais indicam uma menor robustez e capacidade de generalização a variações ambientais ou de imagem. Modelos que dependem de características explicitamente definidas podem ser frágeis a pequenas mudanças nos dados de entrada, o que é um problema crítico para a monitorização contínua em ambientes dinâmicos.

### 3.3. Classificação e Deteção de Objetos em Imagens Aéreas

As Redes Neurais Convolucionais são amplamente aplicadas na análise de imagens aéreas, abrangendo tarefas de classificação, deteção de objetos e segmentação. Na classificação, o objetivo é identificar a que classe um objeto pertence numa imagem, resultando num vetor de probabilidades para cada classe. Por outro lado, na deteção de objetos, a rede percorre a imagem com uma janela deslizante para prever a classe e fornecer a localização do objeto através de uma caixa delimitadora.

A versatilidade e a capacidade de adaptação intrínseca das arquiteturas de CNNs para lidar com problemas complexos de visão computacional são evidenciadas pela sua ampla gama de aplicações em imagens aéreas. Por exemplo, na ocupação do território, foi investigada a utilização das redes CaffeNet e GoogLeNet para classificar imagens aéreas de satélite (ótica e infravermelhos), onde a técnica de *fine-tuning* de redes pré-treinadas demonstrou os melhores resultados, com a GoogLeNet a alcançar 97.10% de *precision* para imagens óticas [23]. Sameen et al. [24] desenvolveram uma CNN para a classificação de sete classes de ocupação do território, atingindo 97.3% de *precision* geral através da combinação de *dropout* e *batch normalization*. Ševo e Avramović [25] ajustaram a GoogLeNet para detetar e classificar objetos em cinco classes, obtendo aproximadamente 98% de *precision*.

Na deteção de veículos, apresentaram uma solução automática para detetar carros em imagens de VANTs, combinando o algoritmo *Mean-shift* (para extrair regiões candidatas) com uma CNN para extração de características e um classificador SVM, o que resultou numa redução do tempo computacional [26]. Tang et al. [27] propuseram um modelo adaptado do Faster R-CNN (VPN e AODN) para detetar veículos e estimar a sua orientação em imagens aéreas. Embora rápido, este modelo apresentou um desempenho inferior na deteção em larga escala e com falsos positivos. Para a deteção de aviões, Radovic et al. [28] utilizaram uma CNN baseada no algoritmo YOLO, alcançando 97.5% de *precision* e 86% em tempo real com um VANT, demonstrando a capacidade de detetar objetos oclusos.

O desafio da escala e detalhe na visão computacional aérea, especialmente para objetos de pequena dimensão, é uma questão persistente. Deng et al. [29] desenvolveram um método para deteção multi-escala de objetos (MS-OPN e AODN), utilizando um conjunto de filtros de convolução de dimensões diferentes (*inception module*), o que melhorou a *precision* na deteção de objetos de pequeno tamanho. Esta observação de que certos modelos apresentam

pior desempenho em larga escala ou para objetos mais pequenos sublinha a importância de arquiteturas que possam lidar com a variabilidade de tamanho dos objetos, como as árvores invasoras, que podem variar significativamente em dimensão e estar densamente agrupadas. A utilização de módulos *inception* e a técnica de janela deslizante são estratégias desenvolvidas para abordar esta questão. No reconhecimento de pássaros, Liu et al. [30] avaliaram e compararam diversos algoritmos de aprendizagem profunda (SSD, SSH, Tiny Face, YOLOv2 para deteção; U-Net, Mask R-CNN para segmentação) em imagens aéreas, concluindo que Tiny Face e SSH apresentavam melhor desempenho para detetar pássaros em imagens com fundos complexos.

### **3.4. Deteção e Mapeamento de Espécies Invasoras da Flora com Aprendizagem Profunda e VANTs**

A aplicação de Aprendizagem Profunda e VANTs para a deteção e mapeamento de espécies invasoras da flora representa uma área de investigação em crescimento. Wang et al. [31] desenvolveram um algoritmo de segmentação de super-píxel para segmentar imagens aéreas com base na presença da espécie *Ulva prolifera*. Utilizaram o algoritmo SEED para dividir a imagem em conjuntos de píxeis que representam o mesmo objeto, classificando-os posteriormente com uma CNN (ALexNet) para deteção binária. Este método demonstrou um desempenho superior e pode ser usado para distinguir diferentes tipos da alga *Ulva prolifera*.

Fan et al. [32] propuseram um algoritmo para a deteção de plantas de tabaco em imagens aéreas obtidas por VANTs. A solução extrai regiões candidatas através de operações morfológicas e utiliza um algoritmo de segmentação para isolar as plantas de interesse com base na intensidade das cores. Cada segmento é então classificado por uma CNN. Os autores concluíram que as CNNs com camadas de sub-amostragem alcançam melhor desempenho e que a quantidade de amostras de treino influencia significativamente o desempenho da rede.

Onishi e Ise [33] desenvolveram um algoritmo baseado na rede GoogLeNet para classificar automaticamente imagens aéreas de sete espécies de árvores. A aplicação de geração sintética de dados (*data augmentation*) foi crucial para reduzir o problema de falsos negativos em classes com número reduzido de amostras, atingindo 89% de classes classificadas corretamente. Este trabalho sublinha que a técnica de aumento de dados

melhora o desempenho da rede, especialmente em casos de desequilíbrio no número de amostras por classe.

Sa et al. [34] realizaram a segmentação de imagens aéreas multi-espectrais em três classes (áreas de cultivo, ervas selvagens e outros) utilizando a rede SegNet. Verificaram que o treino com redes pré-treinadas não teve um impacto tão significativo quanto esperado, devido ao número reduzido de amostras em comparação com os grandes conjuntos de dados usados para o treino original das redes. Os mesmos autores, num estudo posterior (Sa et al. [34]), utilizaram mapas multi-espectrais de nove canais com uma janela deslizante, alcançando maior *precision* na segmentação.

Huang et al. [35] empregaram redes convolucionais para a segmentação de imagens aéreas de VANTs, separando os píxeis em três classes: planta de arroz, ervas selvagens e outros. Compararam AlexNet, VGGNet e GoogLeNet, e modificaram a VGGNet com uma camada convolucional e módulos de ligação direta para compensar a perda de resolução, concluindo que esta abordagem incrementa a *precision* na deteção e classificação por classe. Safonova et al. [36] utilizaram uma CNN para identificar estágios de saúde de árvores invadidas por uma espécie invasora da fauna, atingindo altas taxas de *precision* (92.75% a 88.89%) e superando outros classificadores.

Apesar da vasta aplicação de algoritmos de aprendizagem automática profunda na agricultura para classificação da ocupação do território, monitorização e classificação de plantações agrícolas, e contagem de árvores ou animais, a deteção e classificação de espécies invasoras da flora em imagens aéreas capturadas por VANTs, utilizando redes neuronais convolucionais, ainda é uma área com exploração limitada. O trabalho de N. C. de Sá [22], que incide na identificação da *Acacia longifolia*, utilizou o algoritmo *Random Forest*, um classificador clássico que requer extração prévia de características. As CNNs, ao descobrirem automaticamente as características mais representativas das classes durante o treino, oferecem uma abordagem mais eficaz e com maior potencial de precisão, preenchendo uma lacuna importante na área [22].

### **3.5.Reconhecimento Automático de Espécies Invasoras a partir de Imagens Aéreas**

O trabalho desenvolvido por Gonçalves [37] apresenta uma abordagem inovadora para a deteção automática da espécie invasora *Acacia longifolia* em imagens aéreas, explorando o

potencial da aprendizagem profunda (DL) e, em particular, das redes neuronais convolucionais (CNN). A relevância desta investigação prende-se com a necessidade urgente de monitorizar e controlar a proliferação desta espécie em território português, dado o seu impacto negativo na biodiversidade e nos ecossistemas locais.

A solução proposta integra várias etapas: aquisição de imagens através de veículos aéreos não tripulados (VANT), pré-processamento das imagens para segmentação e preparação dos dados, e classificação automática baseada em modelos de CNN. Foram concebidas duas variantes da arquitetura, ambas com entradas compostas por amostras de 100×100 píxeis extraídas por uma janela deslizante. A primeira variante foi treinada para classificação binária (presença ou ausência de *Acacia longifolia*), enquanto a segunda utilizou uma classificação multi-classe (nove categorias distintas, incluindo diferentes tipos de vegetação e elementos do terreno), posteriormente convertida para binária para análise comparativa de desempenho.

Os resultados experimentais demonstraram elevadas taxas de acerto (98,5% para a abordagem multi-classe convertida e 98,7% para a abordagem binária), evidenciando que a inclusão de múltiplas classes não compromete a eficácia na deteção da espécie invasora e fornece informação adicional relevante sobre o ambiente. Adicionalmente, o estudo explorou um mecanismo de validação assistida por especialistas, no qual previsões com menor confiança são revistas manualmente. Este procedimento mostrou-se eficaz para melhorar a precisão final, equilibrando o custo de intervenção humana com o ganho em desempenho.

O contributo desta investigação reside na demonstração de que é possível desenvolver sistemas robustos, rápidos e escaláveis para o reconhecimento precoce de plantas invasoras a partir de imagens aéreas de alta resolução, combinando metodologias de *computer vision* e inteligência artificial. Este tipo de abordagem tem aplicação direta na gestão ambiental e no apoio à tomada de decisão para controlo e erradicação de espécies invasoras.

### 3.6. Classificação de Espécies de Florestas Secundárias com Imagens Hiperespectrais de VANT e Dados LiDAR

A classificação precisa de espécies arbóreas é um pilar fundamental para a gestão de recursos florestais e a avaliação da biodiversidade. Este processo é essencial para a estimativa da biomassa florestal e para o acompanhamento da biodiversidade [38]. Embora os avanços na tecnologia de deteção remota, como as imagens Hiper espectrais (HSI, do inglês *Hyperspectral Imaging*) e os dados de *Light Detection and Ranging* (LiDAR), ofereçam otimismo para o mapeamento detalhado da distribuição de espécies arbóreas em vastas regiões, a complexidade das florestas secundárias naturais, com as suas copas densas e sobrepostas, apresenta desafios significativos [39], [40], [41]. A integração de dados LiDAR, que capturam informações estruturais tridimensionais (3D), complementa as HSI, permitindo a extração de dados mais detalhados ao combinar informações espectrais e estruturais. A delimitação precisa da copa de árvores individuais (ITCD, do inglês *Individual Tree Crown Delineation*) é um pré-requisito crucial para a classificação de espécies arbóreas baseada em objetos [42]. Abordagens baseadas em DL, particularmente as CNN, têm demonstrado capacidades de aprendizagem superiores, e a incorporação de mecanismos de atenção, como o Módulo de Atenção de Bloco Convolutivo (CBAM, do inglês *Convolutional Block Attention Module*), pode melhorar o desempenho da classificação ao focar na informação espacial e de canal [43].

Esta página foi intencionalmente deixada em branco

## 4. Recolha e pré-processamento de dados

Este capítulo dedica-se a explorar as metodologias utilizadas na recolha e no pré-processamento dos dados. O rigor nesta fase inicial é crucial, uma vez que dados incompletos, inconsistentes ou ruidosos podem comprometer significativamente a validade e a fiabilidade dos resultados obtidos. São abordadas detalhadamente as estratégias utilizadas para adquirir os dados, bem como as técnicas aplicadas para limpar, transformar e preparar os dados para as etapas subsequentes de modelação. O objetivo primordial é garantir que o *dataset* final seja robusto, representativo e adequado de modo a obter os melhores resultados possíveis.

Para este projeto, foram realizadas gravações no dia 27 de outubro de 2024 com recurso a um drone (VANT) na zona de Pedrógão Grande. A localização exata das gravações está representada na Figura 4. As condições meteorológicas foram, no geral, favoráveis, embora no início das gravações se registasse a presença de nevoeiro e orvalho. Posteriormente, o céu tornou-se limpo, com predomínio de sol ao longo do dia. O drone utilizado foi um *Dji Mavic 2 enterprise advanced* (Figura 5) cujas características contam com um sensor de 48 megapixéis, um FOV (do inglês, *field of view*) de 84°, uma abertura de lente de  $f/2.8$  e com uma resolução de gravação de 1920x1080 com uma taxa de 29.97 (aproximadamente 30) *frames* por segundo. Foram efetuadas um total de cinco gravações (Tabela 1) de alturas que variaram entre 40 e 80 metros, com os respetivos meta dados e respetivas gravações térmicas, a preto e branco. Algumas destas gravações foram iniciadas enquanto o drone ainda estava no chão (pronto para levantar voo) e outras foram apenas iniciadas quando o drone já se encontrava em voo. O ângulo da câmara do drone foi de 90° perpendicular ao solo visualizando assim apenas o cume das árvores. As espécies arbóreas encontradas nas gravações são *Acacia dealbata*, eucalipto, alguns, mas poucos sobreiros e outras espécies de árvores/plantas de pequeno porte.

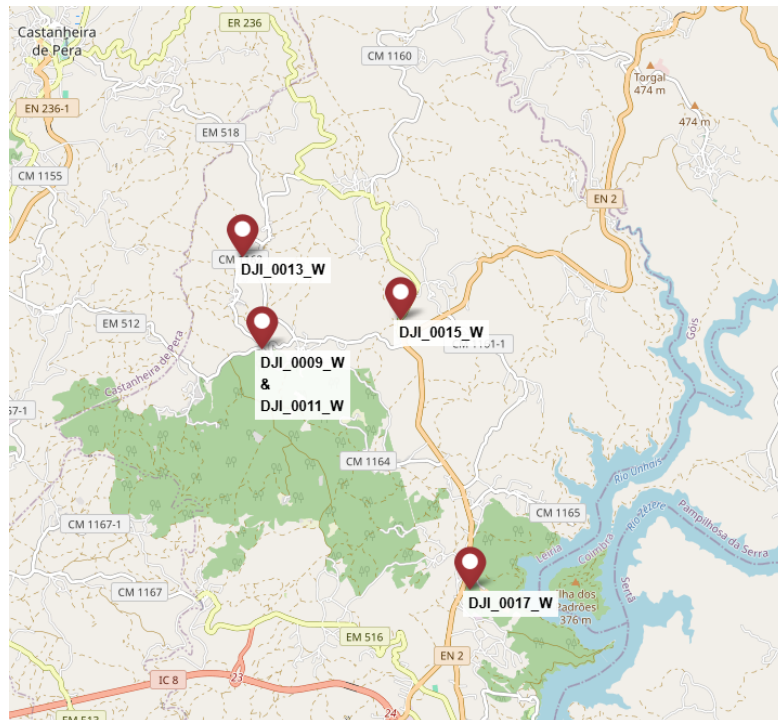


Figura 4 - Localizações das gravações efetuadas



Figura 5 - Drone Dji Mavic 2 enterprise advanced [44]

**Tabela 1 - Vídeos, tempos e imagens respetivas**

<b>Gravação</b>	<b>Tempo (hh:mm:ss)</b>	<b>Frames</b>
DJI_0009_W	00:13:51	24914
DJI_0011_W	00:05:55	10650
DJI_0013_W	00:12:38	22732
DJI_0015_W	00:07:25	13339
DJI_0017_W	00:11:19	20370
<b>Total</b>	<b>00:51:08</b>	<b>92005</b>

Neste projeto são apenas utilizadas as gravações a cores (RGB, do inglês *Red, Green and Blue*) já que as gravações de infravermelhos tornam praticamente impossível detetar a espécie arbórea apenas pelo seu cume.

As sequências de vídeo apresentaram particularidades relevantes. Por um lado, a natureza temporal das gravações leva a que *frames* consecutivas sejam muito semelhantes entre si, o que poderia introduzir redundância no *dataset*. Por outro lado, a realização de múltiplos voos em diferentes momentos do dia possibilitou captar condições variadas de luminosidade, contrastando áreas de sol direto com zonas de sombra, bem como diferenças de altitude.

Outro aspeto observado foi a variação da qualidade visual das imagens. Entre as *frames* extraídas, algumas apresentavam elevada nitidez, enquanto outras revelavam diferentes graus de desfocagem. Esta perda de qualidade podia ser atribuída ao movimento rápido do drone ou a fatores atmosféricos, como vento ou humidade, que afetaram a estabilidade das gravações. Assim, a construção do *dataset* exigiu uma seleção criteriosa das imagens, de modo a equilibrar a diversidade necessária para a análise com a consistência mínima em termos de qualidade visual.

Na etapa de pré-processamento de dados e construção do conjunto de dados (*dataset*) foram construídos alguns *scripts* em Python e uma aplicação web com o objetivo de facilitar a limpeza de dados, classificação e construção do *dataset*.

Foram ainda identificados três passos principais para o pré-processamento de dados:

1. Extração e redimensionamento de *frames*;
2. Agrupamento de *frames* em conjuntos, que designamos por *chunks*, e classificação;
3. Balanceamento de classes e criação do *dataset*.

## 4.1.Extração e redimensionamento de *frames*

Nesta secção, é descrito o processo de extração de *frames* de cada vídeo e o seu posterior redimensionamento, etapas essenciais para otimizar o desempenho e a eficiência do treino do modelo de classificação.

Para cada vídeo do *dataset*, todas as *frames* são extraídas sequencialmente e numeradas de forma única. Esta numeração garante que a ordem temporal das *frames* é preservada, o que pode ser crucial para futuras análises ou para a criação de sequências de treino, se necessário.

Após a extração, cada *frame*, originalmente com as dimensões de 1920x1080 píxeis, é redimensionada para 512x512 píxeis. Esta redução de dimensionalidade é de extrema importância por duas razões:

- **Eficiência Computacional:** *Frames* de dimensões mais pequenas exigem menos poder de processamento e memória, acelerando o tempo de treino do modelo;
- **Compatibilidade com Modelos de *Deep Learning*:** Muitos modelos de *Deep Learning*, particularmente os que são pré-treinados, operam com tamanhos de entrada específicos (como 512x512 ou 224x224). Redimensionar as *frames* assegura a compatibilidade e permite aproveitar o conhecimento aprendido por estes modelos.

Esta fase de extração e redimensionamento garante que o *dataset* esteja numa forma otimizada para o treino do modelo de classificação, equilibrando a retenção de informação relevante com a eficiência computacional, como representado na Figura 6.

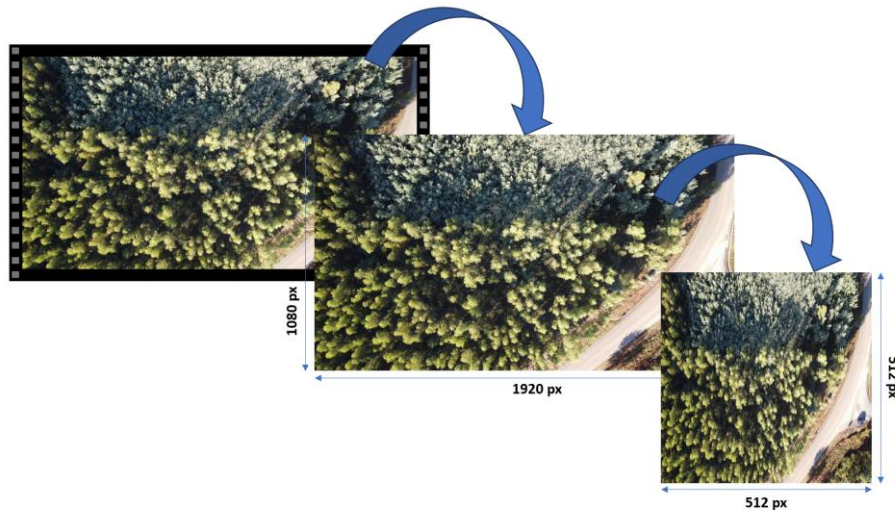


Figura 6 - Extração e redimensionamento das *frames*

## 4.2. Agrupamento de *frames* e classificação

Nesta etapa, é descrito o agrupamento das *frames* semelhantes numa diretoria denominada de “chunk” e a classificação dos mesmos. Com o objetivo de facilitar a etapa de classificação das *frames*, observou-se uma elevada semelhança entre *frames* temporalmente próximas. Assim, foi introduzida uma etapa de agrupamento de *frames*, com o intuito de otimizar o processo manual de classificação. Em vez de classificar todas as *frames* individualmente, passou-se a classificar conjuntos de *frames* semelhantes, designados por “chunks”, classificação essa que foi essencial para o treino do modelo de *Machine Learning* de classificação de imagens.

Para a etapa de agrupamento, é empregue o modelo de *Machine Learning* ResNet50V2, uma arquitetura de Rede Neural Convolutacional (CNN) pré-treinada, amplamente reconhecida pela sua capacidade de extrair características robustas de imagens.

A ResNet50V2 é utilizada para gerar *embeddings* (vetores de características) para cada *frame*. Estes *embeddings* representam a informação visual de cada *frame* num espaço de alta dimensão, onde *frames* semelhantes terão vetores de características próximos. Para determinar a semelhança entre as *frames*, calculamos a similaridade de cosseno entre os seus *embeddings*.

O critério de agrupamento é definido por um grau de semelhança de pelo menos 95%. Isto significa que as *frames* consecutivas são agrupadas se a similaridade de cosseno entre os seus *embeddings* for igual ou superior a este valor. Quando uma *frame* subsequente apresenta

uma similaridade inferior a 95% em relação à *frame* anterior, um novo *chunk* é iniciado como representado na Figura 7. Esta percentagem foi escolhida empiricamente. Um valor demasiado alto resultaria num menor número de *frames* por *chunk*, o que aumentaria a carga de trabalho no processo de classificação manual. Por outro lado, um valor de semelhança demasiado baixo aumentaria o risco de classificações incorretas de certas *frames*.

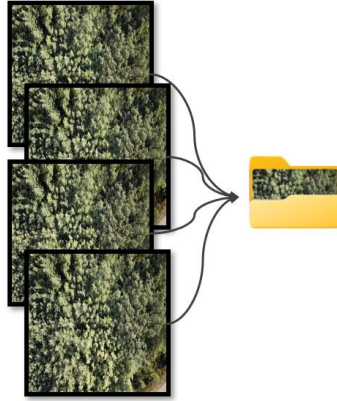


Figura 7 - Agrupamento de imagens semelhantes

Cada um destes agrupamentos de *frames* visualmente semelhantes é designado por "chunk". A criação de *chunks* é um passo fundamental para condensar a informação temporal dos vídeos, focando-se em momentos onde há uma alteração visual significativa. É importante notar que cada *chunk* é propriamente numerado temporalmente, preservando a sequência original dos eventos no vídeo. Esta abordagem permite-nos trabalhar com unidades de informação mais compactas e significativas, otimizando o processo para as fases de análise e classificação.

Após o agrupamento das *frames* em *chunks*, os resultados obtidos para cada conjunto de *frames* dos vídeos são os apresentados na Tabela 2 cujas colunas correspondem respetivamente ao nome do vídeo, a quantidade de *chunks* resultante do mesmo, o número mínimo de imagens por *chunk*, o número máximo, a sua média e mediana.

Tabela 2 - Resultados do agrupamento de *frames* em *chunks* para cada vídeo

Vídeo	Nº de <i>chunks</i>	Mínimo	Máximo	Média	Mediana
DJI_0009_W	311	8	558	80.11	45.00
DJI_0011_W	311	1	383	34.24	19.00
DJI_0013_W	238	1	889	95.51	21.50
DJI_0015_W	317	1	653	42.08	11.00
DJI_0017_W	208	1	1660	97.93	44.50

Para a etapa de classificação das *frames*, ainda de modo a facilitar o processo, foi criada uma aplicação web bastante simples com o único objetivo de ajudar a classificar *chunks* associando *labels* às *frames* pertencentes aos mesmos. Para isto, os *labels* possíveis foram simplificados, não contabilizando assim as espécies arbóreas que têm pouca frequência nos vídeos gravados pelo drone.

Como *labels* possíveis definidos temos:

- Acácia;
- Eucalipto;
- Estrada;
- Outros.

O *frontend* deste website é bastante simples e intuitivo. Tem como página inicial uma lista de diretorias que corresponde às diretorias que necessitam de classificação dos *chunks* de imagens como representado na Figura 8.



Figura 8 - Aplicação web - lista de diretorias

Após a seleção de uma das diretorias, aparece uma das imagens do *chunk* seguinte a classificar e um conjunto de *checkboxes* para escolher quais dos elementos referidos estão presentes na imagem. Existe também a opção de selecionar aleatoriamente uma outra imagem do *chunk* atual (para o caso de alguma das imagens estar desfocada ou se entender mal os elementos nela presentes). Existe também a possibilidade de remover todas as imagens do *chunk* atual no caso de as imagens serem desnecessárias para o nosso *dataset*

através do botão “Remover chunk”. Foi adicionado ainda outro botão denominado “Saltar chunk” que serve apenas para passar o *chunk* atual à frente, com a possibilidade de voltar ao mesmo mais tarde. Por fim, existe um botão de submeter a seleção e irá aparecer uma nova imagem correspondente ao *chunk* seguinte (Figura 9).



**Figura 9 - Aplicação web - classificação de imagens**

Após isto, as imagens pertencentes ao *chunk* são copiadas para a pasta ou pastas correspondentes à classificação “label” e, as imagens que não pertencem a um *label* são copiadas para a pasta “not\_label”.

Por fim, a diretoria com a designação “chunkName” é removida, assim como todo o seu conteúdo. Isto pode acontecer pois todo o seu conteúdo já foi copiado para as pastas necessárias. Assim, todo este processo que anteriormente teria que ser efetuado de forma manual, passou a ser muito mais automatizado, simples e rápido de efetuar. Como resultado final da classificação de cada vídeo temos os resultados da Tabela 3 onde as colunas “Acácia”, “Eucalipto”, “Estrada” e “Outros” correspondem à quantidade de imagens presentes para cada categoria.

**Tabela 3 - Resultados da classificação de cada vídeo em quantidades por classe**

Vídeo	Acácia	Eucalipto	Estrada	Outros
DJI_0009_W	24108	20331	11046	14065
DJI_0011_W	9635	7846	6009	9604
DJI_0013_W	21436	18892	9792	14418
DJI_0015_W	8494	12319	9792	12126
DJI_0017_W	14029	7444	5301	20339

### 4.3. Balanceamento de classes e criação do *dataset*

A preparação de um *dataset* robusto e bem balanceado é um passo crítico para garantir o desempenho e a generalização de qualquer modelo de *Machine Learning*. Nesta secção, detalhamos as estratégias empregadas para balancear as classes e para a criação do *dataset* final, assegurando a sua integridade e prevenindo a contaminação de dados.

Embora a classificação original do *dataset* seja do tipo multiclasse, optou-se por preparar o conjunto de dados para uma tarefa de classificação binária, de forma a simplificar o problema e otimizar o desempenho do modelo.

Importa salientar que, dado que o *dataset* foi construído a partir de imagens extraídas de vídeos, as *frames* consecutivas apresentam um elevado grau de semelhança. Para evitar que imagens quase idênticas fossem distribuídas simultaneamente pelos conjuntos de treino, validação e teste, criando assim uma contaminação de dados, foi necessário adotar um método de seleção e agrupamento que garantisse a separação destas imagens semelhantes entre os diferentes conjuntos, minimizando assim o risco de sobreposição de informação e de enviesamento dos resultados.

Neste contexto, procedeu-se ao reagrupamento das imagens em *chunks*, para cada uma das duas classes, de forma a concentrar em cada grupo apenas imagens com uma semelhança mínima de 80%. A razão da decisão desta percentagem é explicada mais abaixo, após a apresentação da metodologia de amostragem sequencial com saltos.

Após este novo processo de agrupamento, os *chunks* foram concatenados num único *dataset* que inclui todos os *chunks* de cada vídeo. Os resultados desta divisão estão representados na Tabela 4.

Tabela 4 - Quantidades de *chunks* do reagrupamento para cada vídeo

Vídeo	<i>Chunks</i> acácia	<i>Chunks</i> não acácia
DJI_0009_W	46	4
DJI_0011_W	32	6
DJI_0013_W	24	4
DJI_0015_W	24	19
DJI_0017_W	17	16
<b>Total</b>	<b>143</b>	<b>49</b>

De modo a dar continuação à construção do *dataset*, foi posteriormente adotada uma metodologia de amostragem sequencial com saltos, desenhada para prevenir a contaminação de dados e assegurar que as partições fossem representativas de todo o conjunto.

A abordagem é a seguinte:

- **7 *chunks* para treino:** Os primeiros 7 *chunks* consecutivos de um determinado segmento de dados são alocadas para o conjunto de treino;
- **Salto de 1 *chunk*:** Ignora-se o *chunk* seguinte para criar uma separação e evitar dependências temporais diretas entre os conjuntos;
- **2 *chunks* para validação:** Os próximos 2 *chunks* são atribuídos ao conjunto de validação, que será usado para avaliar o desempenho dos modelos durante o processo de definição de hiperparâmetros;
- **Salto de 1 *chunk*:** Novamente, salta-se um *chunk*;
- **1 *chunk* para teste:** O *chunk* seguinte é guardado para o conjunto de teste, que serve para a avaliação final do desempenho do modelo em dados nunca antes vistos;
- **Salto de 1 *chunk*:** E, por fim, salta-se um último *chunk* antes de repetir todo o processo.

Este ciclo é repetido sequencialmente sobre a totalidade dos grupos de imagens balanceadas até que todas as *frames* tenham sido devidamente reorganizadas e atribuídas a um dos conjuntos. Esta estratégia de salto entre os grupos é crucial para garantir que não haja sobreposição de informação entre os conjuntos de treino, validação e teste, assegurando

assim uma avaliação imparcial e robusta do modelo, bem como um treino otimizado do mesmo (Figura 10).

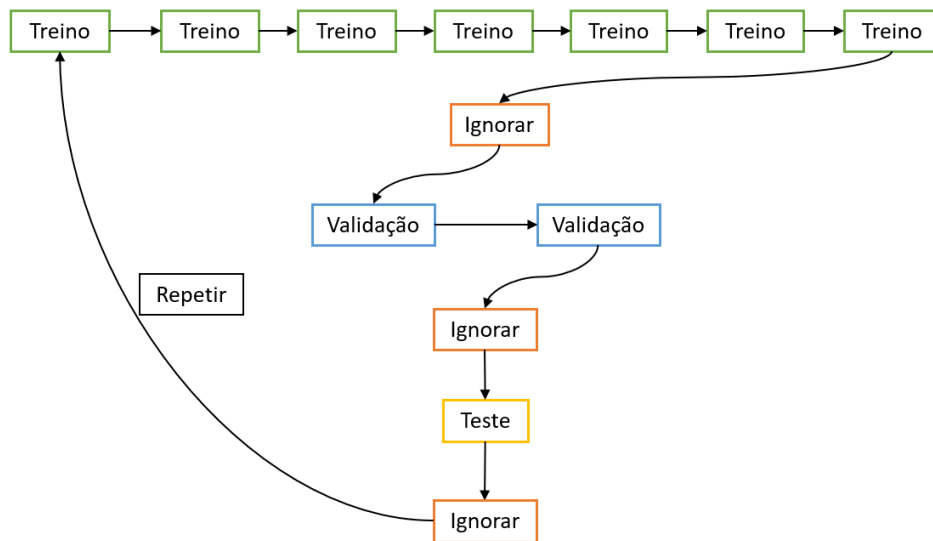


Figura 10 - Estratégia de divisão do dataset

Como referido acima, o reagrupamento das imagens em *chunks* foi realizado utilizando uma percentagem mínima de semelhança entre imagens de 80%. Este valor foi determinado de acordo com dois fatores. Se a percentagem de semelhança mínima fosse muito grande, a possibilidade de contaminação entre treino-validação e validação-teste seria muito grande, o que significa que as últimas imagens de cada grupo de 7 *chunks* do conjunto de treino iriam ser muito semelhantes às primeiras imagens do grupo seguinte de 2 *chunks* do conjunto de validação. De igual modo, as últimas imagens de cada grupo de 2 *chunks* do conjunto de validação iriam ser muito semelhantes às primeiras imagens de cada *chunk* do conjunto de teste. Para além disso, se a percentagem de semelhança fosse inferior a 80%, iriam existir poucos *chunks* (com mais imagens cada um), insuficientes, para a metodologia de amostragem sequencial com saltos.

Após esta etapa, foram obtidos os resultados representados na Tabela 5.

Tabela 5 - Resultados de contagem de imagens após amostragem sequencial com saltos

Vídeo	Treino		Validação		Teste	
	Acácia	Não acácia	Acácia	Não acácia	Acácia	Não acácia
Total	20834	7212	5620	1876	8139	1209

Como podemos verificar na Tabela 5, existe uma grande falta de balanceamento de dados que se traduz numa maior quantidade de dados da classe *Acácia*. Assim, foi ainda necessário criar um *script* para equilibrar as classes. Este processo consistiu em eliminar imagens aleatoriamente da classe maioritária até atingir a mesma quantidade de imagens da classe minoritária. Designamos este *dataset* por *dataset\_1* e as quantidades de imagens finais presentes neste *dataset* estão apresentadas na Tabela 6.

Tabela 6 - Resultados de contagem de imagens do *dataset\_1* balanceado

	Treino	Validação	Teste
<b>Acácia</b>	7212	1876	1209
<b>Não acácia</b>	7212	1876	1209

Na tentativa de obter melhores resultados, foi ainda criado um segundo *dataset* que designamos por *dataset\_2* cujo objetivo foi reduzir o número de imagens semelhantes, mantendo as imagens com melhor qualidade. Para isso, são executados os mesmos processos que o *dataset\_1*, no entanto, após a etapa da amostragem sequencial com saltos, foi criado um novo *script* que, mais uma vez, compara todas as imagens com um grau de semelhança mínimo de 95%, ordena as imagens semelhantes por ordem crescente de qualidade. Esta qualidade é medida com três métricas sendo estas o desfoque ou *blur*, ruído ou *noise* e *brisque* (BRIEF, do inglês *Binary Robust Independent Elementary*). Após esta ordenação por qualidade, as cinco melhores imagens de cada *chunk* são escolhidas descartando todas as outras, resultando assim nas contagens da Tabela 7.

Tabela 7 - Resultados de contagem de imagens do *dataset\_2* não balanceado

	Treino	Validação	Teste
<b>Acácia</b>	2337	992	766
<b>Não acácia</b>	1109	253	118

Ainda após este processo e para finalizar o *dataset\_2* foi necessário mais uma vez equilibrar as classes. Assim, foi aplicada a mesma técnica utilizada para o *dataset\_1* onde se elimina da classe maioritária imagens aleatoriamente até alcançar o mesmo número de imagens da classe minoritária, resultando assim nas contagens da Tabela 8.

Tabela 8 - Resultados de contagem de imagens do *dataset\_2* balanceado

	Treino	Validação	Teste
<b>Acácia</b>	1109	253	118
<b>Não acácia</b>	1109	253	118

Na Figura 11 está presente uma comparação visual da diferença entre o *dataset\_1* e o *dataset\_2*.

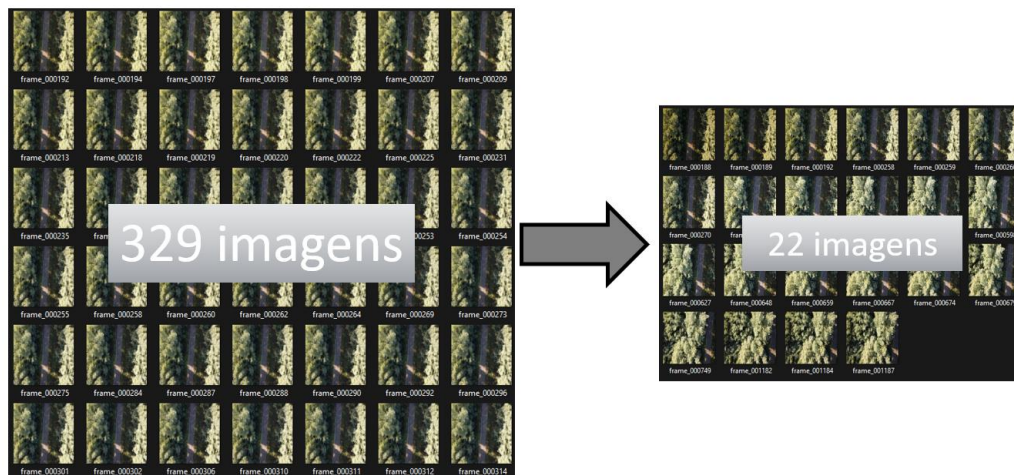


Figura 11 - Comparação visual entre *dataset\_1* e *dataset\_2*

Como podemos observar na Figura 11, o *dataset\_2* contém muito menos imagens semelhantes entre si comparativamente com o *dataset\_1*. Esta seleção criteriosa resultou na diminuição das redundâncias e na obtenção de um *dataset* mais equilibrado, essencial para o treino dos modelos de *Deep Learning*.

Esta página foi intencionalmente deixada em branco

## 5. Modelação

De modo a efetuar o treino dos modelos de *Deep Learning* foi fornecido pela ESTG acesso a um servidor que possui duas placas gráficas da NVIDIA cujos modelos são “RTX A4000” e “A100 80GB PCIe”, das quais foi apenas utilizada a placa gráfica A100 cujo poder de processamento é muito maior, especialmente para treino de modelos de *Deep Learning*.

Para o reconhecimento automático da espécie invasora *Acacia dealbata* foi desenvolvida e treinada uma Rede Neuronal Convolutiva para classificação binária. Para efeitos desta investigação, considerou-se como classe positiva a categoria *not\_Acacia* (não acácia) e como classe negativa a categoria *Acacia*. Como referido no Capítulo 2, é bastante importante realizar vários testes com valores dos hiperparâmetros diferentes de modo a otimizar o treino e controlar o *overfitting* do modelo.

Os hiperparâmetros alterados nos vários testes deste projeto foram:

- *Dropout*;
- Número de camadas convolucionais;
- Dimensão dos filtros de convolução;
- Número de épocas de treino;
- *Kernel regularizer*;
- *Early stopping*;
- *Learning rate*;
- Redutor da taxa de aprendizagem.

As secções 5.1 e 5.2 descrevem, respetivamente, a modelação realizada para o *dataset\_1* e para o *dataset\_2*.

### 5.1. Modelação para o *dataset\_1*

Inicialmente foram realizados testes com CNNs, recorrendo à arquitetura VGG16, previamente treinada em *ImageNet*, para a técnica de *fine-tuning*. O processo consistiu em, primeiramente, congelar a base convolutiva da rede VGG16 para treinar apenas o classificador (como representado no primeiro modelo). Em seguida, as camadas da rede foram descongeladas progressivamente, começando pelas camadas mais próximas da saída e avançando gradualmente para as camadas mais próximas da entrada. Esta abordagem

permitiu ajustar os pesos do modelo às características específicas do *dataset* utilizado, procurando um equilíbrio entre o aproveitamento do conhecimento previamente adquirido e a adaptação ao novo problema de classificação. As características destas CNNs, que foram treinadas com um *early stopping* com paciência de 30 e um máximo de 150 épocas, estão apresentadas na Tabela 9 com os seus diversos parâmetros e camadas descongeladas.

**Tabela 9 - Características das CNNs com arquitetura VGG16 para o *dataset\_1*. Notas para a leitura da tabela: A camada Dense(1) integra a função de ativação *sigmoid* para produzir uma probabilidade de saída.**

Nome	Camadas descongeladas	Learning rate	Camadas adicionadas
VGG16_V1	Nenhuma	0.0001	Dense(128, relu) → Dropout(0.5) → Dense(1)
VGG16_V2	block5_*, block4_*, block3_*	0.0001	Flatten → Dense(256) → Dropout(0.5) → Dense(1)
VGG16_V3	block5_*, block4_*, block3_*, block2_*	0.00001	Flatten → Dense(256) → Dropout(0.5) → Dense(1)
VGG16_V4	block5_*, block4_*, block3_*	0.00001	Flatten → Dense(256) → Dropout(0.5) → Dense(1)
VGG16_V5	block5_*, block4_*, block3_*	0.0001	Flatten → Dense(256, relu,l2=0.001) → Dropout(0.5) → Dense(1)
VGG16_V6	block5_*, block4_*, block3_*, block2_*	0.0001	Flatten → Dense(256, relu,l2=0.001) → Dropout(0.5) → Dense(1)
VGG16_V7	block5_*, block4_*, block3_*, block2_*, block1_*	0.0001	Flatten → Dense(256, relu, l2=0.001) → Dropout(0.5) → Dense(1)

Posteriormente, dando continuidade aos testes anteriores, foram realizadas novas experiências utilizando a arquitetura ResNet50V2, cujas principais características se encontram resumidas na Tabela 10. Esta arquitetura foi escolhida por apresentar um bom equilíbrio entre desempenho e eficiência computacional, sendo particularmente adequada para tarefas de classificação de imagens em contextos com recursos limitados. O objetivo destes testes foi avaliar se uma arquitetura mais recente e otimizada, como a ResNet50V2, poderia superar os resultados obtidos anteriormente com a VGG16, tanto em termos de

*precision* como de capacidade de generalização. Tal como nos testes anteriores, foi aplicada a técnica de *fine-tuning*. O processo iniciou-se com o congelamento da base convolucional da rede, permitindo que apenas as camadas de classificação fossem treinadas no primeiro modelo. De seguida, os restantes hiperparâmetros foram ajustados para maximizar o desempenho no *dataset* em estudo. Para evitar o *overfitting*, foi aplicado um método de *early stopping* com uma paciência de 30 e um máximo de 150 épocas de treino.

**Tabela 10 - Características das CNNs com arquitetura ResNet50V2 para o *dataset\_1*. Notas para a leitura da tabela: A camada Dense(1) integra a função de ativação *sigmoid* para produzir uma probabilidade de saída.**

Nome	Camadas descongeladas	Learning rate	Camadas adicionadas
ResNet50_V1	0	0.0001	Dense(128, relu) → Dropout(0.5) → Dense(1)
ResNet50_V2	Últimas 16	0.00001	Dense(64, relu, l2=0.001) → Dropout(0.5) → Dense(1)
ResNet50_V3	Últimas 12	0.00001	Dense(64, relu, l2=0.001) → Dropout(0.5) → Dense(1)
ResNet50_V4	Últimas 13	0.000001	Dense(32, relu, l2=0.001) → Dropout(0.4) → Dense(1)
ResNet50_V5	0	0.000001	Dense(32, relu, l2=0.001) → Dropout(0.4) → Dense(1)
ResNet50_V6	Últimas 14	0.000001	Dense(64, relu, l2=0.001) → Dropout(0.5) → Dense(1)
ResNet50_V7	Últimas 3	0.000001	Dense(32, relu, l2=0.001) → Dropout(0.4) → Dense(1)
ResNet50_V8	Últimas 6	0.000001	Dense(64, relu, l2=0.001) → Dropout(0.4) → Dense(1)
ResNet50_V9	0	0.000001	Dense(32, relu, l2=0.001) → Dropout(0.4) → Dense(1)

Após os testes com a arquitetura ResNet50V2, foi desenvolvida uma rede neuronal convolucional (CNN) criada de raiz, com o objetivo de construir uma solução mais leve e personalizada para o problema de classificação binária. Esta abordagem permitiu um controlo total sobre a arquitetura da rede, incluindo o número e o tipo de camadas, funções

de ativação, estratégias de regularização e métodos de normalização. As principais características desta CNN encontram-se descritas na Tabela 11.

**Tabela 11 - Características das CNNs criadas de raiz para o *dataset 1*. Notas para a leitura da tabela: DA corresponde a *data augmentation*, MP corresponde a *MaxPooling* e *Drop* corresponde a *dropout* e a camada Dense(1) integra a função de ativação *sigmoid* para produzir uma probabilidade de saída.**

Nome	Learning rate	Arquitetura
CNN_V1	0.0001	DA → Conv2D(32,3x3) → MP → Conv2D(64,3x3) → MP → Conv2D(128,3x3) → MP → Conv2D(128,3x3) → MP → Flatten → Dense(512) → Dense(1)
CNN_V2	0.00001	DA → Conv2D(32,3x3) → MP → Drop(0.25) → Conv2D(64,3x3) → MP → Drop(0.25) → Conv2D(128,3x3) → MP → Drop(0.25) → Conv2D(128,3x3) → MP → Drop(0.25) → Flatten → Dense(256) → Drop(0.5) → Dense(1)
CNN_V3	0.00001	DA → Conv2D(32,3x3) → MP → Drop(0.4) → Conv2D(64,3x3) → MP → Drop(0.4) → Conv2D(128,3x3) → MP → Drop(0.4) → Conv2D(128,3x3) → MP → Drop(0.4) → Flatten → Dense(256) → Drop(0.6) → Dense(1)
CNN_V4	0.00001	DA → Conv2D(32,3x3) → MP → Drop(0.3) → Conv2D(64,3x3) → MP → Drop(0.3) → Conv2D(128,3x3) → MP → Drop(0.3) → Conv2D(128,3x3) → MP → Drop(0.3) → Conv2D(256,3x3) → MP → Drop(0.3) → Flatten → Dense(256) → Drop(0.5) → Dense(1)
CNN_V5	0.00001	DA → Conv2D(32,3x3) → MP → Drop(0.3) → Conv2D(64,3x3) → MP → Drop(0.3) → Conv2D(128,3x3) → MP → Drop(0.3) → Conv2D(128,3x3) → MP → Drop(0.3) → Conv2D(256,3x3) → MP → Drop(0.3) → Flatten → Dense(256) → Drop(0.5) → Dense(1)
CNN_V6	0.0001	DA → Conv2D(64,7x7) → Conv2D(64,7x7) → MP → Drop(0.3) → Conv2D(128,5x5) → Conv2D(128,5x5) → MP → Drop(0.3) → Conv2D(256,5x5) → Conv2D(256,5x5) → MP → Drop(0.3) → Flatten → Dense(512) → Drop(0.5) → Dense(1)
CNN_V7	0.00001	DA → Conv2D(32,7x7) → Conv2D(32,7x7) → MP → Drop(0.3) → Conv2D(64,5x5) → Conv2D(64,5x5) → MP → Drop(0.3) → Conv2D(128,5x5) → MP → Drop(0.3) → Flatten → Dense(512) → Drop(0.5) → Dense(1)

Foram realizados vários testes experimentais, representados na Tabela 11, de modo a melhorar o desempenho da rede, o tamanho dos filtros convolucionais, a dimensão dos *feature maps*, bem como a estrutura da camada densa. Todos os treinos foram realizados com *early stopping* com uma paciência de 30, um máximo de 150 épocas. Além disso, todas as camadas de *MaxPooling* têm um *pool size* de 2.

Foi ainda realizado um conjunto de testes com recurso à biblioteca *Optuna*, uma *framework* de otimização de hiperparâmetros baseada em *Bayesian optimization*, com o objetivo de melhorar o desempenho do modelo e, ao mesmo tempo, reduzir o esforço manual associado ao *fine-tuning* de hiperparâmetros. Esta ferramenta permitiu a exploração automática de um vasto espaço de procura, ajustando diversos parâmetros, tais como a taxa de aprendizagem (*learning rate*) com valores entre 1e-3 e 1e-7, a taxa de *dropout* com valores entre 0.1 e 0.5, a taxa de regularização L2 com valores de 1e-2 a 1e-5 e ainda dois optimizadores diferentes, *adam* e *rmsprop*.

## 5.2. Modelação para o *dataset\_2*

Optou-se ainda por desenvolver uma nova CNN personalizada para a classificação binária mais leve e adaptada ao *dataset\_2*. Esta abordagem permitiu um controlo completo sobre a arquitetura da rede, incluindo o número de camadas, as funções de ativação e as estratégias de regularização.

Os detalhes desta CNN estão descritos na Tabela 12. A otimização da rede foi realizada através de testes experimentais, ajustando o tamanho dos filtros convolucionais, a dimensão dos *feature maps* e a estrutura da camada densa.

**Tabela 12 - Características das CNNs criadas de raiz para o *dataset\_2*. Notas para a leitura da tabela: DA corresponde a *data augmentation*, MP corresponde a *MaxPooling* com um *pool size* de 2, *Drop* corresponde a *dropout* e RS corresponde a *rescaling* com um fator de (1./255), todas as camadas Conv correspondem a Conv2D e a camada Dense(1) integra a função de ativação *sigmoid* para produzir uma probabilidade de saída.**

Nome	Learning rate	Arquitetura
CNN_2_V1	0.00001	RS → Conv (32,3x3) → MP → Conv (62,3x3) → MP → Conv (128,3x3) → MP → Conv (256,3x3) → MP → Conv (256,3x3) → Flatten → Dense(1)
CNN_2_V2	0.0001	RS → Conv (16,3x3,l2(0.001)) → Drop(0.2) → MP → Conv (32,3x3,l2(0.001)) → Drop(0.2) → MP → Conv (64,3x3,l2(0.001)) → Drop(0.2) → MP → Conv (128,3x3,l2(0.001)) → Drop(0.2) → MP → Conv (128,3x3,l2(0.001)) → Drop(0.2) → Flatten → Dense(1)
CNN_2_V3	0.00001	RS → Conv (16,3x3,l2(0.001)) → MP → Conv (32,3x3,l2(0.001)) → MP → Conv (64,3x3,l2(0.001)) → MP → Conv (128,3x3,l2(0.001)) → MP → Conv (256,3x3,l2(0.001)) → Flatten → Dense(1)
CNN_2_V4	0.00001	RS → Conv (32,3x3,l2(0.001)) → Drop(0.1) → MP → Conv (64,3x3,l2(0.001)) → Drop(0.1) → MP → Conv (128,3x3,l2(0.001)) → Drop(0.1) → MP → Conv (128,3x3,l2(0.001)) → Drop(0.1) → Conv (256,3x3,l2(0.001)) → Drop(0.1) → Flatten → Dense(1)
CNN_2_V5	0.00001	DA → Conv (32,7x7) → Conv (32,7x7) → MP → Drop(0.3) → Conv (64,5x5) → Conv (64,5x5) → MP → Drop(0.3) → Conv (128,5x5) → MP → Drop(0.3) → Flatten → Dense(512) → Drop(0.5) → Dense(1)
CNN_2_V6	0.00001	DA → Conv (32,3x3) → MP → Drop(0.25) → Conv (64,3x3) → MP → Drop(0.25) → Conv (128,3x3) → MP → Drop(0.25) → Conv (128,3x3) → MP → Drop(0.25) → Flatten → Dense(256) → Drop(0.5) → Dense(1)
CNN_2_V7	0.0001	DA → Conv (64,7x7) → Conv (64,7x7) → MP → Drop(0.3) → Conv (128,5x5) → Conv (128,5x5) → MP → Drop(0.3) → Conv (256,5x5) → Conv (256,5x5) → MP → Drop(0.3) → Flatten → Dense(512) → Drop(0.5) → Dense(1)

Na compilação e treino do modelo, optou-se pela função de *loss binary\_crossentropy*, apropriada para problemas de classificação binária, e pelo otimizador RMSprop, configurado com uma taxa de aprendizagem inicial de 1e-5. Durante o processo de treino, foram monitorizadas diversas métricas de desempenho, incluindo a *accuracy*, a *loss*, a *precision* e o *recall*, fornecendo uma visão abrangente do progresso do modelo. Para otimizar o treino e garantir a convergência, foram utilizadas estratégias de *callbacks*: o *Early Stopping* foi ativado com uma paciência de 50 épocas, interrompendo o treino se a *loss* de validação não melhorasse, e o *Model Checkpoint* foi configurado para guardar automaticamente o modelo que apresentasse o melhor desempenho no conjunto de validação. Adicionalmente, foi implementado um *ReduceLROnPlateau* para ajustar dinamicamente a *learning rate*, reduzindo-a por um fator de 0.1 se a *loss* não melhorasse após 7 épocas, com uma *learning rate* mínima de 1e-7 com o objetivo de melhorar ao máximo o treino do modelo criado.

Após os resultados obtidos com a rede CNN personalizada, explorou-se o potencial da arquitetura VGG16 pré-treinada para a tarefa de classificação binária com o novo *dataset\_2*. O objetivo foi avaliar o desempenho de uma arquitetura de referência, comparando-a com o modelo personalizado desenvolvido. A abordagem de *transfer learning* foi adotada, utilizando a VGG16 como modelo base e congelando as suas camadas iniciais para aproveitar os *features* já aprendidos. Para a classificação, foi adicionado um cabeçalho de classificação composto por camadas densas, conforme a seguinte estrutura: uma camada *GlobalAveragePooling2D*, para reduzir as dimensões dos *feature maps*; uma camada *Dense* com 128 neurónios e função de ativação ReLU; uma camada *Dropout* com uma taxa de 0.5, para regularização; e uma camada *Dense* de saída com 1 neurónio e função de ativação sigmoid, adequada para a classificação binária. Os modelos foram treinados com um otimizador Adam, e a função de *loss* utilizada foi a *binary cross-entropy*. Os detalhes de cada modelo testado com a arquitetura VGG16, incluindo as camadas descongeladas e os *learning rates* utilizados, estão descritos na Tabela 13. De notar que o primeiro modelo apresentado não inclui a camada de *Dropout*, servindo como base de comparação para os restantes testes.

**Tabela 13 - Características das CNNs com arquitetura VGG16 para o *dataset\_2***

<b>Nome</b>	<b>Camadas descongeladas</b>	<b><i>Learning rate</i></b>
VGG16_2_V1	Nenhuma	0.0001
VGG16_2_V2	block5_*	0.0001
VGG16_2_V3	block5_*	0.00001
VGG16_2_V4	block5_*	0.000001
VGG16_2_V5	block5_*, block4_*	0.0001
VGG16_2_V6	block5_*, block4_*	0.00001
VGG16_2_V7	block5_*, block4_*	0.000001
VGG16_2_V8	block5_*, block4_*, block3_*	0.0001
VGG16_2_V9	block5_*, block4_*, block3_*	0.00001
VGG16_2_V10	block5_*, block4_*, block3_*	0.000001

Após a análise da arquitetura VGG16, explorou-se o potencial de um modelo mais recente e eficiente, a ResNet50V2, para a tarefa de classificação binária com o *dataset\_2*. O objetivo desta nova fase de testes foi avaliar o desempenho de uma arquitetura mais recente, conhecida por otimizar a largura, a profundidade e a largura de banda de redes neurais. A abordagem de *transfer learning* foi novamente adotada. A ResNet50V2 foi utilizada como modelo base e as suas camadas iniciais foram congeladas. Para a classificação, foi adicionado um cabeçalho de classificação composto por camadas densas, conforme a seguinte estrutura: uma camada *GlobalAveragePooling2D*, para reduzir as dimensões dos *feature maps*; uma camada *Dense* com um número variável de neurónios e função de ativação *ReLU*; e uma camada *Dense* de saída com 1 neurónio e função de ativação *sigmoid*, adequada para a classificação binária. Os modelos foram treinados com um otimizador *Adam*, e a função de *loss* utilizada foi a *binary cross-entropy*. Os detalhes de cada modelo testado com a arquitetura ResNet50V2, incluindo as camadas descongeladas, as *learning rates* e o número de neurónios na camada densa, estão descritos na Tabela 14. As colunas da Tabela 14 “LR” e “Neurónios” correspondem respetivamente a *learning rate* e número de neurónios.

Tabela 14 - Características das CNNs com arquitetura ResNet50V2 para o *dataset\_2*

Nome	Camadas descongeladas	LR	Neurónios
ResNet50_2_V1	Nenhuma	0.0001	128
ResNet50_2_V2	conv5_block*	0.000001	128
ResNet50_2_V3	conv5_block*, conv4_block*	0.000001	128
ResNet50_2_V4	conv5_block*, conv4_block*, conv3_block*	0.000001	128
ResNet50_2_V5	conv5_block*, conv4_block*, conv3_block*, conv2_block*	0.000001	128
ResNet50_2_V6	conv5_block*	0.000001	256
ResNet50_2_V7	conv5_block*, conv4_block*	0.000001	256
ResNet50_2_V8	conv5_block*, conv4_block*, conv3_block*	0.000001	256
ResNet50_2_V9	conv5_block*, conv4_block*, conv3_block*, conv2_block*	0.000001	256
ResNet50_2_V10	conv5_block*	0.000001	64
ResNet50_2_V11	conv5_block*, conv4_block*	0.000001	64
ResNet50_2_V12	conv5_block*, conv4_block*, conv3_block*	0.000001	64
ResNet50_2_V13	conv5_block*, conv4_block*, conv3_block*, conv2_block*	0.000001	64
ResNet50_2_V14	conv5_block*	0.000001	32
ResNet50_2_V15	conv5_block*, conv4_block*	0.000001	32
ResNet50_2_V16	conv5_block*, conv4_block*, conv3_block*	0.000001	32
ResNet50_2_V17	conv5_block*, conv4_block*, conv3_block*, conv2_block*	0.000001	32
ResNet50_2_V18	conv5_block*	0.000001	Nenhuma
ResNet50_2_V19	conv5_block*, conv4_block*	0.000001	Nenhuma
ResNet50_2_V20	conv5_block*, conv4_block*, conv3_block*	0.000001	Nenhuma
ResNet50_2_V21	conv5_block*, conv4_block*, conv3_block*, conv2_block*	0.000001	Nenhuma

Desta forma, a modelação apresentada neste capítulo reuniu os recursos computacionais adequados e diferentes arquiteturas de *Deep Learning*, incluindo VGG16, ResNet50V2 e a CNN desenvolvida especificamente para esta investigação. A utilização destes modelos permitiu abordar de forma sistemática o problema da classificação automática de imagens do cume das árvores, distinguindo a presença da espécie invasora *Acacia dealbata*. A definição criteriosa das classes, bem como o ajuste dos hiperparâmetros, assegurou a construção de modelos robustos e preparados para uma análise comparativa do seu desempenho.

## 6. Avaliação de resultados

Este capítulo foca-se na avaliação dos resultados de todos os testes de treino de modelos de *Deep Learning*, realizados para a classificação binária da espécie invasora *Acacia dealbata*. As duas classes abordadas em todo este capítulo são “acácia” e “não acácia” sendo que se considerou como positiva a classe “não acácia”.

As métricas de avaliação utilizadas para a comparação dos resultados foram a *Accuracy*, *Precision*, *Recall*, *F1-score* e AUC (*Area Under the Curve*). Estas medidas são amplamente empregues em problemas de classificação, uma vez que permitem avaliar diferentes aspetos do desempenho do modelo.

A *Accuracy* corresponde à proporção de previsões corretas em relação ao total de previsões realizadas, oferecendo uma visão geral da taxa de acerto do modelo. A *Precision* indica a proporção de casos classificados corretamente como positivos dentro de todas as instâncias previstas como positivas, ou seja, mede o grau de confiança nas previsões positivas. O *Recall* reflete a capacidade do modelo em identificar corretamente as instâncias realmente positivas, mostrando o quão bem o modelo reconhece a classe de interesse. O *F1-score* combina *Precision* e *Recall* numa média harmónica, oferecendo uma medida equilibrada. Por fim, o AUC calcula a área sob a curva ROC (do inglês, *Receiver Operating Characteristic*). Para um modelo de classificação binária, o AUC representa a probabilidade de o modelo classificar uma amostra positiva selecionada aleatoriamente com um valor mais elevado do que uma amostra negativa também selecionada aleatoriamente.

As fórmulas matemáticas de cada métrica são apresentadas a seguir, considerando as variáveis: Verdadeiros Positivos (TP), Falsos Positivos (FP), Verdadeiros Negativos (TN) e Falsos Negativos (FN).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

Para um primeiro teste, a CNN foi treinada utilizando a arquitetura VGG16 e o *dataset\_1*. Os resultados, apresentados na Tabela 15, refletem uma variedade de configurações de hiperparâmetros, bem como o "descongelamento" gradual de camadas da arquitetura VGG16, desde as últimas camadas convolucionais até às iniciais (processo conhecido como *fine-tuning*). A coluna "Modelo" da tabela refere-se ao nome do modelo identificado no Capítulo 5. Apenas os seis melhores resultados destes testes foram registados para análise.

Tabela 15 - Resultados de testes com arquitetura VGG16 com o *dataset\_1*

Modelo	Loss	Accuracy	AUC	Precision	Recall	F1-Score
VGG16_V1	0.6807	0.6857	0.7893	0.6385	0.8561	0.7315
VGG16_V2	2.5894	0.7701	0.7790	0.8077	0.7089	0.7551
VGG16_V3	4.0191	0.4235	0.3838	0.3548	0.1869	0.2449
VGG16_V4	5.2992	0.5624	0.6928	0.7525	0.1861	0.2985
VGG16_V5	2.3621	0.7113	0.7472	0.7201	0.6915	0.7056
VGG16_V6	4.1502	0.6820	0.6707	0.8152	0.4706	0.5968
VGG16_V7	3.7697	0.5380	0.7440	0.7255	0.1224	0.2095

Ao observar os resultados apresentados na Tabela 15, conseguimos notar que, apesar de a *Accuracy* atingir valores de 0.7701 no segundo registo, o valor da *Loss* em algumas das experiências é substancialmente alto. Por exemplo, valores de *Loss* de 4.0191, 5.2992 e 4.1502 sugerem que o modelo está a ter dificuldades em generalizar bem para dados novos.

Métricas como *Precision* e *Recall* mostram um desempenho inconsistente. Embora alguns modelos atinjam uma *Precision* de valor 0.8152, o seu *Recall* de 0.4706 demonstra que falham em identificar a maioria das instâncias da classe positiva. O *F1-Score*, que equilibra ambas as métricas, é, por conseguinte, baixo nestes casos, refletindo o fraco desempenho do modelo.

No entanto, os modelos VGG16\_V1, VGG16\_V2 e VGG16\_V5 destacam-se, apresentando o melhor desempenho em termos de *Accuracy* de valores 0.6857, 0.7701 e 0.7113, *F1-Score* e um bom equilíbrio entre todas as métricas, indicando uma maior capacidade de generalização.

Adicionalmente, foram efetuados testes utilizando a arquitetura ResNet50V2. Os resultados destes testes são apresentados na Tabela 16. Tal como nos testes com a VGG16, estes também exploraram diferentes configurações de hiperparâmetros e estratégias de *fine-tuning*.

Tabela 16 - Resultados de testes com arquitetura ResNet50V2 com o *dataset\_1*

<b>Modelo</b>	<b>Loss</b>	<b>Accuracy</b>	<b>AUC</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
ResNet50_V1	3.3829	0.5227	0.5806	0.5615	0.2076	0.3032
ResNet50_V2	0.8758	0.6869	0.7228	0.7354	0.5840	0.6511
ResNet50_V3	1.0369	0.7039	0.7295	0.7302	0.6468	0.6860
ResNet50_V4	0.7271	0.7841	0.7398	0.8690	0.6691	0.7561
ResNet50_V5	0.7316	0.5910	0.6161	0.5851	0.6253	0.6046
ResNet50_V6	0.6001	0.6481	0.7974	0.6559	0.6228	0.6390
ResNet50_V7	0.5705	0.6697	0.8374	0.6758	0.6523	0.6639
ResNet50_V8	0.6326	0.6664	0.7406	0.6831	0.6209	0.6506
ResNet50_V9	0.5584	0.8079	0.8726	0.8029	0.8162	0.8095

Ao analisar os resultados da Tabela 16, é possível observar uma melhoria no desempenho geral dos modelos treinados com a arquitetura ResNet50V2 em comparação com os modelos baseados em VGG16. Os valores de *F1-Score* são mais robustos e em dois casos, ligeiramente mais altos, variando entre 0.3032 e 0.8095, o que indica que os modelos ResNet50V2 estão a aprender de forma mais eficaz e a convergir para soluções com menos erro.

A *Accuracy* também demonstra uma performance superior e mais consistente. O melhor resultado de *Accuracy* atingiu 0.8079, que é superior ao melhor resultado de *Accuracy* obtido com a VGG16 (0.7701). De igual modo, os valores de *Precision*, *Recall*, *AUC* e *F1-Score* são geralmente mais elevados, sugerindo uma melhor capacidade de discriminação e generalização por parte dos modelos ResNet50V2. O desempenho superior, especialmente da versão ResNet50\_V9, indica que esta arquitetura é mais adequada, para o *dataset\_1*, pois consegue extrair padrões relevantes, apresentando boa capacidade de generalização.

Foram ainda realizados alguns testes com uma CNN criada de raiz ainda com a utilização do *dataset\_1*. Os resultados destes testes estão presentes na Tabela 17.

Tabela 17 - Resultados de testes com uma nova CNN com o *dataset\_1*

<b>Modelo</b>	<b>Loss</b>	<b>Accuracy</b>	<b>AUC</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-Score</b>
CNN_V1	6.1156	0.3870	0.4064	0.3030	0.1737	0.2209
CNN_V2	1.8599	0.5029	0.5074	0.5015	1.0000	0.6680
CNN_V3	0.8496	0.3958	0.2864	0.4270	0.6096	0.5023
CNN_V4	16.4224	0.4868	0.5461	0.4932	0.9586	0.6514
CNN_V5	1.0234	0.4322	0.5353	0.4530	0.6543	0.5354
CNN_V6	0.8851	0.6646	0.7268	0.6718	0.6435	0.6574
CNN_V7	2.9461	0.8151	0.9462	0.7301	1.0000	0.8440

A análise dos resultados para a arquitetura CNN criada de raiz revela uma grande disparidade de desempenho entre as diferentes versões. Enquanto algumas configurações, como CNN\_V1 e CNN\_V4, falharam de forma significativa, apresentando valores de *Loss* de 6.1156 e 16.4224 respetivamente e *Accuracy* de 0.3870 e 0.4868, outras demonstraram uma melhoria nas suas métricas.

Métricas como o *Recall* e o *AUC* evidenciam a inconsistência dos modelos. O CNN\_V2 alcançou um *Recall* perfeito, mas com um *AUC* e *Accuracy* baixos, indicando que, embora tenha conseguido identificar todas as instâncias da classe positiva, também fez muitas previsões incorretas.

Por outro lado, o modelo CNN\_V7 destacou-se como a versão mais otimizada. Com uma *Accuracy* de 0.8151 e uma *AUC* de 0.9462, este modelo demonstrou uma capacidade de desempenho superior. O seu *Recall* de 1.0000, combinado com uma *Precision* de 0.7301, resultou num *F1-Score* de 0.8440. Estes resultados sugerem que, com uma otimização adequada, uma CNN criada de raiz pode superar o desempenho das arquiteturas pré-treinadas, provando ser a abordagem mais eficaz para o *dataset\_1* e os desafios que este apresenta.

Foram ainda realizados testes adicionais utilizando uma CNN criada de raiz, com o *dataset\_2*. Os resultados destes testes são apresentados na Tabela 18. A redução do *dataset* obtendo apenas as cinco melhores imagens de cada grupo de semelhantes foi uma medida crucial adotada para potenciar os resultados dos testes anteriores.

Tabela 18 - Resultados de testes com uma nova CNN com o *dataset\_2*

Modelo	Loss	Accuracy	AUC	Precision	Recall	F1-Score
CNN_2_V1	0.6005	0.8008	0.8369	0.8087	0.7881	0.7983
CNN_2_V2	0.5079	0.8305	0.9290	0.7826	0.9153	0.8438
CNN_2_V3	0.4767	0.7754	0.9187	0.7152	0.9153	0.803
CNN_2_V4	0.5386	0.8136	0.9048	0.7606	0.9153	0.8309
CNN_2_V5	1.7080	0.8178	0.8345	0.7329	1.0000	0.8459
CNN_2_V6	3.2618	0.6610	0.6103	0.6187	0.8390	0.7123
CNN_2_V7	1.1882	0.6356	0.8163	0.5792	0.9915	0.7313

A análise da Tabela 18 revela os melhores resultados obtidos ao longo de todo o processo de testes. Os valores de *Loss* são notavelmente baixos e consistentes, com o valor mínimo a atingir 0.4767. Simultaneamente, a *Accuracy* demonstrou um melhor desempenho, com valores que chegam a 0.8305, no entanto o *F1-Score* demonstrou um ligeiro menor desempenho em comparação com o modelo CNN\_V7.

A melhoria mais expressiva, e que valida a importância desta redução do *dataset*, é visível na robustez das métricas como *Recall* e *F1-Score*. O *Recall* alcança valores de 0.9153 em diversas execuções e 1.0000 para o modelo CNN\_2\_V5, um aumento geral de robustez em comparação com os testes anteriores. Este valor elevado de *Recall* indica que o modelo CNN\_2\_V5 é altamente eficaz na identificação das amostras positivas o que é crucial para aplicações onde a deteção de todas as instâncias das espécies não invasoras é prioritária. Consequentemente, o *F1-Score*, que representa a média harmónica entre *Precision* e *Recall*, também atinge valores muito elevados (até 0.8459), confirmando um equilíbrio robusto entre a capacidade de não classificar incorretamente exemplos negativos como positivos (*Precision*) e a capacidade de encontrar todos os exemplos positivos (*Recall*). A AUC também apresenta valores relativamente elevados, acima de 0.8, o que demonstra uma boa capacidade do modelo em distinguir as duas classes.

Adicionalmente, foram efetuados testes utilizando a arquitetura VGG16. Os resultados destes testes são apresentados na Tabela 19.

Tabela 19 - Resultados de testes com arquitetura VGG16 com o *dataset\_2*

Modelo	Loss	Accuracy	AUC	Precision	Recall	F1-Score
VGG16_2_V1	0.6194	0.7076	0.7880	0.6581	0.8644	0.7473
VGG16_2_V2	7.9599	0.4873	0.5173	0.4919	0.7712	0.6007
VGG16_2_V3	3.8096	0.5932	0.6353	0.5902	0.6102	0.6001
VGG16_2_V4	4.4990	0.4195	0.4482	0.3273	0.1525	0.2081
VGG16_2_V5	13.6863	0.4237	0.4879	0.3875	0.2627	0.3132
VGG16_2_V6	2.5515	0.6822	0.6011	0.6937	0.6525	0.6725
VGG16_2_V7	3.0148	0.5847	0.5955	0.6250	0.4237	0.5051
VGG16_2_V8	3.0629	0.6229	0.6825	0.6028	0.7203	0.6564
VGG16_2_V9	4.4443	0.7331	0.7118	0.7957	0.6271	0.7015
VGG16_2_V10	3.3580	0.6059	0.6279	0.6374	0.4715	0.5421

O VGG16\_2\_V1 destaca-se como o modelo com o melhor desempenho geral, alcançando a menor *Loss* (0.6194) e o maior *F1-Score* (0.7473), que indica um equilíbrio entre a *Precision* (0.6581) e o *Recall* (0.8644). Este resultado sugere que esta versão foi a mais eficaz na tarefa de classificação, apresentando uma alta capacidade de identificar instâncias positivas e uma taxa aceitável de previsões corretas. O seu AUC (0.7880) reforça a sua robustez.

Em contraste, algumas versões como o VGG16\_2\_V4 e o VGG16\_2\_V5 apresentaram resultados muito fracos, com valores de *Loss* elevados (4.4990 e 13.6863, respetivamente) e valores de *Accuracy* muito baixos (0.4195 e 0.4237, respetivamente). O VGG16\_2\_V4 destaca-se negativamente com o menor *F1-Score* (0.2081), resultado da combinação de uma baixa *Precision* (0.3273) e *Recall* (0.1525).

Outras versões, como o VGG16\_2\_V6 e o VGG16\_2\_V9, apresentaram um desempenho sólido, com valores de *Accuracy* e *F1-Score* consideravelmente bons, mas sem atingir o patamar do VGG16\_2\_V1. Estes resultados demonstram que, mesmo dentro de uma arquitetura pré-treinada como a VGG16, a otimização dos hiperparâmetros é crucial para alcançar um desempenho superior no *dataset\_2*.

Por fim, foram efetuados testes utilizando a arquitetura ResNet50V2. Os resultados destes testes são apresentados na Tabela 20.

Tabela 20 - Resultados de testes com arquitetura ResNet50V2 com o *dataset\_2*

Modelo	Loss	Accuracy	AUC	Precision	Recall	F1-Score
ResNet50_2_V1	5.4150	0.6144	0.5303	0.5746	0.8814	0.6957
ResNet50_2_V2	1.0137	0.6441	0.5977	0.6466	0.6356	0.6411
ResNet50_2_V3	0.9522	0.6356	0.6336	0.6270	0.6695	0.6476
ResNet50_2_V4	0.8742	0.7203	0.7120	0.7453	0.6695	0.7054
ResNet50_2_V5	0.9203	0.6864	0.7070	0.6746	0.7203	0.6968
ResNet50_2_V6	1.1497	0.6483	0.7049	0.6549	0.6271	0.6407
ResNet50_2_V7	1.1130	0.6356	0.6798	0.6569	0.5678	0.6092
ResNet50_2_V8	1.1352	0.6059	0.6493	0.6190	0.5508	0.5830
ResNet50_2_V9	1.2722	0.4831	0.5157	0.4831	0.4831	0.4831
ResNet50_2_V10	0.7631	0.7076	0.7260	0.7248	0.6695	0.6961
ResNet50_2_V11	0.7829	0.6314	0.6698	0.6505	0.5678	0.6064
ResNet50_2_V12	0.8216	0.6610	0.6308	0.7262	0.5169	0.6040
ResNet50_2_V13	0.7971	0.6610	0.6460	0.6610	0.6610	0.6610
ResNet50_2_V14	0.7165	0.6822	0.6966	0.7009	0.6356	0.6667
ResNet50_2_V15	0.8519	0.5932	0.5956	0.6774	0.3559	0.4667
ResNet50_2_V16	0.6955	0.6992	0.7215	0.6942	0.7119	0.7030
ResNet50_2_V17	0.7586	0.6017	0.6591	0.5952	0.6356	0.6148
ResNet50_2_V18	0.8940	0.6186	0.6029	0.6400	0.5424	0.5872
ResNet50_2_V19	0.7095	0.6271	0.5837	0.6829	0.4746	0.5601
ResNet50_2_V20	0.6103	0.6483	0.7103	0.6699	0.5847	0.6245
ResNet50_2_V21	0.7611	0.5805	0.5645	0.5960	0.5000	0.5438

O ResNet50\_2\_V4 destaca-se como a versão mais otimizada, apresentando um *F1-Score* elevado de 0.7054. Este valor é resultado de um bom equilíbrio entre *Precision* (0.7453) e *Recall* (0.6695), sugerindo que este modelo é eficaz a classificar corretamente as instâncias positivas, ao mesmo tempo que minimiza os falsos positivos. O seu *Accuracy* de 0.7203 e *AUC* de 0.7120 reforçam o seu bom desempenho geral.

Em contraste, o ResNet50\_2\_V9 e o ResNet50\_2\_V15 apresentam resultados inferiores. O ResNet50\_2\_V9 tem uma *Accuracy* de apenas 0.4831 e um *F1-Score* de 0.4831. Por sua vez, o ResNet50\_2\_V15 obtém o pior *F1-Score* (0.4667), que é resultado de um *Recall*

bastante baixo (0.3559), o que indica que o modelo teve dificuldade em identificar a maioria das instâncias positivas.

Outras versões como o ResNet50\_2\_V2, ResNet50\_2\_V3 e ResNet50\_2\_V16 também mostraram um desempenho sólido, com valores de *F1-Score* acima de 0.64.

Com base na análise exaustiva dos testes realizados com diferentes arquiteturas de redes neuronais e configurações de *datasets*, pode concluir-se que a escolha da arquitetura e a preparação dos dados têm um impacto determinante na performance dos modelos de classificação da espécie invasora *Acacia dealbata*. Inicialmente, os resultados obtidos com a VGG16 revelaram limitações e inconsistência nas métricas de avaliação. Em contrapartida, a arquitetura ResNet50V2 demonstrou um desempenho mais robusto e consistente, superando a VGG16 em algumas das métricas, destacando-se especialmente nos testes com o modelo ResNet50\_V8.

No entanto, foi com a CNN desenvolvida de raiz que se observaram os maiores avanços. Após uma primeira fase de testes com o *dataset\_1*, os resultados foram melhorados no modelo CNN\_2\_V5 com a criação do *dataset\_2*, conseguindo assim obter valores de *Accuracy* e *F1-Score* de 0.8178 e 0.8459 respetivamente.

Apesar dos diversos testes das arquiteturas VGG16 e ResNet50V2 com o *dataset\_2*, o modelo CNN\_2\_V5 permaneceu o melhor. Os seus valores elevados de *Recall*, *F1-Score* e *AUC*, confirmam não só a eficácia da rede em identificar instâncias da classe “not\_Acácia”, mas também a sua fiabilidade global.

Assim, conclui-se que, apesar das vantagens das arquiteturas pré-treinadas, no nosso caso, uma CNN personalizada e devidamente ajustada, juntamente com um tratamento de dados do *dataset*, representa a solução mais eficaz para a tarefa de classificação binária da espécie invasora *Acacia dealbata*, garantindo um bom desempenho e uma boa capacidade de generalização.

## 7. Conclusão e trabalho futuro

O objetivo desta dissertação consistiu no desenvolvimento e validação de uma arquitetura de Redes Neurais Convolucionais (CNN) para o reconhecimento automático da espécie invasora *Acacia dealbata* em imagens de VANTs. Isso foi possível através da automação do pré-processamento de dados e do desenvolvimento de uma CNN binária otimizada.

A criação de um *dataset* a partir de gravações realizadas por um drone sobre o cume de uma floresta constituiu um processo que exigiu especial atenção, sobretudo devido à natureza e às condições em que os dados foram recolhidos. O ponto de partida baseou-se num número limitado de vídeos, o que implicou um aproveitamento criterioso de cada um deles e uma análise detalhada das particularidades presentes nas imagens.

As sequências de vídeo revelaram características relevantes. A natureza temporal das gravações faz com que *frames* consecutivas apresentem uma elevada semelhança, o que poderia resultar em redundância e reduzir a diversidade do *dataset*. Para minimizar este efeito, recorreu-se a imagens captadas em diferentes voos e momentos do dia, o que permitiu incorporar variações de luminosidade e mudanças de altitude.

Outro aspeto importante foi a oscilação da qualidade visual das imagens. Entre as *frames* extraídas, algumas apresentavam grande nitidez, enquanto outras se encontravam marcadas por graus distintos de desfocagem. Estas diferenças eram frequentemente consequência do movimento acelerado do drone ou das condições atmosféricas, como vento e humidade, que afetaram a estabilidade das gravações. Assim, a construção do *dataset* exigiu um processo de seleção cuidadoso, de modo a garantir simultaneamente diversidade e consistência mínima em termos de qualidade visual.

Os testes iniciais com VGG16 mostraram limitações e inconsistência nas métricas de avaliação. A ResNet50V2 trouxe melhorias referentes à robustez dos dados, mas o melhor desempenho foi alcançado com uma CNN criada de raiz, combinada com um *dataset\_2* mais reduzido que o *dataset\_1*.

O modelo com melhor desempenho, com uma *Accuracy* de 0.8178, demonstrou eficácia, particularmente no *Recall* (1.0000) e *F1-Score* (0.8459), confirmando a sua robustez e capacidade de generalização na deteção da *Acacia dealbata*.

Este estudo apresenta um contributo ao demonstrar que a utilização de CNNs, aliada a dados adquiridos por veículos aéreos não tripulados (VANTs), constitui uma abordagem eficaz para a identificação da *Acacia dealbata*. A metodologia desenvolvida oferece um avanço no campo da classificação automática de espécies invasoras e representa um passo concreto para apoiar estratégias de gestão e controlo florestal.

Como trabalho futuro poderá ser efetuada uma expansão e diversificação do *dataset*, o que implica efetuar novas gravações e, possivelmente, ainda gravações em localizações geográficas diferentes de modo a encontrar uma maior diversidade de espécies. Ainda no seguimento da expansão e diversificação do *dataset* será também importante adaptar o modelo de *Deep Learning* para uma nova classificação multi-classe onde se poderá identificar mais do que uma espécie numa única *frame* de vídeo, assim como identificar-se também os vários estágios de desenvolvimento da espécie invasora. Será ainda importante criar uma base de dados com o objetivo de armazenar a localização geográfica das espécies invasoras, sendo que esta localização pode ser obtida através dos metadados de cada gravação efetuada pelo VANT.

## Referências Bibliográficas

- [1] L. J. R. Nunes, M. A. M. Raposo, C. I. R. Meireles, C. J. P. Gomes, and N. M. C. A. Ribeiro, “The Impact of Rural Fires on the Development of Invasive Species: Analysis of a Case Study with *Acacia dealbata* Link. in Casal do Rei (Seia, Portugal),” *Environments*, vol. 8, no. 5, Art. no. 44, 2021.
- [2] “*Acacia dealbata*,” *Plantas Invasoras em Portugal – Universidade de Coimbra*. [Online]. Disponível em: <https://invasoras.pt/en/invasive-plant/acacia-dealbata>. [Accessed: 15-Sep-2025].
- [3] C. Shearer, “The CRISP-DM model: The new blueprint for data mining,” *Journal of Data Warehousing*, vol. 5, no. 4, pp. 13–22, 2000.
- [4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [5] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012.
- [6] S. D. Baek, J. Lee, S. Kim, H.-T. Song, and Y. H. Lee, “Artificial Intelligence and Deep Learning in Musculoskeletal Magnetic Resonance Imaging,” *Investigative Magnetic Resonance Imaging*, vol. 27, no. 2, pp. 67–74, Jun. 2023.
- [7] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proc. Proceedings of the 32<sup>nd</sup> International Conference on International Conference on Machine Learning (ICML'15)*, Volume 37, pp. 448 – 456, 2015.
- [8] J. Yun, Y. H. Cho, S. M. Lee, Y. H. Kim, and N. Kim, “Convolutional neural network (CNN) model architecture,” *Scientific Reports*, vol. 11, no. 1, p. 1102, Jul. 2021.
- [9] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

- [10] G. Hinton, “Lecture 6.5—RMSprop: divide the gradient by a running average of its recent magnitude,” Coursera: Neural Networks for Machine Learning, 2012.
- [11] N. Srivastava et al., “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
- [12] C. Hui, D. M. Richardson, et al., “Macroecology meets invasion ecology: linking the native distributions of Australian acacias to invasiveness”, *Diversity and Distributions*, vol. 17, no. 5, pp. 771-787, 2011.
- [13] S. Vicente, H. Trindade, C. Máguas, J. J. Le Roux, “Highly diverse and highly successful: invasive Australian acacias have not experienced genetic bottlenecks globally”, *Annals of Botany*, vol. 128, no. 2, pp. 149-157, 2021.
- [14] D. M. Richardson, P. Pyšek, M. Rejmánek, M. G. Barbour, F. D. Panetta, and C. J. West, “Naturalization and invasion of alien plants: concepts and definitions,” *Divers. Distrib.*, vol. 6, no. 2, pp. 93–107, Mar. 2000.
- [15] P. Lorenzo, L. González e M. J. Reigosa, “The genus *Acacia* as invader: the characteristic case of *Acacia dealbata* Link in Europe,” *Ann. For. Sci.*, vol. 67, art. no. 101, Jan. 2010.
- [16] S. Madonsela, M. A. Cho, R. Mathieu, O. Mutanga, A. Ramoelo, Z. Kaszta, R. Van de Kerchove, and E. Wolff, “Multi-phenology WorldView-2 imagery improves remote sensing of savannah tree species,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 58, pp. 65–73, 2017.
- [17] M. Onishi, S. Watanabe, T. Nakashima, and T. Ise, “Practicality and Robustness of Tree Species Identification Using UAV RGB Image and Deep Learning in Temperate Forest in Japan,” *Remote Sens.*, vol. 14, no. 7, p. 1710, 2022. doi: 10.3390/rs14071710.
- [18] F. D. Martins, “Utilização de técnicas de deteção remota na identificação de *Acacia* sp. na Região Centro Sul de Portugal Continental,” Ph.D. dissertation, IPCB. ESA, 2012.
- [19] T. S. G. Mendes and A. P. Dal Poz, “Classificação de imagens aéreas de alta-resolução utilizando redes neurais artificiais e dados de varredura a laser,” in *Simpósio Brasileiro de Sensoriamento Remoto*, 2011, vol. 15, pp. 7792–7799.
- [20] S. Tudorache, D. Popescu, and L. Ichim, “Combining efficient textural features with cnn—based classifiers to segment regions of interest in aerial images,” in *2017 5th*

International Symposium on Electrical and Electronics Engineering (ISEEE), 2017, pp. 1–6.

[21] J. Lopatin, K. Dolos, T. Kattenborn, and F. E. Fassnacht, “How canopy shadow affects invasive plant species classification in high spatial resolution remote sensing,” *Remote Sens. Ecol. Conserv.*, vol. 5, no. 4, pp. 302-317, 2019.

[22] N. C. de Sá, P. Castro, S. Carvalho, E. Marchante, F. A. López-Núñez, and H. Marchante, “Mapping the flowering of an invasive plant using unmanned aerial vehicles: is there potential for biocontrol monitoring?” *Front. Plant Sci.*, vol. 9, p. 293, 2018.

[23] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, “Land use classification in remote sensing images by convolutional neural networks,” *arXiv preprint arXiv:1508.00092*, 2015.

[24] M. I. Sameen, B. Pradhan, and O. S. Aziz, “Classification of very high resolution aerial photos using spectral-spatial convolutional neural networks,” *J. Sensors*, vol. 2018, p. 6571583, 2018.

[25] I. Ševo and A. Avramović, “Convolutional neural network based automatic object detection on aerial images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 740–744, 2016.

[26] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, “Deep learning approach for car detection in uav imagery,” *Remote Sens.*, vol. 9, no. 4, p. 312, 2017.

[27] T. Tang, S. Zhou, Z. Deng, L. Lei, and H. Zou, “Fast multidirectional vehicle detection on aerial images using region based convolutional neural networks,” in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 1844–1847.

[28] M. Radovic, O. Adarkwa, and Q. Wang, “Object recognition in aerial images using convolutional neural networks,” *J. Imaging*, vol. 3, no. 2, p. 21, 2017.

[29] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, “Multi-scale object detection in remote sensing imagery with convolutional neural networks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 3–22, 2018.

- [30] Y. Liu et al., “Performance comparison of deep learning techniques for recognizing birds in aerial images,” in 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018, pp. 317–324.
- [31] S. Wang, L. Liu, L. Qu, C. Yu, Y. Sun, F. Gao, and J. Dong, "Accurate Ulva prolifera regions extraction of UAV images with superpixel and CNNs for ocean environment monitoring," *Neurocomputing*, vol. 348, pp. 158–168, 2019.
- [32] Z. Fan, J. Lu, M. Gong, H. Xie, and E. D. Goodman, “Automatic tobacco plant detection in uav images via deep neural networks,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 876–887, 2018.
- [33] M. Onishi and T. Ise, “Automatic classification of trees using a uav onboard camera and deep learning,” arXiv preprint arXiv:1804.10390, 2018.
- [34] I. Sa et al., “Weedmap: a large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming,” *Remote Sens.*, vol. 10, no. 9, p. 1423, 2018.
- [35] H. Huang, J. Deng, Y. Lan, A. Yang, X. Deng, and L. Zhang, “A fully convolutional network for weed mapping of unmanned aerial vehicle (uav) imagery,” *PLoS ONE*, vol. 13, no. 4, p. e0196302, 2018.
- [36] A. Safonova, S. Tabik, D. Alcaraz-Segura, A. Rubtsov, Y. Maglinets, and F. Herrera, “Detection of fir trees (*abies sibirica*) damaged by the bark beetle in unmanned aerial vehicle images with deep learning,” *Remote Sens.*, vol. 11, no. 6, p. 643, 2019.
- [37] C. C. L. Gonçalves, “Identificação Automática de Plantas Invasoras em Imagens Aéreas,” *Dissertação de Mestrado, Departamento de Ciências e Tecnologias da Informação, Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal*, 2019.
- [38] X. Shang and L. A. Chisholm, “Classification of Australian native forest species using hyperspectral remote sensing and machine-learning classification algorithms,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2481–2489, 2013. doi: 10.1109/JSTARS.2013.2282166.

- [39] B. Fu, “Classifying vegetation communities karst wetland synergistic use of image fusion and object-based machine learning algorithm with jilin-1 and uav multispectral images,” *Ecol. Ind.*, vol. 140, p. 108989, 2022. doi: 10.1016/j.ecolind.2022.108989.
- [40] H. Qin, “Individual tree segmentation and tree species classification in subtropical broadleaf forests using Uav-based lidar, hyperspectral, and ultrahigh-resolution Rgb data,” *Remote Sens. Environ.*, vol. 280, p. 113143, 2022. doi: 10.1016/j.rse.2022.113143.
- [41] H. Yu, “Improving grassland classification accuracy using optimal spectral-phenological-topographic features in combination with machine learning algorithm,” *Ecol. Ind.*, vol. 158, p. 111392, 2024. doi: 10.1016/j.ecolind.2023.111392.
- [42] S. E. Franklin and O. S. Ahmed, “Deciduous tree species classification using object-based analysis and machine learning with unmanned aerial vehicle multispectral data,” *Int. J. Remote Sens.*, vol. 39, no. 15, pp. 5236–5245, 2018. doi: 10.1080/01431161.2017.1363442.
- [43] Y. Ma, Y. Zhao, J. Im, Y. Zhao, and Z. Zhen, “A deep-learning-based tree species classification for natural secondary forests using unmanned aerial vehicle hyperspectral images and LiDAR,” *Ecol. Ind.*, vol. 159, p. 111608, 2024.
- [44] HP Drones, “Mavic 2 Enterprise Advanced: perfeito para aplicações industriais e de primeiros socorros,” HP Drones, 5 jun. 2022. [Online]. Disponível em: <https://hp-drones.com/en/mavic-2-enterprise-advanced-perfect-for-industrial-and-first-aid-applications>. [Acedido: 15-set-2025].