Universidade da Beira Interior
Departamento de Informática

Nº 43395 - 2022: *Where Am I? Image-based Geolocalization from Google Maps Data*

Elaborado por:

José Marques

Orientador:

Hugo Proença

10 de julho de 2022

Agradecimentos

A conclusão deste trabalho, bem como grande parte da minha vida não seria possível sem a ajuda dos meus pais e de grandes amigos. Quanto ao meu percurso académico, não seria o mesmo sem a ajuda de professores dedicados dos quais ressalvo o meu orientador Hugo Proença.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

UBI Universidade da Beira Interior

ResNet Uma rede neural residual

GPS Global Positioning System

Tensorflow plataforma de código aberto para aprendizagem de máquinas

SAFA spatial-aware feature aggregation/agregação de características com consciência espacial

first-in-first-firstout Primeiro a entrar primeiro a sair

MLP Multilayer Perceptron/Perceptrão de múltiplas camadas

softmax A função de ativação softmax é usada em redes neurais de classificação.

CNN convolutional neural networks(Rede neural convolucional)

ML Machine Learning

IA Intelegência artificial

DL Deep Learning

API Application Programming Interface

GPU unidades de processamento gráfico

TPU unidades de processamento tensorial

CUDA Compute Unified Device Architecture

RNN Recurrent neural network

VGG Very Deep Convolutional Networks for Large-Scale Visual Recognition(Redes convulucionais muito profundas para reconhecimento visual em grande escala)

Capítulo

1

Introdução

1.1 Enquadramento

O projeto chama-se *"Where Am I? Image-based Geolocalization from Google Maps Data"* e como o nome sugere, o objetivo deste projeto é desenvolver "uma intelegência"artifical capaz de através uma imagem devolver as coordenadas (latitude, longitude) de onde esta foi tirada.Devido à dificuldade de acesso a imagens com geolocalização, pois o google não fornece estas imagens de forma gratuita , a aplicação deste projeto vai se limitar à area da covilhã , principalmente ao centro e zonas perto dos polos da Universidade da Beira Interior (UBI).

1.2 Motivação UBI

Motivação UBI O que me motivou a propor este projeto foi o facto de sempre ter tido interesse na área da intelegência artificial , e o facto de isto ser um bom problema para a sub-àrea de classificação de imagem.Também acho que deva haver uma alternativa de um indivíduo conseguer descobrir a sua localização sem ter que estar 100% dependente do *google maps* ou outras aplicações semelhantes.

1.3 Objetivos

O objetivo deste projeto é a realização de um modelo capaz de devolver um par de cordenadas (latitude e longitude) ,de uma imagem fornecida de forma correta ou o mais próximo possivel da zona de onde a foto foi tirada.

1.4 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.

2. O segundo capítulo – **Estado da arte** – descreve conceitos importantes no âmbito deste projeto , bem como todo o trabalho de investigação para a concretização do projeto.

3. O terceiro capítulo – **Tecnologias** – descreve conceitos importantes no âmbito deste projeto, bem como as tecnologias utilizadas durante do desenvolvimento da aplicação.

4. O quarto– **Implementação e Testes** – Crição de modelos , analisar modelos , comparar modelos e escolher o modelo final.

5. O quinto– **Conclusão e Trabalho futuro** – conclui-se o projeto e são enumeradas ideias de como melhorar o projeto em si.

Capítulo

2

Estado da Arte

2.1 Introdução

Neste capítulo apresentam - se os modelos que serviram de inspiração e foram usados para investigação no âmbito da realização deste projeto.

2.2 Modelo Estimativa de geolocalização de fotos usando um Modelo Hierárquico e Classificação de Cenas

Nesta secção, apresenta-se o quadro de aprendizagem profunda proposto para a estimativa da geolocalização. Segundo os artigos [1],[2] a tarefa é tratada como um problema de classificação subdividindo a terra em parcelas geográficas que contêm um número semelhante de imagens.

2.2.1 Particionamento Adaptativo da Terra em parcelas quadrangulares

Mais detalhadamente, a superfície da terra é projectada sobre um cubo envolvente com seis lados que representam as parcelas iniciais.É aplicada uma subdivisão hierárquica adaptativa baseada nas coordenadas GPS das imagens, onde cada parcela é o nó de uma árvore quartenária. A partir dos nós de raiz, a respectiva parcela é dividida recursivamente até que todas as parcelas contenham um número máximo de imagens . Depois, todas as parcelas resultantes com menos de um número mínimo definido de fotos são descartadas, porque muito provavelmente cobrem áreas como pólos ou oceanos que são

difíceis de distinguir parcelas em áreas bem cobertas fotograficamente são criadas. Isto permite uma previsão mais precisa da localização das imagens que muito provavelmente descrevem regiões interessantes tais como pontos de referência ou cidades.

2.2.2 Classificação Visual de Cenas

Para classificar cenários e extrair etiquetas para os cenários, aplica-se o modelo Uma rede neural residual (ResNet) com 152 camadas . O modelo foi treinado em mais de 16 milhões de imagens de treino de 365 categorias de locais diferentes. Isto encaixa bem nesta abordagem, uma vez que o classificador resultante já distingue imagens que retratam ambientes específicos. As etiquetas da cena são previstas com base no conjunto de vistas de todas as imagens de treino, utilizando a probabilidade máxima do vector de saída. Com base na hierarquia de cenários fornecidos, são extraídas , adicionalmente, etiquetas dos outros conjuntos contendo mais categorias de vistas superiores. Adiciono as probabilidades de todas as classes que são atribuídas à mesma categoria superordenada e gero a etiqueta correspondente. Contudo, algumas cenas como um celeiro são atribuídas a múltiplas categorias (ar livre, natural, feitas pelo homem), porque se sobrepõem visualmente. Por esta razão, A probabilidade destas classes é dividida pelo número de categorias atribuídas para manter a normalização.

2.2.3 Estimativa de Geolocalização

Nesta secção, são introduzidas várias abordagens baseadas em redes neurais convolucionais para uma geolocalização à escala planetária sem restrições. Em primeiro lugar, é apresentada uma abordagem em que o modelo que serve como base que é treinado sem utilizar informação relativa às paisagens e múltiplas partições geográficas. A seguir, é descrito como a informação para diferentes resoluções espaciais, bem como conceitos ambientais, são integrados no processo de formação. Neste contexto, são propostas duas abordagens diferentes para a utilização de rótulos de cenários.

Linha de referência: Para avaliar o impacto das abordagens sugeridas para a geolocalização, é apresentado primeiro um sistema de base que não é influenciado em informações sobre o ambiente e diferentes resoluções espaciais.Por conseguinte, é gerada uma única partição da parcela. Para a classificação, é adicionada uma camada no topo da camadada global da arquitectura ResNet, onde o número de neurónios de saída corresponde ao número de parcelas. Durante o treino, a perda de entropia-cruzada das parcelas é baseada na distribuição de probabilidades e na etiqueta da parcela.

Variante Multi-Partições: É proposta a aprendizagem simultânea da estimativa de geolocalização em múltiplas resoluções espaciais de acordo com o artigo [3]. Em contraste com a abordagem da Linha de referência,é adicionada uma camada totalmente ligada às parcelas geográficas de todas as partições. A perda de classificação multi-partições é calculada utilizando a média dos valores das perdas para cada partição. Como consequência, a rede neural convolutiva é capaz de aprender características geográficas em diferentes escalas, resultando numa classificação mais discriminatória. Contudo, em contraste com o artigo [3], é explorado ainda mais o conhecimento hierárquico para a previsão final.

Redes de Cena Individual: Numa primeira tentativa de incorporar informação contextual sobre o cenário ambiental para a geolocalização fotográfica, são formadas redes individuais de imagens que retratam uma vista específica. Para cada fotografia, são extraídas as probabilidades utilizando a classificação de cenário apresentada anteriormente. Durante a formação, cada imagem com uma probabilidade de cenário superior a um limiar pré-definido é utilizada como entrada para a respectiva Rede de Cenas Individuais . Seguir esta abordagem oferece a vantagem da rede ser exclusivamente treinada em imagens que descrevem paisagens ambientais específicas. Reduz significadamente a diversidade no espaço de dados subjacente e permite à rede aprender características mais específicas. No entanto, é necessário treinar modelos individuais para cada etiqueta de cenário, o que é difícil de gerir se o número de etiquetas diferentes se tornar maior. Por esta razão, sugiro afinar o modelo, que foi inicialmente treinado sem restrições de vista, com imagens da respectiva categoria ambiental.

Rede de classes múltiplas: Uma vez que o método referido de estimativa de geolocalização pode tornar-se inviável para uma grande quantidade de categorias ambientais diferentes, o objectivo é uma abordagem mais aplicável utilizando uma rede que trate a geolocalização fotográfica e o reconhecimento de cenas como um problema de classes múltiplas. A fim de encorajar a rede a distinguir entre imagens de diferentes vistas ambientais, são treinados simultaneamente dois classificadores para estas tarefas complementares. A adição de outra tarefa (complementar) provou ser eficiente para melhorar os resultados da tarefa principal. Mais especificamente, é utilizada uma camada adicional totalmente ligada no topo da camada global da ResNet. O número de neurónios de saída desta camada corresponde à quantidade de categorias do cenário. Os pesos de todas as outras camadas da rede são completamente partilhados.

2.2.4 Previção da Geolocalização usando Informação Espacial Hierárquica

A fim de estimar as coordenadas Global Positioning System (GPS) a partir da saída da classificação, são aplicados os modelos treinados em três amostras de culturas de uma dada imagem de consulta, de acordo com a sua orientação. Em seguida, calcula-se a média das probabilidades de classe resultantes de cada cultura. Note-se que é necessário um passo adicional para testar as Redes de Cena Individual. Neste caso, o rótulo da cena é primeiro previsto utilizando a probabilidade máxima, a fim de alimentar a imagem nas respectivas Redes de Cenas Individuais para geolocalização.

Geoclassificação padrão: Sem depender de informação hierárquica, é utilizado apenas as probabilidades de uma determinada partição de parcelas. A este respeito, atribuo a etiqueta de classe com a probabilidade máxima para prever a parcela geográfica. Aplicando a abordagem multi-particionamento, é possível obter probabilidades de classe em diferentes resoluções espaciais. Na minha opinião, as probabilidades em todas as escalas devem ser exploradas para melhorar a geolocalização e para combinar as capacidades de todas as partições.

Geoclassificação Hierárquica: Para assegurar que cada parcela geográfica possa ser ligada de forma única a uma área superior, é aplicado um parâmetro com um limiar fixo para a subdivisão adaptativa. Assim, é possivel gerar uma hierarquia geográfica a partir das diferentes resoluções espaciais.São Multiplicadas as respectivas probabilidades em cada nível da hierarquia. Consequentemente, a previsão para a melhor subdivisão pode ser refinada através da incorporação do conhecimento de representações mais rudimentares.

Classe para GPS: Dependendo da classe prevista, extraímos as coordenadas GPS da imagem da consulta dada. Utilizando a média da localização de todas as imagens de treino na parcela prevista, em vez do centro geográfico. Isto é mais preciso para regiões que contêm uma área de interesse onde a maioria das fotos é tirada.

2.2.5 Treino da Rede

As abordagens propostas foram treinadas utilizando uma arquitectura ResNet com 101 camadas convolutivas. Os pesos são inicializados por um modelo pré-treinado. Para evitar sobreajustes, os dados são aumentados através da selecção aleatória de uma área que cubra pelo menos 70% da imagem com uma relação de aspecto entre 3/4 e 4/3. Além disso, as imagens de entrada são aleatoriamente invertidas e posteriormente recortadas para 224 × 224 pixels.É utilizado o optimizador de Descida Estocástica Gradiente com

uma taxa de aprendizagem inicial de 0,01, um impulso de 0,9, e uma decadência de peso de 0,0001. A taxa de aprendizagem é exponencialmente reduzida por um factor de 0,5, após cada cinco épocas de treino. Inicialmente as redes são treinadas durante 15 épocas e um lote com tamanho de 128.As redes neurais convolucionais, são validadas em 25,600 imagens. Poderia ser benéfico afinar as Redes de Cena Individual com base num modelo que foi inicialmente treinado sem restrições de cena. Para uma comparação justa, todos os modelos são, portanto, afinados durante cinco épocas ou até que a perda no conjunto de validação convirja. A este respeito, a taxa de aprendizagem inicial é reduzida para 0,001. Finalmente, o melhor modelo do conjunto de validação é utilizado para a realização das experiências. A implementação é realizada utilizando a biblioteca plataforma de código aberto para aprendizagem de máquinas (Tensorflow) [4] em Python.

2.3 Geo-localização de Imagem Cruzada para além da Recuperação Um-a-Um

Nesta secção, apresenta -se o quadro de aprendizagem profunda proposto para a estimativa da geolocalização. Segundo o artigo [5], este modelo destaca-se pelo facto de utilizar imagens a nível do chão/solo e imagens de vista aéreas/satélite para treinar o modelo de forma a classificar com maior precisão a geo-localização das fotos a ser testadas.

2.3.1 Introdução ao modelo

Trabalhos recentes ,[6],[7],[8], mostraram que o desempenho da correspondência de imagens de visão cruzada pode ser significativamente melhorado através da agregação de características e estratégias de extracção de amostras. Quando a orientação da imagem de visualização de rua(fotos a nível do chão/solo) está disponível , os métodos mais avançados podem alcançar uma precisão de recuperação superior a 80% [7], o que mostra a possibilidade de geo-localização precisa em cenários reais.Contudo, os conjuntos de dados existentes simplesmente assumem que cada imagem de visualização de rua, tem uma imagem de visualização aérea de referência correspondente cujo centro está exactamente alinhado no local da imagem. Argumentamos que isto não é prático para aplicações do mundo real, porque a imagem sobre a qual vamos fazer a previsão pode ocorrer em locais arbitrários na área de interesse e as imagens de referência devem ser capturadas antes da previsão. Neste caso, não é garantida uma correspondência perfeita de um para um. Tendo em conta este problema, o artigo [5] propõe uma nova referência para

avaliar a geo-localização transversal num cenário mais realista. Em resumo, dada uma área de interesse, as imagens aéreas de referência são densamente distribuídas para conseguir uma cobertura sem falhas da área de interesse.As Imagens de visão de rua são capturadas em locais arbitrários.

Para além da Recuperação Um-a-Um: Investigações anteriores centram-se principalmente na correspondência um-para-um porque os conjuntos de dados existentes consideram como padrão pares de imagens perfeitamente alinhadas. Contudo, este modelo permite-nos explorar o efeito de amostras de referência que não estão centradas nos locais das imagens , mas ainda cobrir a área de interesse. Como resultado, pode haver múltiplas imagens de referência cobrindo parcialmente o mesmo local de pesquisa, quebrando a correspondência um-para-um. Neste método de geo-localização, é concebida uma nova perda híbrida a tomar vantagem de múltiplas imagens de referência durante a formação.

2.3.2 Localização de visão cruzada bruta a refinada

2.3.3 Estrutura base

Agregação de características: É utilizado o módulo de agregação de características spatial-aware feature aggregation/agregação de características com consciência espacial (SAFA) [7] com a estratégia global de extração negativa(extração neste contexto é o uso de técnicas criadas pela aprendizagem de máquinas para prever os resultados,ao encontrar padrões interessantes entre os itens do conjunto de dados). A agregação de características SAFA [7] é uma combinação de transformação polar e de agregação de blocos de características. No entanto, a transformação polar pressupõe que o GPS de visualização terrestre está no centro da imagem de referência de visualização aérea correspondente, o que não se aplica no nosso caso. Por conseguinte, só adoptamos a agregação de características numa fase inicial do modelo. A ideia principal do bloco de agregação de características é a de voltar a colocar os pesos dos elementos de acordo com as suas posições. O bloco espacialmente consciente proporciona um ganho de desempenho significativo quando a informação de orientação das imagens está disponível. **Estratégia de extração:** É importante para extrair amostras rigorosas durante a formação, pois o modelo sofreria de fraca convergência pois a maioria das amostras mal iria contribuir para a perda total. A ideia chave é construir uma piscina de extração Primeiro a entrar primeiro a sair (first-in-first-firstout) para armazenar integração da amostra e refrescar a piscina juntamente com a propagação posterior de forma eficiente. Num mini-batch, as imagens da primeira metade são seleccionadas aleatoriamente e as amostras globais são extraídas da piscina para formar a outra

metade do lote. Adoptamos esta eficiente estratégia global de extração [8] na estrutura base para melhorar ainda mais o seu desempenho.

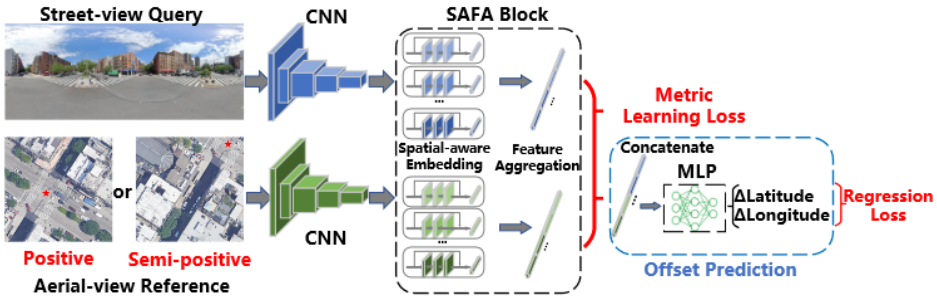


Figura 2.1: Uma visão geral do quadro proposto.https://github.com/Jeff-Zilence/VIGOR/blob/main/data/Architecture.jpg

2.3.4 Previsão

Com a recuperação de imagens, o intervalo mínimo entre as imagens de referência recuperadas do conjunto de dados é metade da largura das imagens aéreas.Para conseguir uma localização mais precisa, aplicamos um Multilayer Perceptron/Perceptrão de múltiplas camadas (MLP) para prever o desvio da localização da consulta em relação ao centro da imagem de referência recuperada. Como mostrado na figura 2.2, o MLP auxiliar consiste em duas camadas totalmente ligadas e recebe como entrada as características de incorporação concatenadas.

2.3.5 Implementação

Todas as experiências são realizadas usando o Tensorflow.As panorâmicas de visão do solo e as imagens de visão aérea são redimensionadas para 640 × 320 e 320 × 320 respectivamente, antes de serem alimentados na rede. [9] é adoptado como o extractor de características de espinha dorsal e 8 Os blocos [7] são utilizados seguindo.

2.4 Aprendizagem Residual Profunda para o Reconhecimento da Imagem ResNet

Nesta secção apresenta - se a rede neural de aprendizagem profunda , usado para o reconhecimento de imagens, segundo o artigo [10].Ao contrário dos

outros artigos mencionados que usei como inpiração para o meu projeto, este não aborda o tema de geolocalização, mas aprofunda o estudo sobre redes neurais de aprendizagem profunda , que podem ser muito uteis no âmbito do reconhicimento da localização através de imagens.

2.4.1 Introdução

Redes neurais convolucionais profundas [11] [12] conduziram a uma série de avanços na classificação de imagens [12] [13]. As redes profundas integram naturalmente características de baixo/médio/alto nível [13] e classificadores de ponta a ponta de múltiplas camadas, e os "níveis"das características podem ser enriquecidos pelo número de camadas empilhadas (profundidade).

Impulsionada pelo significado da profundidade, surge uma questão: Melhorar a aprendizagem das redes de aprendizagem profunda é tão simples empilhar mais camadas? Um obstáculo à resposta desta pergunta foi o notório problema do desaparecimento/exploração de gradientes [14] [15], que impedem a convergência desde o início. Este problema, contudo, foi amplamente abordado pela inicialização normalizada [16] [15] [17] e pelas camadas intermediárias de normalização [18], que permitem que redes com dezenas de camadas comecem a convergir para descidas de gradientes estocásticos com retropropagação [11].

Quando redes mais profundas começam a convergir, um problema de degradação é exposto: com o aumento de profundidade da rede , a precisão fica saturada e depois degrada-se rapidamente. Inesperadamente, tal degradação não é causada por sobreajustamento, e a adição de mais camadas a um modelo adequadamente profundo leva a um erro de treino mais elevado, como foi relatado no artigo [19].

A degradação (da precisão do treino) indica que nem todos os sistemas são igualmente fáceis de optimizar. Consideremos uma arquitectura mais rasa e a sua contrapartida mais profunda que acrescenta mais camadas. Existe uma solução para o modelo mais profundo: as camadas adicionadas são o mapeamento da identidade, e os outros estratos são copiados da camada mais rasa modelo. A existência desta solução construída indica que um modelo mais profundo não deve produzir erros de treino superiores do que a sua contraparte menos profunda.

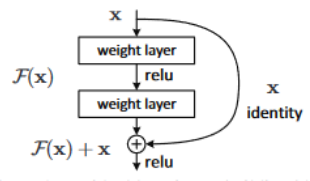


Figura 2.2: Aprendizagem residual: um bloco de construção `https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf`

Neste artigo é abordado o problema da degradação, introduzindo um quadro de aprendizagem residual profundo. Em vez de esperar que cada uma das poucas camadas empilhadas se ajuste directamente a um mapeamento subjacente desejado, deixamos explicitamente que estas camadas se ajustem a conforme um mapeamento residual.

2.4.2 Aprendizagem Residual

Consideremos $H(x)$ como um mapeamento subjacente a ser ajustado por algumas camadas empilhadas (não necessariamente a rede inteira), com x denotando as entradas para as primeiras camadas. Se é possível teorizar que múltiplas camadas não lineares podem aproximar-se assimptoriamente de funções complicadas, então é equivalente fazer a hipótese de que podem aproximar-se assintoticamente das funções residuais, ou seja, $H(x) - x$ (assumindo que as entradas e saídas são das mesmas dimensões). Assim, em vez de esperarmos que as camadas empilhadas aproximem $H(x)$, deixamos que estas se camadas aproximem de uma função residual $F(x) = H(x) - x$. A função original torna-se assim $F(x) + x$. Embora ambas as formas devam ser capazes de aproximar assintoticamente das funções desejadas (como hipótese), a facilidade de aprendizagem pode ser diferente.

Esta reformulação é motivada pelos fenómenos contraintuitivos sobre o problema da degradação . Se as camadas adicionadas puderem ser construídas como mapeamentos de identidade, um modelo com mais profundidade deve ter um erro de formação não superior ao seu homólogo mais superficial. O problema de degradação sugere que os solucionadores podem ter dificuldades em aproximar cartografias de identidade por múltiplas camadas não lineares. Com a reformulação da aprendizagem residual, se os mapeamentos de identidade forem óptimos, os solucionadores podem simplesmente conduzir os pesos das múltiplas camadas não lineares para zero para se aproximarem dos mapeamentos de identidade.

2.4.3 Arquiteturas das redes

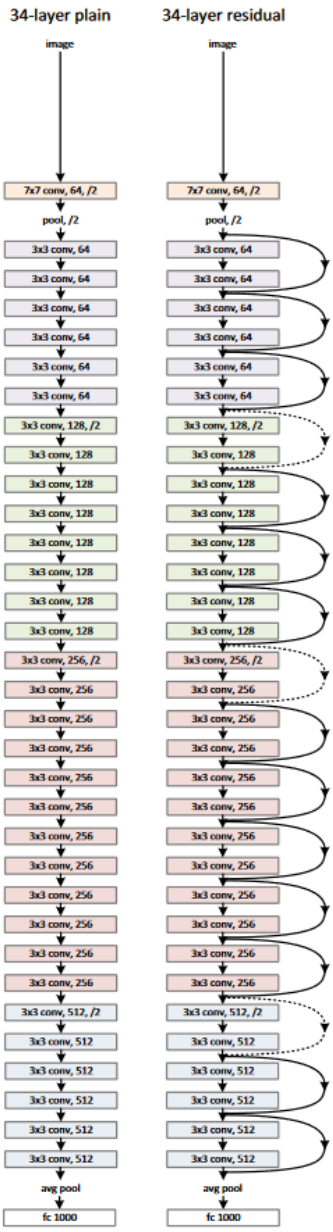


Figura 2.3: 34-camadas simples e 34-camadas residuaishttps://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8

Rede sem blocos residuais: As camadas convolutivas têm na sua maioria filtros 3×3 e seguem duas regras de desenho simples: (i) para o mesmo tamanho de mapa de características de saída, as camadas têm o mesmo número de filtros; e (ii) se o tamanho do mapa das características for reduzido para metade, o número de filtros é duplicado de modo a preservar a complexidade temporal por camada.A amostragem foi utilizada directamente por camadas convolutivas que têm um progresso de valor 2. A rede termina com uma camada média global de agrupamento e um percurso de 1000 camada lidada na totalidade com A função de ativação softmax é usada em redes neurais de classificação. (softmax).O número total das camadas ponderadas são 34 na figura 2.3 (esquerda).

Redes com blocos convuluciaonais: Com base na rede simples acima referida, inserimos ligações de atalho (figura 2.3, à direita) que transformam a rede na sua versão residual homóloga. Os atalhos de identidade podem ser utilizados directamente quando a entrada e saída possuem as mesmas dimensões (atalhos de linha sólida na figura 2.3). Quando as dimensões aumentam (atalhos de linha pontilhada na figura 2.3), consideramos duas opções: (A) O atalho ainda executa o mapeamento de identidade, mas com as dimensões aumentadas. Esta opção não introduz nenhum extra parâmetro; (B) O atalho de projecção é utilizado para dimensões de correspondência (feito por convoluções de 1×1). Para ambas as opções, quando os atalhos atravessam mapas de características de dois são executados com um progresso de valor 2.

2.5 convolutional neural networks(Rede neural convolucional) (CNN) muito profundas para reconhecimento de imagem em grande escala Very Deep Convolutional Networks for Large-Scale Visual Recognition(Redes convulucionais muito profundas para reconhecimento visual em grande escala) (VGG)

2.5.1 Introdução

Nesta secção ,apresenta -se a investigação enunciada no artigo [20], onde se investiga o efeito da profundidade da CNN na sua precisão na configuração

de reconhecimento de imagem em grande escala. A principal contribuição do artigo é uma avaliação minuciosa das redes de crescente profundidade utilizando uma arquitectura com filtros de convolução muito pequenos (3 × 3), o que mostra que uma melhoria significativa sobre as configurações de pré-arte pode ser alcançado empurrando a profundidade para 16-19 camadas de peso.

2.5.2 Arquitetura

Durante o treino, a entrada para as ConvNets é uma imagem de tamanho fixo de 224 × 224 RGB. O único pré-processamento que é feito é subtrair o valor médio RGB, calculado no conjunto de treino, de cada pixel. A imagem é passada através de uma pilha de camadas convolutivas, onde são usados filtros com um campo receptivo muito pequeno: 3 × 3 (que é o tamanho mais pequeno para capturar a noção de esquerda/direita, para cima/baixo, centro). Numa das configurações são utilizados também filtros de convolução de 1 × 1, que podem ser vistos como uma transformação linear dos canais de entrada (seguida de não-linearidade). A etapa de convolução é fixado a 1 pixel; o preenchimento espacial da entrada de camadas convulucionais é tal que a resolução espacial é preservada após a convolução, ou seja, o preenchimento é de 1 pixel para 3 × 3 camadas convulucionais. O agrupamento espacial é realizado por cinco camadas de enchimento máximo, que seguem algumas das camadas convulucionais (nem todas as camadas conv. são seguidas por max-pooling). O Max-pooling é realizado sobre uma janela de 2 × 2 pixels, com o passo 2. Uma pilha de camadas convolutivas (que tem uma profundidade diferente em arquitecturas diferentes) é seguida por três camadas totalmente ligadas: as duas primeiras têm 4096 canais cada uma, a terceira realiza 1000 formas de classificação e, portanto, contém 1000 canais (um para cada classe). A camada final é a camada soft-max.A configuração das camadas totalmente ligadas é a mesma em todas as redes. Todas as camadas ocultas estão equipadas com a rectificação não-linear. Observamos que nenhuma das nossas redes (excepto uma) contém normalização local.

2.5.3 Configuração

São utiliados campos muito pequenos 3 × 3 receptivos em toda a rede, que estão envolvidos com a entrada em cada pixel. É fácil ver que uma pilha de duas camadas de 3 × 3 convulucionais (sem agrupamento espacial no meio) tem um campo receptivo eficaz de 5 × 5; três dessas camadas têm um campo receptivo eficaz de 7 × 7. Então, o que foi ganho com a utilização, por exemplo, de uma pilha de três camadas de 3 × 3 convulucionais em vez de uma única camada de 7 × 7? Primeiro, incorporamos três camadas de rectificação não

linear em vez de uma única, o que torna a função de decisão mais discriminatória. Segundo, diminuímos o número de parâmetros. A incorporação de 1 × 1 camadas convulucionais é uma forma de aumentar a não-linearidade da função de decisão sem afectar os campos receptivos das camadas conv. Ainda que no neste caso a convolução de 1 × 1 seja essencialmente uma projecção linear no espaço da mesma dimensionalidade (o número de canais de entrada e saída é o mesmo), uma não linearidade adicional é introduzido pela função de rectificação.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 2.4: configuração da arquitetura `https://iq.opengenus.org/vgg19-architecture/`

2.6 EfficientNet: Repensar a escala de modelos para redes neurais convolucionais

Nesta secção apresenta -se a rede neural de aprendizagem profunda , usado para o reconhecimento de imagens, segundo o artigo [21].Ao contrário dos outros artigos mencionados que usei como inpiração para o meu projeto, este não aborda o tema de geolocalização, mas aprofunda o estudo sobre redes neurais de aprendizagem profunda , que podem ser muito uteis no âmbito do reconhicimento da localização através de imagens.Neste artigo é utilizada a pesquisa da arquitectura neural para conceber uma nova rede de base e escalá-la para obter uma família de modelos, chamada EfficientNets, que alcança muito melhor precisão e eficiência do que as redes convulucionais anteriores. Em particular, a EfficientNet-B7 atinge uma precisão topo de gama de 84,3% na ImageNet, sendo 8,4 vezes mais pequena e 6,1 vezes mais rápida na inferência do que a melhor rede convulucional existente.

2.6.1 Introdução

A ampliação de redes convulucionais é amplamente utilizada para alcançar uma melhor precisão.No entanto, o processo de expansão da ConvNetshas nunca foi bem compreendido e existem actualmente muitas formas de o fazer. A forma mais comum é escalar ConvNets pela sua profundidade ou largura [22]. Outro método menos comum, mas cada vez mais popular, é escalar os modelos por resolução de imagem [23]. Em trabalhos anteriores, é comum escalar apenas uma das três dimensões - profundidade, largura ou imagem tamanho. Embora seja possível escalar duas ou três dimensões arbitrariamente, a ampliação arbitrária requer uma afinação manual enfadonha e ainda produz frequentemente uma precisão e eficiência sub-óptima.

Neste artigo, o objetivo é estudar e repensar o processo de ampliação de Redes convulucionais.Em particular, é investigada a questão central: existe um método de princípio para aumentar a escala de redes convulucionais que possa alcançar uma maior precisão e eficiência? O nosso estudo empírico mostra que é fundamental equilibrar todas as dimensões de largura, profundidade e resolução da rede, e surpreendentemente tal equilíbrio pode ser alcançado simplesmente escalando cada uma delas com uma relação constante. Com base nesta observação, é proposto um método simples mas eficaz de escalonamento composto. Ao contrário da prática convencional que escalona arbitrariamente estes factores, este método escala uniformemente a largura, profundidade e resolução da rede com um conjunto de coeficientes de escala fixos.Intuitivamente, o método de escala composto faz sentido

porque se a imagem de entrada for maior, então a rede precisa de mais camadas para aumentar o campo receptivo e mais canais para captar padrões mais refinados na imagem maior.

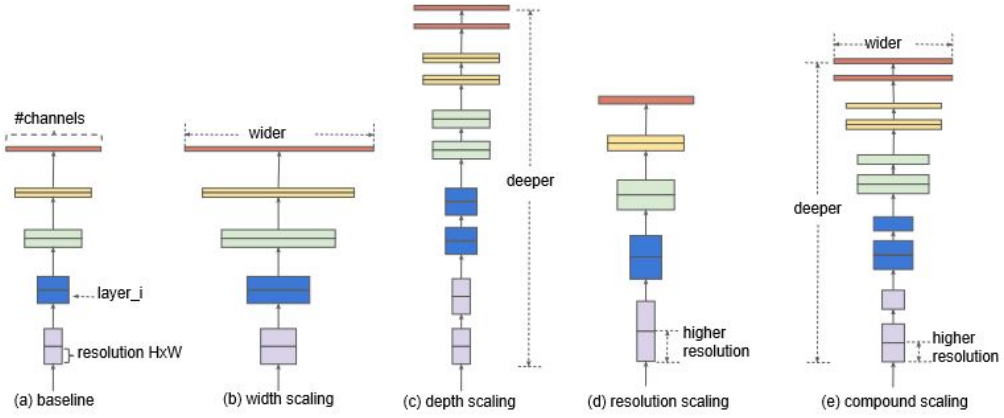


Figura 2.5: Escala do modelo `https://arxiv.org/pdf/1905.11946.pdf`

2.6.2 Escalar as dimensões

A principal dificuldade do problema é que o valor óptimo da profundidade, largura e resolução dependem uns dos outros e os valores mudam sob diferentes restrições dos recursos. Devido a esta dificuldade, os métodos que, na sua maioria, escalam as redes convulucionais numa destas dimensões:

Profundidade: escalar a profundidade da rede é a forma mais comum utilizada por muitas redes convulucionais . A intuição é que a Rede convulucional mais profunda pode capturar características mais ricas e mais complexas, e generalizar bem em novas tarefas. No entanto, as redes mais profundas são também mais difíceis de treinar, devido ao problema do gradiente de desaparecimento. Embora várias técnicas, tais como saltar ligações e normalização de lotes, aliviem o problema de formação, o ganho de precisão de redes muito profundas diminui: por exemplo, a ResNet-1000 tem uma precisão semelhante à da ResNet-101, apesar de ter muito mais camadas.

Largura: escalar a largura da rede è normalmente utilizado para modelos de pequenas dimensões. Redes mais largas tendem a ser capazes de captar características mais finas e são mais fáceis de treinar. Contudo, redes extremamente largas mas pouco profundas tendem a ter dificuldades em captar características de nível mais elevado

Resolução: Com imagens de entrada de alta resolução, ConvNets pode potencialmente capturar padrões mais finos. No início ,as redes convulucio-

nais, usavam imagem com 224x244 de rosulução, redes convulucionais modernos tendem a usar 299x299 ou 331x331 para uma melhor precisão. Resoluções superiores, tais como 600x600, são também amplamente utilizados na detecção de objectos.

O aumento de qualquer dimensão da largura, profundidade ou resolução da rede melhora a precisão, mas o ganho de precisão diminui para modelos maiores. A fim de perseguir uma melhor precisão e eficiência, é fundamental equilibrar todas as dimensões da rede largura, profundidade e resolução durante a ampliação da rede convulucional.

2.7 Conclusões

No final concluo que toda a informação recolhida aqui foi de extrema importância para a conclusão do projeto.

Capítulo

3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Neste capítulo vou abordar e explicar detalhadamente todas as tecnologias e ferramentas que usei durante a fase de implementação e de testagem. Destas ferramentas destaca-se a linguagem de programação Python e as bibliotecas Tensorflow e Keras. Vou também elaborar os temas de Deep Learning (DL) e CNN

3.2 Python

As tecnologias de Intelegência artificial (IA) e a Machine Learning (ML), tornaram-se ultimamente um dos tópicos mais importantes no mundo informático. Uma forma de colocar estes assuntos em prática é usar a linguagem de programação Python.

Uma enorme vantagem do Python é que, em comparação com outras linguagens de programação, já existem enormes quantidades de bibliotecas e a linguagem está disponível para aqueles que nunca escreveram qualquer código para IA.A sintaxe de Python é limpa e o código é bem estruturado. Para algumas pessoas, pode parecer lento, mas o truque principal é que a maioria dos algoritmos exigidos para a programação já foram escritos.

Exemplo de bibliotecas Pyhton úteis para IA e a ML:

Matplotlib: A biblioteca Matplotlib foi concebida para criar visuais científicos. Pode construir um monte de gráficos diferentes, e controlar total-

mente tudo o que pode exibir.

Numpy: Uma vez que trabalhar para criar IA é sempre trabalhar com muitos dados, Numpy em Python será uma grande ajuda. Muitas bibliotecas valiosas operam com o Numpy, por exemplo, análise estatística e visualização de bibliotecas devido ao seu incrível desempenho e velocidade. A menos que tenha um problema relacionado com o desempenho, a aprendizagem do NumPy vale certamente o esforço. Usando NumPy, beneficia de dados convolucionais, histogramas, estatísticas básicas, pesquisas rápidas, álgebra linear, etc.

Scikit-learn: Esta é exactamente a vantagem de uma linguagem conveniente para a programação de IA. Inclui vários algoritmos de classificação, regressão, e agrupamento e permite a interacção com outras bibliotecas de modelação numérica, tais como Pandas, NumPy, e Scipy. As principais vantagens do Scikit-learn são consideradas a sua interface de fácil utilização, muitos algoritmos avançados, documentação detalhada, e integração com outras bibliotecas Python. Por conseguinte, é activamente utilizado em muitos projectos científicos e comerciais.

Tensorflow: O Tensorflow é normalmente utilizado em CNNs e de DL. Contém aulas prontas, informação sobre neurónios, e algoritmos para a sua aprendizagem, o que é muito conveniente quando se trabalha com grandes volumes de dados. É uma plataforma ML flexível para investigação e experimentação com uma interface intuitiva. Permite escrever e depurar imediatamente o código linha a linha, utilizando ferramentas Python padrão.

Esta linguagem é ideal para trabalhar com grandes quantidades de dados, tem o mais amplo conjunto de pacotes e extensões para automatizar o trabalho. Apesar da sua acessibilidade, Python é usado para resolver problemas industriais devido à sua rica disposição, boa estrutura, e modularidade.

Resumindo, Python é a melhor linguagem de programação quando se trata de ML. Usando numerosas bibliotecas desta linguagem, até mesmo os principiantes conseguem desenvolver Inteligência Artificial.

3.3 Tensorflow

O TensorFlow é uma plataforma completa de código aberto para ML. O Tensorflow tem um ecossistema abrangente e flexível de ferramentas, bibliotecas e recursos da comunidade que permite aos investigadores levar adiante

as ML de última geração e aos desenvolvedores criar aplicações com ML.A Application Programming Interface (API) Keras de alto nível que faz parte da plataforma é usada para criar e treinar modelos.

O Keras é uma biblioteca de rede neural de código aberto escrita em Python.As funcionalidades do keras permitem uma experimentação rápida com redes neurais profundas, fácil de usar, modular e extensível.O Keras contém várias funções para construir partes importantes de redes neurais, como camadas, funções de perda, funções de ativação, otimizadores, entre outras.Além das redes neurais padrão, Keras tem suporte para CNN. Ele suporta outras camadas comuns, como camadas de drop-out, normalização em lote e pooling. Também permite distribuir o treinamento dos algoritmos em unidades de processamento gráfico (GPU) e unidades de processamento tensorial (TPU) principalmente em conjunto com a Compute Unified Device Architecture (CUDA).

3.4 Machine Learning

A ML é um tipo de IA) que permite as aplicações de software tornem - se mais precisas na previsão de resultados sem serem explicitamente programadas para o fazer. Os algoritmos de ML utilizam dados históricos como entrada para prever novos valores de saída.

3.4.1 Quais são os diferentes tipos de ML

A ML clássica é frequentemente categorizada pela forma como um algoritmo aprende a tornar-se mais preciso nas suas previsões.Existem quatro abordagens básicas: aprendizagem supervisionada, aprendizagem não supervisionada, aprendizagem semi-supervisionada e aprendizagem por reforço.

Aprendizagem supervisionada: Neste tipo de ML, os programadores fornecem algoritmos com dados de formação rotulados e definem as variáveis que querem que o algoritmo avalie para correlações. Tanto a entrada como a saída do algoritmo são especificadas.

Aprendizagem não supervisionada: Este tipo de ML envolve algoritmos que treinam sobre dados não etiquetados. O algoritmo efectua scans através de conjuntos de dados à procura de qualquer ligação significativa. Os dados sobre os quais os algoritmos treinam, bem como as previsões ou recomendações que emitem, são pré-determinados.

Aprendizagem semi-supervisionada: Esta abordagem à ML envolve uma mistura dos dois tipos precedentes. Os cientistas de dados podem alimentar um algoritmo na sua maioria com dados de formação rotulados, mas

o modelo é livre de explorar os dados por si só e desenvolver a sua própria compreensão do conjunto de dados.

Reforço da aprendizagem: Os desenvolvedores utilizam tipicamente a aprendizagem de reforço para ensinar uma máquina a completar um processo em várias etapas, para o qual existem regras claramente definidas. Os cientistas de dados programam um algoritmo para completar uma tarefa e dão-lhe indicações positivas ou negativas, à medida que se vai descobrindo como completar uma tarefa. Mas, na sua maioria, o algoritmo decide por si mesmo quais os passos a dar ao longo do processo.

3.4.2 Como funciona a aprendizagem supervisionada?

A aprendizagem supervisionada de máquinas requer que o cientista de dados treine o algoritmo tanto com entradas rotuladas como com as saídas desejadas. Os algoritmos de aprendizagem supervisionada são bons para as seguintes tarefas:

Classificação binária: Dividindo os dados em duas categorias.

Classificação multi-classe: Escolhendo entre mais de dois tipos de respostas.

Modelação de regressão: Previsão de valores contínuos.

Ensembling: Combinando as previsões de múltiplos modelos de ML para produzir uma previsão precisa.

3.4.3 Como funciona a aprendizagem não supervisionada?

Os algoritmos de aprendizagem não supervisionada não requer que os dados sejam etiquetados. Eles filtram os dados não rotulados para procurar padrões que possam ser utilizados para agrupar pontos de dados em subconjuntos. A maioria dos tipos de DL, incluindo as redes neurais, são algoritmos não supervisionados. Os algoritmos de aprendizagem não supervisionada são bons para as seguintes tarefas:

Agrupamento: Dividir o conjunto de dados em grupos com base na semelhança.

Detecção de anomalias: Identificação de pontos de dados invulgares num conjunto de dados.

Associação de mineração: Identificação de conjuntos de itens de um conjunto de dados que ocorrem frequentemente em conjunto

Redução da dimensionalidade: Redução do número de variáveis de um conjunto de dados.

3.4.4 Como funciona a aprendizagem semi-supervisionada?

A aprendizagem semi-supervisionada funciona através de cientistas de dados que alimentam uma pequena quantidade de dados de formação rotulados com um algoritmo. A partir disto, o algoritmo aprende as dimensões do conjunto de dados, que pode então ser aplicado a dados novos, não rotulados. O desempenho dos algoritmos melhora tipicamente quando treinam em conjuntos de dados etiquetados. Mas a etiquetagem de dados pode ser demorada e dispendiosa. A aprendizagem semi-supervisionada atinge um ponto intermédio entre o desempenho da aprendizagem supervisionada e a eficiência da aprendizagem não supervisionada. Algumas áreas onde a aprendizagem semissupervisionada é utilizada incluem:

Tradução automática: Algoritmos de ensino para traduzir linguagem baseados em menos do que um dicionário completo de palavras.

Detecção de fraudes: Identificação de casos de fraude quando se tem apenas alguns exemplos positivos.

Rotulagem de dados: Algoritmos treinados em pequenos conjuntos de dados podem aprender a aplicar automaticamente etiquetas de dados a conjuntos maiores.

3.4.5 Como é que a aprendizagem de reforço funciona?

O reforço da aprendizagem funciona através da programação de um algoritmo com um objectivo distinto e um conjunto de regras prescritas para a realização desse objectivo. Os cientistas de dados também programam o algoritmo para procurar recompensas positivas - que recebe quando executa uma acção que é benéfica para o objectivo final - e evitar castigos - que recebe quando executa uma acção que o afasta mais do seu objectivo final. O reforço da aprendizagem é frequentemente utilizado em áreas como:

Robótica: Os robôs podem aprender a executar tarefas no mundo físico utilizando esta técnica.

Jogabilidade de videojogos: A aprendizagem de reforço tem sido utilizada para ensinar os robots a jogar uma série de jogos.

Gestão de recursos: Dados os recursos finitos e um objectivo definido, a aprendizagem de reforço pode ajudar as empresas a planear como atribuir recursos.

3.4.6 Como escolher o modelo de ML

O processo de escolha do modelo de ML para resolver um problema pode ser demorado se não for abordado estrategicamente.

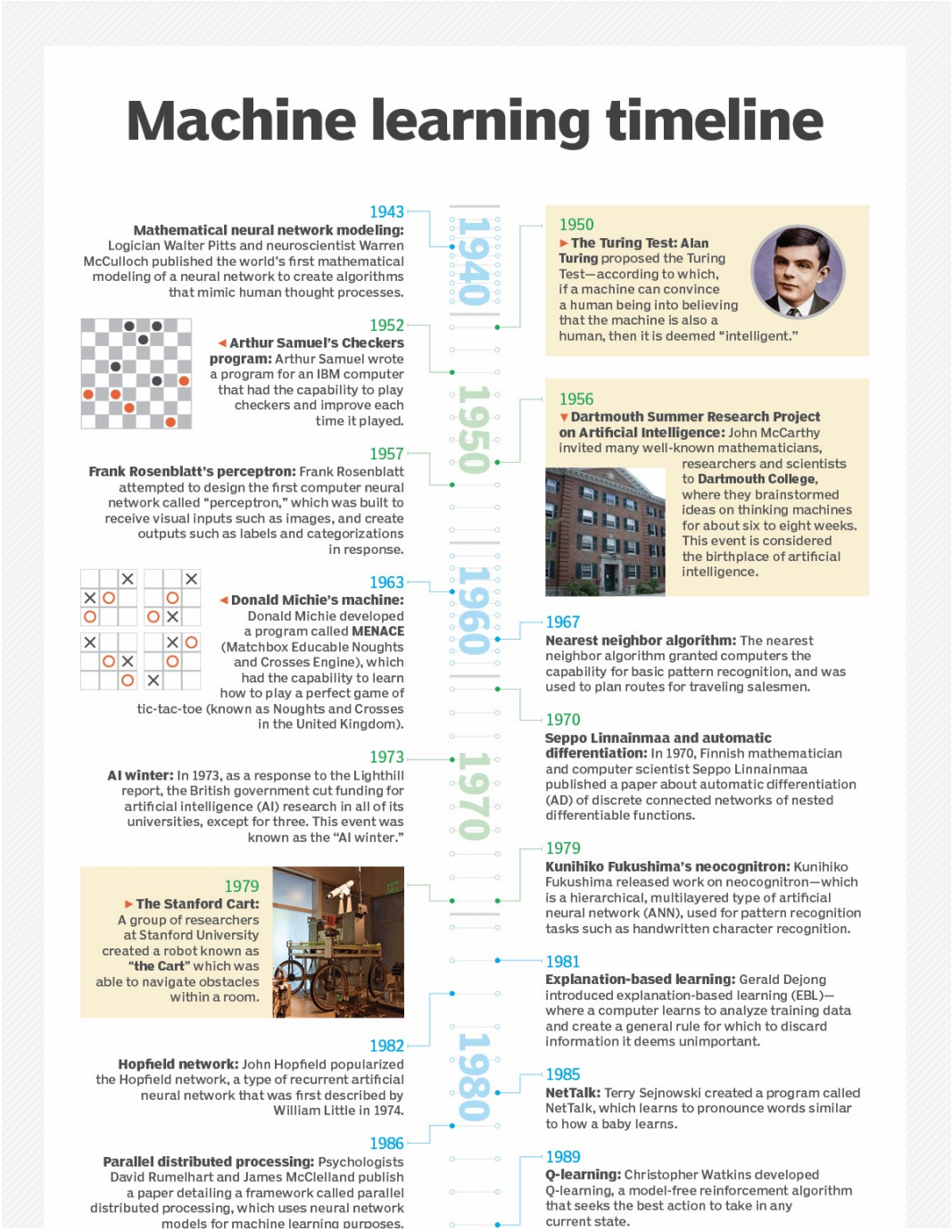
Passo 1: Alinhar o problema com potenciais entradas de dados que devem ser consideradas para a solução. Esta etapa requer a ajuda de cientistas de dados e peritos que tenham uma compreensão profunda do problema.

Etapa 2: Recolher dados, formatá-los e rotulá-los, se necessário. Esta etapa é normalmente conduzida por cientistas de dados, com a ajuda de quem se ocupa de dados.

Etapa 3: Escolher o(s) algoritmo(s) a utilizar e testar para ver o seu bom desempenho. Esta etapa é normalmente levada a cabo por cientistas de dados.

Etapa 4: Continuar a afinar os resultados até atingirem um nível aceitável de precisão. Esta etapa é normalmente executada por cientistas de dados com feedback de peritos que têm uma compreensão profunda do problema.

3.4.7 Como é que a ML evoluiu?



3.4.8 Como é que a ML evoluiu?

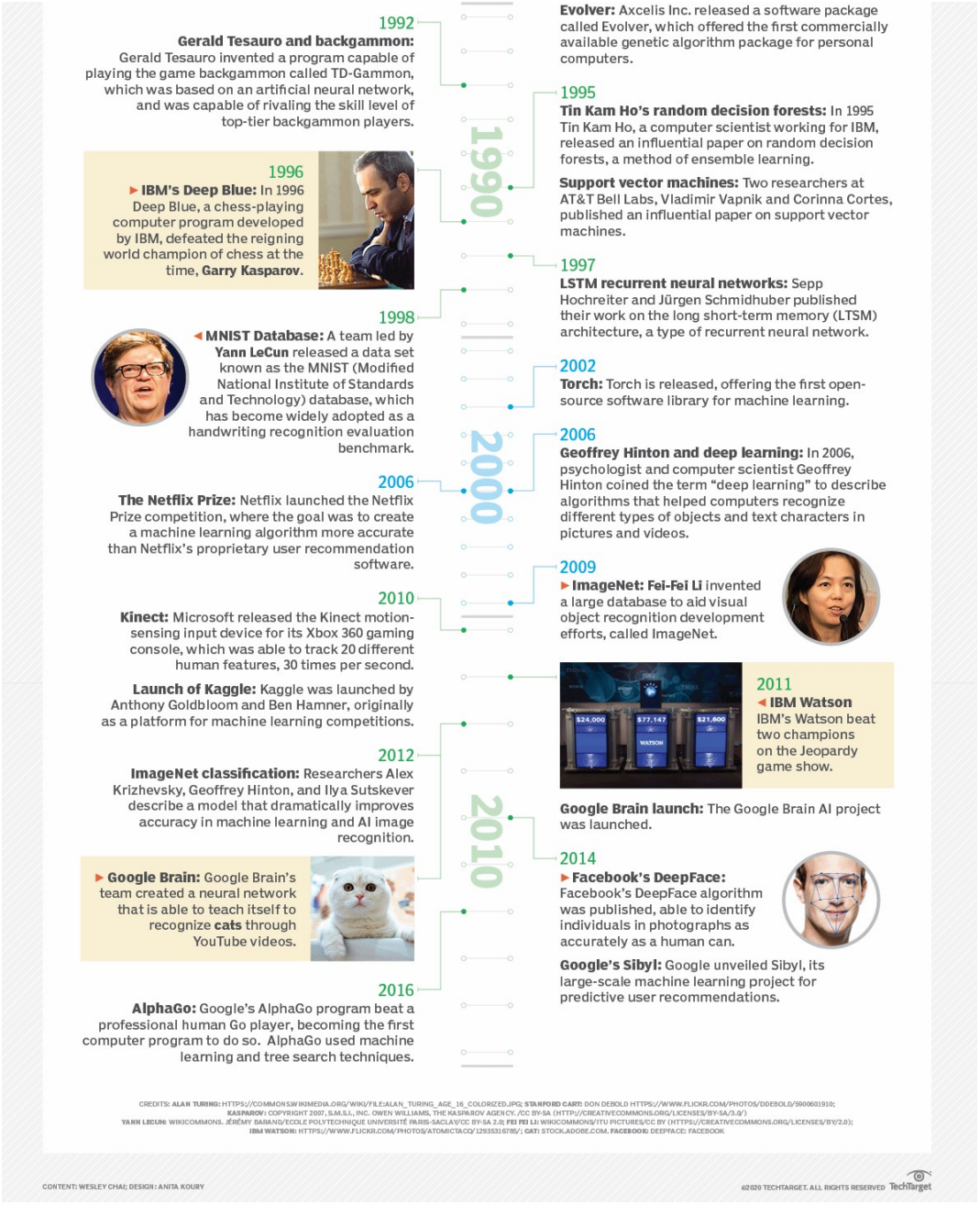


Figura 3.1: Evolução das ML `https://cdn.ttgtmedia.com/rms/onlineimages/whatis-machine_learning_timeline-i.png`

3.5 Deep Learning

DL tenta imitar o cérebro humano - embora longe de corresponder à sua capacidade - de agregar dados e fazer previsões com uma precisão incrível.

3.5.1 O que é DL

A DL é um subconjunto da ML, que é essencialmente uma rede neural com três ou mais camadas. Estas redes neuronais tentam simular o comportamento do cérebro humano - embora longe de corresponder à sua capacidade - permitindo-lhe "aprender"com grandes quantidades de dados. Enquanto uma rede neural com uma única camada pode ainda fazer previsões aproximadas, camadas adicionais ocultas podem ajudar a optimizar e aperfeiçoar para uma maior precisão.

3.5.2 DL vs ML

Se a DL é um subconjunto da ML, como é que diferem? A DL distingue-se da ML clássica pelo tipo de dados com que trabalha e pelos métodos com os quais aprende.

Os algoritmos de ML utilizam dados estruturados e rotulados para fazer previsões - o que significa que as características específicas são definidas a partir dos dados de entrada para o modelo e organizadas em tabelas. Isto não significa necessariamente que não utiliza dados não estruturados; significa apenas que se o fizer, geralmente passa por algum pré-processamento para o organizar num formato estruturado.

A DL elimina alguns dos pré-processamentos de dados que estão tipicamente envolvidos na ML. Estes algoritmos podem ingerir e processar dados não estruturados, como texto e imagens, e automatiza a extracção de características, removendo alguma da dependência de peritos humanos. Por exemplo, digamos que tínhamos um conjunto de fotos de diferentes animais de estimação, e queríamos categorizar por "gato", "cão", "hamster", et cetera. Algoritmos de DL podem determinar que características (por exemplo, orelhas) são mais importantes para distinguir cada animal de outro. Na ML, esta hierarquia de características é estabelecida manualmente por um perito humano.

Depois, através dos processos de descida por gradiente e retropropagação, o algoritmo de DL ajusta-se e ajusta-se à precisão, permitindo-lhe fazer previsões sobre uma nova fotografia de um animal com maior precisão.

A ML e os modelos de DL também são capazes de diferentes tipos de aprendizagem, que são geralmente categorizados como aprendizagem supervisionada, aprendizagem não supervisionada, e aprendizagem de reforço.

3.5.3 Como funciona uma DL

Redes neurais de aprendizagem profunda, ou redes neurais artificiais, tenta imitar o cérebro humano através de uma combinação de entradas de dados, pesos, e enviesamento. Estes elementos trabalham em conjunto para reconhecer, classificar, e descrever com precisão os objectos dentro dos dados.

As redes neurais profundas consistem em múltiplas camadas de nós interligados, cada uma construindo sobre a camada anterior para refinar e optimizar a previsão ou categorização. Esta progressão dos cálculos através da rede é chamada propagação para a frente. As camadas de entrada e saída de uma rede neural profunda são chamadas camadas visíveis. A camada de entrada é onde o modelo de aprendizagem profunda ingere os dados para processamento, e a camada de saída é onde a previsão ou classificação final é feita.

Outro processo chamado retropropagação utiliza algoritmos, como a descida de gradiente, para calcular erros nas previsões e depois ajusta os pesos e enviesamentos da função, deslocando-se para trás através das camadas, num esforço para treinar o modelo. Juntos, a propagação para a frente e a retropropagação permitem a uma rede neural fazer previsões e corrigir quaisquer erros em conformidade. Com o tempo, o algoritmo torna-se gradualmente mais preciso.

O acima descrito descreve o tipo mais simples de rede neural profunda nos termos mais simples. No entanto, os algoritmos de aprendizagem profunda são incrivelmente complexos, e existem diferentes tipos de redes neurais para abordar problemas específicos ou conjuntos de dados. Por exemplo,

As CNNs, utilizadas principalmente em aplicações informáticas de visão e classificação de imagens, podem detectar características e padrões dentro de uma imagem, permitindo tarefas, como a detecção ou reconhecimento de objectos. Em 2015, uma CNN venceu pela primeira vez um humano num desafio de reconhecimento de objectos.

As Recurrent neural network (RNN) são tipicamente utilizadas em aplicações de linguagem natural e de reconhecimento da fala, uma vez que aproveitam dados sequenciais ou de séries temporais.

3.6 Conclusões

Toda a informação recolhida durante este capítulo, foi vital para a implementação do projeto.

Capítulo

4

Implementação e Testes

4.1 Introdução

Neste capítulo aborda-se a componente prática desenvolvida especificando quais as tecnologias usadas, os trabalhos seguidos e a sua implementação.Embora a estimativa bem sucedida da geolocalização de uma fotografia permita uma série de aplicações interessantes, é também um grande desafio.

Neste projeto vai considerar-se este problema, como um problema de classificação de imagens em que cada imagem tem como rótulo um par coordenadas(latitude , longitude). Este modelo vai ser apenas capaz de prever a localização de uma zona especifica, neste caso essa zona vai ser a covilhã, dado uma foto que sabemos que foi tirada na covilhã o modelo de classificação de imagens vai ser capaz de devolver com precisão a geolocalização da imagem em forma de um par de coordenadas(latitude,longitude).

4.2 Conjunto de dados

Um dos maiores obstáculos a ultrapassar neste projeto foi adquirir um conjunto de imagens grande o suficiente com informação geográfica(latitude e longitude) para poder treinar os modelos. A ideia inicial era utilizar as milhares de fotos que a Google utiliza na sua aplicação Google maps, mas a utilização destas imagens fora do contexto da aplicação não é permitida pela empresa. Devido à escassez de conjuntos de dados que contêm imagens com informação geográfica relativa à sua latitude e longitude , abertas para o público , decidiu -se montar um conjunto de dados manualmente, utilizando um telemóvel e uma aplicação que armazena nas fotos as coordenadas de latitude e longitude de onde estas mesmas foram tiradas. Após alguns dias a

recolher fotos acabou-se com um "pequeno"conjunto de dados constituído por 2967 fotos do centro da Covilhã e arredores. Considero este conjunto pequeno pois segundo o capitulo 2 os modelos propostos em todos os artigos utilizaram conjuntos de dados gigantescos(milhares, milhões de imagens).

Figura 4.1: Imagem do conjunto de dados, Exemplo 1

Figura 4.2: Imagem do conjunto de dados, Exemplo 2

Figura 4.3: Imagem do conjunto de dados, Exemplo 3

Após o conjunto de dados estar reunido o próximo passo era agrupar as imagens por classes/rótulos para o modelo poder vir a classificar cada imagem consoante a sua classe, esse rótulo vai ser o conjunto de par de cordenadas que cada imagem representa. Como ao utilizar o conjunto de par de coodernadas completo ia acabar com apenas uma imagem por classe pois foi necessário arredondar as coodernadas a 4 casas décimais. Por exemplo uma imagem que foi tirada no par de coordenadas: (40.28054404973569, -7.504332847557753) vai ser representado pela classe das coordenadas(40.2805, -7.5043). Embora esta solução ajude com o problema do conjunto de dados não estar equilibrado vai eleminar um pouco de precisão à previsão do modelo pois um conjunto de coodernadas reduzido a 4 casas decimais vai abranger uma área circular com 10 metros de raio(se reduzir para 3 casas decimais vai abranger uma área circular com 100m de raio).

Destas 2967 imagens foram retiradas 228 para formar o conjunto de teste, este conjunto de teste vai se manter igual ao longo do período de triagem dos modelos formados , as restantes imagens vão ser usadas para constituir o grupo de treino e validação. Estes ultimos não irão ser iguais ao longo dos testes mas irão ser semelhantes, pois antes de começar o treino, o modelo escolhe 80% das 2739 imagens restantes, de forma aleatória, para formar o grupo de treino e as 20% que sobram para desenvolver o grupo de validação.

Como foi referido anteriormente nesta secção , a base de dados recolhida pessoalmente é relativamente pequena face a outros modelos que tentam resolucionar problemas semelhantes. Uma forma de combater este obstáculo é utilizar "image augmentation"(aumento das imagens em português) nas imagens do conjunto de treino. O aumento das imagens é adquirir imagens "di-

ferentes"a partir da imagem original através de transformações , como por exemplo rotação , translação , alteração da luminosidade na imagem entre outras.

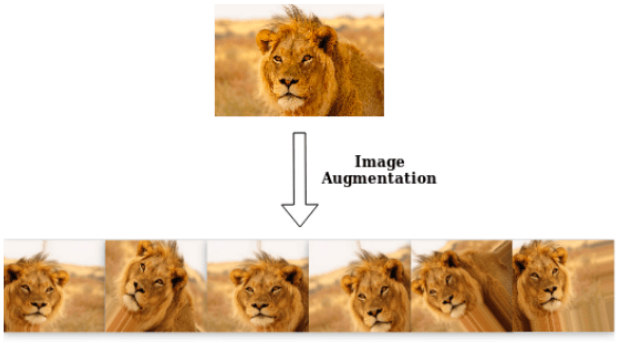


Figura 4.4: Image augmentation, exemplo `https://towardsdatascience.com/machinex-image-data-augmentation-using-keras-b459ef87cd22`

4.3 Modelos Iniciais

Nesta face inicial do projeto optou-se por desenvolver 3 modelos com arquiteturas diferentes para treinar o conjunto de dados , e no final do treino de cada um desses modelos,os resultados obtidos foram comparados entre si e foi escolhido o modelo que apresentava melhor resultados e maior potencial para resolver o problema proposto neste projeto, para ser afinado e otimizado.

Das arquitetura apresentadas no 2.1 decidiu - se formular modelos com base nas arquiteturas: ResNet 2.4, VGG 2.5 e Efficentnet 2.6.

4.3.1 ResNet152

Neste modelo vai-se utilizar uma rede residual constituida por 152 camadas residuais 2.4.

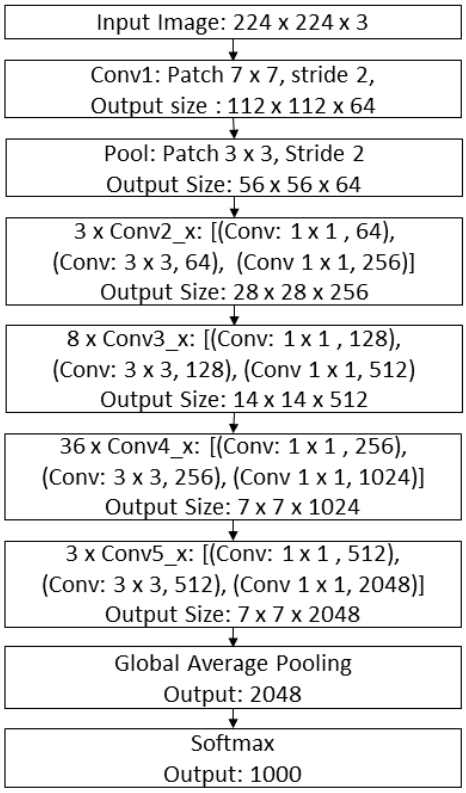


Figura 4.5: Arqitetura das ResNet152

https://www.researchgate.net/figure/ResNet152-v2-Architecture-and-its-Residual-Unit_fig5_345757036

A imagem de 224x244 com três canais de cor é apresentada como o input.A primeira camada é a camada convolucional com 64 filtros diferentes para cada canal. O tamanho da primeira *filter layer* é 7x7, usando uma passada de 2 tanto na direção horizontal como na vertical. As caraterísticas dos mapas resultantes são:(i)passagem pela ReLU(não mostrado),(ii) max pooled (dentro de regiões 3x3, usando passada 2) e (iii) contraste normalizado nos mapas de recursos caraterísticos para obter 64 mapas diferentes de 56x56.Operações semelhantes mas de diferentes dimensões/filtros são repetidas nas seguintes camadas. As últimas duas camadas estão totalmente conectadas, sendo o penúltimo *layer* o seu input. A layer final é uma função softmax, com um output diferente para cada uma das 1484 classes existentes no conjunto de dados. Todos os filtros e mapas de recurso caraterísticos são quadrados. Para além deste comportamento entre todas as camadas do modelo é aplicada uma função de aprendizagem residual enunciado na sub-secção 2.4.2 e ilustrado pela figura 2.2. O modelo vai usar 2798 imagens, repartidas por 1484 classes diferentes , para treinar e 113 para validar, estas imagens já quadradas vão ser redimensionadas para 224x224 pixeis, estas imagens vão ser passadas para o modelo em lotes(batches) de 19 unidades durante 500 épocas.O modelo vai ser otimizado utilizando o otimizador Adagrad com o valor de taxa de aprendizagem 0.002 e o valor acumulador de 0.1 e utliza-se como valor métricos a precisão , a precisão das top 5 previsões do modelo e a perda do modelo.Obtêm-se desta forma os seguintes gráficos após o treino:

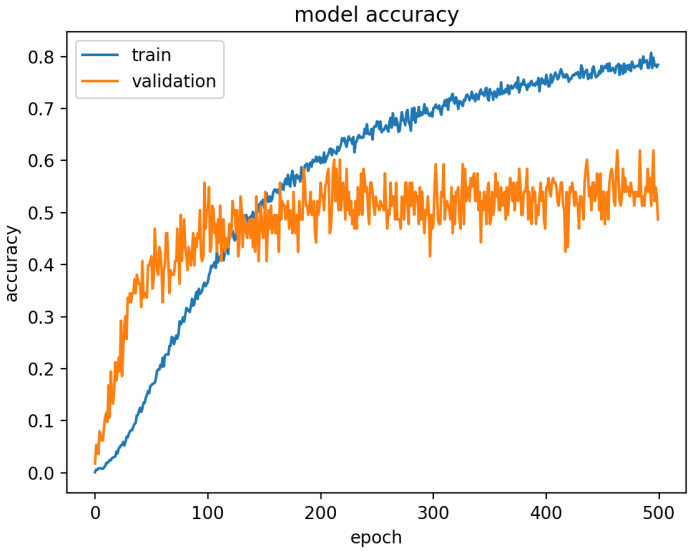


Figura 4.6: Resnet152 precisão

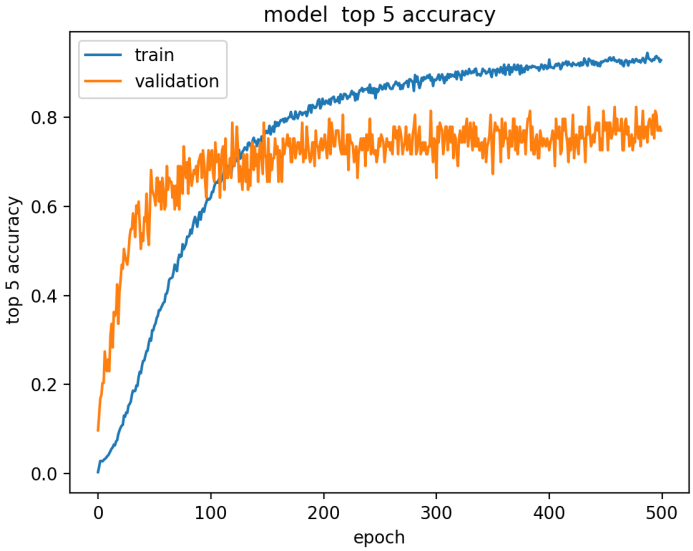


Figura 4.7: Resnet152 precisão das top5 previsões

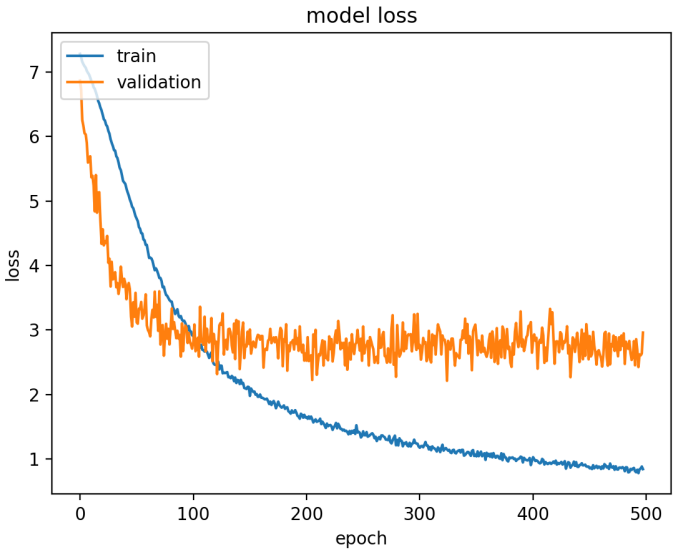


Figura 4.8: Resnet152 perda

Dos gráficos pode-se concluir que as linhas referentes aos valores obtidos do treino estão a comportar-se como esperado , a cada época a precisão aumenta e o valor da perda diminui.

Distancia	Percentagem
5< metros	0.00%
10< metros	0.88%
20< metros	1.76%
50< metros	4.41%
100< metros	7.93%
250< metros	18.06%
500< metros	41.85%
800< metros	65.20%
1000< metros	81.06%
1500< metros	95.59%

Tabela 4.1: Resultados do modelo ResNet152 , **Desvio Padrão: 413.53 metros**

Os valores obtidos do grupo de validação são deveras interessante, até à época 100 a linha do grupo de validação deixa de acompanhar a linha de treino e os valores de precisão e perda deixam de variar significamente o que indica que o modelo a partir da época 100 deixa de aprender padrões do conjunto de treino para conseguir generalizar melhor as imagem novas, introduzidas ao modelo no conjunto de validação , estagnando o seu valor aproximadamente em 0.45(45%) na precisão,0.7(70%) na precisão das top5 previsões e 3 na perda.

Para a avaliação final, testa-se o modelo face ao conjunto de teste , o modelo vai prever a coordenadas da imagem e de seguida vai compara -las às coordenadas reais da imagem e por conseguinte mede-se a distancia entre a coordenadas devolvidas pelo modelo e as coordenadas reais da imagem:

4.3.2 VGG19

Neste modelo vai-se utilizar uma rede VGG com 19 camadas de profundidade 2.5.

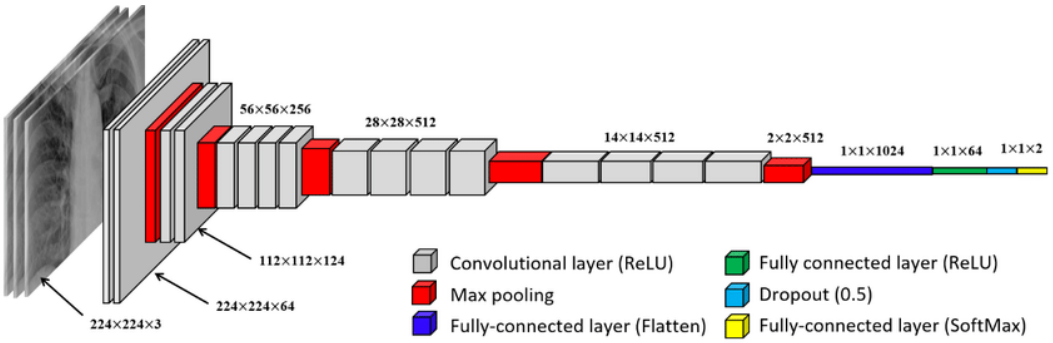


Figura 4.9: Arqitetura da VGG19 `https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-fig3_322512435`

A imagem de 224x244 com três canais de cor é apresentada como o input.A primeira camada é a camada convolucional com 64 filtros diferentes para cada canal. O tamanho da primeira *filter layer* é 7x7, usando uma passada de 2 tanto na direção horizontal como na vertical. As caraterísticas dos mapas resultantes são:(i)passagem pela ReLU(não mostrado),(ii) max pooled (dentro de regiões 3x3, usando passada 2) e (iii) contraste normalizado nos mapas de recursos caraterísticos para obter 64 mapas diferentes de 56x56.Operações semelhantes mas de diferentes dimensões/filtros são repetidas nas seguintes camadas. As últimas duas camadas estão totalmente conectadas, sendo o penúltimo *layer* o seu input. A layer final é uma função softmax, com um output diferente para cada uma das 1484 classes existentes no conjunto de dados. Todos os filtros e mapas de recurso caraterísticos são quadrados.

O modelo vai usar 2798 imagens, repartidas por 1484 classes diferentes , para treinar e 113 para validar, estas imagens já quadradas vão ser redimensionadas para 224x224 pixeis, estas imagens vão ser passadas para o modelo em lotes(batches) de 19 unidades durante 500 épocas.O modelo vai ser otimizado utilizando o otimizador Adagrad com o valor de taxa de aprendizagem 0.002 e o valor acumulador de 0.1 e vamos utlizar como valor métricos a precisão , a precisão das top 5 previsões do modelo e a perda do modelo. Obtêm - se desta forma os seguintes gráficos após o treino:

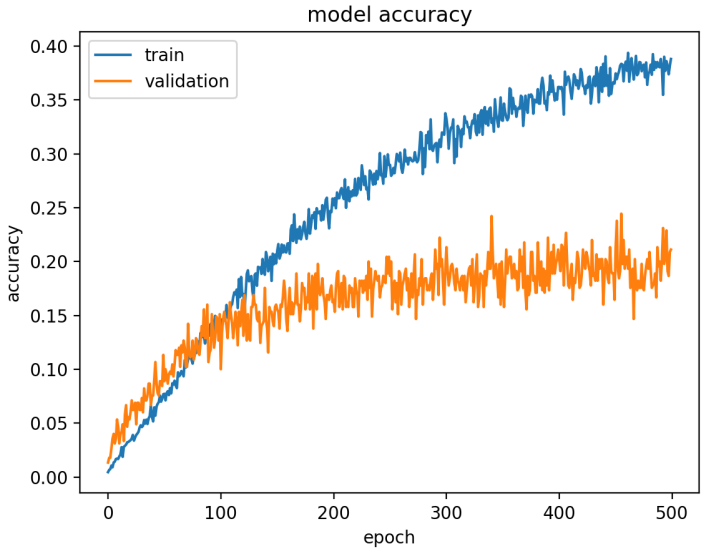


Figura 4.10: VGG19 precisão

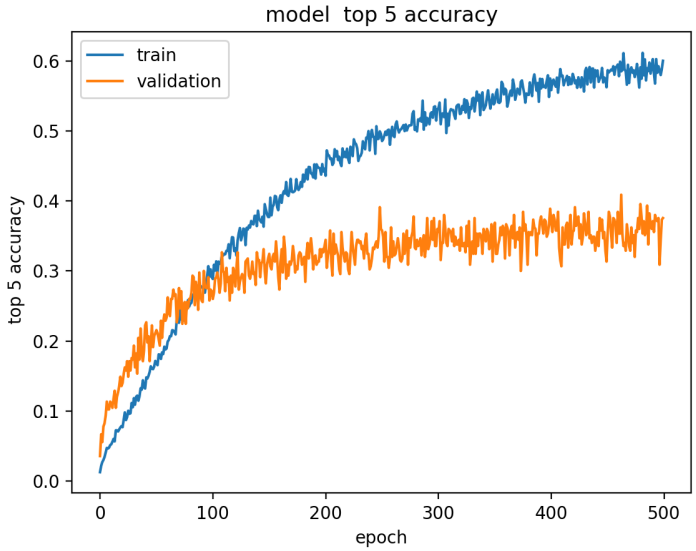


Figura 4.11: VGG19 precisão das top5 previsões

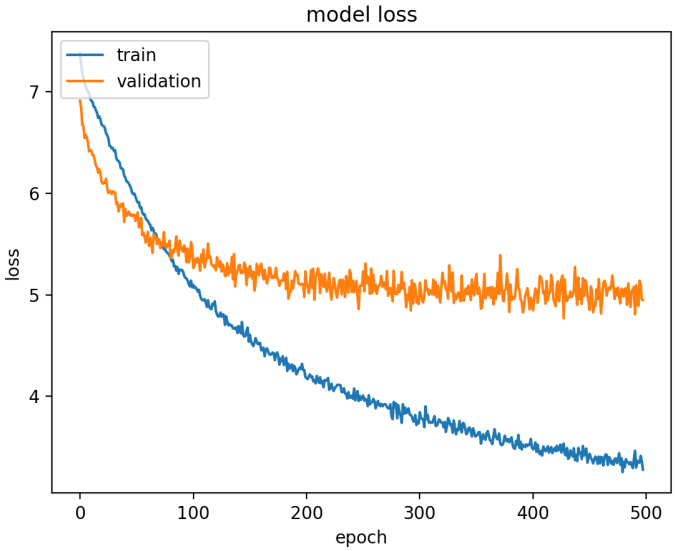


Figura 4.12: VGG19 perda

Dos gráficos pode-se concluir que as linhas referentes aos valores obtidos do treino estão a comportar-se como esperado , a cada época o a precisão aumenta e o valor da perda diminui.

Os valores obtidos do grupo de validação são deveras interessante, até á época 100 a linha do grupo de validação deixa de acompanhar a linha de treino e os valores de precisão e perda deixam de variar significamente o que indica que o modelo a partir da época 100 deixa de aprender padrões do conjunto de treino para conseguir generalizar melhor as imagem novas introduzidas ao modelo no conjunto de validação , estagnando a o seu valor aproximadamente em 0.15(15%) na precisão,0.3(30%) na precisão das top5 previsões e 4 na perda.

Para a avaliação final, testa - se o modelo face ao conjunto de teste , o modelo vai prever a coordenadas da imagem e de seguida vai compara -las às coodenadas reais da imagem e de seguida mede-se a distancia entre a coordenadas devolvidas pelo modelo e as coodernadas reais:

Distancia	Percentagem
5< metros	14.54%
10< metros	20.26%
20< metros	23.79%
50< metros	27.75%
100< metros	33.92%
250< metros	42.73%
500< metros	58.59%
800< metros	75.77%
1000< metros	82.38%
1500< metros	95.59%

Tabela 4.2: Resultados do modelo VGG19 , **Desvio Padrão: 517.30 metros**

4.3.3 *EfficientNetB2*

Neste modelo vai-se utilizar uma rede da familia *EfficientNet*, mais concretamente a *EfficientNetB2* 2.5.

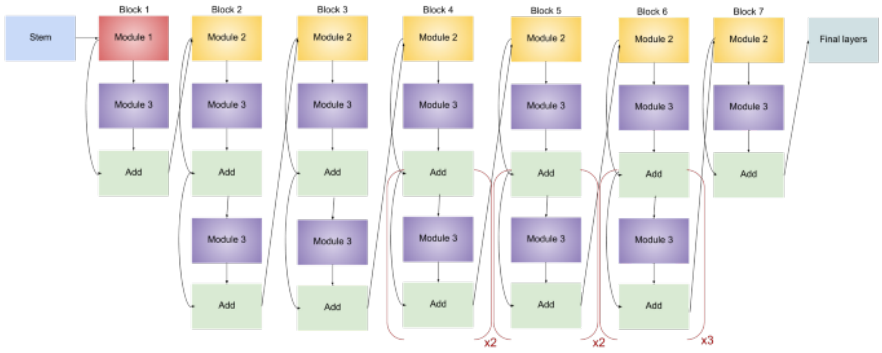


Figura 4.13: Arqitetura da *EfficientNetB2*`https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142`

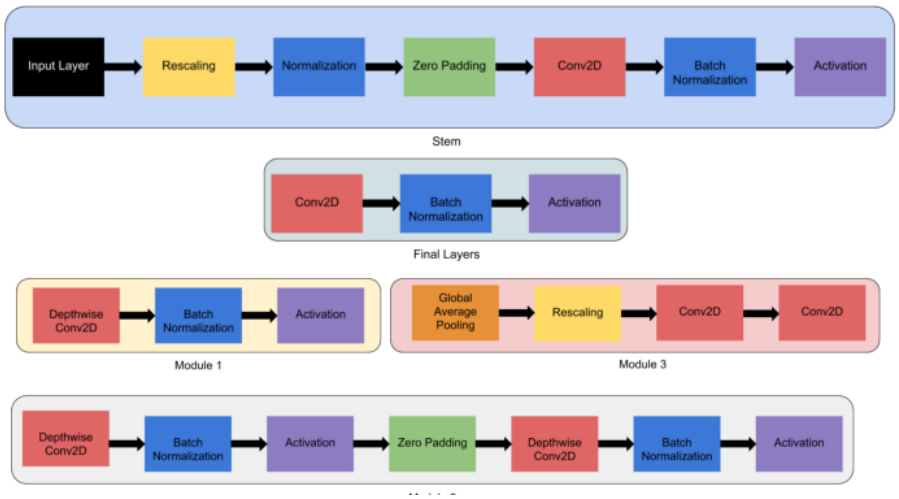


Figura 4.14: Descrição dos módulos e das suas camadas `https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142`

Os blocos residuais ligam o início e o fim de um bloco convolutivo com uma ligação de saltar. Ao adicionar estes dois estados, a rede tem a oportunidade de aceder a activações anteriores que não foram modificadas no bloco

convolutivo. Esta abordagem revelou-se essencial para a construção de redes de grande profundidade.

Quando se olha um pouco mais perto para a ligação de saltar, nota - se que um bloco residual original segue uma abordagem ampla -> estreita -> ampla relativamente ao número de canais. A entrada tem um elevado número de canais, que são comprimidos com uma simples convolução 1x1. Desta forma, a seguinte convolução 3x3 tem muito menos parâmetros. A fim de adicionar entrada e saída no final, o número de canais é novamente aumentado utilizando outra convolução 1x1.Por outro lado, as camadas convolucionais deste modelo segue uma abordagem estreita-> ampla -> estreita. O primeiro passo alarga a rede usando uma convolução de 1x1 porque a seguinte convolução de 3x3 em profundidade já reduz muito o número de parâmetros. Depois, outra convolução 1x1 aperta a rede de modo a corresponder ao número inicial de canais.Esta ideia é denominada de bloco residual invertido porque existem ligações de saltar entre partes estreitas da rede, o que é o oposto de como funciona uma ligação residual original.

Para otimizar a rede é preciso acrescentar uma camada de *Batch Normalization* antes de cada camada convolutiva e usar a função de ativação *ReLU6* em vez de *ReLU*, que limita o valor das activações a um máximo de 6. A activação é linear desde que se situe entre 0 e 6.A razão pela qual são utilizadas funções de activação não linear em redes neurais é que múltiplas multiplicações de matrizes não podem ser reduzidas a uma única operação numérica. Permite construir redes neuronais que têm múltiplas camadas. Ao mesmo tempo, a função de activação *ReLU*, que é normalmente utilizada em redes neurais, descarta valores inferiores a 0. Esta perda de informação pode ser combatida aumentando o número de canais de modo a aumentar a capacidade da rede.Com blocos residuais invertidos, faze-se o oposto e apertam-se as camadas onde as ligações de saltar estão ligadas. Isto prejudica o desempenho da rede. Os autores introduziram a ideia de um estrangulamento linear onde a última convolução de um bloco residual tem uma saída linear antes de ser adicionado às activações iniciais.Ao usarmos a função *ReLU6* ao invés da função *ReLU*, quando se trata de inferência de ponto fixo, a *ReLU6* limita a informação esquerda do ponto decimal a 3 bits, o que significa que tem-se uma precisão garantida à direita do ponto decimal.

O modelo vai usar 2798 imagens, repartidas por 1484 classes diferentes , para treinar e 113 para validar, estas imagens já quadradas vão ser redimensionadas para 260x260 pixeis, estas imagens vão ser passadas para o modelo em lotes(batches) de 19 unidades durante 500 épocas.O modelo vai ser otimizado utilizando o otimizador Adagrad com o valor de taxa de aprendizagem 0.002 e o valor acumulador de 0.1 e vamos utlizar como valor métricos a precisão , a precisão das top 5 previsões do modelo e a perda do modelo. Obtêm-se desta

forma os seguintes gráficos após o treino:

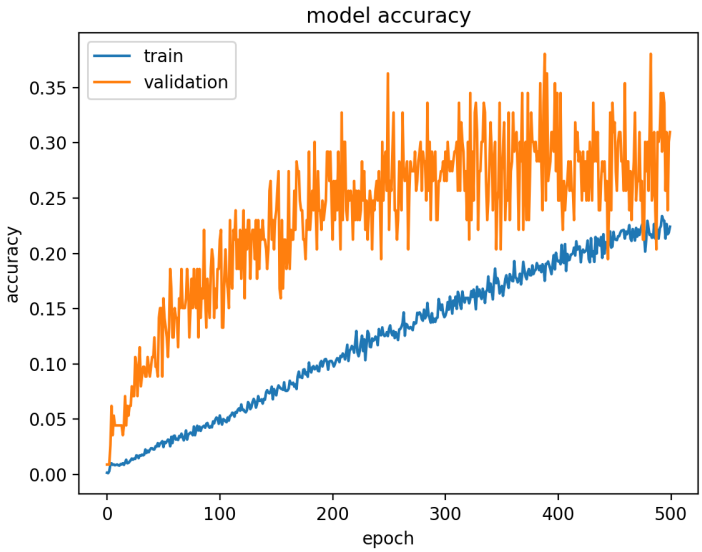


Figura 4.15: *EfficientNetB2* precisão

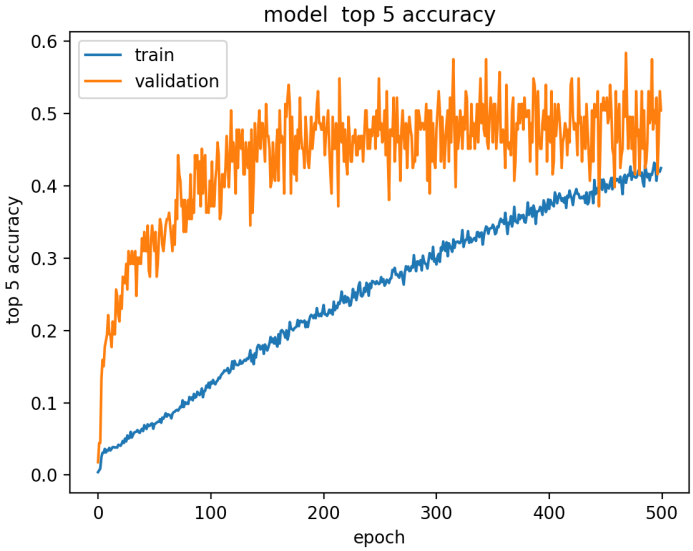


Figura 4.16: *EfficientNetB2* precisão das top5 previsões

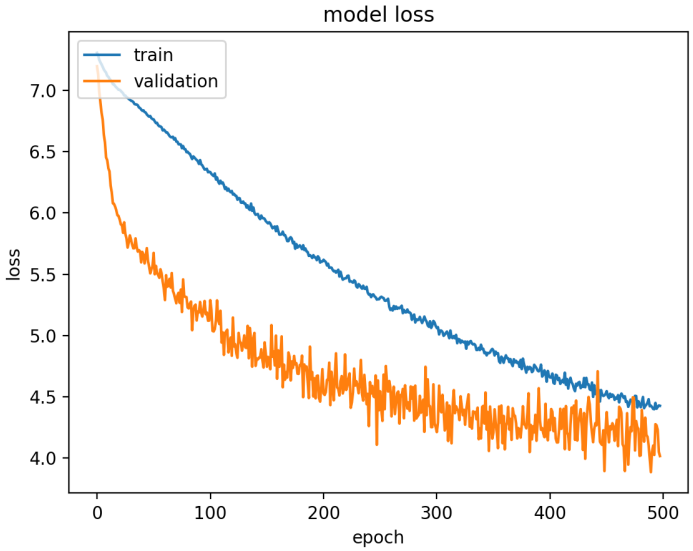


Figura 4.17: *EfficientNetB2* perda

Dos gráficos pode-se concluir que as linhas referentes aos valores obtidos do treino estão a comportar-se como esperado , a cada época a precisão aumenta e o valor da perda diminui, mas o valor de precisão dos valores no grupo validação começam por ser maiores do que no grupo de treino,isto indica que as imagens do grupo de validção são mais fáceis de prever que as imagens do grupo de treino, para obter-se então informação mais concreta deve-se treinar o modelo durante mais épocas. Optou-se treinar o mesmo modelo durante 1500 épocas.

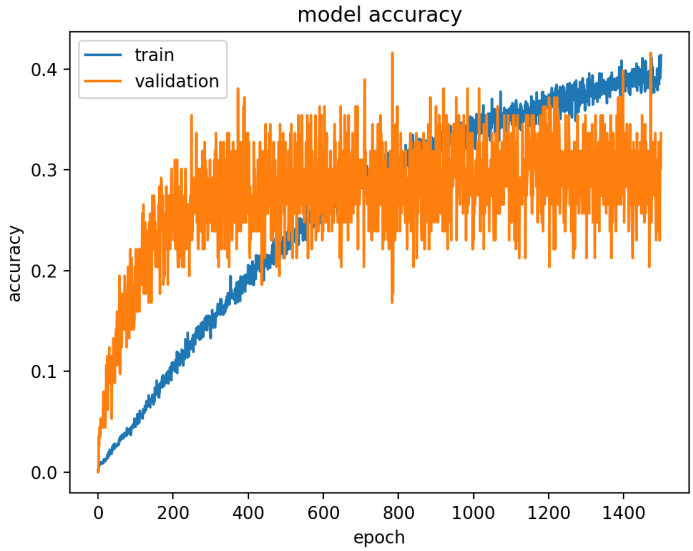


Figura 4.18: *EfficientNetB2* treinada durante 1500 épocas, precisão

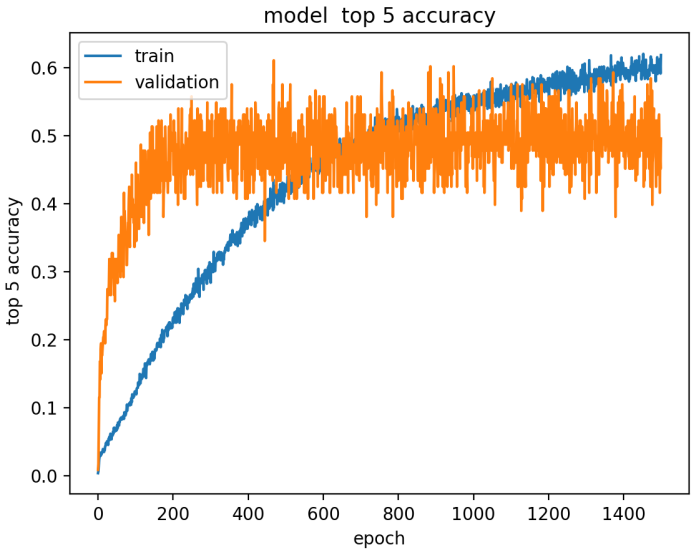


Figura 4.19: *EfficientNetB2* treinada durante 1500 épocas, precisão das top5 previsões

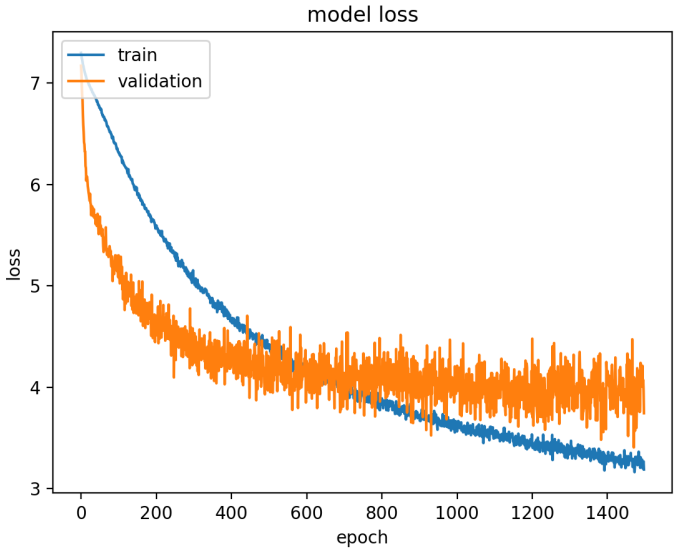


Figura 4.20: *EfficientNetB2* treinada durante 1500 épocas, perda

Estes gráficos tornam-se um pouco dificeis de entrepretar pois existe uma grande oscilação entre os valores obtidos do grupo de validação, o que significa que o modelo considera a quantidade de imagens no grupo de validação baixa para avaliar da melhor forma a capacidade de generalizar do modelo a ser treinado. Infelizmente ao escolher-se aumentar o conjunto de validação vai-se ter que por consequência diminuir o grupo de treino, e ao diminuir este grupo o modelo vai ter menos imagens para treinar e menos preciso irá ser depois no conjunto de teste. Durante esta secção vamos manter o rácio orignal de imagens de 80% e 20% mas mais á frente irão testar - se mais rácios para poder diminuir as oscilações sem sacrificar a qualidade de previsão do modelo, ou sacrificar o mínimo possível.Novamente volto a frisar que estes pequenos precalços ainda são devidos às pequenas dimensões da base de dados.

Apesar destes resultados um pouco inconsistentes dos valores, os valores obtidos do grupo de validação são deveras interessante, Em todos os gráficos na época 600 as linhas pertencentes ao conjunto finalmente intersectam a linha do conjunto de validação, esta embora com grandes oscilações vai se manter constante até ao final do treino estagnando nos valores de 0.28(28%) na precisão , 0.48(48%) nas top5 previsões e 4.3 no valor de perda.Pelo contrário a linha correspondente ao conjunto de treino continua a crescer em termos de precisão e acaba com um valor de 0.4(40%) de precisão e 0.6(60%) nas top5 previsões e com um valor de perda de 3. Segundo o comportamento

destas linhas podemos assumir que estes valores só iriam melhorar caso aumentassemos o valor de épocas.

Para a avaliação final do modelo, este testa-se face ao conjunto de teste , o modelo vai prever a coordenadas da imagem e de seguida vai compara -las às coodenadas reais da imagem e por conseguinte medimos a distancia entre a coordenadas devolvidas pelo modelo e as coodernadas reais, mas nesta situação vamos testar o modelo treinado em 500 épocas face ao modelo treinado em 1500 épocas:

Distancia	Percentagem
5< metros	29.07%
10< metros	41.85%
20< metros	44.49%
50< metros	51.54%
100< metros	59.03%
250< metros	65.64%
500< metros	76.65%
800< metros	83.70%
1000< metros	88.11%
1500< metros	96.92%

Tabela 4.3: Resultados do modelo *EfficientNetB2* 500 épocas , **Desvio Padrão: 480.65 metros**

Distancia	Percentagem
5< metros	33.48%
10< metros	48.02%
20< metros	54.19%
50< metros	64.76%
100< metros	69.16%
250< metros	74.89%
500< metros	84.14%
800< metros	90.31%
1000< metros	92.07%
1500< metros	98.24%

Tabela 4.4: Resultados do modelo *EfficientNetB2* 1500 épocas , **Desvio Padrão: 388.16 metros**

Através destas duas tabelas pode-se afirmar que um número maior de épocas teve um efeito positivo na capacidade de generalizar do modelo.

4.3.4 Conclusões

Após se realizarem estas experiências iniciais nestes três modelos distintos, com base nos resultados consegue-se afirmar que a *EfficientNetB2* foi a que se adaptou melhor ás limitações da dimensão do conjunto de dados, e devolveu melhor resultados em relação ás outras arquiteturas. Por estas razões optou-se por escolher este modelo para realizar mais experiências e para otimizar os seus resultados a fim de encontrar o melhor modelo possível para o problema de geolocalização.

4.4 *Learning rate*, taxa de aprendizagem

Nesta secção vai-se alterar o modelo, modificando a taxa de aprendizem , até encontrar o valor ideal desta. As redes neurais de aprendizagem profunda são treinadas utilizando o algoritmo de descida de gradiente estocástico.A descida de gradiente estocástico é um algoritmo de optimização que estima o gradiente de erro para o estado actual do modelo utilizando exemplos do conjunto de dados de treino, e depois actualiza os pesos do modelo utilizando o algoritmo de retropropagação de erros, referido como simplesmente retropropagação. A quantidade e o número de vezes que os pesos são actualizados durante o treino é referida como taxa de aprendizagem. Especificamente, a taxa de aprendizagem é um hiperparâmetro configurável utilizado no treino de redes neurais que tem um pequeno valor positivo, muitas vezes no intervalo entre 0,0 e 1,0.A taxa de aprendizagem controla a rapidez com que o modelo é adaptado ao problema. Taxas de aprendizagem mais pequenas requerem mais épocas de treino, dadas as menores alterações feitas aos pesos de cada actualização, enquanto taxas de aprendizagem maiores resultam em mudanças rápidas e requerem menos épocas de treino.Uma taxa de aprendizagem demasiado grande pode fazer com que o modelo convirja demasiado depressa para uma solução subóptima, enquanto que uma taxa de aprendizagem demasiado pequena pode fazer com que o processo fique preso.O desafio de treinar redes neurais de aprendizagem profunda envolve a selecção cuidadosa da taxa de aprendizagem. Pode ser o hiperparâmetro mais importante para o modelo. O modelo apresentado na secção 4.3.3 usufruiu de uma taxa de aprendizagem de 0.002, em busca da taxa de aprendizagem que vai melhorar o desempenho da rede , decidiu-se treinar a mesma rede mas com

os seguintes valores de taxa de apredizagem : 0.01,0.1 e 0.25.Realizaram-se testes em perídos de 500 épocas.

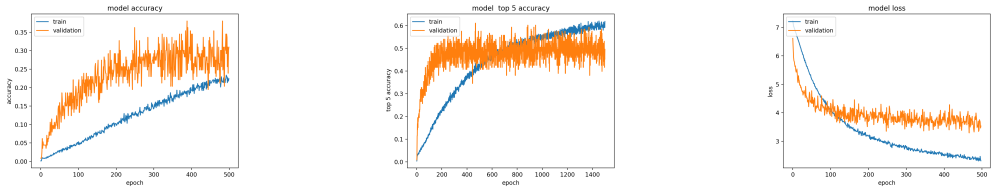


Figura 4.21: Taxa de aprendizagem 0.002

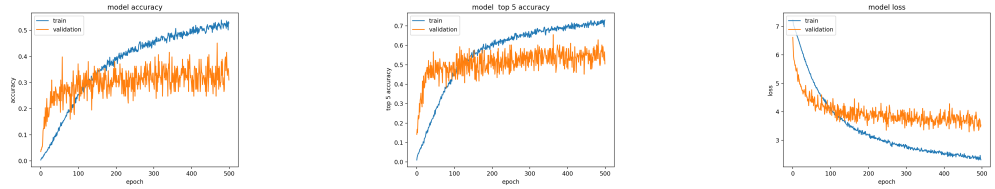


Figura 4.22: Taxa de aprendizagem 0.01

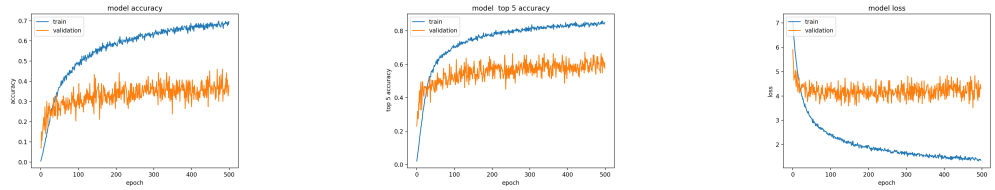


Figura 4.23: Taxa de aprendizagem 0.1

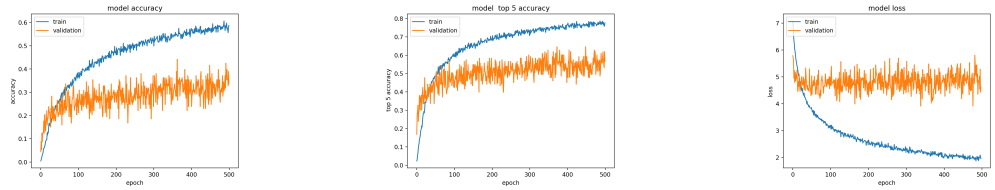


Figura 4.24: Taxa de aprendizagem 0.25

Taxa de aprendizagem	5< m	10< m	25< m	50< m	100< m	Desvio padrão
0.002	29.07%	41.85%	44.49%	51.54%	59.03%	480.65 metros
0.01	40.09%	57.27%	65.20%	75.33%	79.30%	312.94 metros
0.1	37.44%	55.07%	61.23%	69.16%	74.89%	349.36 metros
0.25	38.77%	52.86%	57.27%	65.20%	69.60%	339.01 metros

Tabela 4.5: Resultados dos modelos *EfficientNetB2* com diferentes taxas de aprendizagem

4.4.1 Conclusões

Ao analisar - se os gráficos pode-se assumir que à medida que se aumenta o valor da taxa de aprendizagem a linha referente ao conjunto de treino atinge valores superiores ao modelo orginal representado nas figuras 4.21 e a linha do conjunto de treino interseta e ultrapassa a linha ddo conjunto de validação mais cedo , e segundo a tabela **??** todos os outros modelos são capazes de generalizar melhor que o original que apenas possui um valor de 0.002 de taxa de aprendizagem.Ao prestar-se ainda mais atenção à informação disponibilizada reparamos que os gráficos 4.23 e 4.28 são muito semelhantes, quase indêntico , apesar de possuirem uma diferença de 0.15 entre as suas taxas de aprendizagem.Além do mais a capacidade de generalização do modelo com 0.25 de taxa de aprendizagem generaliza pior que dos modelos com taxa de 0.1 e 0.01, atuando apenas melhor nas categorias de menos 5 metros de distancia e no desvio padrão por uma margem mínima em relação ao modelo com taxa de 0.1.Daqui pode-se retirar que apenas aumentar a taxa de aprendizagem , não melhora a eficácia do modelo, mas pelo contrário , que a partir de um certo valor uma taxa de aprendizagem começa a prejudicar a capacidade do modelo aprender novos padrões e por consequência diminuir a sua capacidade de generalizar.Pode-se ainda verificar que apesar dos gráficos 4.23 parecerem ser mais prometedores ques os dos gráficos 4.22 pois a linha que pertence ao conjunto de treino consegue alcançar melhores valores , porém os valores referentes ao conjunto de validação apresentam valores semelhantes ao longo de todas as épocas, e através da tabela consegue-se afirmar que o modelo com a taxa de aprendizagem 0.01 é capaz de generalizar melhor que o modelo com taxa igual a 0.1 e que os restantes modelos.

Desta forma conclui-se que o valor ideal de taxa de aprendizagem para este problema e para o conjunto de dados disponível deve rondar o valor de 0.01. Para encontrar o valor exato para a taxa de aprendizagem ser ótima teriam que se realizar imensos testes , por isso adotou-se o valor de 0.01 para prosseguir com a otimização do modelo, e conseguir melhorar outros parâ-

metros também importantes.

4.5 *Batch size*

As redes neurais são treinadas utilizando-se o algoritmo de optimização de descida de gradiente estocástico. Isto implica utilizar o estado atual do modelo para fazer uma previsão, comparando a previsão com os valores esperados, e utilizar a diferença como uma estimativa do gradiente de erro. Este gradiente de erro é então utilizado para atualizar os pesos do modelo e o processo é repetido.O gradiente de erro é uma estimativa estatística. Quanto mais exemplos de formação utilizados na estimativa, mais precisa será esta estimativa e mais provável que os pesos da rede sejam ajustados de forma a melhorar o desempenho do modelo. A estimativa melhorada do gradiente de erro tem o custo de ter de utilizar o modelo para fazer muitas mais previsões antes de a estimativa poder ser calculada e, por sua vez, os pesos actualizados.Alternativamente, a utilização de menos exemplos resulta numa estimativa menos precisa do gradiente de erro que depende muito dos exemplos específicos de formação utilizados.Isto resulta numa estimativa turbulenta que, por sua vez, resulta em actualizações turbulentas dos pesos dos modelos, por exemplo, muitas actualizações com estimativas talvez bastante diferentes do gradiente de erro. No entanto, estas actualizações ruidosas podem resultar numa aprendizagem mais rápida e por vezes num modelo mais robusto.O número de exemplos de treino utilizados na estimativa do gradiente de erro é um hiperparâmetro para o algoritmo de aprendizagem chamado *Batch size*.Um tamanho de lote de 32 significa que 32 amostras do conjunto de dados de treino serão utilizadas para estimar o gradiente de erro antes de os pesos do modelo serem actualizados. Uma época de treino significa que o algoritmo de aprendizagem fez uma passagem através do conjunto de dados de treino, onde os exemplos foram separados em grupos *Batch size* seleccionados aleatoriamente.

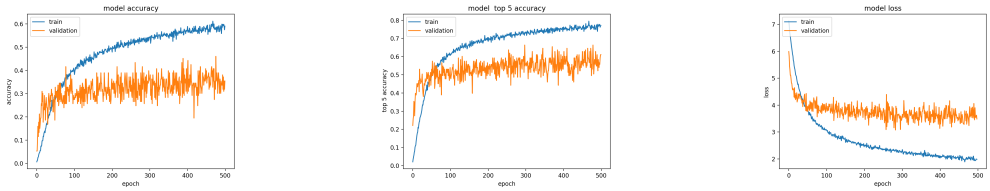


Figura 4.25: *Batch size* de 4

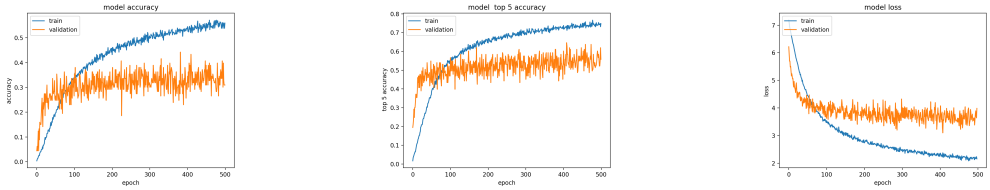


Figura 4.26: *Batch size* de 10

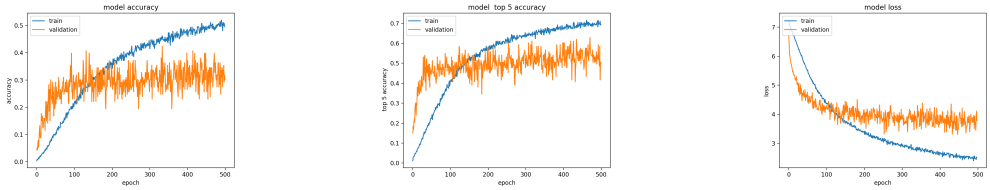


Figura 4.27: *Batch size* de 24

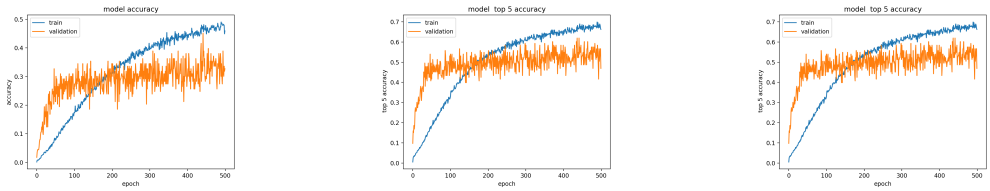


Figura 4.28: *Batch size* de 32

Os modelos anteriores das secções anteriores foram todos treinados com um valor de 19 para o *Batch size*.Nestes modelos vou treinaram-se usando os seguites valores: 4, 10 , 24 , 32. Estes modelos apresentam os seguintes resultados no conjunto de teste:

Batch size	5< m	10< m	25< m	50< m	100< m	Desvio padrão
4	38,77%	56.39%	61.23%	72.69%	78.85%	285.35 metros
10	38.77%	55.77%	62.11%	73.13%	77.97%	296.97 metros
24	38.77%	54.19%	60.79%	71.81%	75.77%	307.27 metros
32	39.21%	54.19%	61.67%	72.25%	75.77%	355.33 metros

Tabela 4.6: Resultados dos modelos *EfficientNetB2* com diferentes taxas de aprendizagem

4.5.1 Conclusão

Como se pode observar pelos resultdados obtidos na tabela 4.6 os resultados não variam muito em termos de precisão , mas através da métrica do desvio padrão consegue-se ver um paradigma, quanto menor o batch size menor o desvio padrão. Após esta observação pode-se afirmar que um *batch size* menor é benéfico para este problema.

4.6 Tamanho do conjunto de validação

Nesta secção aborda-se em como ao sacrificar-se um pouco de imagens do conjunto de treino em prol de termos um conjunto de validação maior.Para esta experiência utilizou-se um rácio de 60/40 para as imagens destinadas aos respetivos conjuntos, ficando assim com 2409 imagens no conjunto de treino e 502 no conjunto de validação.O modelo possui um valor de *batch size* de 4 e taxa de aprendizagem de 0.01.

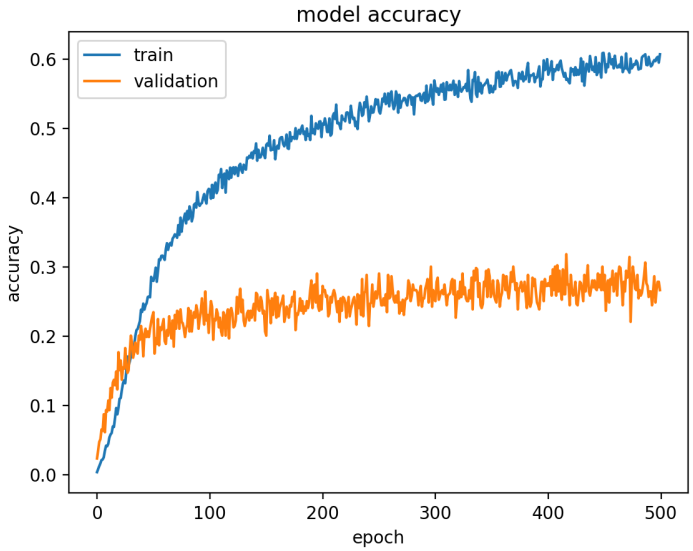


Figura 4.29: grupo de validação com 502 imagens, precisão

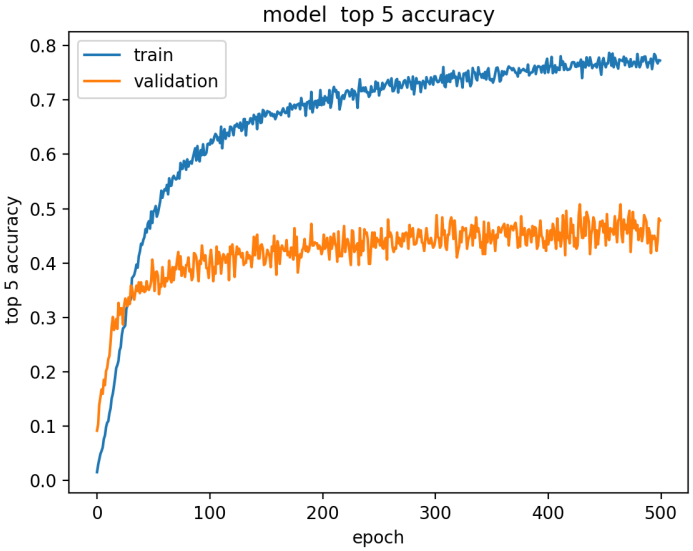


Figura 4.30: grupo de validação com 502 imagens, precisão das top5 previsões

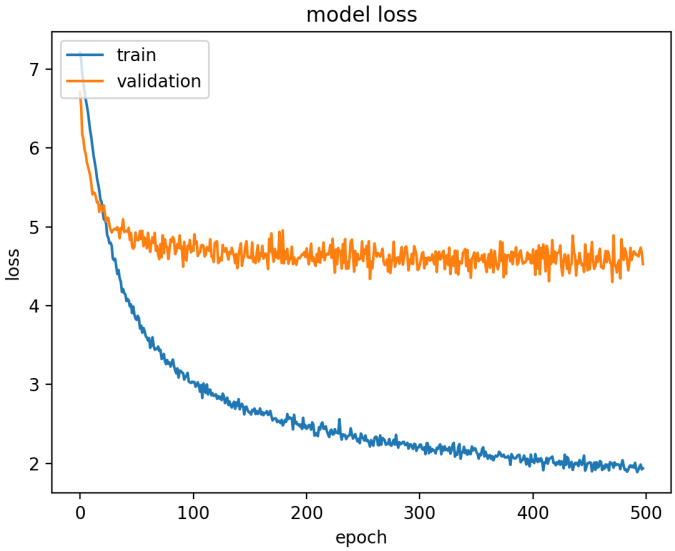


Figura 4.31: grupo de validação com 502 imagens, perda

Distancia	Percentagem
5< metros	36.56%
10< metros	52.42%
20< metros	58.15%
50< metros	70.93%
100< metros	74.45%
250< metros	81.50%
500< metros	87.67%
800< metros	93.39%
1000< metros	95.15%
1500< metros	99.56%

Tabela 4.7: Resultados do modelo com 502 imagens no grupo de validação , **Desvio Padrão: 321.49 metros**

Ao analisar-se o gráfico pode se concluir que a linha correspondente ao grupo de validação não oscila tanto como as linhas de modelos anteriores , mas o custo de diminuir as oscilações da linha e tornar o gráfico mais fácil de intrepretar foi diminuir a precisão do modelo , pois este ficou com menos imagens para treinar em relação a modelos prévios.

4.6.1 Conclusões

No contexto de problema verifica-se que ao usar o maior número possivel de imagens , vai resultar num melhor modelo e que ao usar um grande número de imagens no conjunto de validação vai reduzir as oscilações dos gráficos e desta forma facilitar a sua intrepretação.

4.7 Modelo Final

Face ao conhecimento adquirido nestas últimas secções , propõe-se que o modelo final tenha as seguintes propriedades: *batch size* de 1 , taxa de aprendizagem de 0.01 , o conjunto de treino contenha o número màximo de imagens(não existe conjunto de validação) e vai ser treinado durante 500 épocas.

Após treinado este modelo é testado face ao grupo de teste, obtendo os seguintes resultados:

Distancia	Percentagem
5< metros	42.29%
10< metros	61.23%
20< metros	66.08%
50< metros	77.09%
100< metros	80.62%
250< metros	84.58%
500< metros	89.87%
800< metros	95.15%
1000< metros	96.92%
1500< metros	100%

Tabela 4.8: Resultados do modelo final , **Desvio Padrão: 274.56 metros**

Ao analisar-se estes resultados podemos confirmar que o modelo final é o que melhor generaliza e atua melhor no problema que queriamos resolucionar.

4.8 Dificuldades do modelo

O modelo desenvolvido, dentro do contexto do problema, é capaz de generalizar bem e obter previções coerentes e precisas, mas o modelo está dependente da imagem fornecida para a previsão. As imagens de treino abrangem

várias zonas da Covilhã e estas imagens tentam abrangir um máximo de características de diferentes ruas e àreas de interesse da localidade, as fotos são retiradas a uma certa distancia para captar o máximo de detalhes que tornam aquele setor único e reconhecível.Para além destas foto que compõem a maior parte do conjunto de dados também foram retiradas fotos que se focam não na zona mas sim nos monumentos, grafittis , estátuas ... que são exclusivas à localidade.

Figura 4.32: Exemplo de imagem do conjunto de dados

Figura 4.33: Exemplo de imagem do conjunto de dados

No entanto o modelo é incapaz de fazer uma previsão precisa de a imagem fornecida é pouco abrangente(se a foto foi tirada muito perto) ou pelo contrário se a imagem é demasiado abrangente(uma foto tirada demasiado longe). A primeira é devido ás falta de características descriminatórias suficientes para o modelo ser capaz de atribuir uma classe de forma confiante , e a segunda porque pode ter demasiadas caracteristicas pertencentes a várias classes ou por a imagem foi tirada de um ângulo que não apresentada nenhuma pecularidade em relação às imagens contidas no conjunto de treino.

Para se provar a primeira dificuldade , infelizmente não possuo imagens pouco abrangentes , mas proponho uma alternativa: recortar partes das imagens do grupo de teste e usar o modelo nestas imagens e por fim comparar os resultados das previsões das imagens parciais com os resultados das previsões das imagens na sua totalidade.

Figura 4.34: Imagem original e imagem parcial da imagem original

Ao usar o modelo neste conjunto de teste "parcial"obtemos os seguintes resultados:

Distancia	Percentagem
5< metros	0.00%
10< metros	0.00%
20< metros	0.88%
50< metros	3.98%
100< metros	7.08%
250< metros	31.86%
500< metros	54.42%
800< metros	78.32%
1000< metros	81.86%
1500< metros	95.13%

Tabela 4.9: **Conjunto de imagens parciais,Desvio Padrão: 439.48 metros**

Pode-se afirmar seguramente, que o modelo atua pior neste conjunto de

teste, pois as imagens cobrem características insuficientes para o modelo fazer uma boa previsão.

Pode-se confirmar a segunda dificuldade analisando as tabelas com as distâncias das previsões nas secções anteriores.Embora o modelo nem sempre consiga prever a classe correta consegue escolher um par de cordenadas bastante perto da localização real da imagem, isto é porque certas imagens possuem várias caracteristicas e o modelo considera que existem várias boas possibilidades que representam as coordenadas da foto. Quanto mais longe e/ou mais características a foto possuir de diferentes localizações da cidade, maior o número de potencais classes corretas , e maior a dificuldade de o modelo atingir uma precisão correta.

Ainda outro caso que o modelo atua mal é em fotos tiradas durante a noite, mas este problema è causado por uma falta de fotos noturnas no conjunto de treino.

E claro se as fotos pertencerem a fora da região coberta pela base de dados as previsões vão ser sempre incorretas.Por exemplo uma foto de Lisboa.

4.9 Conclusões

Pode-se concluir que apesar da grande dificuldade do problema a resolver e dos poucos recursos em termos de base de dados disponíveis para a realização de um modelo competitivo com outros modelos com objetivos semelhantes , acredito que foi desenvolvido um modelo decente com uma boa capacidade de generalização para a região da covilhã.

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Na elaboração deste projeto, o problema da *geolocalização* foi abordado como um problema de classificação de imagens.Durante a etapa de investigação apercebi-me que o tópico de redes convulucionais é um assunto muito vasto e que pode ser usados vários problemas e é um assunto que até este momento , está a ser investigado pelos melhores profissionais do ramo de informática. Este projeto despertou muita curiosidade e vontade de continuar a aprofundar o meu conhecimento em redes convulucionais.

5.2 Trabalho Futuro

Para trabalho futuro seria interressante expandir a área de previsão do modelo para algo maior como portugal e/ou o mundo, e adicionar uma base de dados com imagens noturnas para o modelo ser capaz de realizar boas previsões em qualquer hora do dia. Isto não foi realizado devido à falta de recursos, este trabalho foi realizado apenas por um elemento.

Bibliografia

[1] Kader Pustu-Iren Eric Müller-Budack and Ralph Ewerthl. Geolocation Estimation of Photos using a Hierarchical Model and Scene Classification. 2018. [Online] https://openaccess.thecvf.com/content_ECCV_2018/papers/Eric_Muller-Budack_Geolocation_Estimation_of_ECCV_2018_paper.pdf.

[2] James Philbin Tobias Weyand, Ilya Kostrikov. PlaNet - Photo Geolocation with Convolutional Neural Networks. 2016. [Online] https://static.googleusercontent.com/media/research.google.com/pt-PT//pubs/archive/45488.pdf.

[3] Nathan Jacobs Nam Vo and James Hays. Revisiting IM2GPS in the Deep Learning Era. 2017. [Online] https://openaccess.thecvf.com/content_ICCV_2017/papers/Vo_Revisiting_IM2GPS_in_ICCV_2017_paper.pdf.

[4] Paul Barham Eugene Brevdo Zhifeng Chen Craig Citro Greg S. Corrado Andy Davis Jeffrey Dean Matthieu Devin Sanjay Ghemawat Ian Goodfellow Andrew Harp Geoffrey Irving Michael Isard Yangqing Jia Rafal Jozefowicz Lukasz Kaiser Manjunath Kudlur Josh Levenberg Dan Man e Rajat Monga Sherry Moore Derek Murray Chris Olah Mike Schuster Jonathon Shlens Benoit Steiner Ilya Sutskever Kunal Talwar Paul Tucker Vincent Vanhoucke Vijay Vasudevan Fernanda Vi egas Oriol Vinyals Pete Warden Martin Wattenberg Martin Wicke Yuan Yu Mart ın Abadi, Ashish Agarwal and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2016. [Online] https://arxiv.org/pdf/1603.04467.pdf.

[5] Chen Chen Sijie Zhu, Taojiannan Yang. VIGOR: Cross-View Image Geo-localization beyond One-to-one Retrieval. 2021. [Online] https://openaccess.thecvf.com/content/CVPR2021/papers/Zhu_VIGOR_Cross-View_Image_Geo-Localization_Beyond_One-to-One_Retrieval_CVPR_2021_paper.pdf.

[6] Sixing Hu Mengdan Feng Rang M. H. Nguyen Gim Hee Lee. CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial

Geo-Localization. 2018. [Online] https://openaccess.thecvf.com/content_cvpr_2018/papers/Hu_CVM-Net_Cross-View_Matching_CVPR_2018_paper.pdf.

[7] Xin Yu Hongdong Li Yujiao Shi, Liu Liu. Spatial-Aware Feature Aggregation for Cross-View Image based Geo-Localization. 2019. [Online] https://papers.nips.cc/paper/2019/file/ba2f0015122a5955f8b3a50240fb91b2-Paper.pdf.

[8] Chen Chen Sijie Zhu, Taojiannan Yang. Revisiting Street-to-Aerial View Image Geo-localization and Orientation Estimation. 2019. [Online] https://openaccess.thecvf.com/content/WACV2021/papers/Zhu_Revisiting_Street-to-Aerial_View_Image_Geo-Localization_and_Orientation_Estimation_WACV_2021_paper.pdf.

[9] Andrew Zisserman Karen Simonyan. Deep Residual Learning for Image Recognition. 2015. [Online] https://arxiv.org/pdf/1512.03385.pdf.

[10] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. 2015. [Online] https://arxiv.org/pdf/1409.1556.pdf.

[11] J. S. Denker D. Henderson R. E. Howard; W. Hubbard L. D. Jackel Y. LeCun, B. Boser. Backpropagation Applied to Handwritten Zip Code Recognition. 1989. [Online] http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf.

[12] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. Backpropagation Applied to Handwritten Zip Code Recognition. 2012. [Online] https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[13] Rob Fergus Matthew D Zeiler. Visualizing and Understanding Convolutional Networks. 2014. [Online] https://arxiv.org/pdf/1311.2901.pdf.

[14] P.Frasconi Y.Bengio, P.Simard. Learning long-term dependencies with gradient descent is difficult. 1994. [Online] http://www.comp.hkbu.edu.hk/~markus/teaching/comp7650/tnn-94-gradient.pdf.

[15] Yoshua Bengio Xavier Glorot. Understanding the difficulty of training deep feedforward neural networks. 2010. [Online] https://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

[16] Genevieve B. Orr Klaus Robert Müller Yann A. LeCun, Léon Bottou. Efficient backprop. 2012. [Online] http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf.

[17] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015. [Online]https://arxiv.org/pdf/1502.01852.pdf.

[18] Christian Szegedy Sergey Ioffe. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. [Online]https://arxiv.org/pdf/1502.03167.pdf.

[19] Jian Sun Kaiming He. Convolutional Neural Networks at Constrained Time Cost. 2014. [Online] https://arxiv.org/pdf/1412.1710.pdf.

[20] Karen Simonyan Andrew Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. 2015. [Online] https://arxiv.org/pdf/1409.1556.pdf.

[21] Quoc V. Le Mingxing Tan. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2020. [Online] https://arxiv.org/pdf/1905.11946.pdf.

[22] Nikos Komodakis Sergey Zagoruyko. Wide Residual Networks. 2016. [Online] http://www.bmva.org/bmvc/2016/papers/paper087/paper087.pdf.

[23] Ankur Bapna Orhan Firat Mia Xu Chen Dehao Chen HyoukJoong Lee Jiquan Ngia Quoc V. Le Yonghui Wu Zhifeng Chen Yanping Huang, Youlong Cheng. GPipe: Easy Scaling with Micro-Batch Pipeline Parallelism. 2019. [Online] https://arxiv.org/pdf/1811.06965.pdf.