



Categorização de imagem aérea usando deep learning

ALEXANDRE MANUEL SANTOS SILVA

Outubro de 2019

Categorização de imagem aérea usando deep learning

Alexandre Manuel Santos Silva

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

**Orientador: Doutor Carlos Manuel Abreu Gomes Ferreira
Co-Orientador: Doutora Maria de Fátima Coutinho Rodrigues**

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Resumo

O processo de produção de cartografia é demorado, bastante complexo e implica um elevado volume de trabalho humano. A automatização, mesmo que parcial, de partes deste processo, levaria à redução do trabalho e a uma maior eficiência e rentabilidade deste processo.

Neste contexto, a fase deste processo com mais impacto nos custos e com maior potencial de automatização, é aquela onde é efetuada a identificação dos elementos que estão presentes na fotografia aérea, um dos recursos utilizados para produção de cartografia.

Pretende-se com este trabalho estudar diferentes modelos de segmentação semântica, implementados com recurso a *deep learning*, de forma a avaliar a viabilidade da sua utilização para automatização desta fase do processo. Será, também, efetuada uma análise do processo de produção de cartografia, bem como dos recursos disponíveis, para definir o método mais adequado para a utilização dos modelos.

Palavras-chave: Cartografia, Ocupação de solo, Deep Learning, Automatização

Abstract

The process of cartography production it's time-consuming, quite complex and requires a high human workload. The automation, even partial, of some parts of this process would lead to a workload reduction and a greater efficiency and profitability of this process.

The phase of this process with the most impact on costs and greater automation potential, it's the phase where its made the identification of elements existing on aerial photography, one of the resources used to produce cartography.

This works aims to study different semantic segmentation models, implemented using deep learning, to assess whether it is viable to automate this phase of the process. It will also be made an analysis of the cartography production process, as well as the available resources, to define the most appropriate method to use the models.

Keywords: Cartography, Land Cover, Deep Learning, Automation

Agradecimentos

Gostaria de agradecer aos meus pais e irmão pelo apoio durante todo o percurso académico.

Queria também agradecer à InfoPortugal S.A. pela confiança que depositaram em mim para o desenvolvimento deste projeto, e em especial ao Nuno Lopes e ao Edgar Pereira pela sua contribuição neste trabalho, que foi essencial para o sucesso do mesmo.

Também gostaria de agradecer aos meus orientadores, o Professor Doutor Carlos Ferreira e a Professora Doutora Fátima Rodrigues por toda a ajuda dada durante o projeto, quer a nível técnico, quer na produção do presente do documento.

Por fim, e por não ser possível referir toda a gente, gostaria de agradecer a todos que contribuíram direta e indiretamente para o desenvolvimento deste trabalho.

Conteúdo

Lista de Figuras	xiii
Lista de Tabelas	xv
Lista de Algoritmos	xv
Lista de Código	xv
Lista de Acrónimos	xvii
1 Introdução	1
1.1 Enquadramento	1
1.2 Problema	2
1.3 Objetivos	2
1.4 Abordagem	3
1.5 Resultados esperados	4
1.6 Estrutura do documento	5
2 Contexto e Estado da Arte	7
2.1 Contexto	7
2.2 Dados	8
2.2.1 Problemas dos dados	9
2.3 Análise de valor	10
2.3.1 New Concept Development Model	10
Identificação da oportunidade	10
Análise da oportunidade	11
Formação da ideia	11
Seleção da ideia	11
Definição do conceito	11
2.3.2 Valor	11
2.3.3 Proposta de valor	12
2.3.4 Modelo <i>Canvas</i>	12
2.4 Estado da arte	13
2.4.1 <i>Deep Learning</i>	13
2.4.2 Métricas de avaliação	15
Top-N error	15
<i>Accuracy</i>	15
<i>Precision</i>	16
<i>Recall</i>	16
F1	16
Intersection over Union	16
2.4.3 Arquiteturas de modelos	17

	Convolutional Neural Networks	17
	Very Deep Convolutional Neuronal Networks	18
	Fully Convolutional Neuronal Networks	19
	U-Net	20
	DeepLab	21
	Fully Convolutional Dense Networks	22
	Relation-Augmented Fully Convolutional Network	24
2.4.4	Abordagens existentes	26
	Learning Aerial Image Segmentation from Online Maps	26
	Fully Residual Convolutional Neural Networks for Aerial Image Segmentation	26
	DeepGlobe Land Cover Classification Challenge	27
2.4.5	Tecnologias	27
	TensorFlow	27
	PyTorch	27
	Keras	27
	FastAI	27
	Land Cover Mapping API	28
3	Avaliação das soluções/tecnologias existentes	29
3.1	Avaliação das soluções existentes	29
3.1.1	Learning Aerial Image Segmentation from Online Maps	29
3.1.2	Fully Residual Convolutional Neural Networks for Aerial Image Segmentation	32
3.1.3	DeepGlobe Land Cover Classification Challenge	33
3.2	Avaliação das tecnologias existentes	34
4	Design da solução	37
4.1	Arquitetura da solução	37
4.1.1	Modelo de domínio	37
4.1.2	Casos de uso	38
4.1.3	Algoritmo de classificação automática	39
	<i>Pipeline de dados</i>	40
	Melhoria contínua	40
	Conversão do <i>output</i> para ficheiros geográficos	41
4.1.4	Aplicação Web	41
4.1.5	Organização dos componentes	41
5	Construção da solução	43
5.1	Modelo de segmentação semântica	43
5.1.1	Pré-processamento dos dados	43
5.1.2	Seleção de subcategorias	45
5.1.3	Parâmetros de treino dos modelos	46
	Optimizador	46
	<i>Loss function</i>	46
	<i>LR range test</i>	46
	<i>1cycle</i>	47
	<i>Transfer learning</i>	48
	<i>Progressive resizing</i>	48

<i>Data augmentation</i>	48
5.1.4 Modelos utilizados	49
5.1.5 Resultados dos modelos para categorias	49
5.1.6 Resultados dos modelos para subcategorias	50
5.2 Aplicação <i>web</i>	51
6 Avaliação da solução	55
6.1 Métricas	55
6.2 Metodologia de avaliação	55
7 Conclusões	57
7.1 Trabalho futuro	57
Bibliografia	59
A Análise dos dados das categorias	63

Lista de Figuras

1.1	Metodologia Cross-industry standard process for data mining (CRISP-DM) (Chapman et al. n.d.).	4
2.1	Exemplo de fotografia aérea (esquerda) e visualização da informação disponíveis nos <i>shapefiles</i> para a mesma (direita).	9
2.2	Modelo de Canvas	12
2.3	Representação da métrica Intersection over Union (IoU) (Jordan 2019). . .	17
2.4	Representação de uma camada <i>convolutional</i> (<i>A Comprehensive Guide to Convolutional Neural Networks</i> 2019)	18
2.5	Diferença entre uma Fully Convolutional Neuronal Networks (FCNN) e uma Convolutional Neuronal Networks (CNN) (Long, Shelhamer e Darrell 2014). .	19
2.6	Arquitetura da U-Net (Ronneberger, Fischer e Brox 2015).	20
2.7	Diferença entre <i>convolution</i> tradicional (esquerda) e <i>atrous convolution</i> (Gsaxner et al. 2019).	21
2.8	<i>Atrous spatial pyramid pooling</i> (Chen et al. 2016).	22
2.9	Estrutura de um bloco numa <i>Dense Networks</i> (<i>DenseNet</i>) (Huang et al. 2016)	23
2.10	Arquitetura da Fully Convolutional Dense Networks (FCDN) (Jégou et al. 2016).	23
3.1	Arquitetura da rede utilizada em Kaiser et al. (2017).	30
3.2	Tipos de redes utilizados no <i>Fully Residual Convolutional Neural Networks for Aerial Image Segmentation</i> (Sang e Minh 2018).	32
3.3	Arquitetura do modelo vencedor do DeepGlobe Land Cover Classification Challenge (Tian, Li e Shi 2018).	34
3.4	Número de referências ao PyTorch e TensorFlow em artigos publicados no arXiv.org (O'Reilly 2019).	35
4.1	Modelo de domínio	37
4.2	Diagrama de casos de uso	38
4.3	Exemplo de máscara com as categorias Áreas agrícolas (AGRI) (verde) e Construções (CONC) (vermelho) (à esquerda) e sobreposição da mesma sobre a imagem original (à direita).	39
4.4	Diagrama de componentes	41
5.1	Exemplo de geometria de limites/eixo de vias (esquerda) e polígono para a mesma área (direita).	44
5.2	Exemplo da técnica <i>LR range test</i>	47
5.3	Variação da <i>learning rate</i> e do <i>momentum</i> no <i>1cycle</i> (Gugger 2019). . . .	47
5.4	Evolução da <i>accuracy</i> num modelo utilizando a técnica <i>1cycle</i> (a amarelo e laranja) e a técnica tradicional (azul) (Smith 2018).	48
5.5	Resultados obtidos para uma fotografia aérea pelos modelos de categorias. .	50
5.6	Resultados obtidos para uma fotografia aérea pelos modelos de subcategorias. .	51

5.7	Seleção do diretório com imagens a categorizar.	52
5.8	Escolha do tipo de modelo a utilizar.	52
5.9	Processo de categorização das imagens.	53
5.10	Listagem de ficheiros com previsões.	54
5.11	Visualizador de previsões.	54
6.1	Evolução das métricas no processo de treino.	56

Lista de Tabelas

2.1	Valor numa perspetiva longitudinal	11
2.2	Resultados da edição de 2010 do ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krizhevsky, Sutskever e G. E. Hinton 2012).	18
2.3	Precisão de CNN com diferentes níveis de profundidade (Simonyan e Zisserman 2014).	19
2.4	Comparação de FCNN com algoritmos de estado de arte no PASCAL Visual Object Classes (PASCAL VOC) 2012 (Long, Shelhamer e Darrell 2014).	20
2.5	Resultados do International Symposium on Biomedical Imaging (ISBI) cell tracking challenge 2015.	21
2.6	Resultados do DeepLab para o <i>dataset</i> do PASCAL VOC 2012 (Chen, Papandreou, Schroff et al. 2017).	22
2.7	Resultados obtidos com o <i>dataset</i> CamVid (Jégou et al. 2016).	24
2.8	Ilustração das relações entre elementos numa imagem (Mou, Hua e Zhu 2019).	24
2.9	Módulos de relacionamento espacial (a) e relacionamento de canais (b) (Mou, Hua e Zhu 2019).	25
2.10	Resultados do Relation-Augmented Fully Convolutional Network (RA-FCN) para o <i>dataset</i> de Vaihingen (Mou, Hua e Zhu 2019).	25
2.11	Resultados do RA-FCN para o <i>dataset</i> de Postdam (Mou, Hua e Zhu 2019).	26
3.1	Resultados obtidos no <i>Learning Aerial Image Segmentation from Online Maps</i> (Kaiser et al. 2017).	31
3.2	Resultados obtidos no <i>Fully Residual Convolutional Neural Networks for Aerial Image Segmentation</i> (Sang e Minh 2018).	32
5.1	Detalhes das subcategorias selecionadas para treino do modelo.	46
5.2	Comparação dos modelos para as categorias.	49
5.3	Comparação dos modelos para as subcategorias.	51
A.1	Análise dos dados para a categoria Áreas verdes, desportivas e de lazer (ADVR).	63
A.2	Análise dos dados para a categoria AGRI.	64
A.3	Análise dos dados para a categoria CONC.	66
A.4	Análise dos dados para a categoria Cursos hidrográficos (HIDA).	67

Lista de Acrónimos

ADVR	Áreas verdes, desportivas e de lazer.
AGRI	Áreas agrícolas.
CNN	Convolutional Neuronal Networks.
CNN	Very Deep Convolutional Neuronal Networks.
CONC	Construções.
CRISP-DM	Cross-industry standard process for data mining.
CVPR	Computer Vision and Pattern Recognition.
DenseNet	Dense Networks.
DGT	Direção-Geral do Território.
FCDN	Fully Convolutional Dense Networks.
FCNN	Fully Convolutional Neuronal Networks.
FN	Falso Negativo.
FP	Falso Positivo.
HIDA	Cursos hidrográficos.
ILSVRC	ImageNet Large Scale Visual Recognition Challenge.
IoU	Intersection over Union.
ISBI	International Symposium on Biomedical Imaging.
ISPRS	International Society for Photogrammetry and Remote Sensing.
LAS	Laser.
OSM	OpenStreetMap.
PASCAL VOC	PASCAL Visual Object Classes.
PDM	Plano Diretor Municipal.
RA-FCN	Relation-Augmented Fully Convolutional Network.
RGB	<i>Red, Green, Blue.</i>
RGBI	<i>Red, Green, Blue, Infrared.</i>
RTS	Real-time strategy.

SIG	Sistemas de Informação Geográfica.
VN	Verdadeiro Negativo.
VP	Verdadeiro Positivo.

Capítulo 1

Introdução

Neste primeiro capítulo é apresentado, de uma forma resumida, o problema que deu origem a este trabalho, os objetivos e resultados que se pretendem atingir, bem como a abordagem a seguir para o desenvolvimento da solução.

1.1 Enquadramento

A cartografia é um produto essencial no nosso dia a dia, sem mapas a navegação para locais não conhecidos seria uma tarefa extremamente complicada e, apesar de nos dias de hoje o uso de cartografia tradicional ser cada vez menos comum, o uso desta ferramenta de navegação tem vindo a aumentar, dado que qualquer pessoa facilmente possui um *smartphone* ou um automóvel com sistema de navegação integrado. Na área da tecnologia, cada vez mais, a georreferenciação é um fator importante, sendo que agora, para além de se saber o que e o porquê de ter acontecido, é bastante útil saber o local, o "onde".

Outra área onde a cartografia é de extrema importância é a do ordenamento do território, visto ser essencial manter um registo da ocupação e uso de terrenos e solo, e quais os tipos de área existente na sua área de jurisdição (áreas protegidas, zonas de construção, entre outras). Este registo é conhecido por Plano Diretor Municipal (PDM) e é o plano base para todas as decisões sobre gestão de território municipal.

Torna-se então indispensável que a cartografia utilizada esteja sempre o mais atualizada possível. O uso de cartografia com erros pode causar danos graves e grandes perdas ao nível do negócio. Por exemplo, no uso em sistemas de navegação para uso do público geral, um erro na atualização de sentidos em vias de trânsito pode levar a acidentes, e a omissão de estradas pode causar perdas para uma empresa transportadora, enquanto que na utilização por órgãos governamentais um erro no mapeamento de condutas de gás pode causar uma catástrofe, ou então a autorização de construções em zonas não permitidas por estas estarem mal categorizadas.

Enquanto que no uso de sistemas de navegação, a atualização é impulsionada pela necessidade de manter os utilizadores satisfeitos, e conseqüentemente um benefício para o negócio, no caso do uso por órgãos governamentais existe legislação que obriga à atualização periódica da cartografia disponível.

Normalmente, a cartografia é desenvolvida com recurso a fotografia aérea ou imagem de satélite, sendo necessário garantir que os elementos presentes nesta são corretamente representados. Estes elementos podem ser, por exemplo: estradas, construções, áreas agrícolas, entre outros. Para uso de instituições governamentais é comum ser feito um levantamento

dos elementos a diversas escalas, sendo que cada uma tem níveis de detalhe diferentes e necessita de posterior homologação pela Direção-Geral do Território (DGT) para a sua utilização. Este processo é bastante demorado devido à sua minúcia, e normalmente implica que seja feita completagem de campo (deslocação ao local para categorizar elementos que não são possíveis de identificar apenas com o recurso à fotografia e/ou corrigir elementos mal categorizados).

A InfoPortugal é uma empresa que se divide em três grandes áreas de atuação: cartografia e Sistemas de Informação Geográfica (SIG), produção de conteúdos e desenvolvimento de software. A área de cartografia e SIG conta com uma base de dados geográfica com georreferenciação de mais de 130 mil pontos de interesse, 170500 km de eixos de via, 6559 km² de áreas protegidas e 11639 monumentos (*InfoPortugal* 2019). É com recurso a estes dados e aos levantamentos de fotografia aérea que é possível o desenvolvimento da cartografia .

1.2 Problema

No desenvolvimento de cartografia, a fase de categorização de elementos implica um uso intensivo de recursos humanos e, por norma, é a mais demorada e custosa de todo o processo. Esta categorização é feita com recurso a fotografia aérea, sendo posteriormente complementada com uma completagem de campo.

A categorização de elementos é um trabalho bastante repetitivo que é feito ao longo de vários dias, o que torna mais provável o erro humano, devido a se realizar a mesma tarefa constantemente e/ou por outros fatores psicológicos ou emocionais que possam estar a afetar as pessoas responsáveis por este trabalho (segundo estudo feito em Indústrias Transformadoras na Malásia, 48,8% da variação no erro humano pode ser explicado pelo stress, repetição, fadiga e ambiente de trabalho (Yeow et al. 2014)).

Foi identificado pela empresa que a automatização deste processo levaria a uma maior produtividade na produção de cartografia, bem como a redução dos custos deste processo.

1.3 Objetivos

O objetivo deste trabalho passa pela automatização de parte do processo de categorização dos elementos presentes na fotografia aérea, com uma precisão que não difira muito dos resultados obtidos quando o mesmo é feito por pessoas, passando o trabalho destas a ser a validação das categorizações, em detrimento da criação de raiz de todos os elementos.

Para atingir este objetivo será desenvolvido um modelo de segmentação semântica de imagens (categorização pixel a pixel dos elementos presentes na imagem), utilizando algoritmos de aprendizagem automática (*deep learning*), com recurso aos dados obtidos em projetos de cartografia anteriormente desenvolvidos, sendo este o artefacto principal deste trabalho. Também se pretende desenvolver uma aplicação *web*, para visualização das categorizações feitas pelo algoritmo de aprendizagem automática e a sua validação, tendo também disponível algumas ferramentas de análise de resultados (por exemplo: comparação de categorizações de uma mesma área geográfica em momentos diferentes para permitir ver as possíveis diferenças, permitir a análise de zonas verdes com proximidade a habitações para prevenção

de riscos de incêndio). A aplicação *web* apenas servirá como uma *interface* para interação com o modelo de aprendizagem automática, pelo que o seu desenvolvimento será menos prioritário.

De forma a permitir uma melhor integração da solução desenvolvida será analisado o processo de produção de cartografia de forma a otimizar a sua utilização, evitando alterações drásticas nos processos de trabalho da empresa.

1.4 Abordagem

Para desenvolvimento do modelo de categorização automática serão utilizados algoritmos de *deep learning*. O *deep learning* é uma sub-área de *machine learning* que tem vindo a registar um grande crescimento, na qual a aprendizagem da máquina é feita a partir de exemplos, isto é, a máquina aprende com os dados existentes e generaliza um modelo matemático que melhor se adequa a estes (LeCun, Bengio e G. Hinton 2015).

Depois de analisados o catálogo de dados existente na empresa e algumas soluções para problemas semelhantes, foi definido que este seria o tipo de algoritmo a utilizar. A escolha do modelo mais adequado para este problema, será feita com recurso a uma análise experimental aos resultados obtidos com diferentes modelos de redes neuronais já existentes, de modo a escolher-se a mais adequada.

Uma vez que o modelo de categorização automática não é facilmente integrável, de uma forma isolada, no atual fluxo de trabalho, foi definido que deveria ser desenvolvida uma aplicação *web* onde fosse possível visualizar os resultados, bem como algumas ferramentas para análise dos mesmos.

A linguagem de programação a utilizar no desenvolvimento do modelo de segmentação semântica e da aplicação *web* será *Python*, visto ser aquela com mais soluções para a área de *deep learning*. Aliado a isto, o *Python* também permite o desenvolvimento de aplicações *web* e é, atualmente, a mais utilizada na área de desenvolvimento da empresa.

O processo de desenvolvimento do modelo de segmentação semântica terá por base a metodologia Cross-industry standard process for data mining (CRISP-DM). Esta metodologia é cíclica e não existe uma sequência linear entre cada uma das 6 fases que a compõem, já que pode ser necessário retificar fases anteriores do processo em qualquer fase do mesmo.

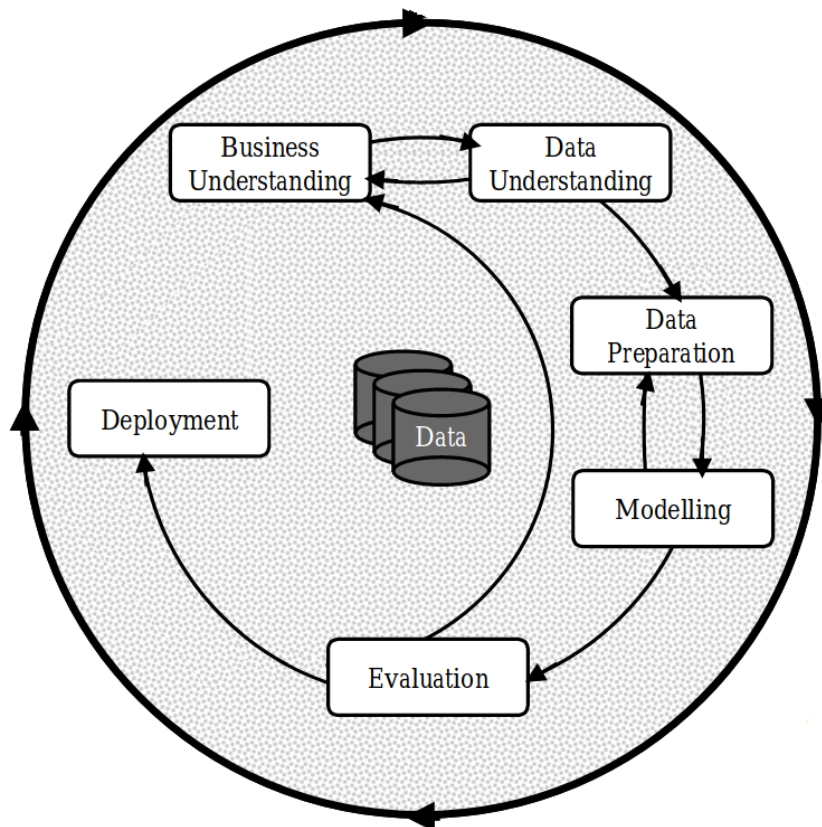


Figura 1.1: Metodologia CRISP-DM (Chapman et al. n.d.).

As 6 fases que constituem o processo são as seguintes:

- **Business Understanding:** Definição dos objetivos do projeto e dos dados a obter, após avaliação e estudo da lógica de negócio.
- **Data Understanding:** Coleção e análise exploratória dos dados.
- **Data Preparation:** Seleção e tratamento dos dados a utilizar.
- **Modelling:** Criação dos modelos.
- **Evaluation:** Avaliação dos resultados obtidos.
- **Deployment:** *Deploy* do projeto.

1.5 Resultados esperados

Pretende-se, com este trabalho, desenvolver um modelo de segmentação semântica capaz de categorizar automaticamente elementos presentes em fotografia aérea, com uma taxa de acerto que permita a substituição de grande parte do trabalho humano (o trabalho humano passaria a ser a validação e correção dos dados e da categorização automática).

Espera-se também que a aplicação *web* desenvolvida permita um uso simples e intuitivo, para que o processo de categorização e validação sejam eficientes e que as ferramentas de análise de resultados disponibilizadas acrescentem valor ao processo existente.

Globalmente, é esperado que este trabalho permita uma melhor rentabilidade e eficiência do processo de produção de cartografia.

1.6 Estrutura do documento

No primeiro capítulo, é feita uma breve apresentação do problema e do contexto em que se inclui. São também traçados os objetivos que se pretendem atingir e qual a abordagem a seguir.

No segundo capítulo, é descrito de forma mais extensa, o contexto do problema, sendo apresentados os dados disponíveis para o desenvolvimento da solução. É também feita a análise de valor do projeto e ao estado da arte, na área da segmentação semântica com recurso a *deep learning*.

No capítulo três, são avaliadas soluções semelhantes para este tipo de problema, bem como tecnologias que possam ser utilizadas para desenvolvimento da solução.

No capítulo quatro, são apresentados de forma breve, os requisitos do projeto e a arquitetura a utilizar na implementação da solução.

No quinto capítulo, apresenta-se a solução desenvolvida.

No sexto capítulo, são apresentadas as metodologias e métricas utilizadas para avaliar os resultados obtidos durante o desenvolvimento do projeto.

No sétimo e último capítulo, apresentam-se as conclusões sobre o trabalho desenvolvido, assim como se mostram caminhos para trabalhos futuros.

Capítulo 2

Contexto e Estado da Arte

Neste capítulo é apresentado o contexto envolvente do projeto, onde é detalhado o processo de trabalho existente, bem como os dados disponíveis para análise. É também feita uma análise de valor ao projeto e, por fim, é avaliado o estado da arte.

2.1 Contexto

A InfoPortugal possui uma área de cartografia, que é responsável pela produção dos seguintes produtos: cartografia topográfica ("cartografia de finalidade múltipla, representando na forma analógica ou digital, os acidentes naturais e artificiais, de acordo com a escala de representação"(*DGTerritório - Cartografia topográfica 2019*)), cartografia temática ("a cartografia de finalidade singular representando fenómenos localizáveis de qualquer natureza, quantitativos ou qualitativos, sobre uma base topográfica mais ou menos simplificada"(*DGTerritório - Cartografia temática de base topográfica 2019*)) e cadastro ("conjunto dos dados que caracterizam e identificam os prédios existentes em território nacional"(*DGTerritório - Cadastro 2019*)).

O processo de produção de cartografia é composto pelos seguintes passos:

1. Recolha de fotografia aérea

Numa primeira fase, é efectuada a recolha de fotografia aérea para a zona que se pretende cartografar. É feita segundo um plano de voo definido previamente, visto ser necessário garantir sobreposição de imagem para permitir a sua visualização em 3D em fases posteriores do processo.

2. Categorização dos elementos da fotografia aérea

Terminada a fase de recolha de fotografia aérea e o seu processamento, é feita a categorização dos elementos nelas presentes. O tipo de elementos a categorizar está dependente da escala da cartografia a produzir (quanto maior a escala, mais detalhada é a categorização). Esta categorização é guardada em ficheiros geográficos (por exemplo *shapefiles*), sendo que a empresa possui um arquivo destes ficheiros e de fotografias aéreas para os projetos desenvolvidos.

Este processo normalmente é feito com recurso a *outsourcing* (nos projetos de pequenas áreas, que são menos comuns, o processo é feito internamente) e envolve um maior uso de recursos humanos, representando a maior fatia do custo de produção de cartografia.

Quando efetuado com recurso a *outsourcing*, é entregue uma categorização simplificada (dado que o elevado número de categorias necessários para homologação implicaria a necessidade de formação das pessoas que realizam este trabalho), sendo necessário adaptar as categorizações feitas e corrigir eventuais falhas. A aceitação do produto, é feita com base na avaliação de um operador de cartografia que verifica a não existência de um grande volume de erros.

3. Completagem de campo

Visto existirem elementos invisíveis (por estarem cobertos por vegetação, por exemplo), ou que não são completamente perceptíveis a partir de fotografia aérea, é executado um trabalho de campo para completar dados em omissão e/ou corrigir dados que estejam errados.

O processo de categorização de elementos é repetido para todos os projetos, mesmo que se trate de zonas que já tenham sido feitas anteriormente, visto ser trabalhoso identificar áreas que tenham sofrido alterações, sendo então necessário repetir todo o processo para garantir que tudo está corretamente categorizado.

Pretende-se então, usar os dados existentes para desenvolver um modelo de aprendizagem automática, capaz de categorizar os objetos da fotografia aérea automaticamente. Desta forma, a categorização manual seria substituída por um trabalho de validação, estando o operador encarregue de validar os dados e corrigir eventuais erros, permitindo agilizar a produção da cartografia, assim como a redução de custos. A viabilidade da sua utilização será feita através da avaliação dos resultados obtidos pelo operador de cartografia (tal como é feito no caso do *outsourcing*).

2.2 Dados

Existem dois tipos de dados disponíveis para treino do modelo de segmentação semântica: fotografias aéreas e ficheiros geográficos com a categorização de elementos.

As fotografias aéreas estão disponíveis, num ou mais, dos seguintes formatos:

- *Red, Green, Blue* (RGB): Contêm dados nos canais vermelho, verde e azul, como as fotografias comuns.
- *Red, Green, Blue, Infrared* (RGBI): Para além dos canais do RGB, inclui um canal de infravermelhos.
- Nuvem de pontos: Restituição da fotografia em formato Laser (LAS), que para além dos dados contidos nos formatos anteriores, contém a altura de cada ponto.

O uso da nuvem de pontos implica a normalização da mesma, já que neste formato se podem obter matrizes irregulares (para cada coordenada pode-se obter múltiplos pontos).

Quanto à categorização, está armazenada em ficheiros geográficos que podem existir em vários formatos digitais, sendo o mais comum, no arquivo existente, o *shapefile*.

Existem múltiplas categorias de elementos, tendo sido definido que para o contexto deste projeto apenas seriam utilizadas as que contêm polígonos (os elementos podem ser pontos, linhas ou polígonos), que são as seguintes:

- Áreas agrícolas (AGRI)

- Áreas verdes, desportivas e de lazer (ADVR)
- Construções (CONC)
- Cursos hidrográficos (HIDA)
- Estradas

Ao contrário das outras categorias, as estradas não estão armazenadas como polígonos, visto ser necessário a categorização dos eixos e limites de via. Apesar disso com o cruzamento destas duas informações é possível obter polígonos para as estradas.

Cada uma destas categorias está armazenada num ficheiro à parte e é composta por múltiplas sub-categorias.



Figura 2.1: Exemplo de fotografia aérea (esquerda) e visualização da informação disponíveis nos *shapefiles* para a mesma (direita).

2.2.1 Problemas dos dados

Apesar do grande volume de dados disponíveis, podem existir alguns problemas na utilização dos mesmos, tendo sido identificados três problemas. A saber:

- Dados incompletos

É possível que em alguns projetos apenas esteja disponível uma das partes dos dados necessários para o treino da rede, visto que alguns clientes apenas adquiriram as fotografias aéreas, enquanto outros apenas a categorização dos elementos. Nestes casos, para se poder utilizar os dados será necessário complementá-los, quer através do uso de fotografias obtidas de outras fontes (por exemplo, os ortofotomapas disponibilizados pela DGT), quer através da categorização manual das fotografias em falta.

- Dados com completagem

Todos os dados armazenados nos ficheiros geográficos correspondem ao produto final, isto é, já foi feita a completagem de campo que completa/corrigue os dados levantados na fotografia aérea, não existindo em arquivo o resultado obtido apenas com o levantamento a partir da fotografia aérea. Isto significa que o treino será feito utilizando dados que contêm algum grau de erro, já que não correspondem ao que é esperado que o algoritmo categorize, visto estarem presentes, na categorização de treino, elementos que não são identificáveis a partir da fotografia, podendo levar a erros de aprendizagem.

Apesar de mais raro é possível que entre o momento em que foi recolhida a fotografia aérea e a completagem de campo, alguns elementos possam ter sofrido alterações (por exemplo, pode ter existido um incêndio florestal) pelo que é possível que a categorização não corresponda ao que está visível na fotografia aérea.

- **Categorização não adequada**

Os dados estão separados pelas categorias definidas pela DGT, mas verificou-se que nem todas as sub-categorias se adequam em termos de parecença ao que seria desejável para o treino do algoritmo (por exemplo a sub-categoria jardins pertence à categoria ADVR, mas em termos de fotografia aérea, a categoria mais adequada seria a AGRI). Para se obter uma separação semântica, poderá ser necessária uma reestruturação das mesmas.

- **Categorização demasiado específica**

No caso das sub-categorias, existem casos em que não é expectável que se consiga obter resultados utilizando o algoritmo de aprendizagem automática, já que são distinções que não são possíveis efetuar apenas com recurso à fotografia aérea. Por exemplo, no caso da categoria CONC, são sub-categorias as tipologias de edifício (centro-comercial, teatro, tribunal) e apenas com recurso à fotografia aérea não é possível especificar a sub-categoria dos mesmos.

Estes problemas nos dados vão exigir que seja feita uma seleção e pré-processamento adequados.

2.3 Análise de valor

Nesta secção será feita uma análise segundo múltiplos modelos do valor de negócio deste trabalho.

2.3.1 New Concept Development Model

O *New Concept Development Model* é um modelo no qual o valor de um produto pode ser obtido respondendo a cinco pontos chave (Koen et al. 2001). Em seguida é feita a análise de valor do presente projeto segundo este modelo.

Identificação da oportunidade

A categorização de elementos em fotografia é uma tarefa manual e repetitiva e que acarreta uma grande parte dos custos de desenvolvimento de cartografia. Com a evolução do *machine learning*, é possível obter cada vez melhores resultados em tarefas deste género, permitindo a substituição do trabalho humano por máquinas. Tudo isto, torna esta tarefa numa excelente oportunidade para estudar a implementação deste tipo de algoritmos.

Análise da oportunidade

Estando os algoritmos de *deep learning* para segmentação de elementos em imagens cada vez mais evoluídos e tendo a empresa uma grande base de dados para treinar um algoritmo desta natureza, pretende-se então, substituir uma tarefa integralmente manual, por uma solução semi-automática em que o trabalho manual passa pela validação e correção dos dados obtidos.

Formação da ideia

Para uma melhor definição de qual a melhor forma de integrar o algoritmo no atual fluxo de trabalho foram realizadas várias reuniões com os vários responsáveis pelo processo, nas quais foram discutidas várias funcionalidades que acrescentariam valor ao algoritmo desenvolvido.

Seleção da ideia

Chegou-se à conclusão que para além da visualização dos resultados obtidos, também seria muito importante a disponibilização de algumas ferramentas de análise dos dados para permitir uma melhor rentabilização dos mesmos, estando o algoritmo desenvolvido integrado numa aplicação *web*.

Definição do conceito

De modo a permitir integrar todos os requisitos levantados e de forma a permitir uma adaptação mais simples das pessoas envolvidas no processo, foi definido que deveria ser desenvolvida uma aplicação *web*, que oferecesse uma *interface* semelhante a outras aplicações SIG já utilizadas.

2.3.2 Valor

Para Woodall (2003), o valor pode ser temporalmente dividido em 4 etapas, desde a sua aquisição até ao pós-uso. O valor do produto para o cliente varia em cada uma destas fases, influenciado pelos benefícios/sacrifícios do mesmo ao longo do tempo. Na tabela 2.1 é possível observar esta divisão no contexto deste trabalho.

	Benefícios	Sacrifícios
Ex Ante VC	Automatização de processo Aumento produtividade	Tempo Custos de hardware
Transaction VC	Confiabilidade	Tempo
Ex Post VC	Suporte Evolução do algoritmo	Tempo
Disposition VC	Otimização de processos Maior rentabilidade	

Tabela 2.1: Valor numa perspetiva longitudinal

2.3.3 Proposta de valor

Desenvolvimento de um algoritmo de categorização de elementos presentes em fotografia aérea com elevada taxa de acerto, que permita substituir o trabalho realizado de forma manual por um processo semi-automático, em que o único trabalho manual seja a validação e correção dos dados obtidos e fazer a sua integração numa aplicação *web* que permita verificar os resultados, bem como fazer análise de alguns dados.

A implementação desta solução permite reduzir os custos, bem como os recursos humanos envolvidos no processo de desenvolvimento de cartografia, tornando-o mais rentável.

2.3.4 Modelo Canvas

Na figura 2.2, é possível ver o modelo de *Canvas* para o contexto deste projeto.

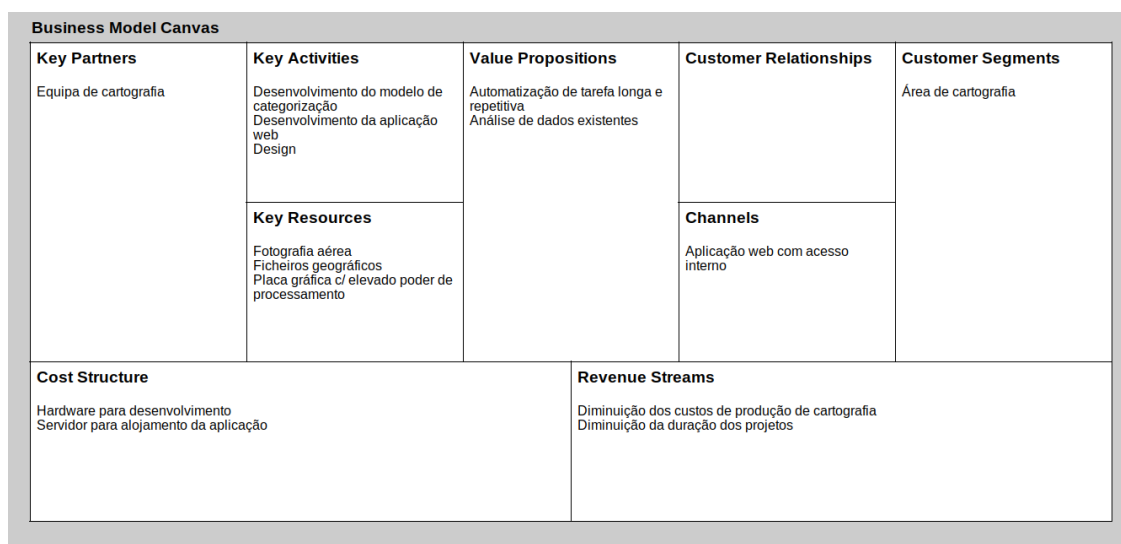


Figura 2.2: Modelo de Canvas

- **Key Partners**

No contexto deste projeto a equipa de cartografia da InfoPortugal é um parceiro chave, pois têm um conhecimento profundo de todo o processo e são os fornecedores dos recursos necessários à execução do projeto.

- **Key Activities**

As principais atividades deste projeto são o desenvolvimento do modelo de categorização utilizando segmentação semântica e da aplicação web para aceder ao modelo. É também contemplado o processo de design de toda a solução.

- **Key Resources**

Os recursos chave para desenvolvimento deste projeto são as fotografias aéreas e os ficheiros geográficos com os elementos categorizados. Também é necessário *hardware* gráfico com elevado poder de processamento para ser possível desenvolver o modelo de segmentação semântica.

- **Value Propositions**

Como propostas de valor têm-se a automatização de um processo que atualmente é realizado manualmente e que acarreta bastantes custos. Também se vê valor na análise dos resultados obtidos com recurso à aplicação web.

- **Customer Relationships**

Dado se tratar de um projeto interno não existe a necessidade de manter e/ou alargar a base de clientes.

- **Customer Segments**

Para além de serem um parceiro chave, no contexto deste projeto a área de cartografia é o cliente, visto serem os utilizadores deste projeto.

- **Channels**

O canal de distribuição do produto é a aplicação web que estará acessível internamente.

- **Cost Structure**

Em termos de custo será necessário investir em *hardware* para o desenvolvimento da solução e também num servidor para alojamento da aplicação web.

- **Revenue Streams**

Apesar de não se obter rendimento direto a partir desta solução, o seu uso reduz os custos de produção de cartografia e o tempo de execução dos projetos, levando ao aumento de rentabilidade dos projetos.

2.4 Estado da arte

Nesta secção é efetuada uma análise ao estado da arte da área de *deep learning*, e mais concretamente a algoritmos de segmentação semântica de imagens.

2.4.1 Deep Learning

O *deep learning* é um ramo de *machine learning* em que se ensina o computador a fazer algo que os humanos fazem naturalmente - aprender por exemplo. Apesar das primeiras referências a este método de aprendizagem por máquinas datar dos anos 80, apenas recentemente têm sido registados grandes avanços e bons resultados na sua aplicação, e isto deve-se principalmente a dois fatores: a existência de um grande volume de dados etiquetados (essencial para a obtenção de bons resultados) e o aumento do poder computacional, dado o elevado custo de processamento necessário para o treino deste tipo de algoritmos (Najafabadi et al. 2015).

Os algoritmos de *deep learning* são bastante flexíveis e podem ser aplicados em diversas áreas de conhecimento com bons resultados, como por exemplo na área automóvel, através dos veículos de condução autónoma e na saúde, para deteção precoce de doenças (LeCun, Bengio e G. Hinton 2015). Uma das formas mais utilizadas para demonstrar a evolução desta tecnologia é através de jogos, onde o objetivo é perceber se a máquina consegue vencer o homem, e neste domínio tem-se tido uma evolução muito notória da capacidade da máquina

em aprender e se tornar especialista em vários jogos. O primeiro grande resultado deste tipo de confronto homem-máquina foi em 1996 quando uma máquina da IBM chamada *Deep Blue* derrotou o campeão em título de xadrez, Garry Kasparov, apesar deste resultado ter sido obtido com recurso a algoritmos de força-bruta e não de *deep learning* (Forbes 2019).

Mais recentemente e já com recurso a *deep learning* a empresa *DeepMind* (subsidiária da Google) desenvolveu o *AlphaGo*, uma máquina feita para jogar o jogo de tabuleiro Go, que ao contrário do xadrez, é demasiado complexo para utilização de força-bruta (existem 10^{170} configurações possíveis do tabuleiro), sendo o treino feito com recurso a milhares de jogos de humanos, amadores e profissionais. Em março de 2016 o *AlphaGo* derrotou o 18 vezes campeão do mundo de Go, Lee Sedol. Em Outubro de 2017, a *Deepmind* lançou uma nova versão do *AlphaGo*, chamado *AlphaGo Zero* e ao contrário do seu antecessor este não necessita de jogos humanos para aprender a jogar, sendo ensinado à máquina as regras do jogo e a aprendizagem faz-se em jogos contra si própria, tendo ao fim de 3 dias de treino ultrapassado a habilidade do *AlphaGo* que derrotou o Lee Sedol (*AlphaGo - DeepMind* 2019). No final de 2017 foi apresentado o *AlphaZero*, sendo que esta versão para além de Go também consegue aprender a jogar xadrez e shogi, batendo um campeão do mundo em cada um deles (*AlphaZero - DeepMind* 2019). Já em 2018, foi apresentado o *AlphaStar*, uma máquina capaz de jogar *Starcraft 2* (um jogo de computador de *Real-time strategy (RTS)*), considerado dos mais complexos do género. Neste tipo de jogo, ao invés dos jogos de tabuleiro, um jogador não tem informação completa do estado atual do jogo, tendo de explorar o mapa para a obter. O treino foi feito inicialmente a partir de repetições de jogos de humanos, e depois jogando contra si próprio. Em dezembro de 2018 o *AlphaStar* bateu um dos melhores jogadores profissionais do mundo, Grzegorz "MaNa" Komincz (*AlphaStar - DeepMind* 2019).

Com este exemplo, é possível aferir a rapidez com que esta área tem evoluído, visto que no espaço de 2 anos uma máquina conseguiu derrotar um campeão do mundo em Go, tornar-se auto-suficiente no processo de treino, ser generalizada para aplicação noutros jogos de tabuleiro e passar do domínio dos jogos de informação completa para os de informação incompleta.

Em relação à área da deteção de elementos presentes em fotografia aérea, na qual este projeto se insere, também se tem registado o crescimento do interesse, bem como evoluções interessantes. O Facebook em parceria com a DigitalGlobe, a Purdue e o MIT, criaram o DeepGlobe, um desafio para evoluir o estado da arte na área de análise de imagens de satélite. Este desafio é composto por três tarefas de identificação de elementos em imagem de satélite: identificação de estradas, identificação de edifícios e classificação da cobertura do solo

Também o Facebook, agora em parceria com o OpenStreetMap (OSM), desenvolveu uma ferramenta (com recurso a *Deep Learning*), o RapiD. Esta ferramenta tem como objetivo a identificação de estradas a partir de imagens de satélite, e funciona como auxiliar para os editores do OSM, já que identifica no mapa troços que têm alta probabilidade de serem estradas. É função do editor aceitar aquele troço e, se necessário, efetuar algum ajuste. Com o RapiD foi possível mapear aproximadamente 480000km de estradas na Tailândia em 18 meses, o que é metade do tempo que demoraria se fosse feito manualmente por uma equipa de 100 especialistas em mapeamento (*Facebook Artificial Intelligence* 2019).

2.4.2 Métricas de avaliação

Pela necessidade de avaliação dos resultados obtidos pelo modelo, para avaliar a sua qualidade e permitir a comparação do seu desempenho com o de outros, são utilizadas métricas que permitem atestar a qualidade, ou falta dela, dos resultados de um modelo. A rápida e constante evolução desta área, leva a que exista uma necessidade constante de comparar os resultados com outros obtidos por modelos semelhantes, para perceber se existe alguma evolução em termos de resultados. Esta necessidade conduz à existência de um conjunto de métricas que são mais utilizadas na generalidade dos casos, sendo aceite a sua validade para atestar a qualidade dos resultados.

Em seguida serão apresentadas algumas das métricas mais comuns na área da *computer vision*, que serão referidos ao longo do documento.

Top-N error

Esta métrica é bastante simples e representa a taxa de erro das previsões. Esta, representa a percentagem de casos em que a categoria correta não se encontra entre as N com maior probabilidade definidas pelo modelo (Shah, Bakrola e Pati 2019a).

Accuracy

A *accuracy* avalia o rácio entre as previsões corretas sobre o total de previsões, sendo que no caso da segmentação semântica, corresponde ao número de pixels corretamente categorizados a dividir pelo total de pixels da imagem (M e M.N 2015). Cada previsão pode ser classificada como como sendo:

- **Verdadeiro Positivo (VP)**: Elemento corretamente identificado como pertencendo a uma dada categoria.
- **Verdadeiro Negativo (VN)**: Elemento corretamente identificado como não pertencendo a uma dada categoria.
- **Falso Positivo (FP)**: Elemento incorretamente identificado como pertencendo a uma dada categoria.
- **Falso Negativo (FN)**: Elemento incorretamente identificado como não pertencendo a uma dada categoria.

Esta métrica é obtida usando a seguinte fórmula:

$$accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

Esta medida pode ser enganadora quando utilizada no âmbito da segmentação semântica, já que se uma categoria aparecer numa porção reduzida em cada uma das imagens avaliadas, os resultados podem ser inflacionados devido a uma elevada taxa de VN.

Precision

Esta métrica pode ser definida como a avaliação dos positivos da *accuracy*, isto é, para todos os elementos para qual a previsão foi uma dada categoria, qual a percentagem de acerto (M e M.N 2015).

$$precision = \frac{VP}{VP + FP}$$

Recall

O *recall* é uma métrica que permite saber qual a percentagem de acerto para uma dada categoria, em relação a todos os elementos que deveriam ter sido previstos como pertencendo a esta (M e M.N 2015).

$$recall = \frac{VP}{VP + FN}$$

F1

Esta métrica é uma média harmónica ponderada da *precision* e do *recall* (M e M.N 2015) e é obtido utilizando a fórmula:

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Intersection over Union

A Intersection over Union (IoU) permite saber a percentagem de sobreposição entre a máscara esperada e a prevista pelo algoritmo, através do rácio dos pixels presentes em ambas relativamente ao total de pixels das mesmas (Shah, Bakrola e Pati 2019b).

$$IoU = \frac{\text{Interseção das áreas das máscaras}}{\text{União das áreas das máscaras}}$$

Esta métrica é utilizada para avaliar a classificação de uma determinada categoria, sendo que é comum também ser apresentada a média dos IoU das múltiplas categorias (representada como mIoU) (Shah, Bakrola e Pati 2019b).

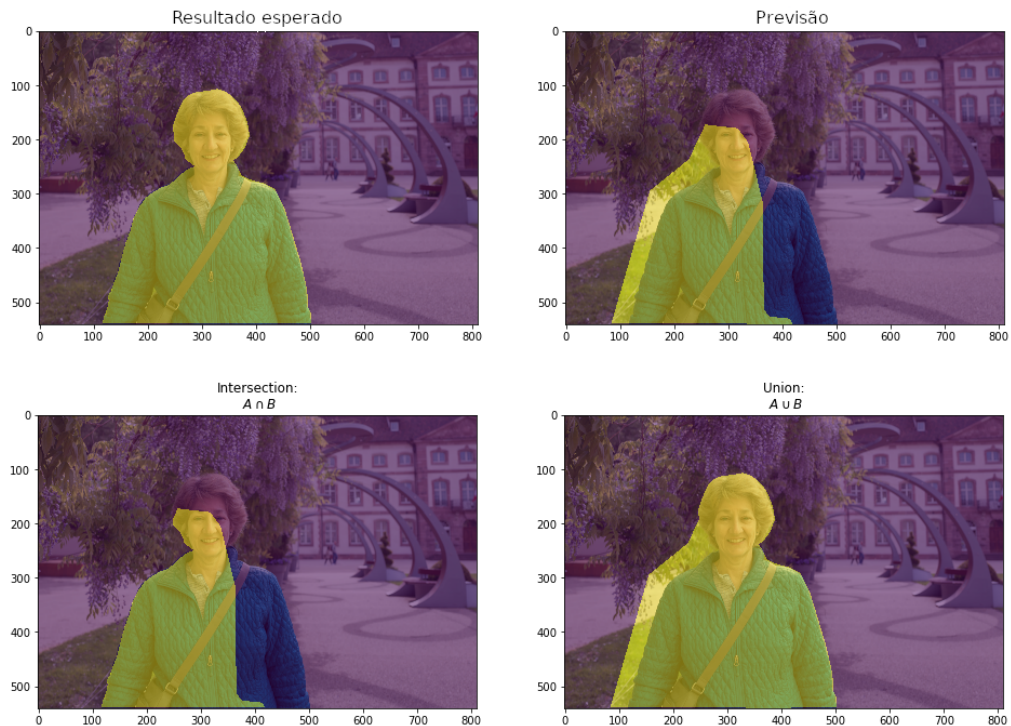


Figura 2.3: Representação da métrica IoU (Jordan 2019).

Esta métrica não sofre do mesmo problema identificado na *accuracy*.

2.4.3 Arquiteturas de modelos

Existem múltiplas arquiteturas de modelos de redes neurais com sucesso comprovado em tarefas de *computer vision*. Estas redes têm como objetivo a classificação de imagens como um todo (cada imagem é classificada com uma e uma só categoria) ou a segmentação semântica dos elementos existentes numa imagem (na mesma imagem podem existir elementos de múltiplas categorias). Seguidamente será feita uma análise de algumas arquiteturas de modelos.

Convolutional Neural Networks

As Convolutional Neural Networks (CNN) têm obtido excelentes resultados na área de detecção de objetos em imagem e vídeo, levando a que este tipo de redes sejam bastante utilizados na área de *computer vision* (Jégou et al. 2016). Esta rede tem este nome dada a sua composição por camadas *convolutional*, que fazem a redução da dimensão do *input* através do varrimento da imagem utilizando uma "janela" com um dado número de pixels, sobre o qual é aplicada uma função para obter o *output* (ver figura 2.4).

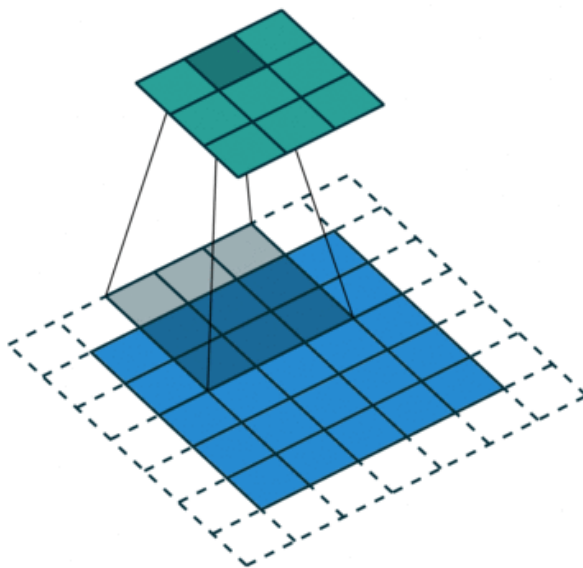


Figura 2.4: Representação de uma camada *convolucional* (A *Comprehensive Guide to Convolutional Neural Networks* 2019)

No ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (um desafio anual com o objetivo de aferir quais os algoritmos com melhor precisão em várias tarefas de reconhecimento de imagem), na edição de 2010, uma rede deste tipo entrou no concurso, obtendo resultados bastante melhores quando comparados com outro tipo de algoritmos, estando os mesmos na tabela 2.2 (Krizhevsky, Sutskever e G. E. Hinton 2012). Este modelo tem como propósito a classificação de imagens e foi avaliado com as métricas *top-1 error* e *top-5 error*.

Modelo	Top-1	Top-5
Sparse coding	47.1%	28.2%
SIFT+ FVs	45.7%	25.7%
CNN	37.5%	17.0%

Tabela 2.2: Resultados da edição de 2010 do ILSVRC (Krizhevsky, Sutskever e G. E. Hinton 2012).

Very Deep Convolutional Neuronal Networks

No artigo de Simonyan e Zisserman (2014) concluiu-se que ao aumentar a profundidade de uma CNN é possível obter uma melhor precisão comparativamente a algoritmos utilizados anteriormente no ILSVRC. Também foi possível atestar a sua aplicação noutros *datasets* de reconhecimento de imagem, também com bons resultados, já que com o aumento da profundidade a capacidade de generalização melhora. Na tabela 2.3 é possível verificar a diminuição da percentagem de erro em função do aumento das camadas de uma CNN. Para este teste foi utilizado o *dataset* da edição de 2012 do ILSVRC (Simonyan e Zisserman 2014). Este modelo tem como propósito a classificação de imagens e foi avaliado com as métricas *top-1 error* e *top-5 error*.

Config. ConvNet	menor lado da imagem		top-1 val. error (%)	top-5 val. error (%)
	treino	teste		
A (11 camadas)	256	256	29.6	10.4
A-LRN (11 camadas)	256	256	29.7	10.5
B (13 camadas)	256	256	28.7	9.9
C (16 camadas)	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D (16 camadas)	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E (19 camadas)	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	25.5	8.0

Tabela 2.3: Precisão de CNN com diferentes níveis de profundidade (Simonyan e Zisserman 2014).

Fully Convolutional Neuronal Networks

Este modelo foi desenvolvido tendo por base o conhecimento obtido nas publicações dos modelos descritos anteriormente, mas aplicando essas técnicas para a segmentação semântica de imagens (Long, Shelhamer e Darrell 2014). Genericamente, a ideia base deste modelo é a substituição da última parte da CNN (camadas *fully connected*), que tem como *output* qual a categoria da imagem, por camadas *fully convolutional* de forma a que em vez de se obter uma classificação da imagem se obtenha o mapa de calor da mesma.

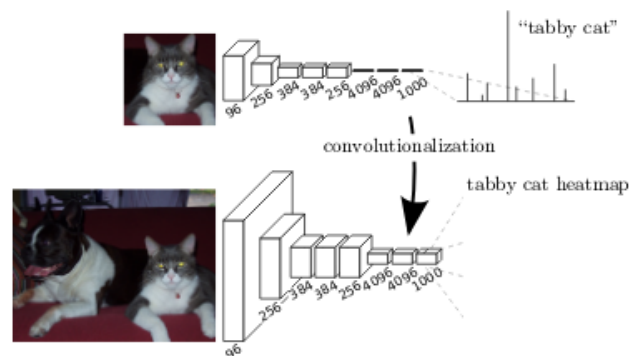


Figura 2.5: Diferença entre uma Fully Convolutional Neuronal Networks (FCNN) e uma CNN (Long, Shelhamer e Darrell 2014).

Este modelo foi avaliado com a métrica IoU média para as várias categorias. Em comparação com os resultados do algoritmo de estado de arte no PASCAL Visual Object Classes (PASCAL VOC) (concurso de classificação de imagem, detecção de objetos e segmentação) de 2012, as FCNN obtiveram melhorias na ordem dos 20% (Long, Shelhamer e Darrell 2014).

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN	47.9	-	-
SDS	52.6	51.6	~ 50 s
FCN-8s	62.7	62.2	~ 175 ms

Tabela 2.4: Comparação de FCNN com algoritmos de estado de arte no PASCAL VOC 2012 (Long, Shelhamer e Darrell 2014).

U-Net

Este modelo foi desenvolvido para segmentação semântica na área da biomédica e tem como base a arquitetura da FCNN (Ronneberger, Fischer e Brox 2015). Nesta arquitetura a rede é composta por duas partes: uma de contração e outra de expansão (lado esquerdo e direito da figura 2.6, respectivamente). A parte de contração tem a arquitetura típica de uma *convolutional network* (pelo que qualquer rede que tenha por base esta arquitetura pode ser usado nesta parte da U-Net), sendo que a diferença desta arquitetura em relação à FCNN consiste na adição da parte de expansão, que aumenta a resolução do *output* da parte de contração através de *upsampling* (Ronneberger, Fischer e Brox 2015). Para permitir a localização dos elementos na imagem, *features* de alta resolução das camadas de contração são combinadas ao *output* das camadas de expansão (setas cinzentas na figura 2.6) (Ronneberger, Fischer e Brox 2015). Esta técnica de passagem de *features* para camadas mais à frente da rede é conhecida como *skip connection* e permite que as camadas mais à frente na rede acessem a detalhes obtidos em camadas anteriores, que de outra forma seriam perdidos. Esta técnica difere da abordagem tradicional, na medida em que cada camada de uma rede neuronal para além de estar conectada a uma ou mais camadas no próximo nível, está também conectada a camadas alguns níveis mais à frente. O seu uso é bastante comum em arquiteturas de modelos para *computer vision*.

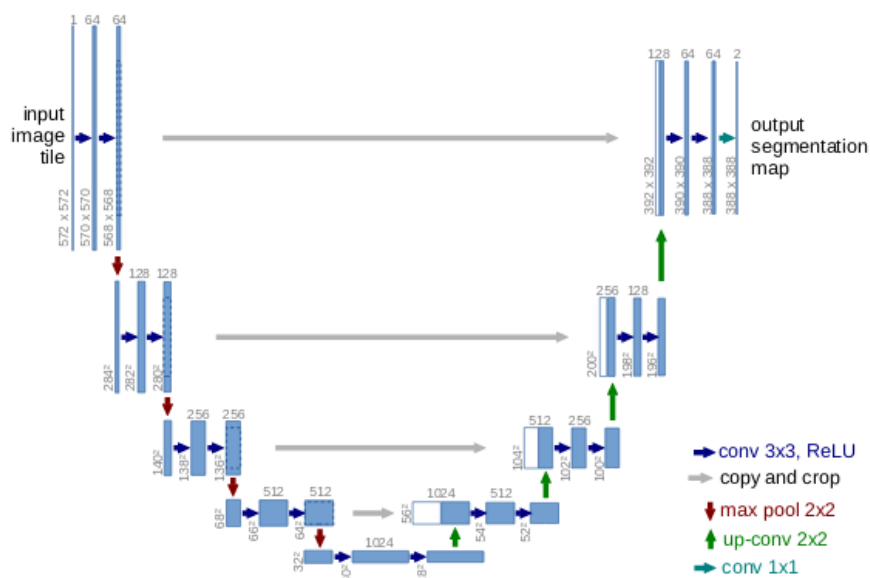


Figura 2.6: Arquitetura da U-Net (Ronneberger, Fischer e Brox 2015).

Este modelo foi avaliado com a métrica IoU média. No International Symposium on Biomedical Imaging (ISBI) cell tracking challenge 2015 foi o melhor entre todos os participantes, como se pode aferir na tabela 2.5.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	0.9203	0.7756

Tabela 2.5: Resultados do ISBI cell tracking challenge 2015.

DeepLab

Este modelo foi desenvolvido para segmentação semântica e encontra-se na 3ª versão, sendo que, a versão original teve por base a Very Deep Convolutional Neuronal Networks (CNN) (Chen et al. 2014). Nesta arquitetura é aplicado o conceito de *atrous spatial pyramid pooling*, que é a combinação de *atrous convolution* e *spatial pyramid pooling*. A *atrous convolution* difere da *convolution* tradicional na medida em que na tradicional a "janela" de varrimento da imagem analisa um dado número de pixels contíguos na imagem, enquanto a *atrous convolution* processa pixels espaçados entre si (ver figura 2.7). Este tipo de *convolution* permite aumentar o contexto processado na "janela" de varrimento da imagem, e oferece o melhor balanceamento entre a localização precisa dos elementos na imagem e a assimilação de contexto (Chen et al. 2016). A *spatial pyramid pooling* é uma técnica que permite uma melhor segmentação de elementos na imagem em várias escalas (ver figura 2.8).

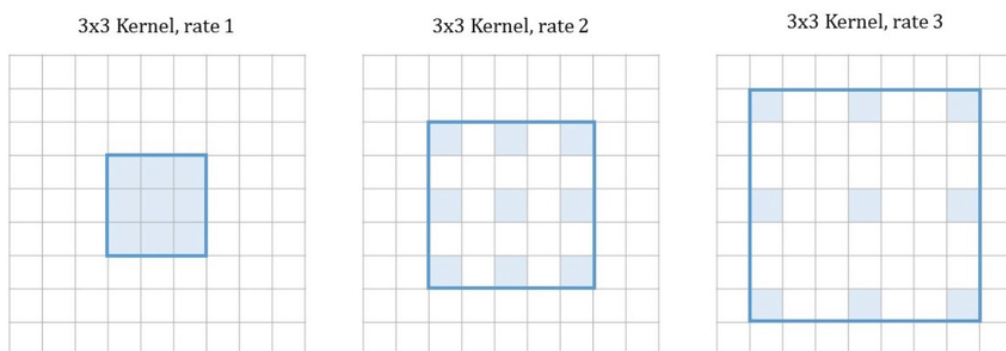


Figura 2.7: Diferença entre *convolution* tradicional (esquerda) e *atrous convolution* (Gsaxner et al. 2019).

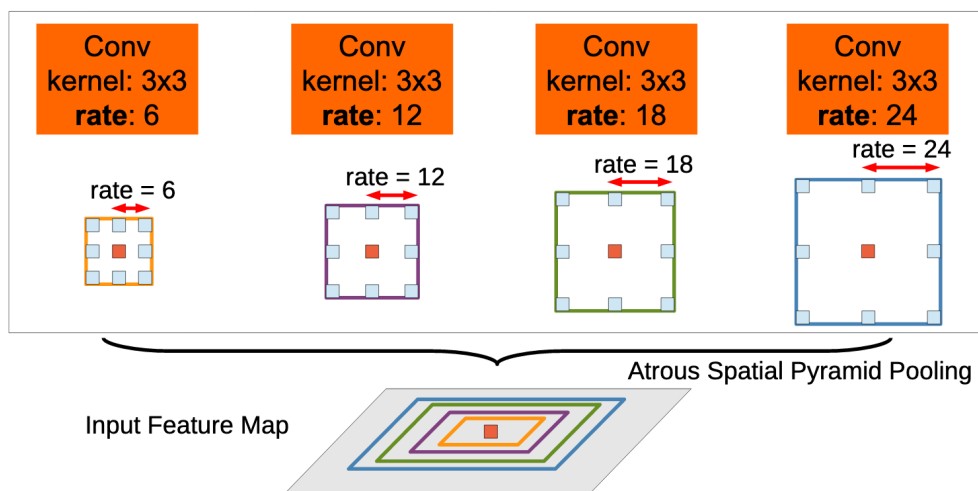


Figura 2.8: *Atrous spatial pyramid pooling* (Chen et al. 2016).

Este modelo foi avaliado com a métrica IoU média, com o *dataset* da edição de 2012 do PASCAL VOC, estando os resultados na tabela 2.6.

Method	mIOU
Adelaide_VeryDeep_FCN_VOC	79.1
LRR_4x_ResNet-CRF	79.3
DeepLabv2-CRF	79.7
CentraleSupelec Deep G-CRF	80.2
HikSeg_COCO	81.4
SegModel	81.8
Deep Layer Cascade (LC)	82.7
TuSimple	83.1
Large_Kernel_Matters	83.6
Multipath-RefineNet	84.2
ResNet-38_MS_COCO	84.9
PSPNet	85.4
IDW-CNN	86.3
CASIA_IVA_SDN	86.6
DIS	86.8
DeepLabv3	85.7
DeepLabv3-JFT	86.9

Tabela 2.6: Resultados do DeepLab para o *dataset* do PASCAL VOC 2012 (Chen, Papandreou, Schroff et al. 2017).

Fully Convolutional Dense Networks

Este modelo foi desenvolvido para segmentação semântica e teve como base um tipo de CNN chamado *DenseNet* (Jégou et al. 2016). Esta arquitetura é caracterizada por uma

aplicação de *skip connections* onde o *output* de cada camada é concatenado ao *output* de todas as camadas seguintes desse bloco (ver figura 2.9) (Huang et al. 2016).

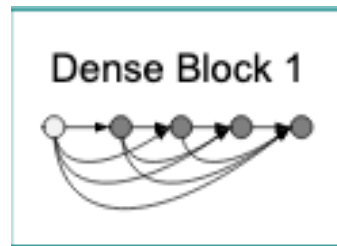


Figura 2.9: Estrutura de um bloco numa *Dense Networks (DenseNet)* (Huang et al. 2016)

A passagem do *output* para camadas seguintes (permitindo a contextualização) e a capacidade de supervisão em multi-escala, tornam a DenseNet bastante adequada para a segmentação semântica (Jégou et al. 2016). A Fully Convolutional Dense Networks (FCDN) foi criada a partir da adição de uma parte de *upsampling* à DenseNet de modo a transformar o retorno original (uma classificação para toda a imagem), num mapa de calor da mesma (Jégou et al. 2016).

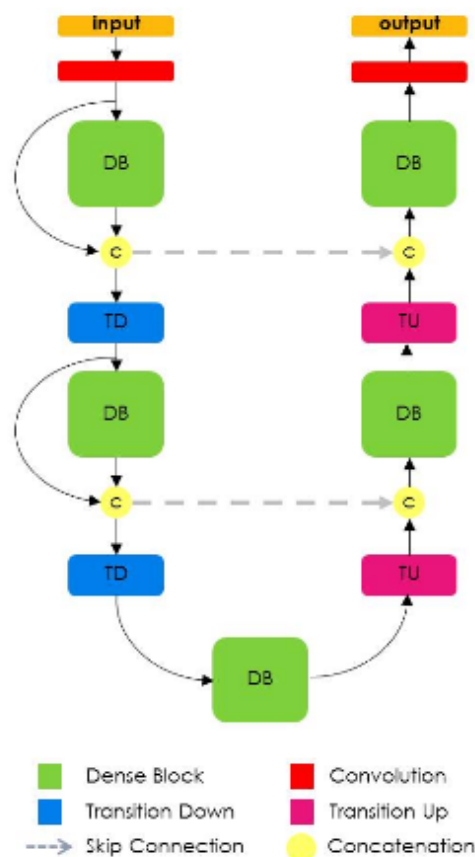


Figura 2.10: Arquitetura da FCDN (Jégou et al. 2016).

Este modelo foi avaliado com as métricas IoU e *accuracy*, utilizando o *dataset* CamVid da *European Conference on Computer Vision* (Jégou et al. 2016).

Model	Pretrained	# parameters (M)	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Cyclist	Mean IoU	Global accuracy
SegNet	✓	29.5	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4	62.5
Bayesian SegNet	✓	29.5						n/a						63.1	86.9
DeconvNet	✓	252						n/a						48.9	85.9
Visin et al.	✓	32.3						n/a						58.8	88.7
FCN8	✓	134.5	77.8	71.0	88.7	76.1	32.7	91.2	41.7	24.4	19.9	72.7	31.0	57.0	88.0
DeepLab-LFOV	✓	37.3	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6	—
Dilation8	✓	140.8	82.6	76.2	89.0	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3	79.0
Dilation8 + FSO	✓	140.8	84.0	77.2	91.3	85.6	49.9	92.5	59.1	37.6	16.9	76.0	57.2	66.1	88.3
Classic Upsampling	✗	20	73.5	72.2	92.4	66.2	26.9	90.0	37.7	22.7	30.8	69.6	25.1	55.2	86.8
FC-DenseNet56 (k=12)	✗	1.5	77.6	72.0	92.4	73.2	31.8	92.8	37.9	26.2	32.6	79.9	31.1	58.9	88.9
FC-DenseNet67 (k=16)	✗	3.5	80.2	75.4	93.0	78.2	40.9	94.7	58.4	30.7	38.4	81.9	52.1	65.8	90.8
FC-DenseNet103 (k=16)	✗	9.4	83.0	77.3	93.0	77.3	43.9	94.5	59.6	37.1	37.8	82.2	50.5	66.9	91.5

Tabela 2.7: Resultados obtidos com o *dataset* CamVid (Jégou et al. 2016).

Relation-Augmented Fully Convolutional Network

Esta arquitetura foi desenvolvida com o objetivo de efetuar segmentação semântica em imagens aéreas. Esta, tem como base a CNN, e apresenta como inovação a introdução de 2 módulos que procuram fazer um relacionamento espacial e de canais nas imagens. Com a aplicação destes módulos é possível relacionar diferentes partes da imagem (através da procura por semelhança de outros elementos em locais diferentes na imagem) e também obter relações entre as diferentes categorias na imagem (é bastante provável que um carro esteja sobre uma estrada, por exemplo) (Mou, Hua e Zhu 2019).



Tabela 2.8: Ilustração das relações entre elementos numa imagem (Mou, Hua e Zhu 2019).

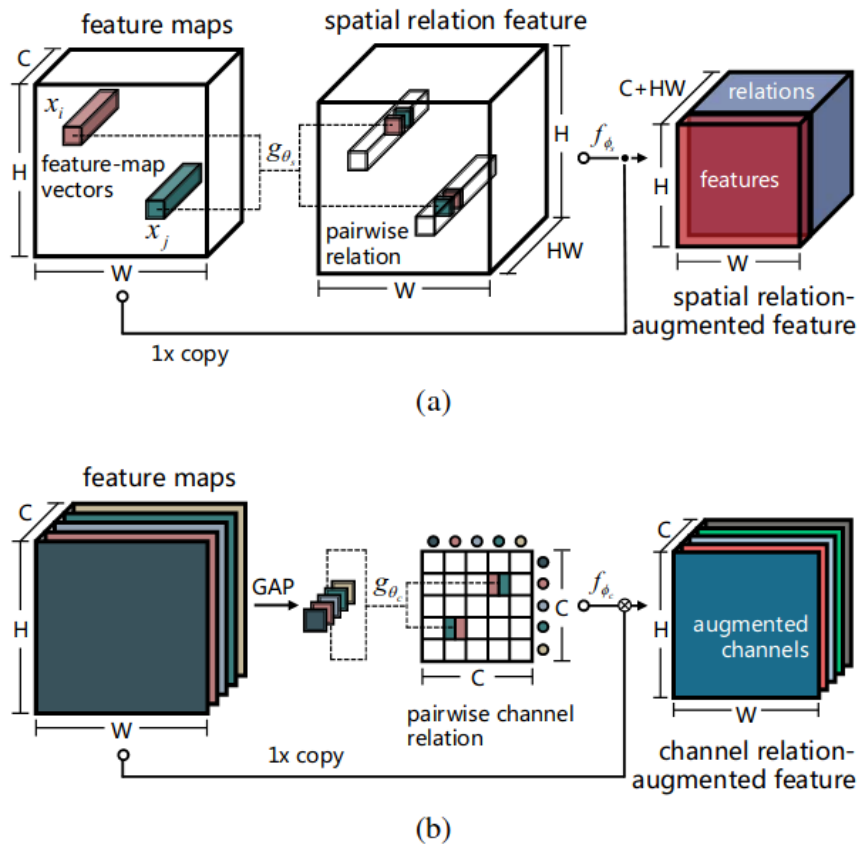


Tabela 2.9: Módulos de relacionamento espacial (a) e relacionamento de canais (b) (Mou, Hua e Zhu 2019).

A utilização destes módulos obteve melhores resultados quando comparado com outros modelos, dado que a sua introdução possibilita uma melhor contextualização espacial dos elementos presentes em imagens aéreas. A métrica utilizada para avaliação foi o IoU, tendo sido efetuados testes em dois *dataset*, das cidades de Vaihingen e Potsdam sendo possível observar os resultados nas tabela 2.10 e 2.11, respetivamente.

Model Name	Imp. surf.	Build.	Low veg.	Tree	Car	mean F_1	mIoU	OA
SVL-boosting+CRF*	86.10	90.90	77.60	84.90	59.90	79.90	-	84.70
RF+dCRF*	86.90	92.00	78.3	86.90	29.00	74.60	-	85.90
CNN-FPL*	-	-	-	-	-	83.58	-	87.83
FCN	88.67	92.83	76.32	86.67	74.21	83.74	72.69	86.51
FCN-dCRF	88.80	92.99	76.58	86.78	71.75	83.38	72.28	86.65
SCNN	88.21	91.80	77.17	87.23	78.60	84.40	73.73	86.43
Dilated FCN	90.19	94.49	77.69	87.24	76.77	85.28	-	87.70
FCN-FR*	91.69	95.24	79.44	88.12	78.42	86.58	-	88.92
PSPNet (VGG16)	89.92	94.36	78.19	87.12	72.97	84.51	73.97	87.62
RotEqNet*	89.50	94.80	77.50	86.50	72.60	84.18	-	87.50
RA-FCN-srm	91.01	94.86	80.01	88.74	87.16	88.36	79.48	89.03
P-RA-FCN	91.46	95.02	80.40	88.56	87.08	88.50	79.72	89.18
S-RA-FCN	91.47	94.97	80.63	88.57	87.05	88.54	79.76	89.23

Tabela 2.10: Resultados do Relation-Augmented Fully Convolutional Network (RA-FCN) para o *dataset* de Vaihingen (Mou, Hua e Zhu 2019).

Model Name	Imp. surf.	Build.	Low veg.	Tree	Car	Clutter	mean F_1	mIoU	OA
FCN	88.61	93.29	83.29	79.83	93.02	69.77	84.63	78.34	85.59
FCN-dCRF	88.62	93.29	83.29	79.83	93.03	69.79	84.64	78.35	85.60
SCNN	88.37	92.32	83.68	80.94	91.17	68.86	84.22	77.72	85.57
Dilated FCN*	86.52	90.78	83.01	78.41	90.42	68.67	82.94	-	84.14
FCN-FR*	89.31	94.37	84.83	81.10	93.56	76.54	86.62	-	87.02
RA-FCN-srm	90.48	93.74	85.67	83.10	94.34	74.02	86.89	81.23	87.61
P-RA-FCN	90.92	94.20	86.64	83.00	94.44	77.88	87.85	81.85	88.30
S-RA-FCN	91.33	94.70	86.81	83.47	94.52	77.27	88.01	82.38	88.59

Tabela 2.11: Resultados do RA-FCN para o *dataset* de Postdam (Mou, Hua e Zhu 2019).

2.4.4 Abordagens existentes

Nesta secção serão referidos alguns casos de estudo na mesma área do presente trabalho.

Learning Aerial Image Segmentation from Online Maps

Estudo realizado por Kaiser et al. (2017), sobre segmentação semântica em imagem aérea de alta resolução. O objetivo deste estudo consistiu na avaliação do desempenho de FCNN com treino a partir de um *dataset* com grande volume de dados, mas que pode não estar corretamente categorizado na sua totalidade, quando comparado com os resultados obtidos a partir de dados totalmente corretos, mas em menor volume. Com esta comparação pretende-se aferir a possibilidade da utilização de dados com menor correção, mas de mais facilmente obtenção, sem prejuízo dos resultados, ao invés de se construir todo o *dataset* manualmente para se obter bons resultados.

Este modelo foi desenvolvido para identificar dois elementos em fotografia aérea, edifícios e estradas, sendo tudo o resto classificado como fundo. Para o estudo foi utilizada uma variante do modelo FCNN e para o treino foram utilizados quatro *dataset* com grande volume de dados a partir de fontes públicas (Google Maps para imagens e OSM para a categorização dos elementos da fotografia, sendo que os dados de categorização podem ser editados por qualquer pessoa, podendo conter erros), para as cidades de Chicago, Paris, Zurique e Berlin. Para além destes *dataset* foram utilizados dois com um menor volume de dados para a cidade de Postdam, sendo um obtido através da mesma fonte dos quatro anteriormente referidos e outro a partir do International Society for Photogrammetry and Remote Sensing (ISPRS) (sendo que estes dados estão totalmente corretos) (Kaiser et al. 2017).

Fully Residual Convolutional Neural Networks for Aerial Image Segmentation

Artigo de Sang e Minh (2018), também sobre segmentação semântica em imagem aérea, no qual foi experimentada uma abordagem a este problema com recurso a uma variante do modelo FCNN. Para treino da rede foi utilizado um *dataset* da ISPRS para a cidade de Vaihingen no qual estão categorizados os seguintes tipos de elementos: pavimentos, edifícios, vegetação baixa, árvores, carros e fundo.

DeepGlobe Land Cover Classification Challenge

Desafio integrado na conferência Computer Vision and Pattern Recognition (CVPR), na qual os participantes têm de desenvolver um modelo capaz de classificar elementos presentes em imagens de satélite com recurso a segmentação semântica. Os organizadores disponibilizaram um *dataset* de alta resolução aos participantes, nas quais é possível identificar áreas com os seguintes tipos: urbana, agricultura, pastagem, floresta, água, árida e desconhecida (todas as que não se incluem em nenhuma das anteriores) (*DEEPGLOBE - CVPR18 2019*).

2.4.5 Tecnologias

Em seguida serão apresentadas as tecnologias que foram consideradas para o desenvolvimento do modelo de segmentação semântica.

TensorFlow

O TensorFlow é uma biblioteca *open-source* desenvolvida pela Google para computação numérica de alto desempenho, com forte suporte para *machine learning* e *deep learning*.

Esta biblioteca disponibiliza APIs em Python, JavaScript, C++, Java, Go e Swift, sendo que a de Python é a mais completa e simples de utilizar (*TensorFlow 2019*).

PyTorch

Biblioteca *open-source* inicialmente desenvolvida pelo Facebook para *deep learning* utilizando GPU e CPU, com recurso a tensores otimizados. Este *software* fornece uma API em Python (*PyTorch 2019*).

Keras

O Keras é uma API de alto nível para o desenvolvimento de redes neuronais. Desenvolvida em Python e com capacidade para internamente usar TensorFlow, CNTK e Theano. Desenvolvida por François Chollet, investigador de *deep learning* da Google (*Keras Documentation 2019*).

O seu principal objetivo é abstrair alguma da complexidade da implementação com as bibliotecas de mais baixo nível.

FastAI

Biblioteca originalmente desenvolvida por Jeremy Howard, investigador científico da Universidade de São Francisco e ex-presidente da plataforma de *data science* Kaggle (*fast.ai 2019*).

É uma biblioteca de alto nível assente no PyTorch, com o objetivo de simplificar a implementação de redes neuronais rapidamente, sem prejuízo dos resultados obtidos, utilizando as melhores práticas disponíveis.

Land Cover Mapping API

API web da Microsoft para classificação de elementos presentes em imagem aérea. Os elementos são classificados segundo as seguintes categorias: água, árvores, herbáceo, árido e sem dados (para tudo o resto). Esta API só funciona com imagens com a extensão tiff, quadradas, no formato RGBI, com tamanho inferior a 4MB e à escala 1:20000 (*AI for Earth* 2019).

Capítulo 3

Avaliação das soluções/tecnologias existentes

Neste capítulo serão avaliadas as soluções apresentadas na secção 2.4.4 e as tecnologias referidas na secção 2.4.5.

3.1 Avaliação das soluções existentes

Em seguida será feita uma avaliação de algumas conclusões feitas nos casos apresentados na secção 2.4.4 que possam ser pertinentes para este trabalho.

3.1.1 Learning Aerial Image Segmentation from Online Maps

No estudo de Kaiser et al. (2017) foi utilizada uma variante de FCNN, tendo sido adicionada mais uma *skip connection* na parte da rede responsável por transformar o *output* uni-dimensional no mapa de calor, já que para o autor, na implementação original as margens dos objetos não estavam suficientemente bem delineadas (Kaiser et al. 2017).

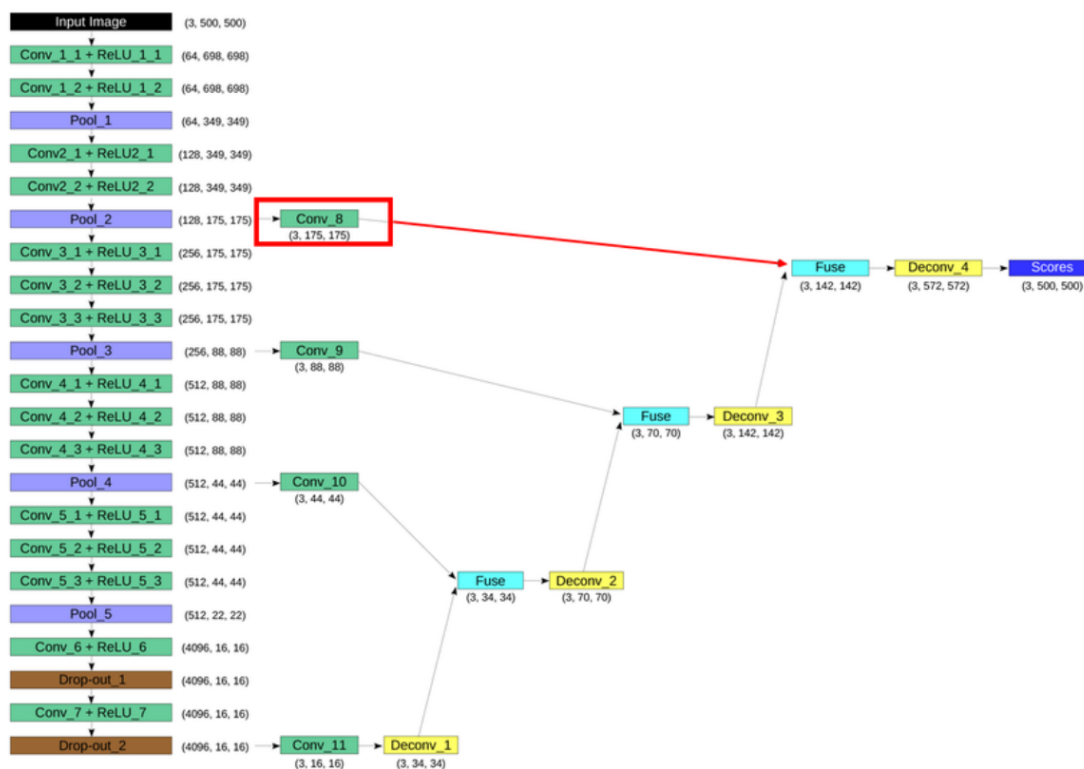


Figura 3.1: Arquitetura da rede utilizada em Kaiser et al. (2017).

Com o algoritmo desenvolvido neste estudo pretendia-se validar se o treino com grandes volumes de dados, mesmo com alguma taxa de erros, resultava em melhores resultados do que com treino com menos dados, mas sem nenhum erro.

Kaiser et al. (2017) realizaram as seguintes experiências:

Ia *Baseline* com dados do ISPRS

Treinou-se a rede com dados de alta precisão do ISPRS, para uma área considerada realística para este tipo de problemas (27ha).

Ib Utilização de rede pré-treinada

Para esta experiência usou-se uma rede pré-treinada para o mesmo *dataset* do Ia.

II *Gold standard* com dados do ISPRS

Neste caso foram novamente utilizados os dados de alta precisão, mas desta vez todo o *dataset* disponível foi utilizado (189 ha). Este é o caso ótimo, no qual existem dados de alta precisão em elevado volume para treino da rede (mais de 200 milhões de pixels categorizados manualmente), sendo que um *dataset* deste tipo implica elevados custos, portanto serão raros os casos práticos onde existirão *datasets* deste gênero.

IIIa *Baseline* com imagens do Google Maps e dados do OSM

Treinou-se a rede com dados de menor precisão da Google e do OSM, para uma área bastante alargada (189ha, a mesma usada no II)

IIIb Aumento do volume de dados de treino

Para testar se o aumento de dados com erro melhora a precisão, adicionou-se a área de outra cidade (Berlim), aos dados utilizados no IIIa.

IV Substituição completa da categorização manual

Neste caso foram utilizadas as imagens de alta qualidade do ISPRS, sendo substituída a classificação manual pelos dados obtidos no OSM.

V *Augmentation* com dados públicos

Para esta experiência foi feito um pré-treino com os dados do Google Maps e do OSM, sendo depois a rede treinada com o mesmo *dataset* utilizado no II.

VI Substituição parcial da categorização manual

Utilizou-se a mesma abordagem do ponto V, mas em vez do uso do *dataset* de alta precisão completo, usou-se apenas uma parte (mesmo *dataset* do I).

Na tabela 3.1 é possível observar os resultados destas experiências (a avaliação foi feita usando as métricas F_1 , *precision* e *recall*).

		av. F_1	av. Precision	av. Recall	F_1 Building	F_1 Road	F_1 Background	train time [h]
Ia	ISPRS baseline	0.764	0.835	0.704	0.793	0.499	0.883	16
Ib	ISPRS baseline pre-trained	0.809	0.853	0.770	0.830	0.636	0.904	16
II	ISPRS gold standard	<u>0.874</u>	0.910	<u>0.841</u>	0.913	<u>0.764</u>	0.923	16
IIIa	Google/OSM baseline P	0.777	0.799	0.756	0.832	0.631	0.845	16
IIIb	Google/OSM baseline P+B	0.797	0.819	0.776	0.828	0.698	0.858	32
IV	Complete substitution	0.779	0.801	0.758	0.796	0.667	0.860	16
V	Augmentation	0.884	<u>0.898</u>	0.870	<u>0.900</u>	0.825	<u>0.922</u>	78
VI	Partial substitution	<u>0.837</u>	<u>0.860</u>	<u>0.816</u>	<u>0.863</u>	<u>0.736</u>	0.899	78

Tabela 3.1: Resultados obtidos no *Learning Aerial Image Segmentation from Online Maps* (Kaiser et al. 2017).

A partir das experiências realizadas, pode-se afirmar que os melhores resultados são obtidos com um grande volume de dados de alta precisão (II), tendo sido atingida uma precisão média de 91%. Na maioria dos casos não é expectável que esteja disponível um *dataset* desta qualidade.

Com um *dataset* público mais alargado (IIIb) foi possível obter uma precisão média bastante próxima da obtida com um *dataset* de alta precisão de menores dimensões (Ia) (81.9% contra 83.5%, respetivamente). Esta diferença reduzida deve-se à boa capacidade de generalização deste tipo de algoritmos.

De notar ainda que os melhores resultados obtidos sem recurso ao *dataset* de grande volume de dados de alta precisão foi aquele onde foi feito um pré-treino com dados públicos (IV) (86%), seguido de um treino com um *dataset* mais pequeno de dados de alta precisão para afinação da rede. Esta é a abordagem recomendada pelos autores do estudo.

Esta conclusão é bastante pertinente para o contexto do trabalho a desenvolver, validando o uso dos dados no seu estado atual, apesar dos problemas encontrados nos mesmos (ver 2.2.1). Desta forma sabe-se que os resultados obtidos para os dados atuais deverão ser satisfatórios, existindo a possibilidade de futuramente se tentar melhorar os resultados através da correção manual de uma porção.

3.1.2 Fully Residual Convolutional Neural Networks for Aerial Image Segmentation

No artigo de Sang e Minh (2018) é experimentado o uso de uma variante de FCNN para segmentação semântica de imagem aérea. É feito o teste de duas tipologias de rede: FCN-ResNet101-8s, com dois blocos residuais (um bloco residual é um tipo de *skip connection*) e FCN-ResNet101-4s com três blocos residuais, estando a sua arquitetura representada na figura 3.2.

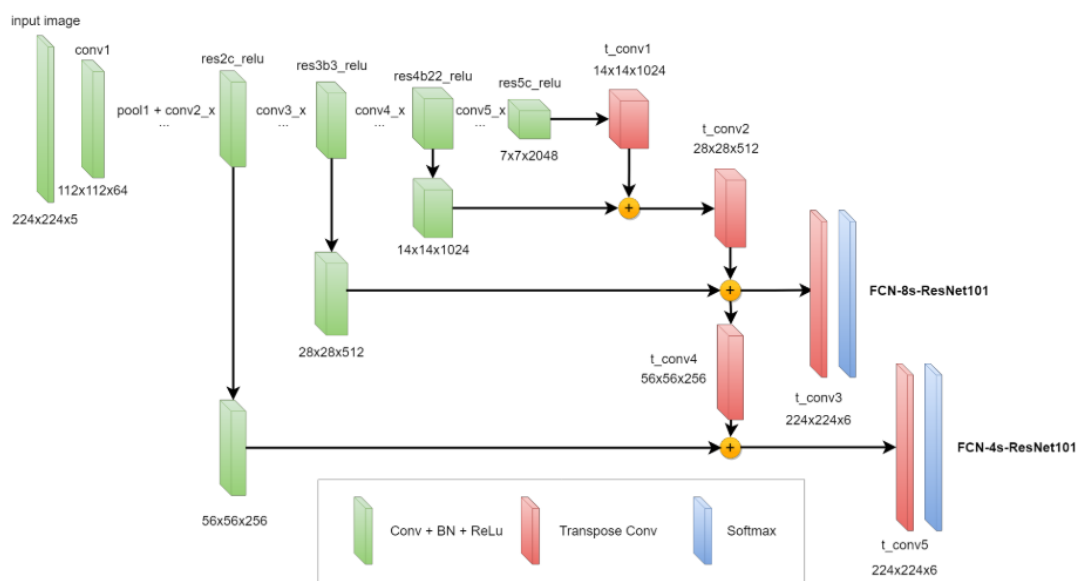


Figura 3.2: Tipos de redes utilizados no *Fully Residual Convolutional Neural Networks for Aerial Image Segmentation* (Sang e Minh 2018).

Para treino da rede foi utilizado o *dataset* do concurso de segmentação semântica de Vaihingen do ISPRS, tendo sido obtidos os resultados disponíveis na tabela 3.2 (a métrica utilizada foi a *accuracy*).

Method	Imp_surf	Building	Low_veg	Tree	Car	Overall
Stair Vision Library (SVL3)	86.6	91.0	77.0	85.0	55.6	84.8
Rule-based	84.3	88.7	74.5	82.0	9.9	81.8
RF + Full CRF (HUST)	86.9	92.0	78.3	86.9	29.0	85.9
CNN ensemble (ONE_5)	90.7	92.2	80.6	90.0	82.7	88.0
FCN (UZ_1)	89.2	92.5	81.6	86.9	57.3	87.3
FCN (UOA)	89.8	92.1	80.4	88.2	82.0	87.6
FCN trained with no-downsampling (DST_1)	90.19	94.49	77.69	87.24	76.77	87.70
FCN trained with no-DS + RF + CRF (DST_2)	90.41	94.73	78.25	87.25	75.57	87.90
FCN_MFS_DSMBBackend	92.3	95.8	83.8	89.6	86.4	90.6
CASIA2 (rank 3)	93.2	96.0	84.7	89.9	86.7	91.1
NLPR3 (rank 2)	93.0	95.6	85.6	90.3	84.5	91.2
HUSTW5 (rank 1)	93.3	96.1	86.4	90.8	74.6	91.6
Our FCN-ResNet101-8s (BKHN10)	92.9	96.0	84.6	89.8	88.8	91.0
Our FCN-ResNet101-4s (BKHN11)	92.9	96.0	84.6	89.9	88.6	91.0

Tabela 3.2: Resultados obtidos no *Fully Residual Convolutional Neural Networks for Aerial Image Segmentation* (Sang e Minh 2018).

A precisão global destas redes (91%) colocam-nas no 4º lugar entre 140 participantes (Sang e Minh 2018). Como trabalho futuro os autores do artigo identificam a utilização de DenseNet e U-Net para tentarem obter melhores resultados.

Com este artigo, mais o estudo anteriormente apresentado, pode-se concluir que as FCNN têm bons resultados na área da segmentação semântica, sendo ainda referidas as FCDN, como potencialmente mais precisas para este tipo de tarefas.

3.1.3 DeepGlobe Land Cover Classification Challenge

Os participantes deste desafio tinham de desenvolver um modelo de segmentação semântica com recurso a imagens de satélite, sendo-lhes fornecido um *dataset* de alta resolução para treino e avaliação dos modelos desenvolvidos. Era também requisito de participação, a submissão de um pequeno artigo com os detalhes da implementação (*DEEPGLOBE - CVPR18 2019*). Analisando o artigo de Tian, Li e Shi (2018), o vencedor deste concurso, é possível verificar que o modelo implementado é uma variante da FCDN (ver figura 3.3). Para os autores este modelo prova o potencial deste tipo de redes, dado que para obter os resultados alcançados não foi necessário o recurso a outros *datasets*, nem a modelos pré-treinados.

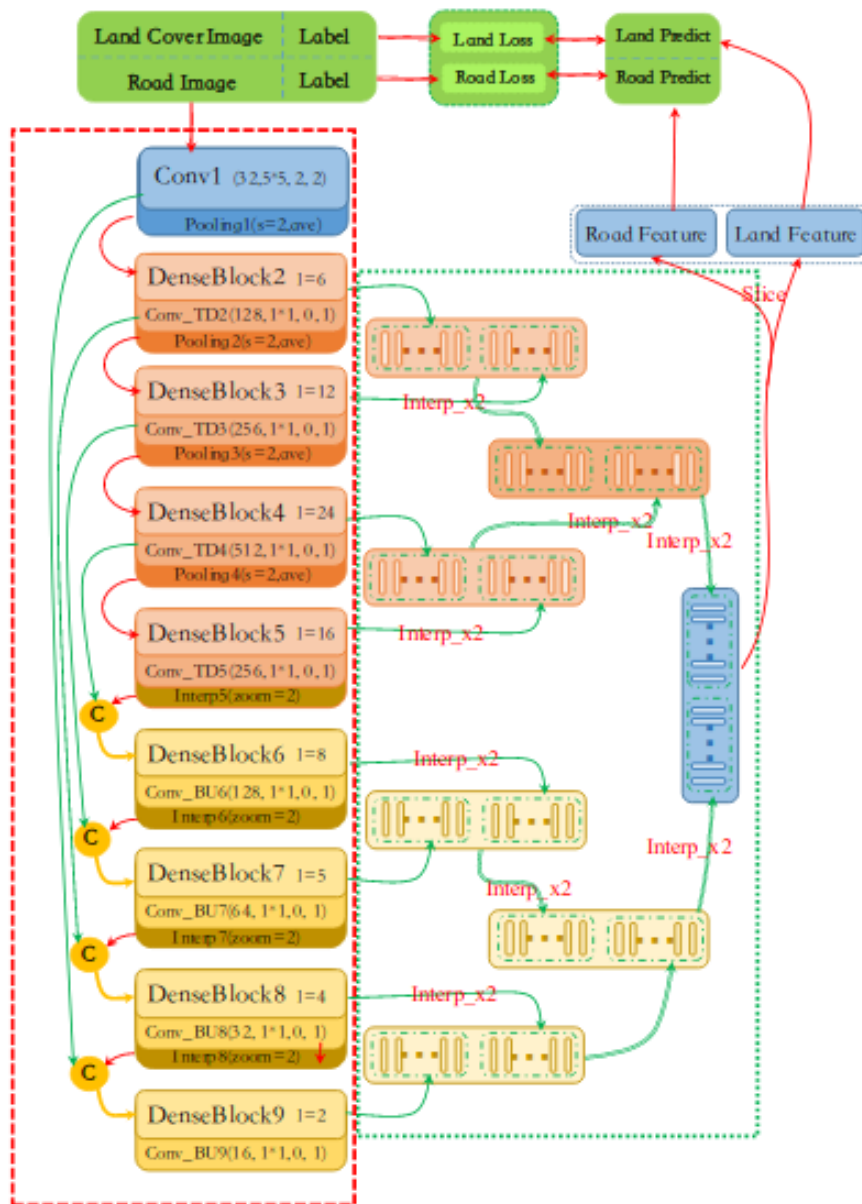


Figura 3.3: Arquitetura do modelo vencedor do DeepGlobe Land Cover Classification Challenge (Tian, Li e Shi 2018).

Para avaliação dos modelos foi utilizada a métrica IoU médio, tendo o mesmo obtido um resultado de 52.24%.

3.2 Avaliação das tecnologias existentes

Nesta secção, será feita a análise para escolha da tecnologia a utilizar no desenvolvimento do modelo de segmentação semântica, o principal artefacto deste trabalho.

Como o principal objetivo do trabalho se prende com a adaptação de modelos já existentes para o problema a resolver, pretende-se escolher uma tecnologia que permita rapidez e facilidade no treino dos modelos e na sua avaliação. É por isso dada uma menor relevância a outros fatores que teriam de ser levados em conta, caso o objetivo passasse pelo desenvolvimento de um modelo assente numa arquitetura criada para este propósito.

A Land Cover Mapping API não foi considerada aplicável no contexto deste trabalho, dadas as elevadas limitações em relação às imagens que consegue classificar (ver 2.4.5) bem como às categorias que conseguem ser identificadas serem bastante reduzidas, não existindo a possibilidade de serem adicionadas novas categorias.

Das restantes tecnologias pode ser feita uma divisão entre TensorFlow e PyTorch, já que o Keras e o FastAI são bibliotecas de mais alto nível, que são desenvolvidas utilizando estas duas como base.

Segundo um artigo da O'Reilly no qual são partilhados dados obtidos pela RISELab da Universidade de Berkeley sobre a menção ao TensorFlow e ao PyTorch em artigos publicados na plataforma arXiv.org, o interesse no PyTorch tem crescido rapidamente (O'Reilly 2019). Como se pode observar na figura 3.4, o número de artigos que referem o PyTorch atingiu o nível do TensorFlow, tendo o crescimento de janeiro a junho de 2019 em relação ao mesmo período de 2018, sido de 194% no caso do PyTorch e 23% no TensorFlow. Estes números demonstram um grande crescimento no interesse nesta tecnologia pela área científica. Como o objetivo deste trabalho passa pelo teste de modelos, que na sua maioria estão detalhados em artigos científicos, optou-se pelo uso do PyTorch.

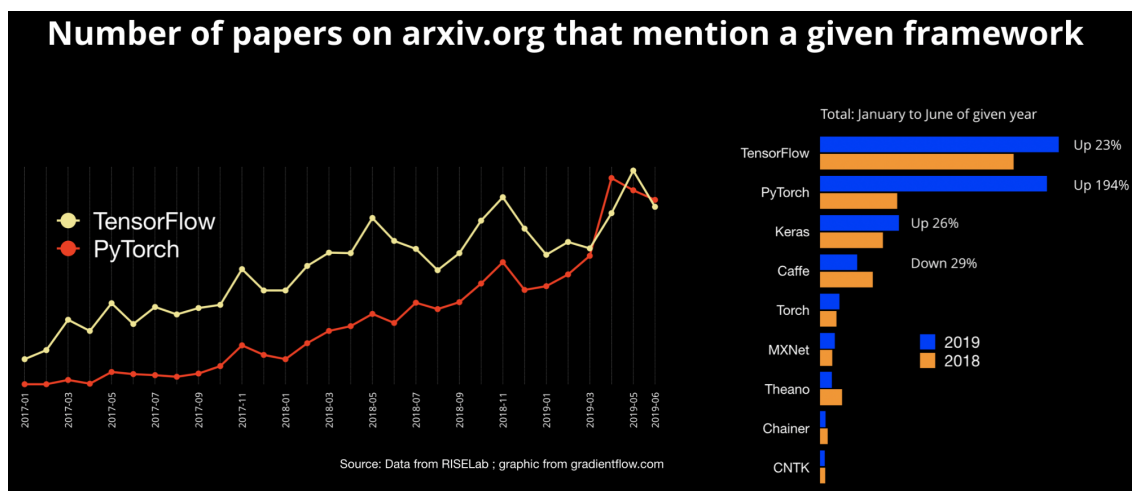


Figura 3.4: Número de referências ao PyTorch e TensorFlow em artigos publicados no arXiv.org (O'Reilly 2019).

A utilização de uma biblioteca de alto nível pode ser útil para acelerar o desenvolvimento em certos casos de uso, e no caso do FastAI concluiu-se que a sua utilização no projeto seria bastante vantajosa. Esta biblioteca traz implementadas várias técnicas de *deep learning* e utilitários que teriam de ser desenvolvidos de raiz, pelo que o seu uso traz melhorias na velocidade de desenvolvimento. As camadas de abstração desta biblioteca permitem que num caso extremo a sua utilização seja exatamente igual à do PyTorch.

Em termos de licenciamento, apenas a Land Cover Mapping API acarreta custos acrescidos, enquanto que todos os outros têm um licenciamento que permite o seu uso sem custos.

No suporte, todos os *software* em causa são bastante similares, visto existir documentação completa e todos têm comunidades bastante ativas.

Capítulo 4

Design da solução

Neste quarto capítulo, é apresentado o design para a solução a implementar, quais os requisitos que foram tidos em conta e as decisões tomadas para chegar à solução encontrada.

4.1 Arquitetura da solução

Após análise do processo de negócio e do problema em questão, efetuou-se um levantamento dos artefactos envolvidos no processo de criação de cartografia e quais os requisitos da solução a desenvolver.

4.1.1 Modelo de domínio

Apesar da complexidade e morosidade deste processo são poucos os tipos de recursos utilizados no processo. Isto dá origem a um modelo de domínio bastante simples, como é possível ver na figura 4.1.

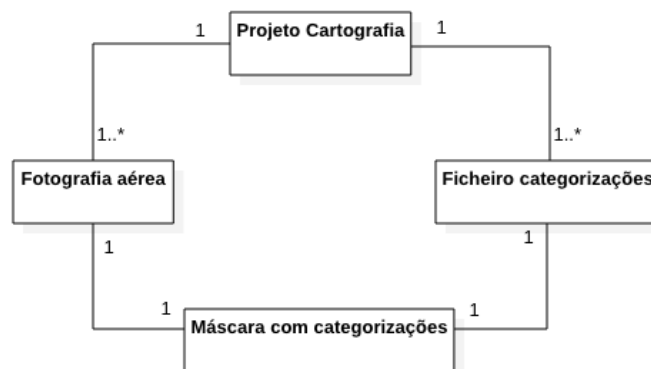


Figura 4.1: Modelo de domínio

- **Projeto de cartografia:** Cada projeto é composto por múltiplas fotografias aéreas e um ficheiro geográfico por cada categoria que tenha sido requisitada para o projeto.
- **Fotografia aérea:** Fotografia de alta resolução da aérea a cartografar com informação geográfica da mesma na meta informação.

- **Ficheiro categorizações:** Um ficheiro por cada categoria, com todos os elementos de todas as fotografias aéreas do projeto.
- **Máscara com categorizações:** Máscara com elementos categorizados presentes numa fotografia aérea. É obtida com recurso aos ficheiros geográficos para a área geográfica disponível na meta-informação da fotografia. Esta máscara é utilizada para treino do modelo de segmentação semântica e *output* do modelo será neste formato.

4.1.2 Casos de uso

Os requisitos para a solução a desenvolver encontram-se definidos no diagrama de casos de uso, disponível na figura 4.2.

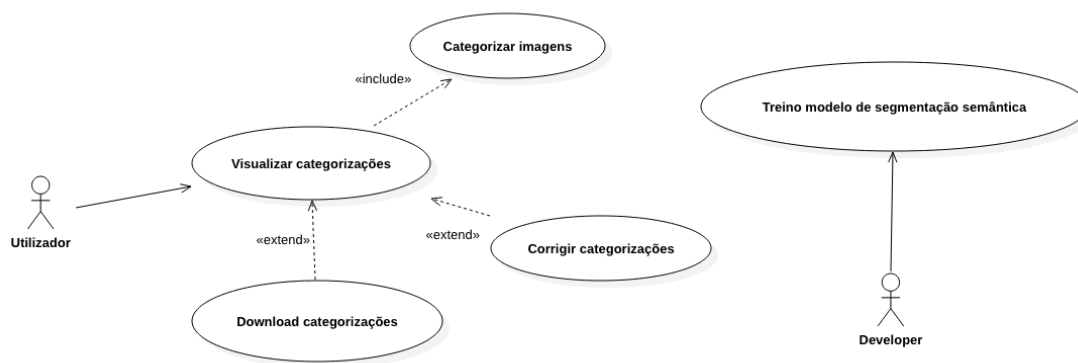


Figura 4.2: Diagrama de casos de uso

1. Treino do modelo de segmentação semântica

Para categorização das fotografias aéreas é necessário treinar um modelo de segmentação semântica, que seja capaz de identificar as diferentes categorias nas imagens. Este treino será feito múltiplas vezes ao longo do tempo (devido ao aumento do volume de dados e de utilização de zonas com morfologia diferente do utilizado até então) para aumento da precisão dos resultados.

2. Visualizar categorizações

Este caso de uso compreende a seleção das imagens a classificar e posterior visualização do resultado das categorizações. Deve ser possível controlar quais as categorias visíveis no mapa para permitir uma análise dos dados. Também deve ser permitida a visualização em simultâneo de categorizações do mesmo local, em momentos diferentes, para verificação de diferenças entre elas.

3. Categorizar imagens

As imagens selecionadas pelo utilizador no caso de uso 1, devem ser categorizadas pelo modelo de segmentação semântica aprendido.

4. Corrigir categorizações

Deve ser possível visualizar os dados em formato vetorial para permitir eventuais correções.

5. Download categorizações

Neste caso de uso, o utilizador pode fazer o *download* das categorizações em formato vetorial.

O caso de uso 2, será implementado com recurso a um algoritmo de classificação automática baseado em *deep learning*, enquanto que o 1, 3 e 4 serão disponibilizados através de uma aplicação *web*.

4.1.3 Algoritmo de classificação automática

Para a classificação automática das imagens serão testadas múltiplas arquiteturas de modelos de segmentação semântica. Optou-se pelo uso de arquiteturas já existentes dada a elevada complexidade de criação destas de raiz. O modelo que tiver um melhor desempenho nas métricas avaliadas, será o escolhido para a classificação de imagens.

Para este tipo de problemas, normalmente existem dois tipos de abordagem: a utilização de uma rede para todas as categorias, ou uma rede por categoria. No primeiro caso o *output* da rede será as N categorias existentes, mais a categoria fundo para todo o resto, enquanto que no segundo o *output* será uma máscara binária composta pela categoria daquela rede e fundo.

No âmbito deste projeto, optou-se pela abordagem de uma rede para todas as categorias, visto ser esperado que esta rede tenha um contexto maior da imagem, e que isso ajude a aprendizagem e o processo de decisão do algoritmo (se o algoritmo conseguir identificar estradas e edifícios, pode fazer exclusão de hipóteses, ou seja, um edifício pode ser qualquer coisa excepto uma estrada, por exemplo).

Foi também definido que deveriam ser desenvolvidos dois modelos: um para as categorias de topo (ADVR, AGRI, CONC, CONC e estradas) e outro para classificação de algumas subcategorias.

O *output* do algoritmo é uma máscara com a mesma dimensão da imagem original onde cada categoria é identificada por uma cor diferente.

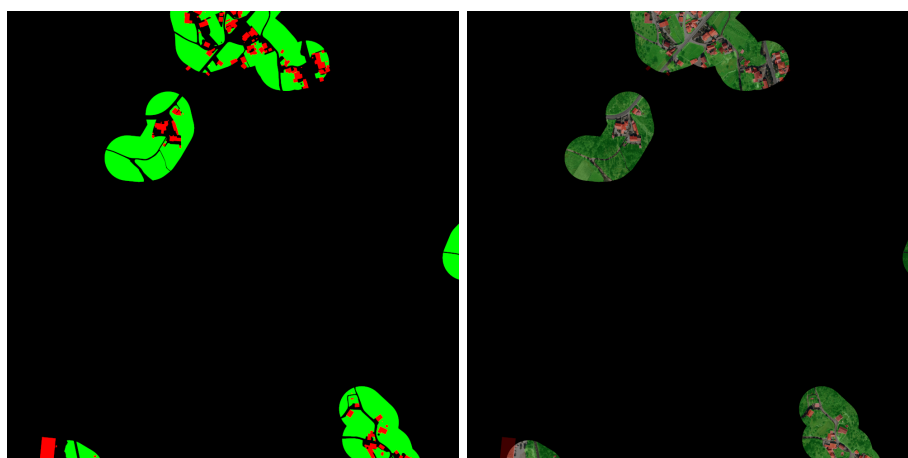


Figura 4.3: Exemplo de máscara com as categorias AGRI (verde) e CONC (vermelho) (à esquerda) e sobreposição da mesma sobre a imagem original (à direita).

Pipeline de dados

Para treinar o algoritmo, é preciso que os dados de treino sejam iguais ao que é esperado obter no algoritmo, sendo então necessário fazer algum pré-processamento dos dados antes de serem utilizados para o treino. De modo a automatizar este processo, que será executado múltiplas vezes, definiu-se uma *pipeline* para a transformação de dados. Na metodologia CRISP-DM, esta *pipeline* enquadra-se na fase de *data preparation*.

A *pipeline* é responsável pelo redimensionamento das imagens, uma vez que as imagens originais são de grandes dimensões (superiores a 10000x10000 pixels), algo que não é confortável pelo algoritmo, tendo-se optado pela divisão das mesmas em imagens mais pequenas (1024x1024 pixels) e pela geração da máscara com as categorias representadas, a partir dos ficheiros geográficos. Cada ficheiro geográfico contém os dados para toda a área do projeto, sendo necessário obter a zona correspondente a cada uma das fotografias, através do uso da sua meta-informação.

Melhoria contínua

É expectável que no futuro existam projetos de cartografia com morfologia diferente das utilizadas em treinos anteriores e, nesse caso, é possível que a qualidade dos resultados do modelo baixe consideravelmente. Para resolver este problema, será feita a categorização manual de uma pequena parte do projeto para novo treino da rede, de modo a aumentar a sua precisão (como demonstrado no estudo *Learning Aerial Image Segmentation from Online Maps*, referido na secção 3.1.1).

Espera-se também que ao longo do tempo o volume de dados da empresa continue a aumentar, e dado que, normalmente neste tipo de algoritmos o aumento do volume e heterogeneidade dos dados significa melhores resultados, torna-se também necessário repetir o processo de treino com os novos dados.

O processo de treino, por norma, será feito na fase inicial e/ou final do desenvolvimento de um projeto de cartografia. No início do projeto, caso os resultados sejam de pior qualidade do que o esperado, será utilizada a abordagem do treino com um pequena porção dos dados classificados manualmente. Após conclusão do projeto, as categorizações corrigidas pelos operadores devem ser usadas para novo treino de forma a tentar melhorar o modelo nas partes em que houve falhas.

Torna-se então essencial que, a solução contemple a evolução iterativa do modelo de classificação, pelo que se pretende que o componente onde o modelo esteja inserido não tenha dependências que dificultem a sua substituição e que o processo de treino seja delineado de forma a que a sua execução seja o mais simples possível, de modo a que o treino da rede possa ser feito sem ser necessário um profundo conhecimento do seu funcionamento interno.

Tal como definido no *CRISP-DM*, este trabalho implica um processo de desenvolvimento cíclico pelo que esta definição de processos torna-se importante para o seu ciclo de vida a longo prazo.

Conversão do output para ficheiros geográficos

O *output* do modelo é uma máscara com a categorização dos elementos nela presente, mas neste formato tem pouco valor para a área da cartografia, já que todo o trabalho de categorização é feito com recurso a ficheiros geográficos vectoriais e apenas com recurso às imagens seria necessário decalcar manualmente o conteúdo das mesmas para o formato apropriado. Para evitar este trabalho, o processo de conversão do *output* para formato vectorial deverá ser automatizado.

4.1.4 Aplicação Web

Para os utilizadores finais desta solução, a interacção direta com o algoritmo de classificação automática não é a mais adequada, já que é um *software* de mais baixo nível que exigiria conhecimentos técnicos avançados. Para contornar esta dificuldade, foi considerado o desenvolvimento de uma aplicação *desktop*, ou uma aplicação *web*. A escolha recaiu sobre a aplicação *web*, visto permitir maior compatibilidade com múltiplos tipos de dispositivo e envolver um menor tempo de desenvolvimento.

4.1.5 Organização dos componentes

Com base nos requisitos e decisões referidos anteriormente, foi identificada a necessidade de implementação de uma arquitetura servidor-cliente para a aplicação *web* e de um módulo que permita o treino de uma forma simples.

Na figura 4.4 é possível observar os componentes desenvolvidos e as suas dependências.

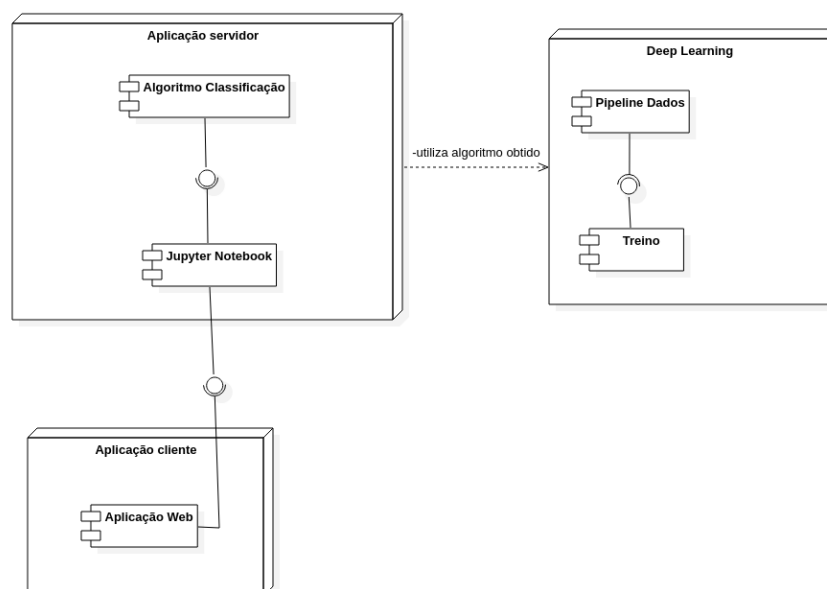


Figura 4.4: Diagrama de componentes

- **Pipeline de dados:** Componente responsável pelo pré-processamento das fotografias aéreas e ficheiros geográficos para poderem ser utilizados no treino do modelo de segmentação semântica. Será desenvolvido com recurso a *scripts* na linguagem de programação *Python*.
- **Treino:** Módulo responsável pelo treino do modelo de segmentação semântica, com recurso a algoritmos de *deep learning*. Este componente será desenvolvido com recurso à biblioteca *FastAI*.
- **Algoritmo categorização:** Componente onde será encapsulado o modelo obtido para permitir a categorização de fotografias aéreas.
- **Jupyter Notebook:** Este componente permite a interação com o modelo de segmentação semântica. Nele, serão categorizadas as imagens e feito o seu armazenamento de forma persistente. Este componente será desenvolvido em *Python*, na *framework Jupyter Notebook*.
- **Aplicação web:** Aplicação para os utilizadores finais carregarem fotografias aéreas e interagirem com as categorizações. Este componente será implementado em *JavaScript*, utilizando a *framework Leaflet*.

Capítulo 5

Construção da solução

Neste capítulo serão abordadas todas as decisões tomadas na implementação da solução para o problema em análise. Também será detalhado todo o trabalho desenvolvido para criação dos vários componentes que constituem esta solução.

5.1 Modelo de segmentação semântica

O modelo de segmentação semântica é o artefacto mais importante deste trabalho, visto que sem ele todos os outros componentes não têm utilidade. O desenvolvimento deste modelo é um processo complexo, que envolve a necessidade de tomada de decisões em cada uma das fases do processo que podem afetar a sua qualidade. Este processo começa com a análise dos dados, sendo o seu final a obtenção do modelo com recurso ao treino das redes com os mesmos dados.

5.1.1 Pré-processamento dos dados

Para o desenvolvimento deste trabalho foram disponibilizados dois projetos de cartografia dos municípios da Batalha e de Melgaço. Ambos os projetos têm disponíveis as fotografias aéreas (no formato RGB à escala 1:2000) e as categorizações dos elementos no formato *shapfile*. Cada fotografia tem a resolução de 10000x10000 pixels. No projeto de Melgaço existem categorização de elementos para uma área de 5282 ha, enquanto que no da Batalha são 1497 ha.

Para os dois projetos, foi necessário transformar os eixos e limites de via em polígonos. Para tal, são gerados polígonos para todas as secções fechadas de limites de vias e excluem-se todos os que não têm um eixo de via no seu interior. Na figura 5.1 é possível ver um exemplo da aplicação deste processo.

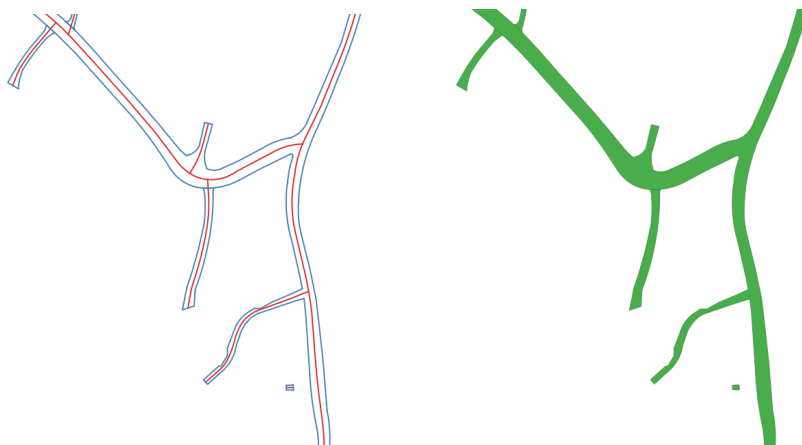


Figura 5.1: Exemplo de geometria de limites/eixo de vias (esquerda) e polígono para a mesma área (direita).

Para que estes artefactos possam ser processados pela rede neuronal, foi identificada a necessidade de efetuar um pré-processamento dos mesmos. A elevada resolução de cada imagem torna-a impossível de ser tratada pela rede neuronal sem a sua divisão em fragmentos mais pequenos, enquanto que as categorizações têm de ser transformadas do formato vetorial para um formato de imagem, para poderem ser utilizadas no treino.

Para o pré-processamento foi desenvolvida uma *pipeline* em que para cada fotografia aérea são feitos os seguintes passos:

1. Geração da imagem com as categorizações

Para cada uma das categorias a identificar, é gerada uma imagem com os pixels onde existem elementos dessa categoria preenchidos com um valor previamente definido para cada uma delas (o valor do pixel varia entre 1 e n , sendo n o número total de categorias). Todos os pixels que não pertençam a uma das categorias definidas para o modelo são agregadas numa pseudo categoria, o *void*, que corresponde a tudo o resto e ficam com o valor 0.

2. Combinação das imagens das diversas categorias para gerar a máscara da imagem

Depois de geradas as imagens para cada uma das categorias é feita uma concatenação de todas de forma a obter a imagem que representa todos os elementos presentes na fotografia aérea. Esta é a máscara usada para treino do modelo.

3. Recortes da fotografia aérea e da máscara com as categorizações

A fotografia aérea e respetiva máscara, são recortadas em fragmentos de 1024×1024 , sendo excluídas todas as fotografias em que a máscara não contém elementos de nenhuma categoria, dado que não existe "nada" que o modelo possa aprender com elas.

4. Geração de análise quantitativa das máscaras geradas

Para permitir uma melhor construção do *dataset* a utilizar durante o treino do modelo, é produzido um relatório de todas as imagens de 1024×1024 que foram obtidas. Para cada imagem é guardada a percentagem de pixels da imagem que não são da categoria *void*.

A seleção do *dataset* a utilizar para treino dos modelos, foi feita com base na análise obtida a partir da *pipeline*, tendo sido definido que apenas seriam utilizadas imagens nas quais pelo menos 50% do seu conteúdo é diferente de *void*. Esta seleção foi efetuada para evitar que sejam utilizadas imagens que apenas servem para o algoritmo aprenda aquilo que não tem interesse categorizar. A execução da *pipeline* resultou na geração de 7731 imagens de 1024x1024, sendo que após aplicação do critério de exclusão referido anteriormente, sobram 5292 imagens. Foi definido que 20% deste *dataset* seria para validação, ficando os restantes 80% para treino do modelo. É comum a utilização da *cross-validation* (uma técnica em que se parte o *dataset* em *n* partes, e em *n* iterações, cada uma das partes é utilizada para teste e as restantes utilizadas para treino do modelo) em *datasets* de pequena e média dimensão. Tendo em conta o caso apresentado na secção 3.1.1, no qual um *dataset* de 189 ha é considerado de grandes dimensões, pode-se afirmar que o *dataset* deste trabalho com 6779 ha será grande o suficiente para não ser necessário aplicar esta técnica.

5.1.2 Seleção de subcategorias

Como referido anteriormente serão desenvolvidos modelos para classificação das categorias da cartografia e para algumas subcategorias. Para além das categorias já existentes na cartografia (ADVR, AGRI, CONC, HIDA) foram adicionadas duas outras: a *roads*, para as estradas, que são obtidas através de um pré-processamento, e o *void* para tudo o que não é expectável que o modelo categorize. Foi também considerado que a inclusão da categoria ADVR poderia causar dificuldades na aprendizagem dos modelos. Por esta razão, os modelos de categorias não incluem a categoria ADVR, tendo sido feita a sua adição nos modelos de subcategorias, após terem sido efetuados alguns testes para perceber a viabilidade da sua inclusão.

Como algumas subcategorias podem não existir num projeto, ou por a sua área total ser quase irrelevante, é necessário um estudo dos dados disponíveis para se perceber quais podem ser interessantes para utilização nos modelos. Foi feita uma análise dos dados disponíveis em ambos os projetos utilizados neste trabalho, tendo sido contabilizada a área total e o total de pixels existentes para cada uma das subcategorias. Destas subcategorias foram pré-selecionadas as que têm uma área extensa, para que o modelo tenha exemplos suficientes para aprender, que sejam facilmente distinguíveis a partir da fotografia aérea. Após discussão com os responsáveis da área de cartografia foram definidas as seguintes categorias para treino do modelo: Vivenda / Casa, Construções em Geral, Área Agrícola e Florestal em Geral, Mato, Mata, Vinha, Pinheiros, Regadio / Horta, Pomar, Sequeiro, Áreas Verdes em Geral. Na tabela 5.1 é possível ver os detalhes das subcategorias selecionadas (a percentagem é calculada em relação ao total da categoria), estando a análise completa disponível no anexo A. A juntar a estas subcategorias mantém-se as categorias principais, onde estão integradas todas as subcategorias restantes.

Categoria	Código	Descrição	Área (m ²)	Pixels	Percentagem
CONC	6020102	Vivenda / Casa	1 735 893,09	173 589 309	43,16%
CONC	06000000	Construções em Geral	750 399,12	75 039 912	18,66%
AGRI	11100102	Área Agrícola e Florestal em Geral	11 491 851,83	1 149 185 183	20,83%
AGRI	11020101	Mato	10 273 369,79	1 027 336 979	18,62%
AGRI	11090301	Mata	8 964 202,19	896 420 219	16,25%

AGRI	11040100	Vinha	6 445 194,61	644 519 461	11,68%
AGRI	11090202	Pinheiros	5 754 323,32	575 432 332	10,43%
AGRI	11010101	Regadio / Horta	3 643 109,86	364 310 986	6,60%
AGRI	11030100	Pomar	2 777 524,95	277 752 495	5,04%
AGRI	11010102	Sequeiro	2 413 759,75	241 375 975	4,38%
ADVR	09020000	Áreas Verdes em Geral	991 464,54	99 146 454	29,51%

Tabela 5.1: Detalhes das subcategorias selecionadas para treino do modelo.

5.1.3 Parâmetros de treino dos modelos

Todos os modelos foram treinados utilizando as mesmas parametrizações e técnicas, sendo que futuramente, podem ser feitos ajustes nestes parâmetros para tentar melhorar os resultados obtidos. Seguidamente será feita uma breve explicação de todos os parâmetros e técnicas utilizadas neste trabalho.

Optimizador

Um dos parâmetros a definir é o otimizador, que é um algoritmo que tem como propósito o ajuste dos pesos da rede neuronal para o modelo atingir melhores resultados. Neste caso foi definido o uso do AdamW, que é um otimizador que consegue obter bons resultados, ainda que não o resultado ótimo, num menor tempo que outros existentes (Loshchilov e Hutter 2017).

Loss function

A *loss function* é utilizada para calcular o desvio entre os resultados obtidos e os resultados esperados. O treino tem como objetivo a minimização dos valores da *loss function* por forma a obter o melhor modelo possível. Neste trabalho foi utilizada a *Cross Entropy* como *loss function* (Janocha e Czarnecki 2017).

LR range test

Para definição da *learning rate*, que é um hiperparâmetro responsável pela velocidade de ajuste dos pesos na rede, foi utilizada a técnica *LR range test*. Tipicamente para se obter um bom valor para este parâmetro é necessário efetuar o treino do modelo utilizando diferentes valores, procedendo-se a ajustes sempre que os resultados melhoram ou pioram. Com esta técnica é feito um treino curto em que a *learning rate* é alterada várias vezes, sendo registados os valores obtidos pelo modelo para cada um deles. Este mini treino torna-se particularmente útil sempre que é pretendido efetuar testes com uma nova arquitetura ou *dataset*, pois permite obter um bom valor para este parâmetro (Smith 2015). Na figura 5.2 é possível observar a aplicação deste método para um dos treinos efetuados no trabalho. O ponto vermelho indica o ponto com maior declive no gráfico, isto é, o valor da *learning rate*

para o qual a diminuição da *loss function* do modelo foi maior. Este é o valor utilizado para treino de todos os modelos em análise.

```
Min numerical gradient: 1.00E-03  
Min loss divided by 10: 1.45E-02
```

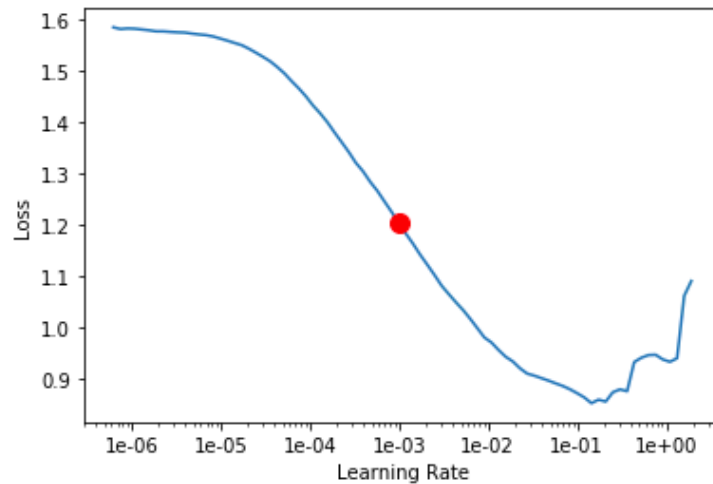


Figura 5.2: Exemplo da técnica *LR range test*.

1cycle

Esta técnica consiste na variação dos parâmetros *learning rate* e *momentum* durante o treino de um modelo, ao invés da utilização de um valor constante. Esta técnica tem este nome, visto o processo de treino passar a assentar num ciclo com dois passos, um ascendente e outro descendente, como se pode ver na figura 5.3. A *learning rate* é ascendente no 1º passo e descendente no 2º, enquanto que o *momentum* é o oposto.

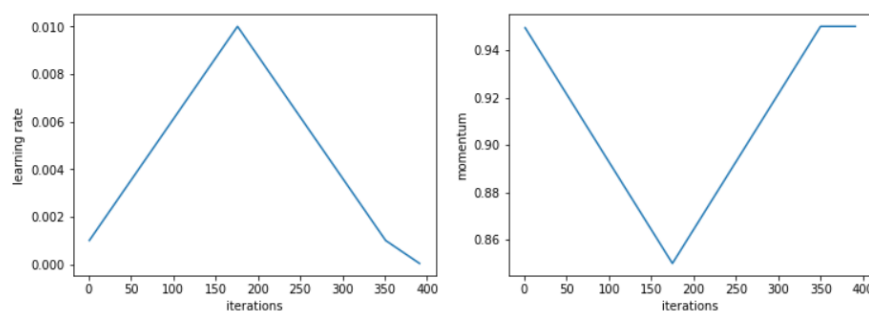


Figura 5.3: Variação da *learning rate* e do *momentum* no *1cycle* (Gugger 2019).

Foi descoberto que, com esta técnica, é possível obter uma super-convergência do modelo, permitindo que se obtenham resultados mais rapidamente do que utilizando o método tradicional (que consiste na utilização de um valor constante para a *learning rate* durante várias *epochs* até a *accuracy* estagnar, sendo feito novo treino após a diminuição da *learning rate* por um fator de 0.1) (Smith 2018). Num dos testes realizados, o modelo obtido utilizando este método tem uma *accuracy* de aproximadamente 91.9% com um treino de 100 *epochs*,

tendo o modelo obtido pelo método tradicional atingido uma *accuracy* de aproximadamente 90.3% em 800 *epochs*.

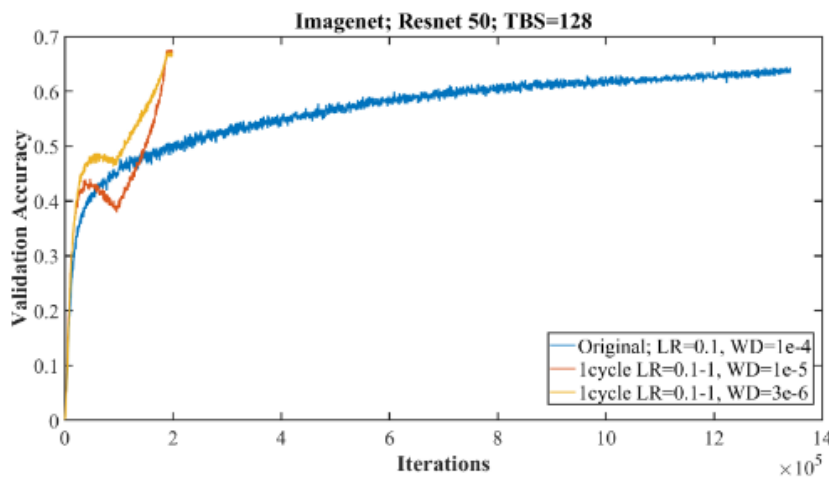


Figura 5.4: Evolução da *accuracy* num modelo utilizando a técnica *1cycle* (a amarelo e laranja) e a técnica tradicional (azul) (Smith 2018).

Transfer learning

O *transfer learning* consiste na utilização dos pesos de um modelo treinado para outra tarefa no desenvolvimento de um modelo com outro propósito, ao invés da utilização de valores iniciais aleatórios para estes pesos. Com o *transfer learning* o treino pode ser visto como um "ajuste" do modelo para adaptação a uma nova finalidade. A utilização do *transfer learning* possibilita, por norma, a obtenção de melhores resultados quando comparado com o uso de valores aleatórios (Kornblith, Shlens e Le 2018).

Progressive resizing

Esta técnica consiste num treino inicial do modelo com imagens de dimensões mais reduzidas, para que o treino seja rápido, mas que permite ao modelo aprender as linhas gerais do que se pretende classificar. Este modelo é novamente treinado com imagens de dimensões maiores, para que possam ser aprendidos os detalhes nelas presentes (fast.ai 2019). Neste trabalho os modelos foram treinados utilizando imagens de 256x256, seguidos de imagens de 512x512. Não foram utilizadas as imagens no tamanho original, dado o elevado tempo de treino necessário e por em alguns testes efetuados as imagens de 1024x1024 não trazerem melhorias significativas aos resultados obtidos.

Data augmentation

O *data augmentation* é uma técnica na qual são aplicadas transformações nos dados de treino, como por exemplo, rotações da imagem, recorte, variações no contraste e luminosidade, entre outras. Com esta técnica é possível aumentar artificialmente o tamanho de um *dataset*, ou no caso de um *dataset* grande, pode servir para aumentar a variedade dos

dados existentes, o que pode levar a uma melhor generalização do modelo. Num teste de comparação de modelos obtidos com e sem *data augmentation*, os resultados dos modelos onde foi aplicada *data augmentation* foram consideravelmente melhores (Perez e Wang 2017). Neste trabalho um modelo de classificação de cães e gatos a aplicação do *data augmentation* levou a que a *accuracy* aumentasse de 85.5% para 91.5%, enquanto que num modelo de classificação de cães e peixes-dourado a *accuracy* passou de 70.5% para 77.0%. Neste trabalho foram aplicadas transformações aleatórias no *dataset* de treino para rotação da imagem, alteração da luminosidade, aplicação zoom da imagem e inversão horizontal e vertical da imagem.

5.1.4 Modelos utilizados

Para obtenção do modelo de segmentação semântica, foram feitos testes com algumas arquiteturas já existentes e que foram abordados na secção 2.4, visto serem as mais comumente utilizadas na área da segmentação semântica. Efetuaram-se testes com a arquitetura U-Net (com *backbone* resnet-34), FCNN (com *backbone* resnet-50), Deeplab (com *backbone* resnet-50) e a FCDN (na versão FCDenseNet56). Escolheram-se as melhores versões destas arquiteturas que o *hardware* consegue suportar, sendo que na maioria dos casos não corresponde à melhor versão da mesma.

5.1.5 Resultados dos modelos para categorias

Em seguida serão apresentados os resultados obtidos para cada um dos modelos, utilizando as categorias da cartografia, sendo feita uma comparação da qualidade dos mesmos. Para cada modelo é avaliado o IoU de cada uma das categorias, o IoU médio, a *accuracy* e o tempo de treino.

Avaliando os resultados (ver tabela 5.2), os modelos FCNN e Deeplab são os que têm um melhor desempenho, sendo que a diferença em todas as métricas ronda o 1%. No entanto, a FCNN demorou menos de metade do tempo a treinar em relação ao Deeplab.

Os outros modelos avaliados, têm desempenhos bastante inferiores e apesar de em termos de *accuracy* os valores não serem muito díspares, na IoU média a diferença é superior a 9%.

Destaca-se o fraco desempenho da FCDN, ao contrário do apresentado na secção 3.1.3, na qual esta rede foi a melhor, sendo importante referir que a versão da rede utilizada é a pior das apresentadas pelos autores da mesma.

Métrica	U-Net	FCNN	Deeplab	FCDN
IoU VOID	0.572473	0.63425	0.632173	0.488232
IoU CONC	0.652151	0.744099	0.746314	0.575815
IoU AGRI	0.925095	0.936333	0.93493	0.918623
IoU Roads	0.571127	0.668757	0.67733	0.501574
IoU HIDA	0	0.184543	0.194128	0
IoU Médio	0.544169	0.633596	0.636975	0.496849
<i>Accuracy</i>	0.912425	0.92891	0.927112	0.903449
Tempo treino	4:10:43	21:49:53	46:50:32	9:22:32

Tabela 5.2: Comparação dos modelos para as categorias.

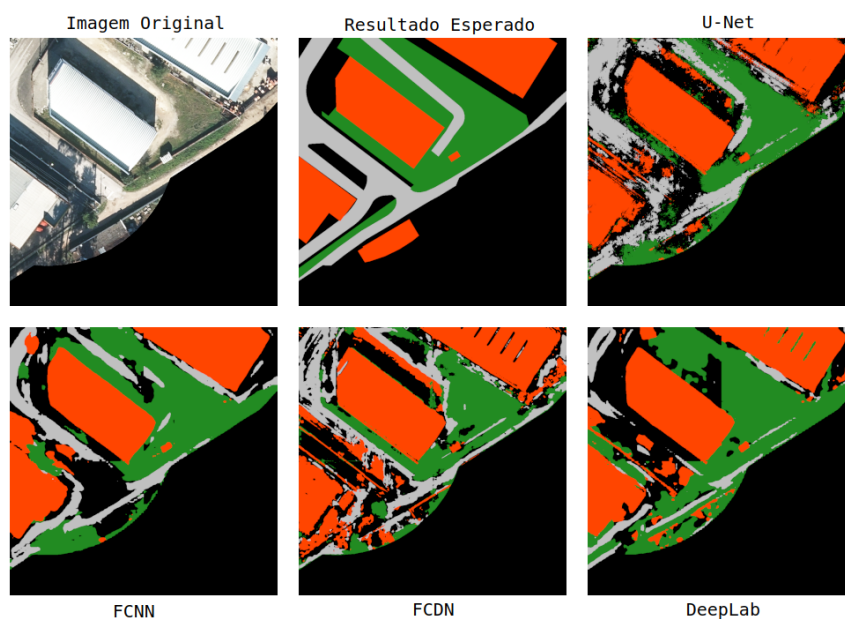


Figura 5.5: Resultados obtidos para uma fotografia aérea pelos modelos de categorias.

5.1.6 Resultados dos modelos para subcategorias

Nesta secção também serão apresentados os resultados dos modelos e feita a comparação dos mesmos, mas desta feita, aplicado ao treino com subcategorias.

Para os modelos que incluem subcategorias, o Deeplab destaca-se claramente dos outros modelos sendo o que tem melhor IoU média (41%, mais 4% que o 2º melhor) e *accuracy* (66.7%, quase mais 3% que o 2º melhor). Em relação ao IoU das outras subcategorias, apenas não é o melhor na HIDA.

A U-Net e a FCNN apresentam resultados semelhantes mas ainda distantes dos obtidos pelo Deeplab. Pela negativa destaca-se os resultados obtidos pela FCDN com apenas 18% de IoU médio e 51% de *accuracy*.

Na tabela 5.3, estão disponíveis os resultados completos para estes modelos.

Métrica	U-Net	FCNN	Deeplab	FCDN
IoU VOID	0.61734	0.63377	0.648249	0.510068
IoU Vivenda/Casa	0.606099	0.57294	0.624407	0.457016
IoU Construções em Geral	0.126305	0.195411	0.209408	0.005199
IoU CONC	0.285011	0.280134	0.336484	0.120492
IoU Pomar	0.196738	0.228248	0.287437	0.031485
IoU Mato	0.402033	0.408061	0.432551	0.224735
IoU Área Agrícola e Florestal	0.494464	0.494153	0.503541	0.368123
IoU Sequeiro	0.050075	0.137604	0.205988	0.008852
IoU Pinheiros	0.283121	0.295353	0.370582	0.055778
IoU Vinha	0.7327	0.73811	0.746906	0.387059
IoU Mata	0.401456	0.422211	0.474167	0.314114
IoU Regadio/Horta	0.330514	0.350289	0.393823	0.111901

Métrica	U-Net	FCNN	Deeplab	FCDN
IoU AGRI	0.078157	0.202471	0.260526	0.015132
IoU Roads	0.629983	0.636631	0.659856	0.438972
IoU HIDA	0.149059	0.115235	0.140456	0.000005
IoU Áreas Verdes em Geral	0.315781	0.303294	0.355535	0.101064
IoU ADVR	0.287764	0.280492	0.321623	0.04505
IoU Médio	0.352153	0.370259	0.410091	0.187944
Accuracy	0.630238	0.638192	0.667596	0.51572
Tempo treino	4:20:28	22:35:05	48:50:12	9:46:04

Tabela 5.3: Comparação dos modelos para as subcategorias.

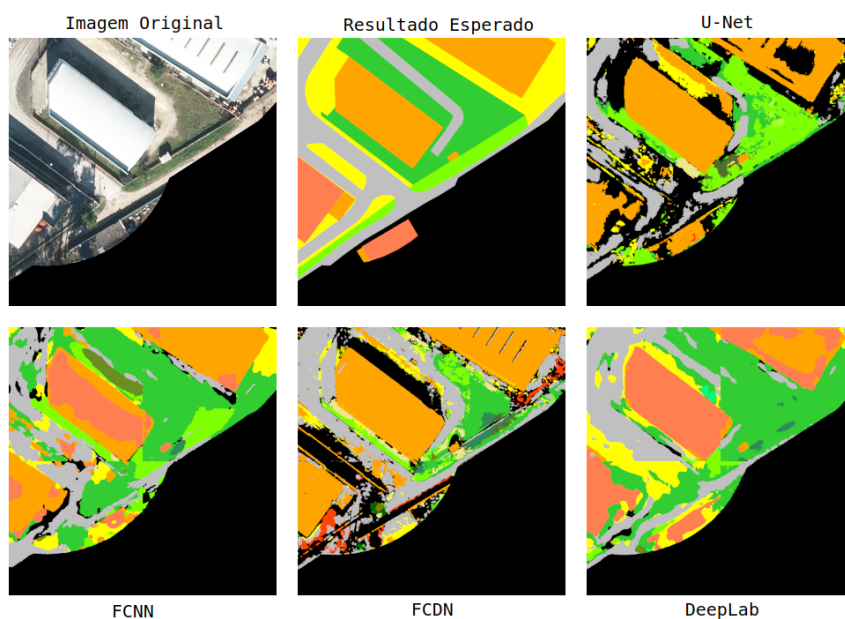


Figura 5.6: Resultados obtidos para uma fotografia aérea pelos modelos de subcategorias.

5.2 Aplicação web

Foi definido que a aplicação *web* deveria ser desenvolvida para permitir a interação com o modelo de categorização, já que o seu uso direto implicaria conhecimentos de programação por parte das pessoas que fazem este trabalho. Para permitir um desenvolvimento mais acelerado e dinâmico, optou-se pelo uso do *Jupyter Notebook*. O *Jupyter Notebook* é um ambiente de desenvolvimento em *web* para *Python*, que permite a edição e execução de códigos em células, como se um documento se tratasse. Esta ferramenta tem tido um uso crescente na área de *data science* e *machine learning* pela facilidade com que se pode testar diferentes soluções e partilhar o *Notebook* com outros utilizadores. Todo o processo de treino e a *pipeline* foram desenvolvidos com recurso ao *Jupyter Notebook*.

O *Notebook* desenvolvido, permite a categorização de imagens utilizando os modelos desenvolvidos. Inicialmente o operador escolhe o diretório onde estão as imagens que pretende

utilizar e qual o tipo de modelo (categorização de categorias / subcategorias). Após definidos estes parâmetros inicia-se o processo de classificação.

```
In [9]: f = FileBrowser()
        f.widget()

        display(f.widget())
        display(next_cell_button('Continuar'))
```

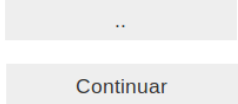


Figura 5.7: Seleção do diretório com imagens a categorizar.

```
In [10]: display(widgets.HTML(f'<h4>Escolha o tipo de classificação a efetuar</h4>'))

        selected_option = widgets.RadioButtons(
            options=['Categorias de topo', 'Subcategorias'],
            #description='Tipo de classificação',
            disabled=False
        )

        display(selected_option)
        display(next_cell_button('Continuar'))
```



Figura 5.8: Escolha do tipo de modelo a utilizar.

```
In [12]: display(widgets.HTML('<h5>A classificar...</h5>'))

output_path = Path(f.path) / 'predictions' / datetime.datetime.now().strftime('%Y%m%dT%H%M%S')
output_path.mkdir(parents=True, exist_ok=True)

for image in tqdm(orthos_to_predict):
    ai_image = open_image(image)

    _, h, w = ai_image.px.shape
    h_pad = 512 - (h % 512)
    w_pad = 512 - (w % 512)

    padded_image = torch.nn.functional.pad(ai_image.px, (0, w_pad, 0, h_pad))

    ### prediction

    row_chunks = []

    for x in tqdm(padded_image.split(512, dim=1)):
        chunk = []

        for y in x.split(512, dim=2):
            prediction = learner.predict(Image(y))
            chunk.append(prediction[1])

        row_chunks.append(torch.cat(chunk, dim=2))

    mask = torch.cat(row_chunks, dim=1)[:, :h, :w]

    # generate tif from mask
    gdal_image = gdal.Open(str(image))

    driver = gdal.GetDriverByName('GTiff')

    outRaster = driver.Create(str(output_path) + '/' + image.stem + '.tif', h, w, 3, gdal.GDT_Byte)
    outRaster.SetGeoTransform(gdal_image.GetGeoTransform())

    colored_image = torch.zeros([h, w, 3])

    for code, color in colormap.items():
        colored_image[mask.squeeze()[:, :] == code] = color

    colored_image = colored_image.numpy()

    outband = outRaster.GetRasterBand(1)
    outband.WriteArray(colored_image[:, :, 0])


    outband = outRaster.GetRasterBand(2)
    outband.WriteArray(colored_image[:, :, 1])

    outband = outRaster.GetRasterBand(3)
    outband.WriteArray(colored_image[:, :, 2])

    outband.FlushCache()

    outband = None
    outRaster = None
```

A classificar...



```
100% 1/1 [05:02<00:00, 302.42s/it]
100% 20/20 [04:48<00:00, 14.31s/it]
```

Figura 5.9: Processo de categorização das imagens.

Como facilmente se observa, o código fonte está disponível para edição, pelo que algo como a alteração dos modelos usados na categorização, é feito de uma forma rápida e simples. Como desvantagem é possível que um lapso na utilização altere o código e leve ao seu mal funcionamento, pelo que é pretendido em versões futuras a ocultação do código fonte quando utilizado por operadores de cartografia. O seu uso estritamente interno faz com que o seu uso com finalidades maliciosas seja pouco provável, pelo que não foi um ponto ainda considerado.

A categorização é armazenada no mesmo local onde estão as imagens a categorizar, sendo criado um diretório para as previsões com um *timestamp* de quando as mesmas foram efetuadas, para que no futuro caso se pretenda comparar os resultados se saiba a data de categorização das mesmas. É ainda possível efetuar o *download* de todas as categorizações feitas.



Figura 5.10: Listagem de ficheiros com previsões.

Foi também implementado um visualizador, para ser possível analisar as máscaras obtidas, tendo imagens aéreas como fundo, como é possível observar na figura 5.11. Com esta ferramenta é possível avaliar os resultados imediatamente após a sua categorização, tornando simples a validação em geral dos resultados, antes do seu *download* e análise mais detalhada. Em termos de funcionalidades especificadas na secção 4.1.2 faltou o desenvolvimento de uma ferramenta de edição das categorizações diretamente no visualizador, bem como a transformação das máscaras em formato vetorial.

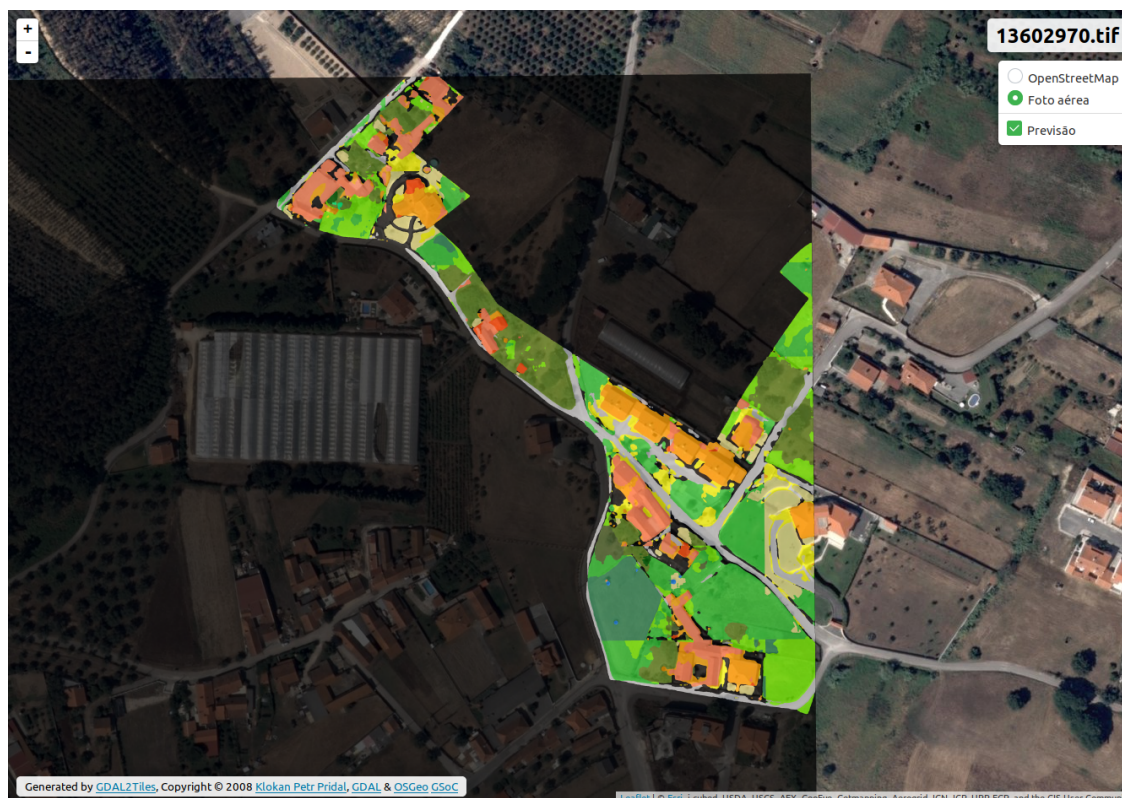


Figura 5.11: Visualizador de previsões.

Capítulo 6

Avaliação da solução

Apresentam-se, neste capítulo, as metodologias e métricas utilizadas para avaliar o trabalho, de forma a garantir que a solução obtida é a melhor possível.

6.1 Métricas

Para validação das classificações obtidas pelo algoritmo foram utilizadas duas métricas, a *accuracy* e a IoU (ambas explicadas em detalhe na secção 2.4.2). Foram escolhidas estas métricas visto serem as mais comumente utilizadas para avaliação dos resultados de algoritmos de segmentação semântica.

6.2 Metodologia de avaliação

Para treino deste tipo de redes, normalmente é feita uma divisão percentual do *dataset*, sendo uma parte utilizada para o treino e a outra, normalmente mais pequena, utilizada para a validação do modelo. O treino é efetuado em várias iterações (*epoch*). Em cada *epoch* o *dataset* de treino é passado à rede em parcelas mais pequenas (*batch size*) para que seja feita a aprendizagem com base nos dados, sendo em seguida feita a avaliação do modelo obtido com o *dataset* de avaliação.

O valor das métricas é calculado na fase de validação de cada *epoch*, onde é feito o cálculo das métricas que se pretendem avaliar utilizando a previsão obtida para a última imagem de cada *batch*. A média dos valores obtidos em cada *batch* é o resultado das métricas para essa *epoch*.

Durante o processo de treino, as métricas de cada *epoch* são úteis para perceber se o modelo esta a evoluir em cada iteração feita sobre o *dataset* de treino. As métricas obtidas na última *epoch* são as utilizadas para avaliar a qualidade do modelo obtido. Na figura 6.1 é possível ver a evolução das métricas durante o processo de treino.

epoch	train_loss	valid_loss	VOID_iou	CONC_iou	AGRI_iou	Roads_iou	HIDA_iou	acc	acc_no_void	time
0	1.000423	0.757116	0.264887	0.026637	0.797844	0.020395	0.000180	0.771962	0.845082	03:29
1	0.560417	0.456575	0.440942	0.345368	0.874914	0.146756	0.000033	0.850941	0.903374	03:29
2	0.425048	0.379720	0.476127	0.440242	0.897827	0.321061	0.000029	0.871459	0.918194	03:27
3	0.376215	0.343075	0.483385	0.499582	0.902580	0.425373	0.000000	0.883304	0.936505	03:26
4	0.331913	0.314220	0.512240	0.546249	0.911085	0.474954	0.000000	0.891367	0.937321	03:28
5	0.323321	0.310640	0.529419	0.564202	0.912249	0.483161	0.000000	0.892189	0.924922	03:27
6	0.303835	0.291569	0.540316	0.571698	0.917044	0.506036	0.000000	0.898123	0.937620	03:28
7	0.287193	0.279557	0.551081	0.596378	0.918759	0.530521	0.000000	0.903093	0.941865	03:28
8	0.285126	0.278472	0.555738	0.615555	0.917927	0.529608	0.000000	0.902944	0.935419	03:29
9	0.275745	0.266892	0.560948	0.624986	0.922130	0.550985	0.000000	0.907153	0.946671	03:28

Figura 6.1: Evolução das métricas no processo de treino.

Capítulo 7

Conclusões

Neste trabalho, foram testados múltiplos modelos de segmentação semântica, para identificação de elementos presentes em fotografia aérea, com objetivo de acelerar o processo de produção de cartografia.

Apesar de visualmente os resultados parecerem bastante interessantes, e as métricas indicarem que os resultados conseguidos estão próximos dos obtidos em casos semelhantes, ainda é necessário a validação da sua utilidade pela equipa da cartografia. Este processo ainda não foi feito, visto o processo de produção de cartografia ser iterativo e, nos projetos já concluídos, não ser possível obter os dados, no estado em que se encontravam na fase em que se pretende integrar o modelo. Está prevista a sua validação no próximo projeto de produção de cartografia. Não obstante ao já referido, elementos da equipa de produção de cartografia consideram os resultados satisfatórios e vêem também potencial para a utilização deste tipo de modelos para outras tarefas na área. Pode-se assim concluir que o objetivo principal deste trabalho, a automatização do processo de categorização de elementos em fotografia aérea, foi parcialmente atingido. Foram desenvolvidos múltiplos modelos de categorização automática com taxas de acerto interessantes, pelo que para se considerar o objetivo totalmente atingido falta então comprovar o benefício da sua utilização num projeto de produção de cartografia.

Com este trabalho também foi possível o desenvolvimento de competências em *deep learning*, estando a empresa interessada em avaliar a sua potencial aplicabilidade em outras áreas de negócio.

O desenvolvimento da aplicação *web* foi contemplado por forma a permitir a centralização das múltiplas tarefas envolvidas a esta fase (descritas na figura 4.2). Para este componente apenas foi possível implementar um protótipo, com algumas das funcionalidades que se pretendiam desenvolver. O investimento nas funcionalidades em falta ainda está sob avaliação, visto existir *software* que já responde a estes requisitos, não sendo necessário despende tempo na sua evolução/manutenção.

7.1 Trabalho futuro

Como trabalho futuro existem algumas tarefas por completar e melhorias que se podem fazer ao trabalho desenvolvido. Em relação ao modelo de segmentação poderão ser feitas as seguintes tarefas/melhorias:

- Validação do modelo pela área de cartografia.

- Transformação do *output* do modelo para o formato vetorial para permitir a sua edição/uso.
- Efetuar um pós-processamento do resultado do modelo para remover a granulação. Na figura 5.6 é possível ver que os limites das áreas classificadas têm alguma granulação, pelo que seria interessante a aplicação de algum processamento para melhoria do resultado.
- Treino das redes com diferentes parâmetros dos utilizados para tentar melhor os resultados.
- Teste com o modelo RA-FCN, que não foi possível utilizar durante o desenvolvimento do trabalho.
- Treino com diferentes formatos de *input* (RGBI e nuvem de pontos).
- Treino com dados antes da completagem de campo, para evitar falsos erros na validação do modelo.
- Treino com um pequeno *dataset* de alta qualidade, feito manualmente, para melhoria dos resultados (ver secção 3.1.1).

Já em relação à aplicação *web* têm-se como trabalho futuro:

- Ferramenta de edição dos dados obtidos pelo modelo, em formato vetorial.
- Ferramenta de análise de dados (comparação de levantamentos em diferentes alturas para uma área, análise espacial).

Bibliografia

- A Comprehensive Guide to Convolutional Neural Networks* (2019). url: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (acedido em 31/08/2019).
- AI for Earth* (2019). url: <https://aiforearth.portal.azure-api.net/docs/services/5ada7702b5f82394929929a9/operations/5ab5905bb8d61f0e48853404> (acedido em 31/08/2019).
- AlphaGo - DeepMind* (2019). url: <https://deepmind.com/research/alphago/> (acedido em 17/02/2019).
- AlphaStar - DeepMind* (2019). url: <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/> (acedido em 17/02/2019).
- AlphaZero - DeepMind* (2019). url: <https://deepmind.com/blog/alphazero-shedding-new-light-grand-games-chess-shogi-and-go/> (acedido em 17/02/2019).
- Chapman, P et al. «The CRISP-DM user guide». Em: *lyle.smu.edu*. url: <https://lyle.smu.edu/%7B%7Dmhd/8331f03/crisp.pdf>.
- Chen, Liang-Chieh, George Papandreou, Florian Schroff et al. (2017). «Rethinking Atrous Convolution for Semantic Image Segmentation». Em: arXiv: 1706.05587. url: <http://arxiv.org/abs/1706.05587>.
- Chen, Liang-Chieh et al. (2014). «Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs». Em: arXiv: 1412.7062. url: <http://arxiv.org/abs/1412.7062>.
- (2016). «DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs». Em: arXiv: 1606.00915. url: <http://arxiv.org/abs/1606.00915>.
- DEEPGLOBE - CVPR18* (2019). url: <http://deepglobe.org/> (acedido em 30/08/2019).
- DGTerritório - Cadastro* (2019). url: <http://www.dgterritorio.pt/cadastro/> (acedido em 17/02/2019).
- DGTerritório - Cartografia temática de base topográfica* (2019). url: http://www.dgterritorio.pt/sistemas%7B%5C_%7Dde%7B%5C_%7Dinformacao/snig/catalogo%7B%5C_%7Dsnig/cartografia%7B%5C_%7Doficial%7B%5C_%7De%7B%5C_%7Dhomologada/cartografia%7B%5C_%7Dtematica%7B%5C_%7Dde%7B%5C_%7Dbase%7B%5C_%7Dtopografica/ (acedido em 17/02/2019).
- DGTerritório - Cartografia topográfica* (2019). url: http://www.dgterritorio.pt/sistemas%7B%5C_%7Dde%7B%5C_%7Dinformacao/snig/catalogo%7B%5C_%7Dsnig/cartografia%7B%5C_%7Doficial%7B%5C_%7De%7B%5C_%7Dhomologada/cartografia%7B%5C_%7Dtopografica/ (acedido em 17/02/2019).
- Facebook Artificial Intelligence* (2019). url: <https://ai.facebook.com/blog/mapping-roads-through-deep-learning-and-weakly-supervised-training> (acedido em 30/08/2019).
- fast.ai* (2019). url: <https://www.fast.ai/> (acedido em 17/02/2019).
- fast.ai* (2019). *Now anyone can train Imagenet in 18 minutes* · *fast.ai*. url: <https://www.fast.ai/2018/08/10/fastai-diu-imagenet/> (acedido em 15/09/2019).

- Forbes (2019). url: <https://www.forbes.com/sites/gilpress/2018/02/07/the-brute-force-of-deep-blue-and-deep-learning/%7B%5C%7D3cbb662349e3> (acedido em 17/02/2019).
- Gsaxner, Christina et al. (2019). «Exploit fully automatic low-level segmented PET data for training high-level deep learning algorithms for the corresponding CT data». Em: *PLOS ONE* 14.3. Ed. por Abir Jaafar Hussain, e0212550. issn: 1932-6203. doi: 10.1371/journal.pone.0212550. url: <http://dx.plos.org/10.1371/journal.pone.0212550>.
- Gugger, Sylvain (2019). *Another data science student's blog – The 1cycle policy*. url: <https://sgugger.github.io/the-1cycle-policy.html> (acedido em 15/09/2019).
- Huang, Gao et al. (2016). «Densely Connected Convolutional Networks». Em: arXiv: 1608.06993. url: <http://arxiv.org/abs/1608.06993>.
- InfoPortugal (2019). url: <https://cartografia.infoportugal.pt/> (acedido em 17/02/2019).
- Janocha, Katarzyna e Wojciech Marian Czarnecki (2017). «On Loss Functions for Deep Neural Networks in Classification». Em: arXiv: 1702.05659. url: <http://arxiv.org/abs/1702.05659>.
- Jégou, Simon et al. (2016). «The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation». Em: arXiv: 1611.09326. url: <http://arxiv.org/abs/1611.09326>.
- Jordan, Jeremy (2019). *Evaluating image segmentation models*. url: <https://www.jeremyjordan.me/evaluating-image-segmentation-models/> (acedido em 25/09/2019).
- Kaiser, Pascal et al. (2017). «Learning Aerial Image Segmentation from Online Maps». Em: doi: 10.1109/TGRS.2017.2719738. arXiv: 1707.06879. url: <http://arxiv.org/abs/1707.06879> <http://dx.doi.org/10.1109/TGRS.2017.2719738>.
- Keras Documentation (2019). url: <https://keras.io/> (acedido em 17/02/2019).
- Koen, Peter et al. (2001). «Providing clarity and a common language to the “fuzzy front end”». *Research Technology Management* 44: 46 – 55». Em: url: <http://citeseerx.ist.psu.edu/viewdoc/summary;jsessionid=5F9B2AD35117F3CA408BEF764737D0DE?doi=10.1.1.459.7796>.
- Kornblith, Simon, Jonathon Shlens e Quoc V. Le (2018). «Do Better ImageNet Models Transfer Better?». Em: arXiv: 1805.08974. url: <http://arxiv.org/abs/1805.08974>.
- Krizhevsky, Alex, Ilya Sutskever e Geoffrey E. Hinton (2012). «ImageNet classification with deep convolutional neural networks». Em: *Advances in neural information processing systems*. doi: 10.1145/3065386. url: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- LeCun, Yann, Yoshua Bengio e Geoffrey Hinton (2015). «Deep learning». Em: *Nature* 521.7553, pp. 436–444. issn: 0028-0836. doi: 10.1038/nature14539. url: <http://www.nature.com/articles/nature14539>.
- Long, Jonathan, Evan Shelhamer e Trevor Darrell (2014). «Fully Convolutional Networks for Semantic Segmentation». Em: arXiv: 1411.4038. url: <http://arxiv.org/abs/1411.4038>.
- Loshchilov, Ilya e Frank Hutter (2017). «Decoupled Weight Decay Regularization». Em: arXiv: 1711.05101. url: <http://arxiv.org/abs/1711.05101>.
- M, Hossin e Sulaiman M.N (2015). «A Review on Evaluation Metrics for Data Classification Evaluations». Em: *International Journal of Data Mining & Knowledge Management Process* 5.2, pp. 01–11. issn: 2231007X. doi: 10.5121/ijdkp.2015.5201. url: <http://www.aircconline.com/ijdkp/V5N2/5215ijdkp01.pdf>.
- Mou, Lichao, Yuansheng Hua e Xiao Xiang Zhu (2019). «A Relation-Augmented Fully Convolutional Network for Semantic Segmentation in Aerial Scenes». Em: arXiv: 1904.05730. url: <http://arxiv.org/abs/1904.05730>.

- Najafabadi, Maryam M et al. (2015). «Deep learning applications and challenges in big data analytics». Em: *Journal of Big Data* 2.1, p. 1. issn: 2196-1115. doi: 10.1186/s40537-014-0007-7. url: <http://www.journalofbigdata.com/content/2/1/1>.
- O'Reilly (2019). *One simple graphic: Researchers love PyTorch and TensorFlow - O'Reilly Media*. url: <https://www.oreilly.com/ideas/one-simple-graphic-researchers-love-pytorch-and-tensorflow> (acedido em 14/09/2019).
- Perez, Luis e Jason Wang (2017). «The Effectiveness of Data Augmentation in Image Classification using Deep Learning». Em: arXiv: 1712.04621. url: <http://arxiv.org/abs/1712.04621>.
- PyTorch* (2019). url: <https://pytorch.org/> (acedido em 17/02/2019).
- Ronneberger, Olaf, Philipp Fischer e Thomas Brox (2015). «U-Net: Convolutional Networks for Biomedical Image Segmentation». Em: arXiv: 1505.04597. url: <https://arxiv.org/abs/1505.04597>.
- Sang, Dinh Viet e Nguyen Duc Minh (2018). «Fully Residual Convolutional Neural Networks for Aerial Image Segmentation». Em: *Proceedings of the Ninth International Symposium on Information and Communication Technology - SolCT 2018*. New York, New York, USA: ACM Press, pp. 289–296. isbn: 9781450365390. doi: 10.1145/3287921.3287970. url: <http://dl.acm.org/citation.cfm?doid=3287921.3287970>.
- Shah, Parth, Vishvajit Bakrola e Supriya Pati (2019a). «Optimal Approach for Image Recognition using Deep Convolutional Architecture». Em: arXiv: 1904.11187. url: <http://arxiv.org/abs/1904.11187>.
- (2019b). «Optimal Approach for Image Recognition using Deep Convolutional Architecture». Em: arXiv: 1904.11187. url: <http://arxiv.org/abs/1904.11187>.
- Simonyan, Karen e Andrew Zisserman (2014). «Very Deep Convolutional Networks for Large-Scale Image Recognition». Em: arXiv: 1409.1556. url: <http://arxiv.org/abs/1409.1556>.
- Smith, Leslie N. (2015). «Cyclical Learning Rates for Training Neural Networks». Em: arXiv: 1506.01186. url: <http://arxiv.org/abs/1506.01186>.
- (2018). «A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay». Em: arXiv: 1803.09820. url: <http://arxiv.org/abs/1803.09820>.
- TensorFlow* (2019). url: https://www.tensorflow.org/api%7B%5C_%7Ddocs (acedido em 17/02/2019).
- Tian, Chao, Cong Li e Jianping Shi (2018). *Dense Fusion Classmate Network for Land Cover Classification*. url: http://openaccess.thecvf.com/content%7B%5C_%7Dcvpr%7B%5C_%7D2018%7B%5C_%7Dworkshops/w4/html/Tian%7B%5C_%7DDense%7B%5C_%7DFusion%7B%5C_%7DClassmate%7B%5C_%7DCVPR%7B%5C_%7D2018%7B%5C_%7Dpaper.html.
- Woodall, Tony (2003). «Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis». Em: *Academy of Marketing Science Review*.
- Yeow, Jian Ai et al. (2014). «Effects of Stress, Repetition, Fatigue and Work Environment on Human Error in Manufacturing Industries». Em: *Journal of Applied Sciences* 14.24, pp. 3464–3471. issn: 18125654. doi: 10.3923/jas.2014.3464.3471. url: <http://www.scialert.net/abstract/?doi=jas.2014.3464.3471>.

Apêndice A

Análise dos dados das categorias

Código	Descrição	Área (m ²)	Pixéis	Porcentagem	Percentil
09020000	Áreas Verdes em Geral	991 464,54	99 146 454	29,51%	29,51%
03020101	Rochas	770 028,61	77 002 861	22,92%	52,43%
07000000	Áreas Industriais Serviços em Geral	679 506,88	67 950 688	20,22%	72,65%
09020100	Parques Jardins em Geral	214 137,29	21 413 729	6,37%	79,03%
13060000	Áreas Utilização Pública e Oficial	185 761,10	18 576 110	5,53%	84,55%
10100104	Parque Estacionamento	185 096,22	18 509 622	5,51%	90,06%
09010000	Área Desportiva em Geral	127 753,42	12 775 342	3,80%	93,87%
09010103	Limite Recinto Prática Desportiva	78 570,02	7 857 002	2,34%	96,20%
13040000	Cemitério	59 172,05	5 917 205	1,76%	97,97%
09010102	Campo Jogos sem Bancadas	21 801,53	2 180 153	0,65%	98,61%
09010600	Piscinas em Geral	8 667,44	866 744	0,26%	98,87%
07050200	Tratamento Resíduos Líquidos	8 174,30	817 430	0,24%	99,12%
09010101	Campo Jogos com Bancadas	8 141,71	814 171	0,24%	99,36%
10070101	Central Camionagem	7 778,32	777 832	0,23%	99,59%
07060101	Central Eléctrica	6 154,89	615 489	0,18%	99,77%
09020401	Parque Campismo	5 553,82	555 382	0,17%	99,94%
09010801	Campo Ténis	1 996,31	199 631	0,06%	100,00%
13030200	Ruínas	89,03	8 903	0,00%	100,00%
Total		3 359 847,48	335 984 748		

Tabela A.1: Análise dos dados para a categoria ADVR.

Código	Descrição	Área (m ²)	Pixeis	Percentagem	Percentil
11100102	Área Agrícola e Florestal em Geral	11 491 851,83	1 149 185 183	20,83%	20,83%
11020101	Mato	10 273 369,79	1 027 336 979	18,62%	39,46%
11090301	Mata	8 964 202,19	896 420 219	16,25%	55,71%
11040100	Vinha	6 445 194,61	644 519 461	11,68%	67,39%
11090202	Pinheiros	5 754 323,32	575 432 332	10,43%	77,83%
11010101	Regadio / Horta	3 643 109,86	364 310 986	6,60%	84,43%
11030100	Pomar	2 777 524,95	277 752 495	5,04%	89,47%
11010102	Sequeiro	2 413 759,75	241 375 975	4,38%	93,84%
11090104	Carvalhos	1 412 514,70	141 251 470	2,56%	96,40%
11090201	Eucaliptos	1 167 769,84	116 776 984	2,12%	98,52%
11030200	Olival	740 617,26	74 061 726	1,34%	99,86%
11090206	Sobreiros	55 434,90	5 543 490	0,10%	99,96%
11070201	Eira	9 606,56	960 656	0,02%	99,98%
11090101	Castanheiros	5 651,40	565 140	0,01%	99,99%
11020102	Canavial	4 931,15	493 115	0,01%	100,00%
11090205	Montado Misto (Sobro+Azinho)	719,29	71 929	0,00%	100,00%
Total		55 160 581,41	5 516 058 141		

Tabela A.2: Análise dos dados para a categoria AGRI.

Código	Descrição	Área (m ²)	Pixeis	Percentagem	Percentil
06020102	Vivenda / Casa	1 735 893,09	173 589 309	43,16%	43,16%
06000000	Construções em Geral	750 399,12	75 039 912	18,66%	61,82%
06020202	Telheiro / Alpendre	361 339,00	36 133 900	8,99%	70,81%
06020203	Anexo / Arrecadação	327 315,51	32 731 551	8,14%	78,95%
06050105	Fábrica	153 376,92	15 337 692	3,81%	82,76%
06070201	Edifício em Ruínas	122 067,70	12 206 770	3,04%	85,80%
06020201	Garagem	105 448,29	10 544 829	2,62%	88,42%
06020103	Prédio	102 069,38	10 206 938	2,54%	90,96%
06050501	Armazém	75 669,53	7 566 953	1,88%	92,84%
06020204	Escadas	50 583,69	5 058 369	1,26%	94,10%
06110210	Espigueiro	20 231,11	2 023 111	0,50%	94,60%
06110211	Estufa Agrícola	16 700,30	1 670 030	0,42%	95,02%
06010307	Escola Preparatória e ou Secundária	15 327,34	1 532 734	0,38%	95,40%
06140000	Edifício em Construção	13 340,61	1 334 061	0,33%	95,73%
06030102	Centro Comercial	13 160,50	1 316 050	0,33%	96,06%
06120101	Pavilhão Gimno Desportivo	12 707,04	1 270 704	0,32%	96,37%
06120102	Piscina	11 725,29	1 172 529	0,29%	96,66%

Código	Descrição	Área (m ²)	Pixéis	Porcentagem	Percentil
06010601	Mosteiro / Convento	10 906,04	1 090 604	0,27%	96,93%
06050201	Hotel	10 590,52	1 059 052	0,26%	97,20%
06010701	Igreja	8 556,29	855 629	0,21%	97,41%
06010702	Capela	7 912,74	791 274	0,20%	97,61%
06020105	Barraca	7 667,61	766 761	0,19%	97,80%
08010901	Painel Solar	6 949,06	694 906	0,17%	97,97%
06010305	Escola Primária	5 764,69	576 469	0,14%	98,11%
06120106	Outras Construções (Instalações Desportivas)	5 164,85	516 485	0,13%	98,24%
06010803	Lar 3ª Idade	4 669,30	466 930	0,12%	98,36%
06090104	Depósito Água Superfície	4 424,29	442 429	0,11%	98,47%
06010403	Centro Saúde	4 119,69	411 969	0,10%	98,57%
06080107	Áreas Serviço (Restaurante / Estação Serviço / etc)	3 617,40	361 740	0,09%	98,66%
06010303	Jardim Infância	3 505,05	350 505	0,09%	98,75%
06120104	Estádio / Bancadas	3 337,88	333 788	0,08%	98,83%
06011001	Quartel Bombeiros	2 937,07	293 707	0,07%	98,90%
06010301	Creche	2 830,10	283 010	0,07%	98,97%
06120201	Edifícios Termas	2 324,94	232 494	0,06%	99,03%
06010406	Centro Recuperação Reabilitação	2 087,75	208 775	0,05%	99,08%
06010311	Universidade / Faculdade	2 016,25	201 625	0,05%	99,13%
06040201	Bancos	1 968,80	196 880	0,05%	99,18%
06050502	Silo	1 934,59	193 459	0,05%	99,23%
06090202	Posto Transformação	1 920,51	192 051	0,05%	99,28%
06010607	Castelo	1 876,37	187 637	0,05%	99,33%
06030104	Hipermercado	1 869,19	186 919	0,05%	99,37%
06010401	Hospital	1 794,21	179 421	0,04%	99,42%
06010309	Escola Formação Profissional	1 629,75	162 975	0,04%	99,46%
06050202	Pensão / Residencial / Albergaria	1 602,77	160 277	0,04%	99,50%
06050207	Casa Rural	1 495,84	149 584	0,04%	99,53%
06010504	Biblioteca	1 294,57	129 457	0,03%	99,57%
06030103	Mercado	1 283,71	128 371	0,03%	99,60%
06080101	Parque Estacionamento	1 253,55	125 355	0,03%	99,63%
06010804	Centro Social	1 193,65	119 365	0,03%	99,66%

Código	Descrição	Área (m ²)	Pixéis	Porcentagem	Percentil
06010201	Câmara Municipal / Secretaria Regional	1 171,37	117 137	0,03%	99,69%
06120702	Pousada Juventude	1 008,20	100 820	0,03%	99,71%
06120301	Edifícios Campismo	944,84	94 484	0,02%	99,74%
06010502	Museu	911,79	91 179	0,02%	99,76%
06010908	Edifício GNR	886,57	88 657	0,02%	99,78%
06100200	Resíduos Líquidos	855,58	85 558	0,02%	99,80%
06080102	Parque Estacionamento em Telheiro	774,42	77 442	0,02%	99,82%
06080105	Estação (Terminal de Camionagem)	663,30	66 330	0,02%	99,84%
06090101	Estação Tratamento Águas	610,37	61 037	0,02%	99,85%
06090302	Gasómetro	587,08	58 708	0,01%	99,87%
06040101	Correios Telégrafos Telefones	584,57	58 457	0,01%	99,88%
06010512	Coreto	511,69	51 169	0,01%	99,90%
06010203	Edifício Administrativo	509,90	50 990	0,01%	99,91%
06050104	Estaleiro	500,00	50 000	0,01%	99,92%
06090103	Depósito Água Elevado	446,13	44 613	0,01%	99,93%
06010204	Tribunal	417,14	41 714	0,01%	99,94%
06070102	Azenha	412,51	41 251	0,01%	99,95%
06050401	Depósito Combustível	318,98	31 898	0,01%	99,96%
06010302	Infantário	295,37	29 537	0,01%	99,97%
06010703	Santuário	282,25	28 225	0,01%	99,97%
06010511	Centro Cultural	247,74	24 774	0,01%	99,98%
06070101	Moinho	222,91	22 291	0,01%	99,99%
10070102	Paragem Auto-carro	172,30	17 230	0,00%	99,99%
06010605	Estátua	111,79	11 179	0,00%	99,99%
06040202	Seguros	102,10	10 210	0,00%	100,00%
06090105	Depósito Água Subterrâneo	101,57	10 157	0,00%	100,00%
06120502	Estufa Botânica	26,29	2 629	0,00%	100,00%
06010507	Observatório	13,95	1 395	0,00%	100,00%
06050701	Mina	10,21	1 021	0,00%	100,00%
Total		4 021 554,33	402 155 433		

Tabela A.3: Análise dos dados para a categoria CONC.

Código	Descrição	Área (m ²)	Pixeis	Porcentagem	Percentil
12020602	Tanque	38 732,46	3 873 246	56,30%	56,30%
12040301	Lagoa	17 174,55	1 717 455	24,96%	81,26%
08010101	Poço	9 169,78	916 978	13,33%	94,59%
12020601	Lago Jardim	3 693,70	369 370	5,37%	99,96%
08010105	Nora	30,63	3 063	0,04%	100,00%
Total		68 801,11	6 880 111		

Tabela A.4: Análise dos dados para a categoria HIDA.