INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Deteção automática de idade através de características faciais

MIGUEL NUNO DE QUEIRÓS BOUÇA RIBEIRINHO MACHADO
Julho de 2022

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Deteção automática de idade através de características faciais

Miguel Nuno de Queirós Bouça Ribeirinho Machado

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas de Informação e Conhecimento

Orientador: Doutora Fátima Rodrigues

Porto, julho 2022

Resumo

Atualmente a automação de tarefas é uma prática cada vez mais comum em diversos sectores, uma vez que permite reduzir a necessidade de mão de obra e aumentar a eficiência de tarefas.

O envelhecimento é um processo natural e complexo no desenvolvimento do ser humano. É afetado por fatores intrínsecos e extrínsecos. A compreensão deste processo é fundamental para viabilizar a deteção de idade baseado em características faciais.

O presente trabalho propõe a construção de um sistema de deteção de idade com base numa imagem facial do utilizador. Este sistema contempla, numa fase inicial o pré-processamento da imagem seguido do desenvolvimento de um modelo de deteção de idade através de uma rede neuronal convolucional. O sistema foi ainda disponibilizado através de uma aplicação web.

Dos vários modelos desenvolvidos com recurso às redes *Xception*, VGG-16 e Inception-V4 o que obteve melhor performance foi o modelo *Xception*. Este modelo, prevendo 4 faixas etárias, apresentou uma taxa de acerto de 88%.

Palavras-chave: Idade, Deteção, Redes Neuronais Convolucionais, Web app

Abstract

Nowadays task automation is a common practice in many sectors. It allows to reduce labor and increase efficiency.

Aging is a natural and complex process in the human being development. This process is affected by intrinsic and extrinsic factors. The comprehension of this process is fundamental to detect age based on facial characteristics.

For this project we propose the creation of an aging detection system, based on a user's facial image. The system comprises, in an initial phase, the pre-processing of the image followed by the development of an age detection model through a convolutional neural network. The system will be available through a web application.

There were many models created based on the *Xception*, VGG-16 and Inception-V4 networks, but the one with the best performance was an *Xception* model. This model, assessing 4 age groups, had and accuracy of 88%.

Keywords: Age, Detection, Convolutional neural networks, Web app

Agradecimentos

A realização de um trabalho desta magnitude envolve direta ou indiretamente várias pessoas às quais pretendo expressar uma palavra de agradecimento, nomeadamente:

Á orientadora, Professora Doutora Fátima Rodrigues, por todo o conhecimento transmitido e sugestões realizadas, pelo apoio incondicional ao longo de todo o desenvolvimento do presente trabalho.

A toda a minha família e amigos, em primeiro lugar pela amizade demostrada e em segundo por toda a ajuda e apoio ao longo do meu percurso académico.

A todos que anteriormente foram referidos, e a todos aqueles que de alguma forma contribuíram para a realização desta dissertação, o mais profundo obrigado.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

AHP	*Analytic Hierarchy Process*
ANN	*Artificial Neural Network*
CE	*Cross Entropy*
CNN	*Convolution Neural Network*
CRISP-DM	*Cross Industry Standard Process for Data Mining*
FAST	*Function Analysis and System Technique*
FN	*False Negative*
FP	*False Positive*
GPU	*Graphic Processing Unit*
GB	*Gigabytes*
HTTP	*Hypertext Transfer Protocol*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
MAE	*Mean absolute error*
MB	*Megabytes*
MSE	*Mean Squared Error*
NCD	*New Concept Development*
QFD	*Quality Function Deployment*
RAM	*Random-access memory*
ReLU	*Rectified Linear Unit*
RMSE	*Root Mean Squared Error*
RGB	Red Green Blue
SAVE	*Society of American Value Engineers*
SGD	Stochastic Gradient Descent

TN	*True Negative*
TP	*True Positive*
YGA	*Yamada Gender and Age*
VGG	*Visual Geometry Group*

1 Introdução

1.1 Contexto

Atualmente existe uma mudança de paradigma onde muitas tarefas, que anteriormente eram desempenhadas por seres humanos, hoje são desempenhadas por computadores. Este facto levou a um aumento de desempenho e ao aperfeiçoamento de outros sistemas. Diversas tarefas do cotidiano são atualmente desempenhadas por computadores de forma mais eficiente, económica e veloz.

Um sistema capaz de definir características demográficas, nomeadamente a idade, com base em imagens possui um imenso potencial no mundo atual. O uso de um sistema de determinação de idade, através de reconhecimento facial, apresenta inúmeras aplicações em diversas áreas. Áreas como segurança, estudos biométricos, tecnologias de entretimento, recomendação de bens, assim como definição de grupos consoante faixas etárias. Ferramentas de marketing podem personalizar a mensagem com base na faixa etária reduzindo assim custos com publicidade para o publico alvo errado.

Para um ser humano este processo é considerado uma tarefa trivial, uma vez que por norma os seres humanos são capazes de identificar facilmente a idade aproximada de outro individuo. Porém, existem diversos fatores que podem dificultar este processo. O uso de maquilhagem, cirurgias plásticas bem como tratamentos estéticos podem alterar o aspeto de um individuo contrastando desta forma a aparência com os padrões de envelhecimento [1].

As redes neuronais convolucionais, nos últimos anos, tornaram-se na técnica de visão da computação com maior popularidade. Têm apresentado excelentes resultados para executar tarefas de classificação de imagens, deteção de objetos e reconhecimento facial. No entanto, o seu uso requer uma elevada capacidade de processamento [2].

1.2 Problema

O processo de envelhecimento humano não se trata de um processo uniforme, uma vez que diversos fatores podem condicionar a aparência humana. A qualidade de vida, o stress, doenças são alguns aspetos que podem afetar o desenvolvimento humano. Para além dos fatores anteriormente mencionados, o próprio procedimento de recolha de imagem também possui variáveis que podem prejudicar o processo de deteção de idade, nomeadamente a pose, a iluminação da imagem, a qualidade da imagem, entre outras.

Para definir um modelo adequado é necessário estabelecer as características e parâmetros fundamentais para a categorização do modelo. A qualidade do *dataset* e a quantidade de imagens a utilizar apresentam uma forte influência sobre a precisão final do modelo. Face aos diversos fatores que podem condicionar a avaliação da idade, a seleção do conjunto de dados e o seu correto tratamento são dos principais desafios neste processo. Atualmente ainda não existe um conjunto de dados de referência para sistemas de deteção de idade. Pode-se constatar em artigos científicos relacionados com o tema, que não só não existe um consenso no melhor conjunto de dados a utilizar como, por norma, existem trabalhos atuais para complementar e aperfeiçoar os conjuntos de dados [2].

Apesar de toda a investigação realizada nos últimos anos, a precisão da estimativa de idade humana, de forma automática, ainda não conseguiu ultrapassar a precisão apresentada por humanos [1].

1.3 Objetivos

O presente trabalho tem como propósito criar um modelo capaz de detetar a idade de um ser humano com base na imagem da sua face. De forma a atingir esta meta, as seguintes tarefas são concretizadas:

- Realizar uma pesquisa bibliográfica relativamente às principais metodologias de reconhecimento facial para apuramento de idade.
- Identificar os *datasets* públicos com maior relevância.
- Desenvolver e avaliar modelos de deteção de idade humana (quer idade numérica quer grupo etário), usando processamento de imagens e algoritmos de *deep learning*.
- Comparação de resultados com outros trabalhos de deteção de idade.

1.4 Resultados expectáveis

Com este projeto serão desenvolvidos modelos de reconhecimento de idade capazes de estimar a idade ou grupo etário de um indivíduo através da imagem da sua face, com uma taxa de acerto competitiva com os demais trabalhos nesta área. O modelo com melhores resultados será integrado numa aplicação *web,* com o intuito de facilitar o seu uso pelos utilizadores.

1.5 Análise de valor

Um modelo capaz de detetar a idade de um ser humano, de forma automática, somente com base na imagem de uma face representa uma mais-valia para diversos sectores do mercado. O sistema, resultante da integração deste modelo, permite a redução de custos de mão-de-obra, bem como aumentar a eficiência de outros processos, nomeadamente campanhas de marketing direcionadas a faixas etárias específicas. Verificam-se ainda outros benefícios, onde a quantificação das mais-valias não é tão simples, como o controlo de acesso a sites por parte de crianças menores, evitando a exposição a conteúdo indesejado.

1.6 Metodologia

A abordagem a adotar, no atual trabalho, contempla o uso de rede neuronais para a construção de modelos de deteção de idade através de imagens de faces humanas. A construção dos modelos de classificação é realizada recorrendo à metodologia CRISP-DM, *Cross-industry standard process for data mining* [3]. Esta metodologia é considerada como o *standard* mais utilizado para suporte a projetos de *data mining*, devido à flexibilidade que apresenta em qualquer domínio.

Assim sendo, de acordo com a referida metodologia, numa fase inicial é necessário analisar e compreender o problema em questão. De seguida procede-se com a recolha de imagens de um ou diversos conjuntos de dados para a deteção de idade. As imagens são posteriormente tratadas através de diversas técnicas, como por exemplo normalização. Na fase de modelação, vão ser criados vários modelos com origem em arquiteturas diferentes, o que permite testar diferentes tipos de redes neuronais convolucionais. Pretende-se recorrer a redes pré-treinadas com o intuito de simplificar a conceção dos modelos e melhorar a sua performance. Os resultados dos diversos modelos vão ser avaliados e comparados entre eles através de testes estatísticos. Estes testes permitem confirmar se as diferenças de desempenho entre os vários modelos são significativas. Para finalizar, o modelo que exiba melhores resultados vai ser integrado na aplicação final.

Para auxiliar o desenvolvimento do modelo e realizar o controlo de versões o sistema escolhido é o GIT. Todo o desenvolvimento é documentado. Pretende-se desta forma documentar os procedimentos, dados e observações de forma lógica e organizada.

No final comparam-se os resultados com os objetivos iniciais. Deve-se ainda efetuar a comparação dos resultados obtidos com os resultados declarados em outros trabalhos científicos semelhantes.

1.7 Estrutura do documento

O capítulo inicial do presente documento apresenta uma introdução ao problema, enquadrando o contexto em que este surge. São ainda apresentados os objetivos que se pretendem alcançar, os resultados expectáveis e é realizada uma análise de valor ao sistema proposto. Para finalizar expõe-se resumidamente a abordagem ao problema.

No capítulo do estado da arte referencia-se o processo de envelhecimento e, as dificuldades que este acarreta para o modelo que se pretende desenvolver. São apresentadas as tecnologias intrínsecas ao problema em estudo. Iniciando-se por uma visão global ao tema aprendizagem de máquina e posteriormente pormenorizando em técnicas de *deep learning*, nomeadamente em redes neuronais convolucionais. São apresentados os *datasets* com maior popularidade para trabalhos de deteção de idade e por último é realizado uma análise a trabalhos científicos da área.

O terceiro capítulo consiste na análise de valor, onde as metodologias de análise de valor da solução são expostas. Desenvolvem-se os modelos *Canvas*, NCD e o AHP para o problema em questão.

A proposta de design e estrutura para a solução proposta é apresentada no quarto capítulo.

No quinto capítulo, é relatado o processo de criação dos vários modelos bem como a sua avaliação. Os modelos são comparados entre eles e com os trabalhos semelhantes analisados no segundo capítulo.

O capítulo seguinte consiste na apresentação final do sistema concebido.

O último capítulo apresenta as conclusões relativamente ao trabalho desenvolvido. São ainda validados os objetivos iniciais, referidas as limitações sentidas ao longo do trabalho, bem como um conjunto de propostas de futuros trabalhos a serem desenvolvidos.

2 Estado da arte

Atualmente diversos sistemas inteligentes são concebidos com o intuito de substituir o fator humano na concretização de tarefas específicas. Estes sistemas permitem, por norma, alcançar melhor performance na execução das tarefas e reduzir custos. De forma a desenvolver um sistema de deteção de idade, capaz e eficiente, é necessário dotar o sistema da capacidade de estimar as características biológicas dos utilizadores. Esta capacidade permite simular aptidões humanas, que tendem a ser fundamentais em qualquer interação social. A face é o atributo com maior relevância no processo de reconhecimento. Quer seja reconhecimento de identidade, de raça, de idade, a face é o elemento mais utilizado nestas análises. Assim sendo, estes sistemas inteligentes devem ser capazes de, com base numa imagem facial do utilizador, extrair as características mais pertinentes e estimar o atributo necessário para a sua função.

No presente trabalho será abordado o processo de deteção automática de idade através de características faciais. A idade é um fator relevante uma vez que, a capacidade de estimar a mesma é fundamental para ferramentas de entretenimento, de segurança digital, de marketing, entre outros. O processo de estimar a idade através das características faciais representa a capacidade de um sistema definir a idade exata ou a faixa etária com base numa imagem facial.

2.1 Processo de envelhecimento

O processo de envelhecimento consiste num processo complexo que pode ser afetado por inúmeros fatores. Numa fase inicial da vida de um ser humano existe um desenvolvimento craniofacial significativo. Para além da expansão do crânio verifica-se uma expansão dos olhos, orelhas, boca e nariz. Com o avançar da idade podem-se constatar outras alterações faciais. Surgem as rugas, bem como a alteração de textura da pele e manchas da idade (hiperpigmentação). Sob a pele, as células que produzem melanina, devido à exposição a raios ultravioleta do sol são danificadas. O que origina uma distribuição desigual de melanina e

consequentemente manchas na face. A maioria das alterações faciais visíveis da fase de adulto para velhice consistem em alterações da pele.

Existem fatores intrínsecos e extrínsecos que afetam o envelhecimento facial. Fatores extrínsecos são fatores externos ao corpo humano, e podem ser aspetos ambientais. Fatores intrínsecos podem ser a estrutura óssea, a genética, entre outros.

A automação de um sistema de deteção de idade, com imagens faciais, apresenta diversas dificuldades. No entanto, um marco significativo para esta tarefa foi a disseminação do *FG-NET Aging Dataset* em 2002. Este conjunto de dados contempla imagens de indivíduos com diferentes idades sendo o primeiro *dataset* deste género [1].

Existem diferentes abordagens usadas para detetar a idade. No entanto, técnicas de *Deep Learning* têm alcançado sucesso em tarefas de análise de faces, nomeadamente deteção de faces, alinhamento de faces, verificação de faces e estudos demográficos [1]. Por norma, este tipo de tarefas contempla numa fase inicial um algoritmo de deteção de faces para auxiliar na remoção de informação desnecessária das imagens. Um dos algoritmos mais populares, em parte devido à sua velocidade, é o detetor de faces de Viola e Jones que utiliza a técnica *Haar-like* para detetar uma face e obter as suas coordenadas [4].

2.2 Aprendizagem máquina

Inteligência artificial (IA) surgiu na década de 50 e pode ser resumida como o esforço de automatizar tarefas intelectuais realizadas por humanos. No âmbito da inteligência artificial, na década de 90, surge o conceito aprendizagem máquina, ou *machine learning,* que rapidamente se torna a área de IA mais popular e com maior sucesso [5].

Os sistemas de *machine learning* são treinados em vez de ser explicitamente programados. Através da análise de um conjunto de dados, com volume significativo onde dados podem estar catalogados ou não, o sistema descobre padrões intrínsecos nos dados que permitem automatizar tarefas.

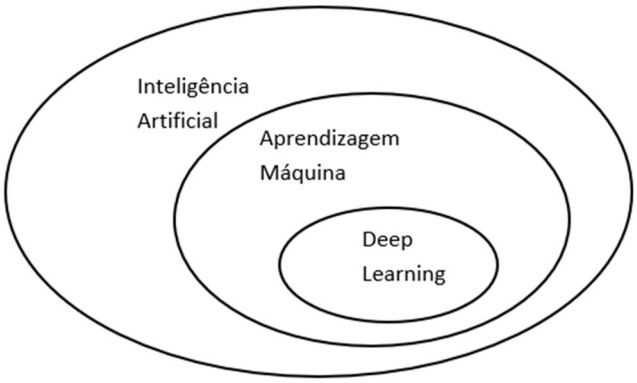


Figura 1 – Diagrama hierárquico de inteligência artificial, aprendizagem máquina e *deep learning* [Adaptado de 5].

Os dois principais paradigmas de aprendizagem máquina são a aprendizagem supervisionada e a aprendizagem não supervisionada. A aprendizagem supervisionada é feita usando dados etiquetados de modo a que os algoritmos aprendam com os dados e desenvolvam modelos capazes de prever o rótulo correto em novos dados não rotulados. Na aprendizagem não supervisionada os dados não são etiquetados e, portanto, os algoritmos de *machine learning* atuam sobre os mesmos sem orientação descobrindo autonomamente padrões nos dados [6].

2.2.1 Classificação vs. Regressão

A tarefa de detetar a idade de um ser humano pode ser alvo de duas abordagens. A primeira consiste em abordar a tarefa como um problema de classificação. Problemas de classificação são empregues quando o problema pode ser definido por um número de classes discretas. Para a tarefa de estimar a idade, cada uma destas classes pode referenciar um intervalo de idades. Assim sendo, neste caso está-se a estimar a faixa etária do indivíduo, consoante os intervalos definidos pelas classes. A segunda abordagem à tarefa consiste em abordar como um problema de regressão. O uso de problemas de regressão está associado a resultados contínuos, mais precisamente, quando o resultado diz respeito a uma quantidade especifica e não um conjunto de possíveis classes. Neste caso pretende-se estimar a idade exata do indivíduo em vez de obter o intervalo de idades em que se encontra [5].

2.3 Redes Neuronais Artificiais

O nome redes neuronais artificiais (ANN) trata-se de uma alusão ao modo de funcionamento do cérebro humano. Alguns conceitos base das redes neuronais foram concebidos com base na compreensão do nosso cérebro [5].

Estas redes consistem em elementos de processamento (nós) organizados em três tipos de camadas interligadas por um elevado número de conexões. Os dados são introduzidos na rede através da camada de *input*, segue-se uma ou mais camadas do tipo oculto (*hidden layers*) e a rede termina na camada de *output* [5].

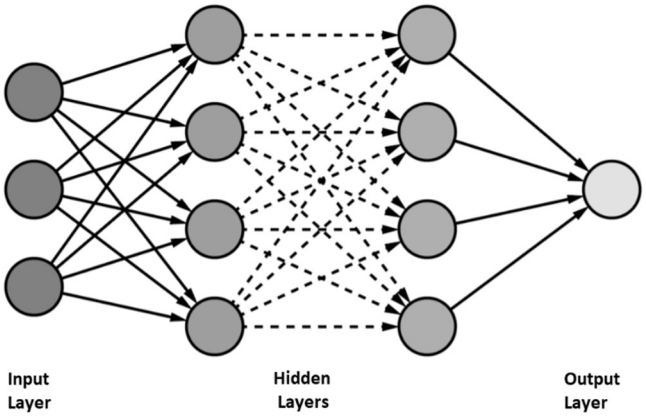


Figura 2 – Esquema de rede neuronal artificial [7].

A camada ou camadas intermédias, do tipo ocultas, é onde o processamento se realiza com base nas conexões, que estão associadas a coeficientes designados por pesos.

Assim sendo, em cada nó é realizado o somatório dos valores de inputs multiplicados pelos respetivos pesos e é adicionado um parâmetro designado por *bias*. Ao resultado final é aplicada uma função, *activation function*, e obtido o valor de saída do nó. A *activation function* permite aumentar a capacidade de aprendizagem da rede com dados complexos, de forma a conseguir lidar com dados complexos não lineares. As funções mais comuns são *Sigmóide* e *ReLU* (*Rectified Linear Unit*) apresentadas de seguida. Tal como se pode constatar na figura 3 a *Sigmóide* produz um valor entre 0 e 1, enquanto que a *ReLU* tem como resultado um valor igual ou superior a 0. Existe ainda a função *Softmax* que é uma generalização da função *Sigmóide* para casos não binários. [8].

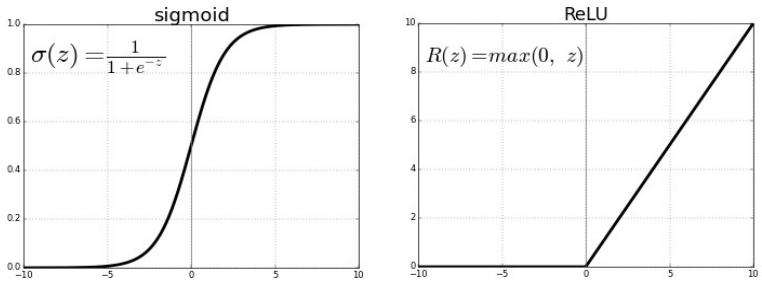


Figura 3 – Representação de funções *Sigmóide* e *ReLu* [9].

Durante o processo de treino os pesos são ajustados, num processo designado *backpropagation*, sendo este processo fundamental para a rede aprender a classificar os valores introduzidos.

Este processo de afinação de pesos consiste num processo iterativo onde repetidamente se ajusta os pesos de cada conexão de forma a minimizar o erro na classificação [5].

Para treinar uma rede neuronal é necessário um elevado volume de dados. O processo de aprendizagem varia consoante o tipo de dados que são introduzidos na rede. No caso de processo de aprendizagem supervisionado, o conjunto de dados de treino inclui o resultado expectável correto. Uma vez que o valor final é conhecido, a rede segundo um processo iterativo, realiza previsões e ajusta o valor dos pesos [8].

A rede neuronal pode ter maior flexibilidade e complexidade aumentando o número de camadas intermédias. Porém este aumento torna a rede mais complexa e mais difícil de otimizar os seus pesos.

Uma rede neuronal com um elevado número de camadas ocultas é designada de rede neuronal profunda. Existem vários tipos de redes neuronais profundas, no entanto as mais populares para processamento de imagem, são as redes neuronais convolucionais (CNN – *Convolutional Neural Network*) [2].

2.4 Deep Learning

Deep learning consiste num segmento de *machine learning*. O que distingue as técnicas de *deep learning* é o seu processo de aprendizagem. O conceito de profundidade (*depth*) em *deep learning* diz respeito às sucessivas camadas de representações. A profundidade do modelo consiste no número de camadas presentes. Estas camadas de representações são desenvolvidas via redes neuronais. Estas redes são capazes de modelar relações entre dados complexos não lineares e, desta forma, criar conhecimento a partir de exemplos.

As técnicas de *deep learning* são muito usadas em trabalhos onde é necessário realizar um reconhecimento de voz ou tarefas de âmbito visual [10].

2.4.1 Convolutional neural networks

Na última década as redes neuronais convolucionais tornaram-se na técnica mais popular para classificação de imagens, deteção de objetos e reconhecimento facial [2].

Uma rede neuronal convolucional, por norma, consiste em camadas convolucionais (*convolution layers*), camadas de *pooling*, camada *flatten* e para finalizar *fully connected layers*. As primeiras são o elemento mais significativo das redes neuronais convolucionais. Quando utilizadas para deteção de imagens, as redes são alimentadas através da camada de *input*, com os valores dos píxeis das imagens, em formato de vetor.

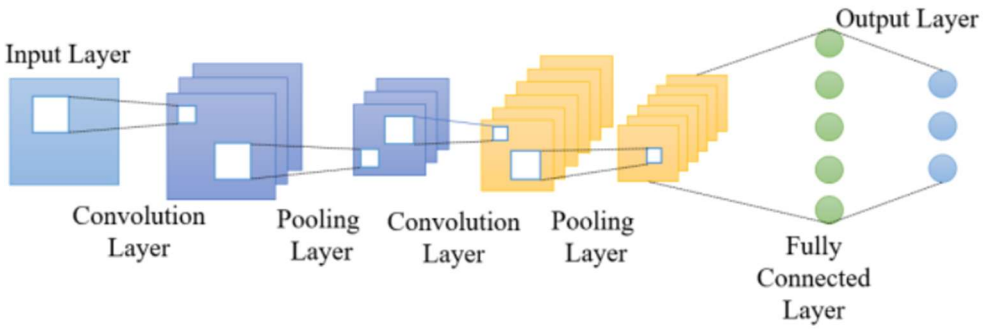


Figura 4 – Arquitetura base de CNN [11].

Em matemática o termo convolução refere-se à combinação de duas funções, produzindo uma terceira. As camadas de convolução são responsáveis pela extração de características a partir dos dados introduzidos (conjuntos de píxeis em caso de imagens) e geram os *feature maps*. Para realizar a extração é aplicado um filtro (*kernel*) que consiste numa matriz de pesos, que procura um padrão. Estes filtros concedem robustez ao modelo, permitindo deste modo solucionar problemas de distorção, rotação e translação de imagens [12].

Os vários *feature maps* são agrupados, formando um tensor com profundidade igual ao número de filtros. Este tensor será o argumento de entrada na próxima camada.

A camada de *pooling* segue-se à camada convolucional e permite reduzir a dimensão da matriz resultante, simplificando a rede e criando um mapa com características condensadas. Os tipos de *pooling* mais utilizados são o *max-pooling* e o *average-pooling*. O primeiro consiste em selecionar o maior valor de uma determinada região, por sua vez, o segundo tipo consiste em obter o valor médio. O *max-pooling* é considerado o mais eficaz [12].

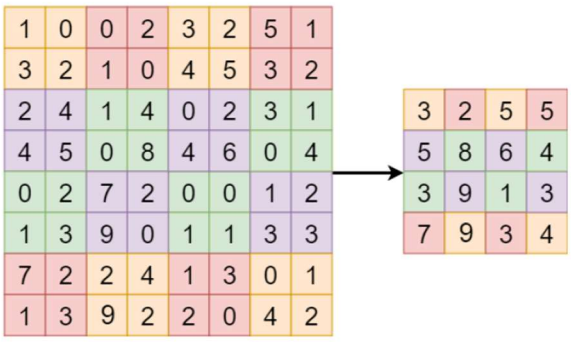


Figura 5 – Exemplo de *max-pooling* [12].

As *fully connected layers* são responsáveis por mapear as características extraídas pelas camadas de convolução e *pooling* para os outputs finais do modelo. Este tipo de camada é capaz de determinar quais as características que se correlacionam com uma classe específica. O seu nome provém do facto de cada nó da camada estar conectado a todos os nós da camada

anterior e a todos os nós da camada seguinte. Antes das *fully connected layers* existe uma camada *flatten* que tem como função alterar o formato dos dados de uma matriz para um *array*.

O funcionamento das redes neuronais convolucionais podem ser divididas em duas fases. A fase de extração de características (*feature extraction*) e a fase de classificação. A primeira fase diz respeito às camadas de convolução e *pooling* e têm como objetivo extrair características relevantes da imagem. As camadas iniciais de uma rede tendem a extrair características mais genéricas como orientações, bordas. Quanto maior é a profundidade na fase de extração maior é a capacidade de extrair características complexas de imagens. A segunda fase consiste em transformar as características obtidas num vetor e em seguida, tal como nas redes neuronais, prosseguir com o processo de classificação. No caso de existirem diversas classes, o resultado final será um vetor com uma distribuição de probabilidades para cada classe [5].

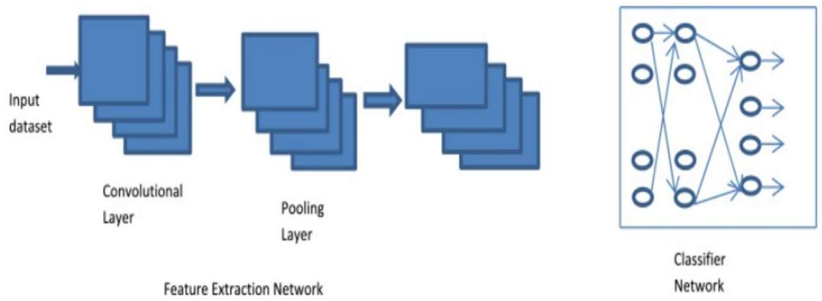


Figura 6 – Fases de rede neuronal convolucional [13].

Existem diversas estratégias capazes de ampliar a performance das redes neuronais convolucionais.

Data Augmentation

Uma quantidade reduzida de imagens na fase de treino pode condicionar a capacidade de generalização do sistema desenvolvido. A técnica *data augmentation* consiste em realizar transformações específicas nas imagens para aumentar a quantidade de imagens disponíveis para o conjunto de treino. Transformações como rotações, translações, aumento ou redução do zoom, entre outras. Estas transformações ajudam a prevenir o sobreajustamento, ou seja, que o modelo se ajuste totalmente com o conjunto de treino e perca a capacidade de generalizar com novas imagens [13].

Dropout

O *dropout* tem como objetivo reduzir o sobreajustamento. Através desta técnica alguns dos nós são aleatoriamente anulados (valor de *input* do nó é estabelecido como zero) bem como as suas conexões. Anulando um nó e as suas conexões durante o processo de treino, previne que o modelo se adapte ao conjunto de dados e não consiga generalizar. Esta técnica é aplicada através da adição de uma camada do tipo *dropout*, onde é definida a *dropout rate*. Esta taxa

tem valores entre 0 e 1. Uma taxa de 0 implica que nenhum nó é anulado, uma taxa de 1 implica que 100% dos nós são anulados. Nas redes neuronais, por norma, as camadas *dropout* são implementadas na fase de classificação [14].

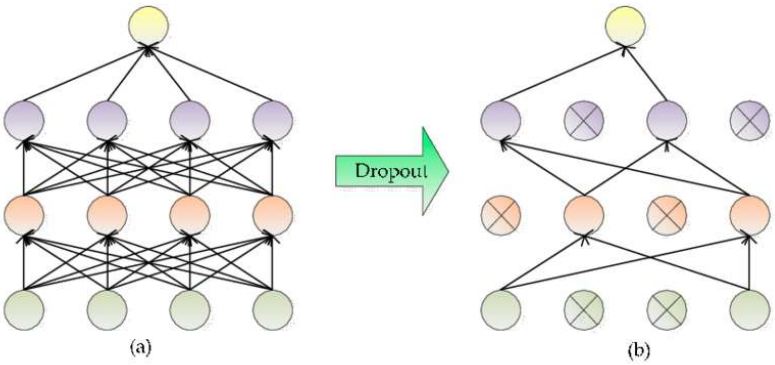


Figura 7 – Efeito aplicação de *dropout* [15].

Epochs

O número de *epochs* diz respeito ao número de iterações da rede neuronal na fase de aprendizagem. Em cada iteração todo o conjunto de treino é processado e, existe um ajuste dos pesos da rede o que, por norma, torna o modelo mais eficaz. A variação no ajuste dos pesos nos primeiros *epochs* deve ser superior aos últimos, devido ao conhecimento obtido pela rede nas iterações anteriores [16].

Loss function

No final de cada *epoch* a performance da rede é avaliada de acordo com o seu objetivo. As funções que avaliam a performance dos modelos, quantificando a diferença entre o resultado de saída e o resultado expectável são as *loss functions*, também designadas por *cost functions*. Se a capacidade preditiva do modelo for boa, o valor da função será reduzido. O valor da função deve ir diminuindo a cada *epoch*. As funções *Mean Squared Error* (MSE) e *Cross entropy* (CE) são as *loss functions* mais populares [16].

Learning Rate

A *learning rate* consiste na taxa de aprendizagem, e através deste parâmetro é possível definir a velocidade a que a rede altera dos seus pesos. Um aumento no valor de *learning rate* implica um aumento na variação dos pesos. Uma aprendizagem mais focada nos últimos padrões identificados pode ser representada por taxa superiores, por outro lado uma aprendizagem mais contínua pode ser representada por taxas inferiores. Por norma, os valores da taxa devem ir reduzindo progressivamente ao longo do processo de aprendizagem.

Assim sendo, é crucial definir um valor apropriado para a *learning rate*. Um valor muito reduzido o modelo irá convergir de forma muito lenta, em contrapartida um valor muito elevado o modelo não será capaz de convergir, tal como se pode constatar na figura 6 [16].

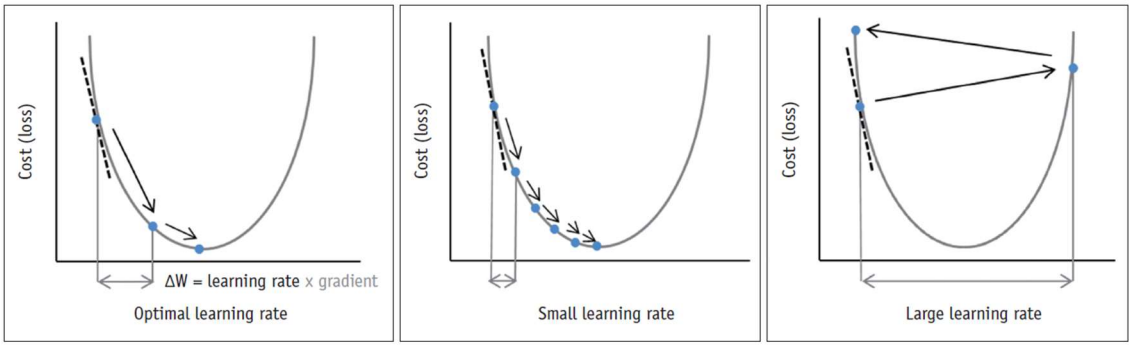


Figura 8 – Efeitos da taxa de aprendizagem no treino do modelo [16].

2.4.2 Redes pré-treinadas

Tal como mencionado anteriormente, numa primeira fase, as redes neuronais convolucionais são responsáveis por extrair características através do processo de convolução. De seguida segue-se o processo de classificação.

A conceção de uma nova CNN é um processo complexo, uma vez que as redes por norma possuem inúmeros parâmetros a serem ajustados e é necessário um elevado grau de conhecimento para criar uma rede eficiente e veloz. De forma a simplificar a criação de novas redes neuronais convolucionais iniciou-se o aproveitamento de modelos anteriormente desenvolvidos. Desta forma, surge o conceito de *transfer learning* que consiste em recorrer a redes pré-treinadas como ponto inicial no processo de classificação, para auxiliar na extração de características. Estas redes possuem diversas camadas onde durante o processo de treino conseguem aprender desde aspetos genéricos como bordas, arestas até aspetos mais detalhados como detetar um animal. Usualmente, estas redes são treinadas com recurso a uma elevada quantidade de imagens [17].

Para além de auxiliar o processo de desenvolvimento de um novo modelo, o *transfer learning* pode ainda ser utilizado em conjuntos de dados que não possuem informação suficiente para treinar um modelo de raiz [5].

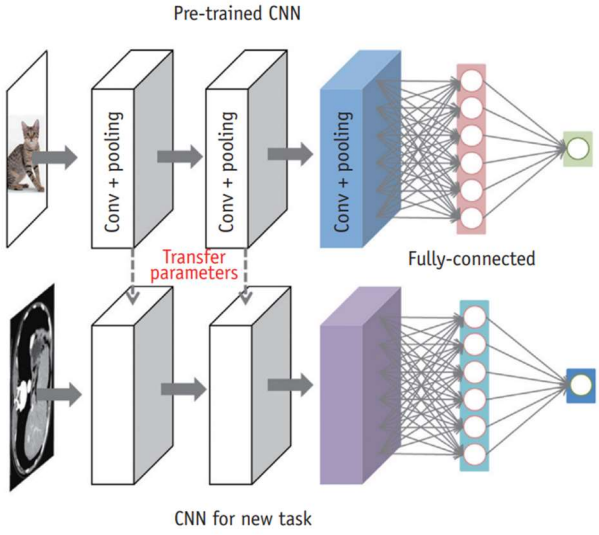


Figura 9 – Exemplo de *transfer learning* e *fine-tuning* [13].

O novo modelo a desenvolver é assim baseado numa rede pré-treinada sendo necessário personalizar as últimas camadas da rede. A técnica de remover as últimas camadas, adicionar novas camadas e congelar as restantes designa-se por *fine-tuning*. Na figura 7 pode-se verificar os processos de *transfer learning* e *fine-tuning* [16].

O desempenho das CNN foi evoluindo ao longo do tempo. De seguida são apresentadas algumas arquiteturas de referência.

LeNet

O modelo *LeNET* consiste num modelo pequeno, simples e pouco profundo, porém consegue bons resultados para problemas simples. Esta arquitetura foi proposta por *LeCun* em 1998 e tem o seu foco no reconhecimento de dígitos [17].

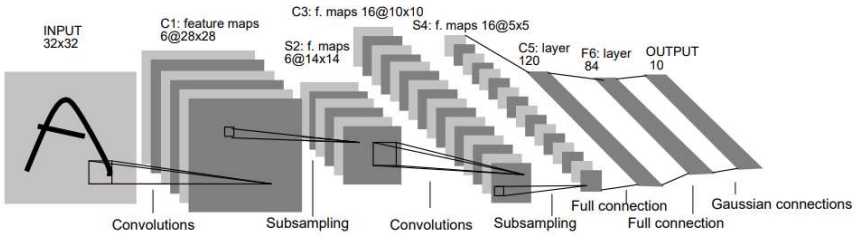


Figura 10 – Arquitetura da rede LeNet [17].

Esta rede possui, no total, duas camadas de cada um dos tipos: convolucionais, Pooling e *Fully-Connected*. Relativamente às camadas convolucionais, possuem 6 filtros e 16 filtros respetivamente [17].

VGG

A arquitetura VGG (*Visual Geometry Group*) é uma das arquiteturas mais populares em modelos de visão computacional. Surge em 2014 e foi usada na competição ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*). A ILSVR é uma competição que tem como objetivo de avaliar algoritmos para deteção de objetos e classificação de imagens.

Comparando esta arquitetura com as arquiteturas anteriormente desenvolvidas, pode-se concluir que existiu uma redução no tamanho dos filtros, bem como o aumento da quantidade de filtros após as camadas de *max-pooling* [18].

Existem duas versões desta arquitetura, VGG-16 e VGG-19. O que diferencia estas duas versões é o número de camadas convolucionais, o modelo VGG-19 contempla 3 camadas convolucionais extra [19].

Este modelo foi concebido para receber imagens RGB com 224x224 píxeis. As camadas de convolução possuem filtros 3x3 seguidas de uma camada de *max-pooling* de filtro 2x2. A função de ativação, exceto na classificação, é do tipo *ReLU*. Na camada final de classificação foi utilizada a função de ativação *Softmax*. A função *Softmax* é aplicada na camada final para normalizar os valores de *output* da rede em probabilidades [19].

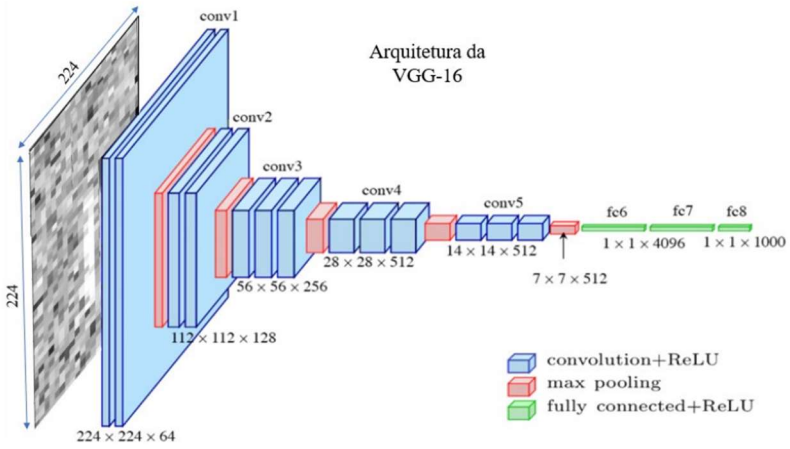


Figura 11 – Arquitetura da rede VGG-16 [19].

Esta arquitetura possui um elevado número de parâmetros o que a torna mais customizável, porém é mais difícil de configurar e aumenta o consumo de memória. De salientar ainda que, as redes VGG são consideradas lentas, pois apresentam uma velocidade reduzida durante a fase de treino de uma rede.

Esta arquitetura serviu de base para o desenvolvimento de várias arquiteturas, e foi fundamental para demonstrar que para a obtenção de melhores resultados em redes convolucionais a profundidade é um aspeto relevante [19].

ResNet

Esta arquitetura foi a vencedora da competição ILSVR no ano de 2015. Esta rede residual possui diversas versões, em que cada versão consiste num número diferente de camadas. O surgimento desta rede implicou um avanço significativo na arquitetura de redes, uma vez que permitiu treinar redes neuronais com mais de 150 camadas. Anteriormente este feito era considerado muito difícil pois após uma certa profundidade o aumento de camadas não implicava o aumento da qualidade do modelo, ou seja, verificava-se uma degradação do modelo com o aumento da profundidade. O modelo que apresentou maior eficácia na competição em 2015 possuía 152 camadas, formadas por blocos básicos residuais [20].

Inception

Anteriormente à da criação da arquitetura *Inception*, a grande maioria das redes neuronais convolucionais desenvolvidas para aumentar a sua performance aumentavam o número de camadas convolucionais, tornando-as mais profundas. A primeira rede do tipo *Inception* foi um importante marco no desenvolvimento das CNN uma vez que, em vez de aumentar a profundidade da rede, a sua estrutura foi revista. Uma das alterações foi a uso de filtros de múltiplas dimensões, o que tornou a rede com maior largura, em vez de maior profundidade.

Existem diversas arquiteturas da família *Inception*. A *Inception-V4* surge como uma evolução em relação às suas antecessoras. O objetivo da sua criação foi reduzir a complexidade da *Inception-V3* e aumentar a sua eficácia. Verificou-se uma alteração dos procedimentos iniciais na rede, introduzindo os *Redution Blocks*, que são utilizados para alterar as dimensões da rede [21].

Xception

A arquitetura designada por *Xception* foi inspirada na arquitetura *Inception* e foi concebida por *Chollet* [22]. Diferencia-se da *Inception* por introduzir o conceito de *depthwise separable convolutions*. É constituída por várias camadas convolucionais do tipo *deepwise separable*. Este tipo resume-se a empregar filtros diferentes para cada canal da imagem. Contrariamente ao caso comum em que se utiliza um só filtro de 3 dimensões. Neste caso utiliza-se um filtro por canal RGB aplicando-se posteriormente uma convolução padrão com um filtro de tamanho 1x1. É composta por 36 camadas de convolução estruturadas em catorze módulos [23].

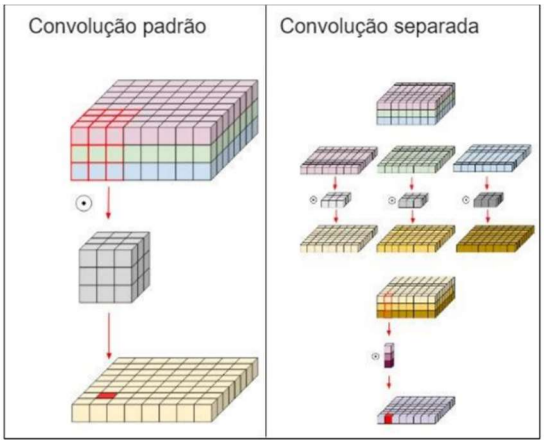


Figura 12 – Diferença entre convolução padrão e convolução separada [24].

Esta arquitetura apresenta maior eficácia no uso dos parâmetros quando comparada com a arquitetura *Inception-V3*. Ambas as arquiteturas possuem o mesmo número de parâmetros, no entanto a *Xception* apresentou melhores resultados na classificação de imagens com o *dataset ImageNet* [23].

2.5 Datasets

De forma a conseguir prever corretamente a idade ou o intervalo de idades é fundamental assegurar a qualidade dos dados. Existem *datasets* que são públicos tais como FG-NET ou IMDB e outros. No entanto, a maioria dos conjuntos de dados não são públicos e para ter acesso é necessário contactar os respetivos autores. Em seguida apresentam-se alguns dos *datasets* mais difundidos nesta área de deteção de idade.

2.5.1 FG-NET Dataset

Este *dataset* público contém um total de 1002 imagens de 82 indivíduos de idades compreendidas entre os 0 e os 69 anos. As imagens são a cores e a preto e branco. Cada indivíduo possui em média 12 imagens. Os indivíduos representam várias raças, no entanto existem grandes inconsistências relativamente às poses, às expressões faciais, bem como iluminação. De salientar ainda que algumas imagens possuem fraca qualidade uma vez que foram digitalizadas [1].

2.5.2 MORPH Dataset

Dataset criado na *North Carolina University* que se divide em dois álbuns. O primeiro álbum contempla 1724 imagens de 515 indivíduos com idades entre 27 e 68 anos. Cerca de 83% das fotografias são de homens o que cria uma disparidade de géneros. O segundo álbum contém

55134 imagens de 13 mil indivíduos obtidas ao longo de 4 anos. Ambos os álbuns contêm metadados relativamente a data de nascimento, raça, género e data da fotografia [20].

2.5.3 Yamada Gender and Age (YGA) Dataset

Este conjunto inclui 8000 imagens a cores de alta resolução pertencentes a 1600 indivíduos, sendo metade homens. Os indivíduos são asiáticos com idades entre 0 e 93 anos. Existem aproximadamente 5 imagens frontais de cada pessoa. As imagens não possuem iluminação constante e pode-se verificar diferentes expressões faciais [1].

2.5.4 Gallagher's Web-collected Dataset

Criado por *Gallagher and Chen* do motor de busca de imagens *Flickr.com*, este conjunto de dados possui 28231 faces em 5080 imagens. Encontra-se dividido por 7 intervalos. É considerado um recurso ideal para estimativa de grupo de idade [1].

2.5.5 Ni's Web-collected Dataset

Criado usando os motores de busca *Google.com* e *Flickr.com*, este conjunto possui 219892 faces em 77021 imagens, sendo um dos maiores *datasets* já reportado. Possui idades entre 1 e 80 anos tornando o conjunto de dados muito completo e, desta forma, ideal para estimar a idade de crianças, adultos e idosos [1].

2.5.6 UTKFace

Este conjunto de dados possui mais de 20000 imagens, contendo informação sobre género, idade e grupo étnico. Possui idades entre os 0 e os 116 anos, e uma elevada diversidade relativamente a poses, expressões faciais e iluminação [25].

2.5.7 APPA-Real

O *dataset* APPA contém um total de 7591 imagens associando faces humanas com a sua idade. Contempla idades entre os 0 e 95 anos. As imagens não possuem a mesma dimensão. Pode-se ainda verificar uma iluminação inconstante e diferentes expressões faciais [26].

2.5.8 IMDB-Wiki

Consiste no *dataset* com maior quantidade de faces humanas. Possui um total de 523051 imagens referentes a 20284 indivíduos. Este *dataset* resulta da junção dos conjuntos de dados IMDB e WIKI. Por norma são utilizados em conjunto, porém podem ser obtidos individualmente e são designados por IMDB e WIKI. As imagens possuem informações relativamente ao género,

idade e nome. Apesar do elevado volume de dados estes não apresentam a qualidade desejada, uma vez que existem erros na informação das imagens bem como imagens vazias [2].

De seguida é apresentado um quadro resumo relativamente à informação dos *datasets* mais relevantes para identificação de idade com base em imagens faciais.

Datasets	Nº indivíduos	Nº imagens	Intervalo de idades
FG-NET	82	1002	0-69
MORPH	13618	55134	27-68
YGA	1600	8000	0-93
Gallagher's	—	28231	0-66
Ni's	—	219892	1-80
UTKFace	—	23708	0-116
APPA-Real	—	7591	0-95
IMDB-Wiki	20284	523051	0-100

Tabela 1 – Sumário dos *datasets.*

2.6 Trabalhos relacionados

2.6.1 Facial Age Estimation Using Convolution Neural Networks [27]

Este trabalho diferencia-se dos restantes trabalhos científicos pois pretende estudar a eficácia de quatro modelos pré-treinados, recorrendo a três tipos de arquiteturas diferentes. As redes a utilizar são *ResNet50*, *ResNet101*, *Sequeeze1.0* e *Inception-V4*.

O conjunto de dados utilizado foi *UTKFace dataset* que possui 9780 imagens com idades compreendidas entre os 0 e 111 anos. Este *dataset* foi dividido em 3 partes, 70% para treino, 20% para validação e 10% para teste. As imagens foram formatas de acordo com as necessidades das redes pré-treinadas. Assim sendo, para o modelo *Inception-V4* as imagens foram formatas com 299x299x3, e para os restantes modelos as imagens foram formatas com 224x224x3.

O conjunto de dados foi treinado com 3 classes (intervalos) de idades: Classe A – 0 até 17, Classe B – 18 até 36, Classe C – maiores 36.

Não se verificaram grandes disparidades na quantidade de imagens para cada uma das classes, ou seja, o *dataset* encontrava-se equilibrado para as classes estabelecidas.

A rede que apresentou melhores resultados foi a *Inception-V4* com um *F1-score* de aproximadamente 84% e uma *accuracy* de 86%. A rede com pior desempenho foi a *ResNET-50* com uma *accuracy* inferior a 79%.

Modelo	Loss	Accuracy	Recall	Precision	F1-score
ResNet-50	0,5024	0,7822	0,7467	0,7583	0,7489
Resnet-101	0,4921	0,8026	0,7815	0,7828	0,7820
SqueezeNet-V1.0	0,4801	0,8026	0,7907	0,7873	0,7850
Inception-V4	**0,3839**	**0,8579**	**0,8357**	**0,8384**	**0,8369**

Tabela 2 – Comparação de modelos através de várias métricas.

Pode-se constatar que todas as redes apresentam bons resultados, porém deve ser salientado que estamos a estimar a idade utilizando somente 3 classes finais, ou seja, 3 conjuntos de faixas etárias. Para além do reduzido número de classes utilizado, a terceira classe contempla indivíduos na idade adulta com mais do que trinta e cinco anos, o que inclui as faixas etárias onde os algoritmos de classificação apresentam piores resultados. Tal como mencionado anteriormente, as alterações físicas na idade adulta são mais superficiais, o que dificulta a diferenciação entre idades e assim sendo afeta o desempenho dos algoritmos.

Como trabalho futuro os autores propõem a criação de mais uma classe, o que perfaz um total de 4, com a finalidade de criar os grupos crianças, jovens, adultos e idosos.

2.6.2 Age Estimation from Face Images Based on Deep Learning [28]

Este trabalho expõe 4 populares datasets. O primeiro é o IMDB-WIKI que possui um elevado número de imagens, no entanto a qualidade das imagens é reduzida e existem algumas imagens vazias. O conjunto *OIU-Adience* possui imagens distribuídas por 8 grupos de idade. Os *datasets* MORPH-II e *FG-Net* são dois dos conjuntos mais mencionados em trabalhos científicos e já anteriormente apresentados.

O autor deste trabalho refere o trabalho de *Agbo-Ajala et al* [24] onde os autores pretendem classificar a idade e género com base em imagens. A metodologia aplicada consistiu em recorrer a um modelo pré-treinado com o conjunto de dados IMDB-WIKI e de seguida foi realizado *fine-tuning* com os dados provenientes do MORPH-II. O processo de *fine-tuning* consistiu em alterar as últimas camadas do modelo para tentar otimizar os resultados.

Este trabalho pretende recorrer a arquiteturas anteriormente desenvolvidas para classificar idades a partir de imagens. As redes usadas foram: *InceptionResNetV2, Xception, DenseNet121, ResNet152V2,* VGG-16 e *InceptionV3*. O objetivo é comparar a eficácia das várias redes e, portanto, não se personalizam as redes. O *dataset* final utilizado, foi criado com base nos *datasets* mencionados, e inclui 4970 imagens de faces e possui idades entre 1 e 70 anos.

Segundo os autores, abordar o problema de deteção de idade como um problema de classificação, ou seja, estimar a faixa etária é mais simples do que estimar a idade exata. Daí que este trabalho tenha usado as várias redes mencionadas para comparar os resultados em

dois cenários diferentes. No primeiro são criados 10 grupos com 7 anos cada e no segundo são criados 70 grupos (1 ano cada).

Após uma breve introdução às diferentes arquiteturas apresentadas, é mencionado que se recorreu ao algoritmo *Adam*. Este algoritmo é baseado num gradiente de primeira ordem de funções objetivas estocásticas e é um dos algoritmos de otimização mais populares. Foi também implementado o mecanismo *Early Stopping* em que, quando a validação de performance não melhora em cinco iterações seguidas, o processo de treino é terminado.

Verificando os resultados pode-se constatar que a rede desenvolvida com base no modelo *Xception* obteve o melhor resultado quer para o cenário com 10 grupos quer para o cenário com 70 grupos. Este modelo apresentou uma *accuracy* de 60% para a classificação do primeiro cenário (10 classes) e aproximadamente 24% para a classificação do segundo cenário (70 classes). O modelo que apresenta piores resultados é o *Inception-V3* apresentando uma *accuracy* de 50% para o primeiro cenário e 19% para o segundo.

Como conclusão o autor refere que existe uma grande disparidade entre os resultados finais obtidos e um resultado satisfatório que possa ser utilizado no mundo real.

2.6.3 Deep Learning Based Real Age and Gender Estimation from Unconstrained Face Image towards Smart Store Customer Relationship Management [18]

A pandemia resultante do COVID-19 alterou o modo como se realiza compras, uma vez que se passou a privilegiar um sistema de compras sem contacto. Para automatizar o processo de compras, a deteção automática de género e de idade é relevante pois permite uma experiência personalizada ao cliente. Através da informação do género e faixa etária existe a possibilidade de restringir determinados produtos, bem como de determinar qual publicidade se adequa ao cliente. Existe ainda a possibilidade de personalizar a comunicação para com o cliente. Assim sendo, um sistema de aprendizagem profunda integrado numa loja inteligente automatizada, oferece garantias para os vendedores num período de incertezas. Para além de representar um bom investimento a longo prazo com a redução de mão-de-obra.

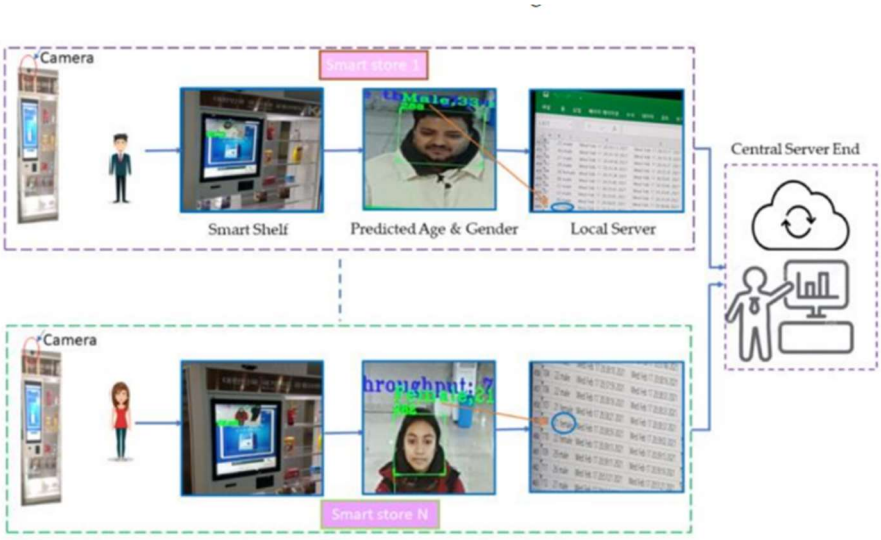


Figura 13 – Processo de caracterização do cliente com um sistema automatizado [18].

O objetivo deste trabalho científico é construir um sistema de deteção de género e idade. Numa fase inicial, as imagens são tratadas para solucionar problemas como expressões, poses ou iluminação. Apesar de as redes neuronais convolucionais serem capazes de lidar com erros de alinhamento em imagens, verificou-se que estes erros podem afetar a performance das redes e, assim sendo, neste trabalho foi realizado um pré-processamento das imagens. Quer a deteção de género quer a deteção de idade são ambas consideradas, neste caso, um problema de classificação. Recorreu-se à arquitetura VGG-16 devido ao facto de ser uma rede que apresenta uma boa performance para reconhecimento de imagens, facilmente disponível e para tarefas de classificação os modelos pré-treinados públicos são uma mais-valia para o início do processo de treino.

Para treinar adequadamente uma rede profunda é necessária uma grande quantidade de dados, caso contrário pode ocorrer *overfitting,* ou seja, que o modelo se ajuste com o conjunto de treino e perca a capacidade de generalizar com novas imagens. Para treinar a rede foi utilizado o *dataset IMDB-WIKI*, sendo que este foi complementado com o intuito de aumentar a qualidade dos dados. Foi assegurado que o *dataset* a utilizar, na fase de treino, consiste num conjunto equilibrado onde todas as classes são aproximadamente representadas. Para concretizar esta condição, foram importadas imagens de outros *datasets* e foi realizado o processo de aumentar a quantidade e diversidade de dados (*Data Augmentation*). Para o conjunto de treino foram selecionadas 80% das imagens, sendo os restantes 20% usados para testar o modelo.

Tal como mencionado anteriormente, antes do processo de extração de características de imagens foi executado um pré-processamento onde as imagens são alinhadas e as suas escalas são ajustadas. As imagens são colocadas no formato 256x256 píxeis e é recortado o seu centro com 224x224 píxeis. O modelo VGG-16 recebe por defeito imagens RGB com 224x224 píxeis. A rede foi iniciada com os pesos obtidos a partir da *ImageNet*. A nova rede recorreu à estrutura da VGG-16 para a extração de características e alterou as camadas finais. Aplicou-se uma

camada *Softmax* para classificação de resultados independentemente da tarefa (deteção género ou idade). No entanto, no caso de deteção de idade existe um passo adicional onde é calculada a idade final prevista.

A deteção de idade consiste num problema de regressão uma vez que idade consiste num valor contínuo em vez de um conjunto discreto de classes. No entanto, por norma, as redes CNN que são treinadas para abordar problemas de regressão, apresentam uma elevada taxa de erros. Estes erros surgem devido à instabilidade dos modelos em lidar com *outliers*. Como resultado, as redes exibem dificuldades em convergir num valor e as previsões tornam-se instáveis. Para evitar estes problemas, a abordagem do presente trabalho consiste em considerar, numa fase inicial, o problema como um problema de classificação. Onde a imagem é classificada consoante um conjunto de grupos de idades. São criados 101 categoriais de 1 ano cada, compreendendo assim as idades de entre os 0 e os 100 anos. De forma a resolver o problema como um problema de regressão a última camada da rede é alterada e é aplicada uma função euclidiana para a tarefa de regressão. Esta função consiste no somatório da multiplicação do valor de cada categoria pela sua probabilidade.

A experiência demonstrou que os resultados obtidos com a aplicação da função euclidiana foram superiores aos resultados de aplicar diretamente a rede CNN como regressão.

Para comparar os resultados de deteção de idade, com outros trabalhos científicos, o modelo foi avaliado usando o *dataset MORPH*. O resultado final para o *Mean Average Error* (MAE) foi de 2.42, que consiste num resultado acima da média dos resultados analisados. Na tabela seguinte são apresentados alguns dos resultados de trabalhos que usaram o *dataset MORPH*.

Métodos	**AGES**	**Rothe el al.**	**Proposta**	**Liu et al.**
MAE	8.83	3.25	2.42	2.32

Tabela 3 – Comparação de resultados de diferentes trabalhos semelhantes.

A funcionalidade de deteção de género, como era de prever, obteve um melhor desempenho uma vez que se trata de um problema com menor complexidade e de classificação binária.

2.6.4 Age Estimation From Facial Image Using Convolutional Neural Network [30]

Neste trabalho desenvolveu-se um modelo com base na arquitetura ResNet50. O trabalho foi abordado com um problema de regressão.

Numa fase inicial as imagens são pré-processadas, o que inclui a deteção da face recorrendo à biblioteca *dlib*, a formatação das imagens de acordo com as necessidades da arquitetura ResNet50 (224x224 píxeis). Aplica-se a técnica *One Hot Encoding*, onde os valores da idade real de cada imagem são convertidos em vetores (*one hot vectors*).

Para treinar o modelo combinaram-se os *datasets APPA-Real* e *UTKFace*, resultando em mais de 27 mil imagens de faces. Para melhorar a dimensão do dataset de treino aplicou-se ainda *data augmentation.* O processo de *data augmentation* abrangeu rotação, alteração do zoom, distorção, alteração de contraste e brilho. Na fase de teste foi utilizado o *dataset FG-NET*.

A camada final da rede baseada na arquitetura *ResNet50* foi removida e foi acrescentada uma nova camada *fully connected,* de tamanho igual ao número de elementos utilizados no treino do modelo. A rede foi iniciada com os pesos adquiridos a partir da *ImageNet*. Foram testadas duas funções de otimização. A *Adam,* que tende a ser a mais popular, e a *SGD.*

Optimizador	MAE
Adam	5.3
SGD	4.5

Tabela 4 – Comparação resultados para diferentes optimizadores.

Tal como se pode constatar o optimizador *SGD* apresenta melhores resultados que o *Adam*. Os valores finais são ainda comparados com outros trabalhos equivalentes e verifica-se que o resultado com o SGD consiste num dos melhores resultados obtidos. É ainda salientada pelos autores a importância do processo de *data augmentation*, uma vez que permitiu ampliar o *dataset* de treino.

2.6.5 Análise Trabalhos

No primeiro trabalho, o autor compara a eficácia de algumas das arquiteturas mais populares num problema de classificação. Neste trabalho é interessante a comparação entre os resultados das várias arquiteturas e não os resultados em si, pois devido ao reduzido número de classes todos os modelos apresentaram bons resultados.

O segundo trabalho para além de analisar alguns dos *datasets* mais populares, o autor tenta comparar a eficácias de várias redes CNN. Em comparação com o primeiro trabalho, a abordagem ao problema foi equivalente (classificação), no entanto neste caso foram definidos dois cenários: 10 classes e 70 classes. Os resultados finais comprovam que quando é maior o número de classes, resulta num menor intervalo de idades, pior é a taxa de acerto das redes. A rede *Xception*, que apresentou os melhores resultados, obteve somente 60% de *accuracy* para o primeiro cenário (10 classes), o que não pode ser considerado um valor adequado. Comparando este trabalho com outros trabalhos científicos, constatamos que o resultado tende a ser inferior, porém foram criadas 10 classes uniformes de 7 anos cada. Ao contrário de diversos trabalhos da área que tendem a selecionar conjuntos de faixas etárias específicas de acordo com o processo de envelhecimento (lactentes, crianças, adolescentes, jovens, adultos, idosos).

O terceiro trabalho apresenta maior complexidade que os anteriores. A tarefa foi abordada como um problema de regressão. Neste trabalho para além do pré-processamento de imagens, aplicou-se a técnica de *fine-tuning* sobre uma rede baseada na arquitetura VGG. As últimas

camadas da rede foram alteradas e o algoritmo *Softmax* é utilizado. A última camada da rede aplica uma função euclidiana, que converte os valores obtidos com o algoritmo Softmax no valor final da previsão da idade. É ainda mencionado que o uso deste sistema alternativo em que, a tarefa é inicialmente abordada com um problema de classificação e posteriormente através da função euclidiana convertida num problema de regressão, apresentou melhores resultados do que aplicar diretamente a CNN como um problema de regressão. Analisando a comparação final com outros trabalhos análogos podemos concluir que o resultado final foi satisfatório.

No último trabalho, tal como o terceiro, a tarefa foi abordada como um problema de regressão. Este trabalho inclui o pré-processamento das imagens, *data augmentation* e *fine-tuning* sobre uma rede baseada na arquitetura ResNet50. Realiza-se a comparação entre duas funções de otimização distintas, sendo que a SGD apresenta um melhor resultado final. Os resultados obtidos são favoráveis quando comparados com trabalhos semelhantes.

Os resultados entre os quatro trabalhos não são comparáveis uma vez que os trabalhos, para além de usarem diferentes conjuntos de dados, têm como objetivo estudar aspetos distintos.

3 Análise de valor

A automatização do processo de deteção de idade com base em fotografias de faces humanas possui inúmeras utilizações no mundo real.

A pandemia resultante do COVID-19 veio evidenciar a importância das lojas inteligentes onde, não existe necessidade de interação entre o vendedor e os clientes. Para facilitar o método de compras e promover o futuro das lojas inteligentes, um sistema capaz de estimar a idade e género dos clientes é indispensável. A construção e integração de um sistema capaz de detetar a idade de um ser humano, de forma automática, somente com base numa imagem permite diminuir os custos de uma organização. Para além da redução dos custos de mão-de-obra, este sistema permite aumentar a eficiência de outros processos, nomeadamente as campanhas de marketing conseguem promover produtos específicos para a faixa etária dos clientes.

Os benefícios previamente mencionados permitem, a uma organização, realizar um estudo da viabilidade financeira do projeto. No entanto, existem benefícios em que é de extrema dificuldade quantificar as mais valias económicas. Como por exemplo, o controlo de acesso a sites por parte de crianças menores, evitando a exposição a conteúdo indesejado.

Com base nos excelentes resultados apresentados pelas redes neuronais convolucionais em tarefas semelhantes, este tipo de rede é o candidato ideal para o sistema a conceber. O sistema pode recorrer a uma rede pré-treinada para melhorar o desempenho e reduzir a complexidade na sua conceção. Porém, para selecionar uma rede pré-treinada é necessário comparar os resultados científicos publicados para apurar a escolha ideal. Para selecionar a melhor opção de forma coerente, pode-se recorrer a uma análise multicritério, sendo necessário definir os parâmetros base da avaliação.

3.1 Valor

A *Society of American Value Engineers* (SAVE) define análise de valor como uma abordagem sistemática e estruturada para melhorar projetos, produtos e processos. Com esta abordagem é possível alcançar um equilíbrio entre qualidade, função, desempenho, segurança e custo [31].

A fórmula utilizada, por SAVE, para definir valor consiste em:

$$Valor = \frac{Função}{Custo}$$

Assim sendo, podemos constatar que o valor é igual às funcionalidades do produto sobre os recursos necessários para a sua conceção [31].

3.2 New Concept Development Model

O *New Concept Development Model* (NCD) consiste num modelo para guiar o processo de desenvolvimento de um novo produto ou ideia. Este modelo subdivide-se em 3 elementos:

- O centro do modelo, o motor, que é responsável pela estratégia e visão holística.
- Os elementos que fazem parte do *Front End of Inovation*. Estes cinco elementos descrevem os procedimentos desde uma fase inicial em que se está a identificar as oportunidades até à fase de implementar a ideia.
- Os fatores de influência que representam os fatores externos.

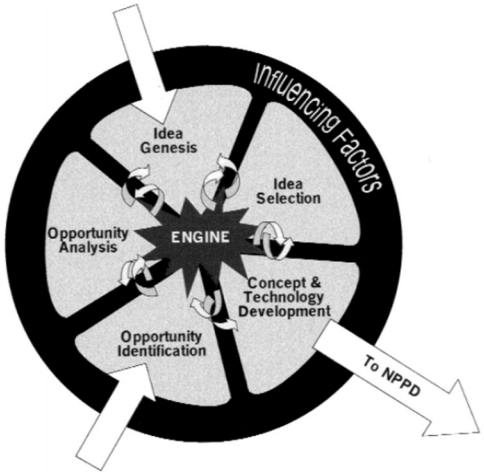


Figura 14 – Representação do *New Concept Developemnet Model* (NCD) [32].

3.2.1 Identificação da oportunidade

Esta fase inicial caracteriza-se pela identificação das oportunidades que se enquadrem na visão estratégica da organização. Inclui diversas técnicas nomeadamente o *brainstorming* ou mapeamento mental.

Um sistema capaz de definir características demográficas, nomeadamente a idade, com base em imagens possui uma imensa aplicabilidade nos mercados. Diversas tarefas do cotidiano são executadas por computadores com maior velocidade, eficácia e de forma mais económica. A automação do processo de deteção de idade com base em fotografias e, ou imagens pode ser aplicada em ferramentas de marketing para personalizar a mensagem consoante a idade do publico alvo. Pode ser aplicada em equipamentos com limites de idade, automatizando o processo e reduzindo os custos, um exemplo são as máquinas de tabaco.

3.2.2 Análise de oportunidades

Numa segunda fase é relevante estudar o mercado e perceber como a oportunidade identificada por ser aproveitada por uma organização.

Um investimento no desenvolvimento de um projeto de deteção de idade permitiria automatizar tarefas que hoje em dia são realizadas por seres humanos. Este facto pode permitir a redução em custos de mão-de-obra e aumento de produtividade, o que implica valor para o mercado. Uma outra possibilidade para este investimento seria bloquear o acesso a websites consoante a idade do utilizador, prevenindo assim o acesso de crianças a conteúdo sensível ou impróprio.

3.2.3 Conceção e desenvolvimento de ideias

Após estudar diversos trabalhos científicos publicados neste tópico, pode-se concluir que existem diversas metodologias para atingir o objetivo final, deteção de idade com base em imagens. Entre as soluções mais populares encontram-se as técnicas de *deep learning*, bem como técnicas de extração de características seguidas de aplicação de algoritmos de classificação. Apesar dos bons resultados apresentados por todos estas técnicas, nos dias que correm a precisão da estimativa de idade humana, de forma automática, ainda não conseguiu ultrapassar a precisão apresentada por humanos.

3.2.4 Seleção de ideias

Esta fase destaca-se pela seleção e maturação da ideia de forma a alcançar a melhor solução.

Face aos excelentes resultados apresentados pelas técnicas de *deep learning*, nomeadamente redes neuronais convolucionais, optou-se por recorrer a esta técnica. Este tipo de redes

neuronais tem-se distinguido em trabalhos de *Computer Vision*, tornando-a na mais popular, devido à sua eficácia. O desenvolvimento de uma rede neuronal convolucional eficaz de raiz implica um alto nível de complexidade e um conjunto de dados de elevada qualidade e dimensão. De forma a otimizar o produto final e tirar partido de trabalhos anteriormente desenvolvidos, pode-se recorrer a redes pré-treinadas. Estas redes podem ser utilizadas para auxiliar o processo de extração de características. De salientar que existem inúmeras redes pré-treinadas.

3.2.5 Definição do conceito

Para finalizar deve ser desenvolvido o modelo de negócio e uma correta definição do conceito.

Pretende-se, desenvolver um sistema capaz de detetar imagens de faces, em tempo real, e com base nas imagens estimar a idade de seres humanos. O sistema é baseado em redes neuronais convolucionais e recorre a redes pré-treinadas para agilizar a sua construção. Tal como mencionado, este sistema pode ser aplicado em diversos tipos de negócio.

De seguida apresenta-se o modelo *Canvas* para o sistema descrito.

3.3 Modelo Canvas

O modelo *Canvas* surge em 2008 através de *Alexander Osterwalder*, e tem sido amplamente utilizado na definição de planos de negócio. Consiste numa ferramenta prática e versátil que permite uma visão holística, e deste modo detetar aspetos fundamentais de um negócio [33].

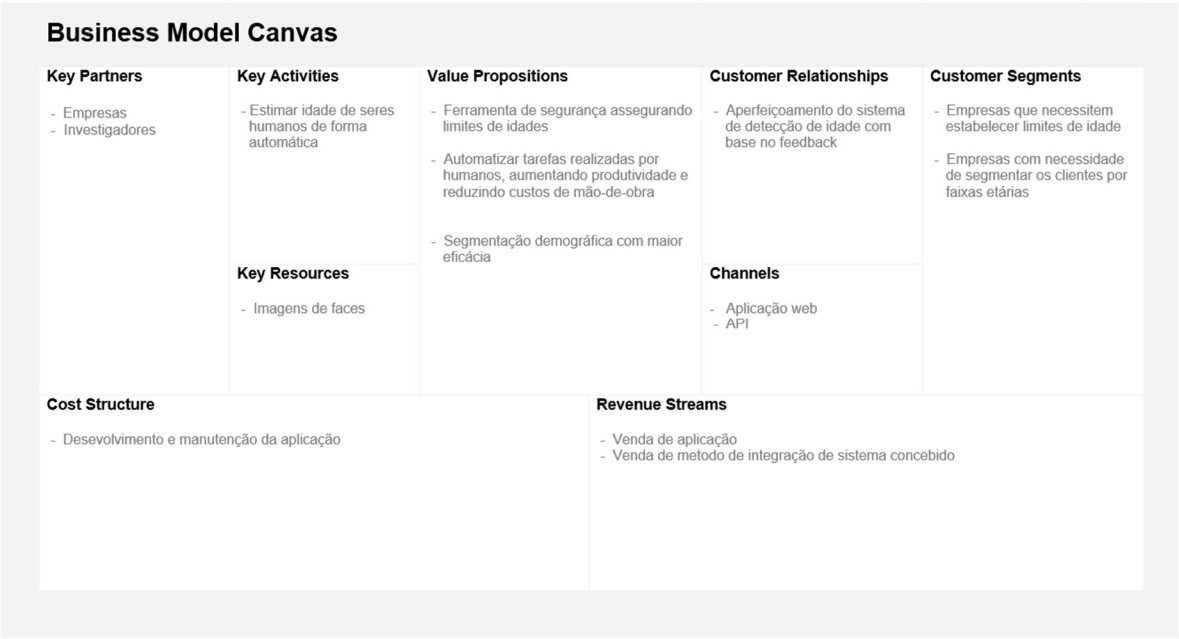


Figura 15 – Modelo de negócio *Canvas*.

3.4 Avaliação de hipóteses

A implementação ideal do sistema está dependente da escolha da melhor rede pré-treinada a utilizar na rede neuronal convolucional. Esta escolha pode influenciar a performance do modelo e o seu desenvolvimento. Será realizada uma análise multicritério de apoio à decisão para selecionar a melhor opção.

A análise de decisão multicritério tem como objetivo minimizar a complexidade de uma tomada de decisão, auxiliando os decisores a selecionar uma opção de um conjunto de alternativas. Baseia-se numa abordagem quantitativa para apoiar o processo de tomada de decisão, de modo a potenciar aos responsáveis uma visão abrangente do problema em questão [34].

Um dos principais métodos desenvolvidos no ambiente das decisões multicritério discretas é o Método de Análise Hierárquica (AHP – *Analytic Hierarchy Process*). Este método, desenvolvido na década de 70 por *Thomas Saaty*, permite tomar uma decisão fundamentada recorrendo a cálculos matemáticos. Este método baseado em critérios qualitativos e quantitativos divide o problema em níveis hierárquicos com o intuito de facilitar a compreensão e avaliação [35].

Com a aplicação do método AHP pretendemos determinar qual a melhor abordagem ao problema e selecionar a rede pré-treinada que melhor se enquadra. Os critérios a utilizar serão:

- Complexidade
- Custo desenvolvimento
- Eficácia

O critério complexidade é relativo à dificuldade em aplicar a rede e o tempo necessário para a sua implementação. O Custo de desenvolvimento diz respeito à capacidade de processamento necessária para usar o modelo. Por fim, o último critério tem o intuito de comparar a eficácia apresentada pelas redes em trabalhos científicos publicados.

Existem 4 alternativas:

- VGG-16
- Inception-V4
- ResNet50
- Xception

Todas estas redes são mencionadas no capítulo anterior em maior detalhe.

Para comparar os diversos critérios é necessário atribuir um grau de importância aos conjuntos. Para esta atribuição recorre-se à escala fundamental dos números absolutos, também

conhecida como escada de *Saaty*. Esta escala numérica possui valores entre 1 e 9. Onde 1 representa igual importância entre os critérios e 9 um dos critérios é extremamente mais relevante [35].

	Complexidade	**Custo desenvolvimento**	**Eficácia**
Complexidade	1,00	2,00	0,33
Custo desenvolvimento	0,50	1,00	0,25
Eficácia	3,00	4,00	1,00
TOTAL	**4,50**	**7,00**	**1,58**

Tabela 5 – Matriz de comparação de critérios

Após a criação da tabela é necessário normalizar os seus valores. Assim sendo procede-se com a divisão de todos os valores pelo total da sua coluna. Os valores normalizados permitem calcular o vetor de prioridades, que indica a ordem de importância de cada critério. Este vetor consiste na média aritmética dos valores de cada um dos critérios.

	Complexidade	**Custo desenvolvimento**	**Eficácia**
Complexidade	0,22	0,29	0,21
Custo desenvolvimento	0,11	0,14	0,16
Eficácia	0,67	0,57	0,63

Tabela 6 – Matriz de comparação de critérios normalizada

	Prioridade Relativa
Complexidade	(0,22+0,29+0,21)/3 = **0,24**
Custo desenvolvimento	(0,11+0,14+0,16)/3 = **0,14**
Eficácia	(0,67+0,57+0,63)/3 = **0,62**

Tabela 7 – Vetor de prioridades.

Analisando os resultados da tabela anterior podemos concluir que o critério com maior relevância é a eficácia apresentada. O critério com menor relevância, dos três, é o custo de desenvolvimento. Uma justificação para o reduzido valor do critério custo desenvolvimento é o facto de atualmente diversas empresas permitirem, de forma gratuita, acesso a equipamentos na *cloud* com processadores GPU.

Face aos valores atribuídos a cada par de critérios, é necessário validar a consistência dos valores, através do rácio de consistência (RC). Os valores são considerados consistentes se o valor de RC for menor que 0,10 (10%). O rácio da consistência é calculado através da divisão entre o índice de consistência (IC) e o índice de consistência aleatória (IR).

Por sua vez, o CI é calculado através da seguinte equação:

$$CI = \frac{\lambda\, max - n}{n-1} = \frac{3{,}03 - 3}{3-1} = 0{,}013$$

Onde:

$\lambda\, max$ – representa o maior valor próprio da matriz.

n – número de critérios.

$$\lambda\, max = 0{,}24 * 4{,}5 + 0{,}14 * 7 + 0{,}62 * 1{,}58 = 3{,}03$$

Assim sendo,

$$RC = \frac{IC}{IR} = \frac{0{,}013}{0{,}58} = 0{,}02 < 0{,}1$$

Verificamos que o valor do rácio da consistência é inferior a 10% o que indica que os valores das prioridades relativas estão consistentes.

De forma a decidir a melhor abordagem com base nos critérios é necessário criar matrizes de comparação paritária para cada critério.

Complexidade

	VGG-16	**Inception-V4**	**ResNet50**	**Xception**	**Vetor Prioridade**
VGG-16	1	3	3	5	0,52
Inception-V4	1/3	1	1	3	0,20
ResNet50	1/3	1	1	3	0,20
Xception	1/5	1/3	1/3	1	0,08

Tabela 8 – Matriz comparação critério Complexidade

Custo Desenvolvimento

	VGG-16	Inception-V4	ResNet50	Xception	Vetor Prioridade
VGG-16	1	1/3	1/4	1/2	0,10
Inception-V4	3	1	1/2	2	0,28
ResNet50	4	2	1	3	0,47
Xception	2	1/2	1/3	1	0,16

Tabela 9 – Matriz comparação critério Custo Desenvolvimento

Eficácia

	VGG-16	Inception-V4	ResNet50	Xception	Vetor Prioridade
VGG-16	1	1/2	2	1/3	0,16
Inception-V4	2	1	3	1/2	0,28
ResNet50	1/2	1/3	1	1/4	0,10
Xception	3	2	4	1	0,47

Tabela 10 – Matriz comparação critério Eficácia

De seguida pretende-se obter a prioridade composta para as alternativas

$$\begin{matrix} 0{,}52 & 0{,}1 & 0{,}16 \\ 0{,}71 & 0{,}16 & 0{,}25 \\ 0{,}71 & 0{,}25 & 0{,}08 \\ 0{,}25 & 0{,}1 & 0{,}38 \end{matrix} \quad \mathbf{x} \quad \begin{matrix} 0{,}24 \\ 0{,}14 \\ 0{,}62 \end{matrix} \quad = \quad \begin{matrix} 0{,}24 \\ 0{,}25 \\ 0{,}17 \\ \mathbf{0,33} \end{matrix}$$

A alternativa Xception surge como a mais indicada para desenvolver o modelo, em função dos critérios definidos e das suas respetivas importâncias. De seguida, com praticamente o mesmo valor, surgem as alternativas VGG-16 e Inception-v4. A alternativa considerada menos indicada é a rede ResNet50.

Face à inexistência de pré-requisitos no atual projeto, foi acordado com a Docente Susana Nicola que não fazia sentido implementar as técnicas QFD (*Quality Function Deployment*) e FAST (*Function Analysis and System Technique*).

4 Análise e design

Neste capítulo realiza-se uma análise ao sistema do ponto de vista estrutural e definem-se suas funcionalidades.

A solução proposta consiste numa aplicação web onde, através do uso da *webcam*, será possível recolher uma imagem, como *input* do sistema. Como alternativa ao uso da *webcam*, é possível também realizar o *upload* de uma fotografia. O utilizador somente necessita de utilizar o *browser* para aceder à aplicação, sendo que o processamento será realizado num servidor. Face às necessidades do sistema, não será necessário incluir qualquer tipo de sistema de *login*, nem será necessário guardar dados dos utilizadores. De seguida são assim apresentados os dois casos de uso possíveis da aplicação para o utilizador.

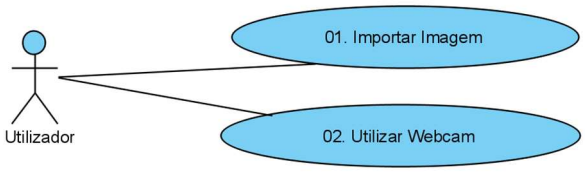


Figura 16 – Casos de usos.

4.1 Estrutura do sistema

O funcionamento do sistema pode ser subdivido em 4 módulos.

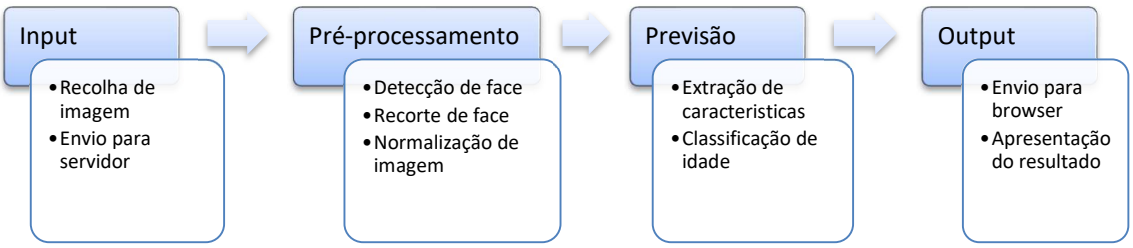


Figura 17 – Estrutura geral do sistema.

4.1.1 Input

O primeiro módulo tem o intuito de recolher a imagem do individuo através da *webcam* de um computador ou por *upload* e transferir para o segundo módulo. Para realizar estas tarefas o utilizador deve utilizar a aplicação web.

4.1.2 Pré-processamento

O segundo módulo, de pré-processamento, é responsável por preparar a imagem. Neste módulo a face é selecionada, recortada e de seguida a imagem é normalizada, de acordo com as necessidades da rede CNN seguinte. Este módulo permite tratar as imagens, de tal modo que para o modulo seguinte, são enviadas imagens de faces sem informações extra desnecessárias, tais como ruído de fundo. Através da biblioteca *OpenCV* a imagem é processada pela classe *CascadeClassifier* que, utiliza a técnica *Haar-like* para detetar a imagem e obter as suas coordenadas [4]. Esta técnica é também usada para validar se a imagem é válida e se contém uma face humana. De seguida com base nas coordenadas a face é recortada e formatada de acordo com as necessidades das arquiteturas a implementar. Por fim a imagem é normalizada reduzindo o valor dos seus pixéis à mesma escala.

4.1.3 Previsão

O terceiro módulo é constituído por uma rede neuronal convolucional, que com base na imagem da face consegue prever a idade do indivíduo. Este módulo tem por base uma rede pré-treinada de forma a abstrair a complexidade do processo de conceção da rede neuronal convolucional de raiz. De forma a otimizar o desempenho do sistema, serão utilizadas diferentes redes pré-treinadas. A rede que apresentar melhores resultados será integrada na aplicação final. As redes pré-treinadas a utilizar são:

- VGG-16

- Xception
- Inception-V4

Pretende-se estimar a idade do indivíduo em análise aplicando duas abordagens distintas, são elas regressão e classificação. A regressão tem como objetivo estimar um valor numérico específico. A classificação resume-se a estimar o intervalo de idades em que o utilizador se enquadra. O uso de ambas as abordagens, permite criar modelos distintos e implementar a melhor opção no sistema final

4.1.4 Output

No último modulo a previsão final é enviada para o browser e o resultado é apresentado ao utilizador através da aplicação web.

4.2 Arquitetura

O projeto será baseado numa arquitetura Cliente-Servidor. Este tipo de arquitetura garante fácil acesso à aplicação por parte dos utilizadores, não sendo necessário instalar uma aplicação especifica. Através do *browser* será possível ter acesso à aplicação, que comunicará com o servidor através de pedidos HTTP. A linguagem de programação do lado do servidor será *Python*.

O servidor será responsável pelo pré-processamento das imagens e por implementar o modelo desenvolvido. Do lado do servidor será necessário desenvolver uma API.

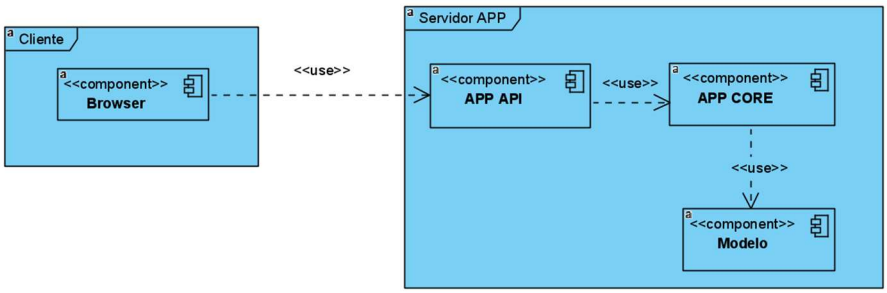


Figura 18 – Arquitetura do sistema.

Para o desenvolvimento deste sistema será necessário treinar o modelo. Para executar o treino do modelo recorreu-se ao *Google Colaboratory (Colab)*, que consiste numa plataforma online que permite executar código *Python* no browser. Esta plataforma garante acesso, de forma gratuita, a processadores GPU com 12 GB de RAM. O que face ao elevado poder computacional necessário para treinar o modelo é fundamental. No final do treino do modelo pretende-se compilar os resultados, criar um ficheiro no formato H5 e guardar no servidor. Este ficheiro contém os pesos e a configuração do modelo.

5 Implementação e avaliação dos modelos

O presente capítulo tem o intuito de descrever o trabalho experimental realizado para tentar criar o modelo com o melhor desempenho.

Numa fase inicial será descrito o trabalho realizado na preparação dos dados, sendo posteriormente apresentados e avaliados os vários modelos de classificação e de regressão.

Relativamente ao problema de classificação foram definidas oito faixas etárias que se pretendem prever. As faixas etárias definidas tiveram em conta idades com relevância para campanhas de marketing e promoção de produtos. Os 8 grupos são:

- Grupo1: 0 a 2 anos
- Grupo2: 3 a 5 anos
- Grupo3: 6 a 10 anos
- Grupo4: 11 a 17 anos
- Grupo5: 18 a 25 anos
- Grupo6: 26 a 40 anos
- Grupo7: 41 a 65 anos
- Grupo8: Maiores que 66 anos

Relativamente ao problema de regressão, não é necessário definir grupos uma vez que se pretende estimar a idade exata do indivíduo. Porém, foi definido que o intervalo de idades relevante a prever seria entre os 0 e os 85 anos.

De acordo com a metodologia CRISP-DM adotada neste trabalho, passa-se em seguida a descrever as suas várias fases.

5.1 Seleção dos dados

Para o desenvolvimento do trabalho tentou-se obter os melhores conjuntos de dados uma vez que, tal como mencionado no segundo capítulo, a quantidade e qualidade de dados possui influência nos resultados finais.

Os *datasets* utilizados foram o *UTKFace*, APPA-*Real*, *FacialAge e* IMDB. Foram ainda realizados alguns testes com imagens do FGNET.

O *dataset* designado por *FacialAge* é baseado no *dataset* WIKI e resulta de uma limpeza dos dados, removendo imagens sem faces ou com idades incorretas. Este conjunto de dados conta com idades dos 0 aos 110 e com um total de 9778 imagens. O conjunto de dados original não possui a qualidade desejada, sendo que seria necessário realizar uma limpeza profunda ao mesmo. Recorrendo a este *dataset*, agilizou-se o processo de seleção de imagens e obteve-se diretamente um conjunto de dados com melhor qualidade.

O *dataset* MORPH, considerado um dos melhores e mais completos *datasets* para deteção de idades com base em faces, não foi possível obter uma vez que o mesmo não está disponível de forma gratuita.

O *dataset* IMDB apesar de se tratar de um conjunto de dados volumoso, consiste no conjunto de dados com pior qualidade. Possui inúmeras imagens sem faces humanas, possui imagens em que a idade indicada não está correta. Consequentemente, e de forma a tentar aproveitar alguns dados, filtraram-se as imagens com melhor qualidade. Os dados foram tratados e verificou-se uma melhoria relativamente a imagens vazias ou incorretas, no entanto a qualidade dos dados existentes não é garantida, portanto utilizou-se este *dataset* em primeiro lugar. A utilização deste conjunto de dados em primeiro lugar tem o intuito de realizar um pré-treino na rede e desenvolver assim uma versão base. Este *dataset* será referenciado posteriormente como *dataset A*.

De seguida os *datasets FacialAge*, *APPA-Real* e *UTKFace* foram tratados e posteriormente foram combinados de forma a aumentar o volume de dados. Uma vez que se verificava uma disparidade na quantidade de dados para cada classe, os *datasets* foram balanceados. A combinação destes três *datasets* será posteriormente referenciada como *dataset B*. Inicialmente os *datasets* foram utilizados individualmente, no entanto, durante o processo de uniformização de classes, verificou-se que existia um desaproveitamento de dados na eliminação de imagens nas classes com maior representatividade.

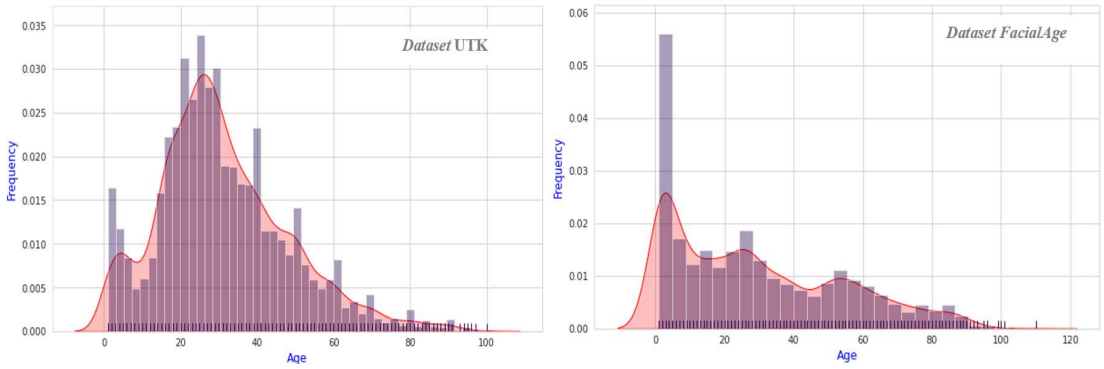


Figura 19 - Distribuição de idades nos *datasets UTKFace* e *FacialAge.*

Este desaproveitamento deve-se ao facto dos *datasets* possuírem diferentes percentagens para cada classe e, deste modo, a sua combinação auxilia na redução de algumas das discrepâncias verificadas.

5.2 Limpeza dos dados

A limpeza dos dados no *dataset A* (*IMDB*) contempla a remoção de imagens com idade negativa ou valores demasiado elevados (>101 anos). Bem como, através do uso de um detetor de faces, a remoção de imagens que não possuem faces ou que possuem mais do que uma face (uma vez que só existe um registo de idade para cada imagem). Após este processo foi realizada uma inspeção às restantes imagens, para tentar averiguar se persistiam imagens vazias ou imagens com valores incorretos. Apesar de todo este processo, a qualidade das imagens não está assegurada, nem a certeza no registo de idade para cada imagem.

Para o *dataset APPA-Real*, foi obtida uma lista de identificadores referentes a imagens indesejadas ou com dados errados. Estas imagens foram removidas com um script. Após uma inspeção visual aos restantes dados foram encontradas imagens sem faces. Estas imagens foram adicionadas ao script anterior, otimizando-o.

Relativamente ao *FacialAge*, foi também obtida uma lista de identificadores de imagens que não possuem a idade correta. Essa lista de imagens foi removida, evitando assim erros no processo de treino.

O conjunto de dados *UTKFace* não foi alvo de qualquer intervenção, uma vez que através de inspeção visual não foram detetados problemas.

Nos cenários criados em que o problema foi interpretado como regressão, todo o processo é semelhante, no entanto todas as imagens com idade superior a 85 anos são descartadas. Após os 85 anos a quantidade de faces por idade é escassa, sendo que existem idades que não possuem qualquer registo fotográfico.

Após o processo de limpeza a distribuição de dados por cada grupo é muito irregular para ambos os *datasets*, tal como se pode constatar de seguida.

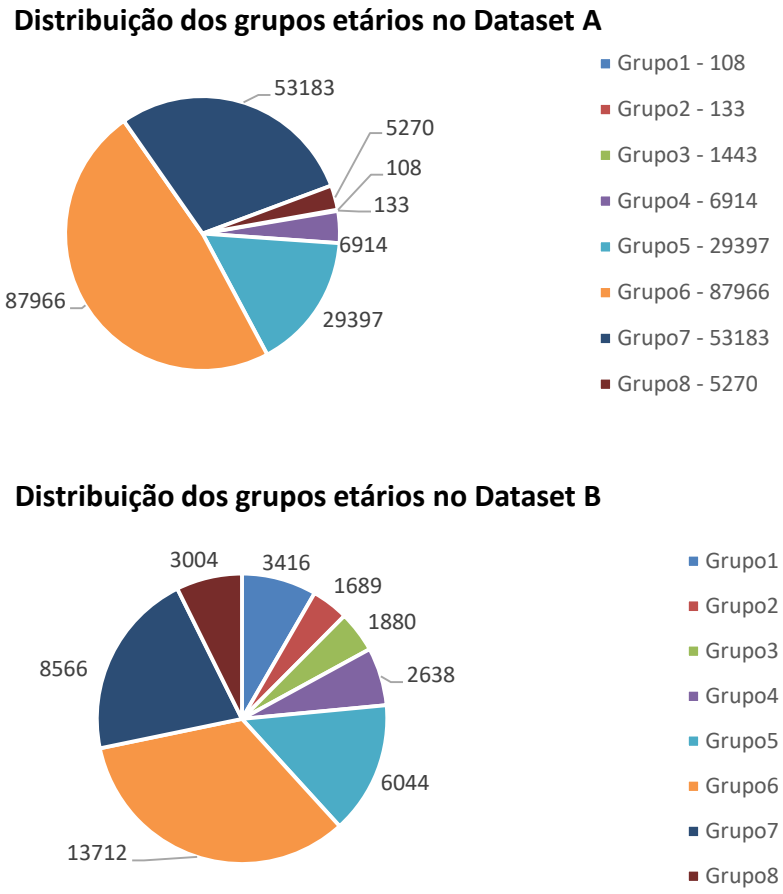


Figura 20 - Distribuição dos grupos etários nos *datasets* iniciais

5.3 Preparação dos dados

Após a limpeza dos dados iniciou-se o processo de preparação dos mesmos.

Quer para o *dataset A* quer para o *dataset B* o processo de preparação de dados foi semelhante. Em ambos os casos as imagens foram redimensionadas de acordo com a arquitetura a utilizar.

Os primeiros testes realizados com os *datasets* indicaram um problema baseado nos dados. O modelo final tendencialmente indicava como solução as classes com maior representatividade. Verificou-se que o facto de existir uma discrepância na representatividade entre classes, após a limpeza, prejudicava os resultados finais. Por exemplo, no *dataset A* as imagens de crianças com menos de 6 anos são escassas, por outro lado, indivíduos entre os 18 e 65 anos representam mais de 85% do total dos dados. Face a estes valores e de forma a tentar equilibrar os dados, foram removidas imagens nas classes com excessiva representatividade e foram geradas imagens para as classes menos representadas.

Este processo de eliminação e criação de novas imagens tratou-se de um processo iterativo onde foram testados diversos valores até chegar a um resultado satisfatório.

A criação de novas imagens foi realizada recorrendo a *data augmentation.* Para criar novas imagens foram introduzidas as seguintes alterações às imagens existentes:

- Rotação até 45%;
- Inversão horizontal da imagem;
- Ampliação da imagem até 20%;
- Deslocamento da imagem;

Dataset A

No *dataset A*, para reduzir as discrepâncias, foi calculado o valor médio de imagens por grupo. Numa fase inicial foram eliminadas imagens dos grupos com quantidade superior à média. Após alguns testes verificou-se que os melhores resultados foram obtidos através da condição: Cada grupo não pode ter mais do que 30% do valor médio calculado. Caso possuam são eliminadas, de forma aleatória, as imagens excedentes.

No processo de geração de novas imagens, numa fase inicial foram geradas imagens para igualar a quantidade para todos os grupos. Esta situação condicionou o modelo criado, pois este apresentava elevado *overfitting*. O facto de terem sido geradas demasiadas imagens para alguns grupos, implicou que esses mesmos grupos não possuíam dados com a qualidade pretendida. A geração de imagens através de *data augmentation* deve ser realizada com precaução, pois neste processo são geradas imagens baseadas em imagens existentes. Estas imagens resultantes apresentam semelhanças com as imagens base. Aumentar significativamente a quantidade de imagens recorrendo a esta técnica implica a presença de dados pouco diversificados e consequentemente uma dificuldade acrescida para o modelo estimar imagens de faces desconhecidas.

Após testar algumas condições neste processo, as condições finais são:

- Foi definida como meta o valor 66% da classe com maior representação.
- Somente foram geradas imagens para as classes com menor valor que a meta.
- São geradas imagens nas classes até atingir o valor da meta. Não podendo a quantidade de imagens geradas numa classe ser superior a 100%. Ou seja, uma classe no máximo duplica a quantidade de imagens, mesmo que não atinja o valor da meta.

No final no *dataset A* restaram 37.825 imagens. Estes dados foram divididos em conjunto de treino e de validação (70% e 30% respetivamente). Não foi concebido o conjunto de teste uma vez que este dataset tem o intuito de realizar um pré-treino da rede.
Face ao tamanho do dataset, foi possível constatar que o consumo de memória era significativo e somente recorrendo ao *Google Colab* foi possível continuar com o projeto.

Dataset B

Para o conjunto de dados resultante da combinação dos 3 *datasets* o processo de preparação foi semelhante. No entanto, e uma vez que a discrepância entre as classes era inferior, as condições aplicadas foram, em parte, diferentes. Tal como no primeiro conjunto de dados, foram realizados testes para tentar obter os melhores resultados. Foi calculado o valor médio para as classes e foram eliminadas as imagens excedentes ao valor médio para cada grupo de idades. Antes de gerar imagens foi necessário gerar os conjuntos de treino, validação e teste. A importância da execução desta separação antes de gerar imagens deve-se à necessidade de não criar imagens no conjunto de teste para que este reflita o conjunto original de dados. Tal como mencionado anteriormente através desta técnica são geradas imagens com semelhanças à imagem base e, esse facto pode afetar os resultados dos testes que pretendemos realizar. De seguida, foram geradas imagens para as classes menos representadas nos conjuntos de treino e validação. As condições seguidas foram as mesmas do *dataset A*. Todo este processo foi realizado quer para problemas de regressão quer para problemas de classificação.

Este processo permitiu reduzir as discrepâncias acentuadas que se verificavam inicialmente entre representatividade de classes, tal como se pode verificar de seguida.

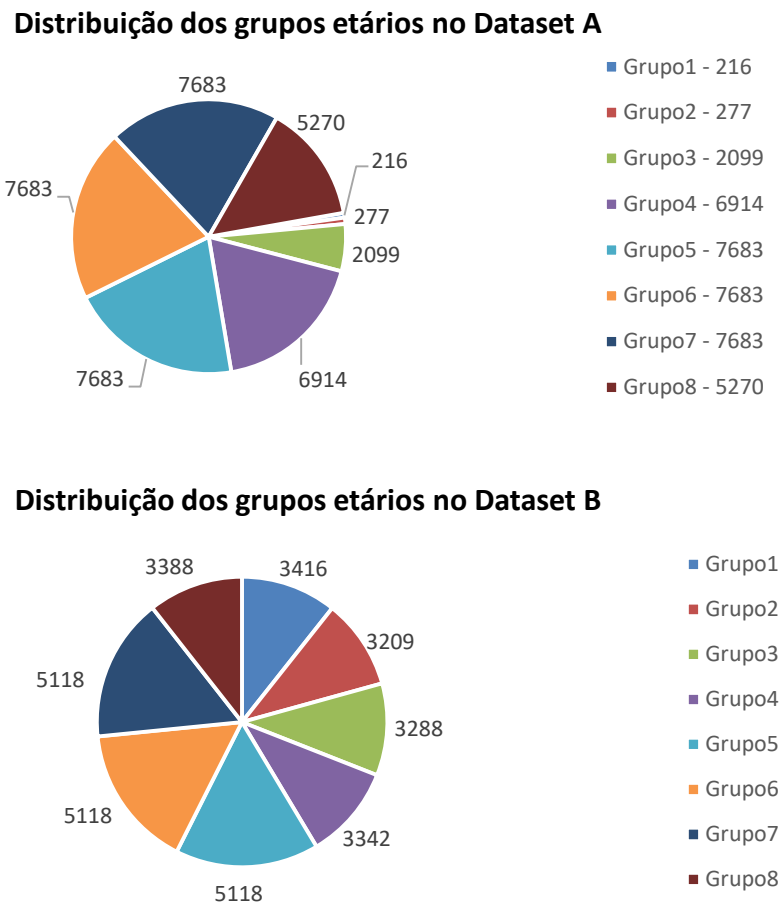


Figura 21 - Distribuição dos grupos etários nos *datasets*

5.4 Pré-processamento dos dados

Após o processo de preparação dos dados, estes foram importados recorrendo a funções da classe *ImageDataGenerator* que pertence à biblioteca *Keras*. Esta classe foi fundamental no desempenho do trabalho, uma vez que permitiu agilizar o processo de importação dos dados e criação de novas imagens. Em problemas de classificação foi utilizada a função *flow_from_directory*, no entanto esta função não pode ser aplicada, de forma direta, para problemas de regressão. Nesse caso foi necessário recorrer à função *flow_from_dataframe*.

Através da classe *ImageDataGenerator* foi também possível realizar um pré-processamento dos dados, de forma a adequar as imagens ao formato ideal para a rede. Por exemplo, ao especificar no parâmetro de pré-processamento o tipo *Xception*, os dados vão ser alvo de alterações que aceleram e otimizam o processo de treino para redes que derivem desta arquitetura. Neste caso uma das alterações é a normalização dos píxeis para possuírem valores entre -1 e 1. Outras arquiteturas são mais eficazes quando a normalização é realizada entre 0 e 1.

5.5 Divisão dos dados

Existem várias formas de dividir os dados, sendo as mais populares o *holdout* e o *cross-validation*. O *holdout* divide aleatoriamente o conjunto inicial em dois subconjuntos, conjunto de treino, com cerca de 70% dos dados iniciais e conjunto de teste com os restantes 30%. O *cross-validation* divide o conjunto inicial de dados em K-subconjuntos de igual tamanho, em cada iteração são usados (k-1) subconjuntos para treino e o restante k subconjunto para teste. O *holdout* possui menor complexidade e requer menos poder computacional do que o *cross-validation*, no entanto reduz a quantidade de dados testados.

Na fase experimental será utilizado o método *holdout* para dividir os dados. Com recurso aos *datasets* disponíveis serão obtidos os dados para treinar, validar e testar o modelo. O primeiro conjunto é usado para treinar o modelo. O segundo, os dados de validação, são usados para obter uma estimativa da performance do modelo e para afinar os valores dos *hyperparameters*. Os dados de teste são usados numa fase posterior para testar a capacidade de previsão final do modelo.

Os dados serão divididos da seguinte forma:

- 70% – Treinar modelo
- 20% – Validar o modelo
- 10% – Testar o modelo

Após a seleção dos melhores modelos estes serão testados com o método *cross-validation* para obter estimativas mais fiáveis.

5.6 Medidas de avaliação dos modelos

Após treinar os modelos e afinar os seus parâmetros com os dados de validação, é necessário testar a sua performance. A definição de quais métricas utilizar para avaliar o modelo varia consoante o tipo de problema em questão.

5.6.1 Medidas para avaliação de modelos de classificação

Em problemas de classificação para avaliar o modelo, por norma numa fase inicial, constrói-se uma matriz de confusão. Esta matriz permite, de uma forma simples, uma análise dos resultados e apurar a performance do modelo.

Com base na matriz de confusão é possível calcular métricas relevantes para a avaliação do modelo. A métrica mais popular é a *accuracy*, que permite compreender num determinado número de acontecimentos qual a proporção de resultados corretos. Porém analisar somente este valor pode induzir em erro, sobretudo se a representação das classes for desequilibrada e, assim sendo, outras métricas como *precision, recall* e *F1-score* são calculadas [36]. Para sistemas de classificação multiclasse existem duas versões para cada uma destas métricas, a micro e a macro. A diferença corresponde a que no caso da macro é calculada a métrica de forma independente para cada classe e em seguida é obtida a média da mesma, tratando todas as classes de igual forma. No caso da micro são calculadas as métricas tendo em conta a contribuição de cada classe. No presente trabalho vamos proceder com o cálculo da *micro precision* e *micro recall* e *micro f1-score* [37].

$$Micro\ precision\ = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} Total\ Coluna_k} = \frac{\sum_{k=1}^{K} TP_k}{Total\ Geral}$$

$$Micro\ recall = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} Total\ Linha_k} = \frac{\sum_{k=1}^{K} TP_k}{Total\ Geral}$$

$$Micro\ f1 - score = \frac{\sum_{k=1}^{K} TP_k}{Total\ Geral}$$

$$Accuracy\ = \frac{\sum_{k=1}^{K} TP_k}{Total\ Geral}$$

Tal como se pode constatar com base nas fórmulas, as 4 métricas terão o mesmo valor. Este facto é usual quando se trata de um problema multiclasse em que cada imagem possui somente um resultado (*single-label*). Não seria possível constatar esta situação caso se tratasse de um problema de classificação multiclasse e multi-valor (*multi-label*) [37].

Outra métrica que por vezes é relevante calcular é o *Matthews Correlation Coefficient* (MCC). É considerada uma das medidas mais robustas pois permite avaliar, problemas multiclasse, com *datasets* não equilibrados, onde as classes possuem diferente representatividade. O valor do coeficiente encontra-se entre -1 e 1. Sendo que 1 representa uma previsão perfeita [36].

$$MCC = \frac{c * s - \sum_k^K p_k * t_k}{\sqrt{(s^2 - \sum_k^K p_k^2)\,(s^2 - \sum_k^K t_k^2)}}$$

A interpretação do valor deste coeficiente é semelhante à do *Pearson Correlation Coefficient* [38]. Deste modo os valores podem ser divididos em cinco grupos quando o valor é positivo. Caso seja negativo o valor é o inverso do positivo, ou seja o -1 representa uma correlação inversa muito forte [39]:

- [0 – 0,2] - Correlação muito fraca
- [0,2 – 0,4] - Correlação fraca
- [0,4 – 0,6] - Correlação média
- [0,6 – 0,8] - Correlação forte
- [0,8 – 1] - Correlação muito forte

5.6.2 Medidas para avaliação de modelos de regressão

Para avaliar problemas de regressão as métricas mais populares são a raiz do erro quadrático médio (RMSE) e o erro médio absoluto (MAE). O MAE mede a distância de um determinado valor de *y* ao valor médio da variável de resposta real. A métrica MAE dá um peso igual a todos os erros, enquanto a métrica RMSE dá um peso extra para grandes erros. Em princípio o RMSE é sempre maior que o MAE, mas se o RMSE for igual ao MAE significa que todos os erros são da mesma ordem de grandeza. Tanto o MAE como o RMSE podem variar de 0 a infinito, mas como se trata de uma medida de erro, quanto menor for o valor de MAE ou RMSE, melhor é a performance do modelo [37].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_j)^2}$$

$$MAE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - y_j|}$$

5.7 Modelação

Recorreu-se a redes pré-treinadas, baseadas em três tipos de arquiteturas diferentes, com o intuito de agilizar e otimizar a conceção dos modelos. As camadas finais da rede foram personalizadas de forma a tentar obter um melhor desempenho.

O processo de criação das redes convolucionais consistiu num processo iterativo, onde se realizaram diversos testes e caso os resultados fossem superiores as alterações eram mantidas, caso contrário eram descartadas. Foram criados diversos cenários para tentar obter o modelo com melhor desempenho na tarefa de deteção de idade com faces humanas.

Cenários baseados nos *datasets*:

- Treinar com cada um dos *datasets* individualmente (IMDB, *UTKFace*, *FacialAge* e *APPA-Real*).
- Treinar com todos os *dataset* juntos, balanceando os dados.
- Treinar com o *dataset A* numa fase inicial, e de seguida treinar com o *dataset B*, sem balanceamento dos dados.
- Treinar com o *dataset A* numa fase inicial, e de seguida treinar com o *dataset B*, balanceando os dados.

Para os três tipos de arquitetura, os melhores resultados foram sempre obtidos com o último cenário. Para além dos cenários acima descritos, foram realizados diversos testes na rede nomeadamente:

- Modificar ou remover camadas da rede.
- Adicionar camadas no final da rede (*dropout*, *max-pooling*, etc.).
- Alterar a quantidade de camadas a treinar.
- Utilizar diferentes algoritmos de otimização (SGD e *Adam*).
- Interpretação como problema de classificação e regressão.

A grande maioria dos testes foram realizados com a rede baseada na arquitetura *Xception*, pois após alguns testes foi definida como a arquitetura ideal. Para além de possuir bons resultados, o modelo é leve o que é relevante quando o objetivo é gravar o modelo num servidor. O modelo baseado na arquitetura *Xception* possui em média um quinto da dimensão de um modelo baseado na arquitetura VGG-16. Em termos de performance os resultados obtidos em modelos baseados nas arquiteturas VGG-16 e *Xception* foram semelhantes, e superiores aos resultados obtidos em modelos baseados na arquitetura *Inception-V4,* como se demonstrará em seguida.

5.7.1 Xception

As redes concebidas com base na arquitetura *Xception* foram obtidas através da biblioteca *TensorFlow*. Recorrendo a esta biblioteca foi possível obter uma rede já com os pesos

resultantes do treino com o *dataset ImageNet*. Esta situação é benéfica pois, permite à rede aprender a distinguir alguns aspetos básicos de imagens (vértices, contornos, etc.).

Classificação

Numa fase inicial somente foi substituída a última camada da rede *Xception* (camada *output*) por uma semelhante, adaptando-a para as 8 classes (igual ao número de grupos definidos), com a função de ativação *Softmax*. A função de otimização maioritariamente utilizada neste projeto foi *Adam*. Com esta estrutura o modelo foi aplicado aos vários cenários descritos.

Os resultados recorrendo aos *datasets* individualmente não foram os esperados. Esta rede quando treinada com o *dataset UTKFace*, apresentava uma *accuracy* de 0,58 e um *loss value* superior a 2. Analisando a matriz de confusão podemos constatar a dificuldade em distinguir algumas faixas etárias, bem como na reduzida quantidade de dados de teste utilizando somente 1 conjunto de dados.

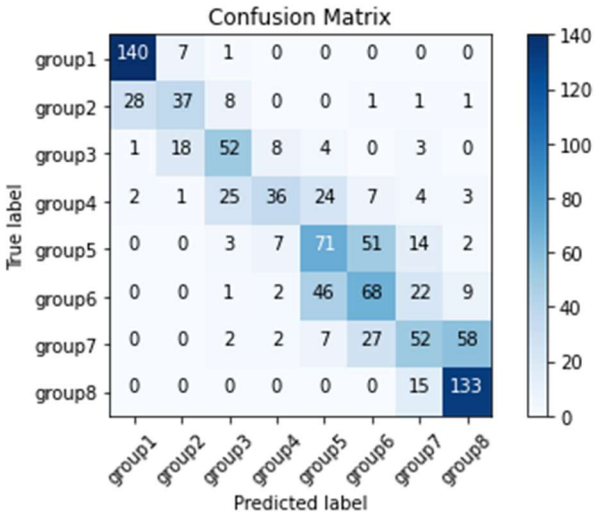


Figura 22 - Matriz de confusão com dataset *UTKFace*

Agrupando os conjuntos de dados *UTKFace*, *APPA-Real* e *FacialAge (dataset B)*, os resultados melhoraram consideravelmente, obtendo uma *accuracy* de 0,66. De seguida verificou-se que a inclusão do IMDB no conjunto de dados não resultava numa melhoria de desempenho. No entanto, utilizando este *dataset* numa fase inicial seguido da utilização do *dataset B* a taxa de acerto do modelo era superior (*accuracy* de 0,725).

Após verificar que os melhores resultados foram obtidos treinando a rede numa fase inicial com o *dataset A* e em seguida com o *dataset B*, foi estabelecido que este seria o cenário ideal para iniciar alguns testes à estrutura da rede.

De seguida foram realizados alguns testes alterando as camadas finais. Por exemplo, as últimas 5 camadas foram removidas. Estas camadas consistiam numa camada final do tipo *fully connected*, numa camada de *average-pooling* e em três camadas do tipo *blocks*. Foram adicionadas três novas camadas, uma camada *max-pooling* (que tal como mencionado

anteriormente tende a apresentar melhores resultados), uma camada do tipo *flatten* e para finalizar a camada *de output* já mencionada. O número de *epochs* foi incrementado e este modelo apresentou resultados semelhantes relativamente à versão anterior. Este modelo possuía as seguintes características:

Nº camadas	132
Total de parâmetros	18.000.944
Parâmetros a treinar	17.950.512

Tabela 11 - Estrutura da rede inicial *Xception* classificação

Relativamente ao processo de treino das camadas da rede, a primeira abordagem foi treinar todas as camadas, ou seja, nenhuma camada foi congelada durante o processo de treino. Quando uma camada é congelada no processo de treino, esta não altera os seus pesos, e por essa razão diz-se que a camada não foi treinada. O treino de todas as camadas torna o processo de treino demorado e aumenta os recursos necessários. Os ensaios seguintes consistiram em congelar camadas no processo de treino, progressivamente, e verificou-se que os resultados finais eram semelhantes. Verificou-se que treinar apenas as últimas 40 camadas não acarretava qualquer perda de desempenho e o processo era agilizado.

Para quantificar a diferença entre o resultado de saída e o resultado expectável recorreu-se, para os modelos de classificação, à função *cross entropy.* Nesta fase o modelo apresentava uma performance satisfatória, no entanto analisando o gráfico com as curvas de aprendizagem constatou-se, através da evolução das *loss functions* no treino com o dataset B, que o modelo apresentava sinais evidentes de *overfitting.* Uma vez que após o segundo *epoch* o valor do *validation loss* aumentava. Podemos concluir que o modelo se sobreajustou com o conjunto de treino e perdeu parcialmente a capacidade de generalizar com novos dados.

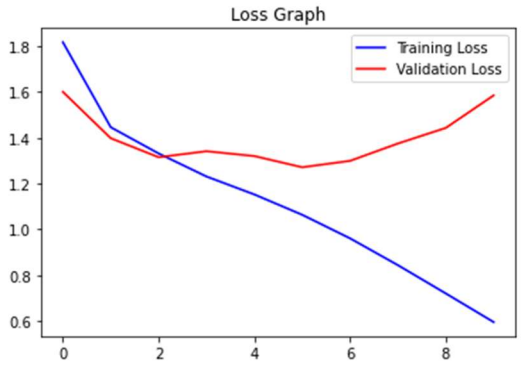


Figura 23 – Curvas de aprendizagem iniciais do modelo.

Para tentar solucionar esta situação, foi adicionada uma nova camada do tipo *dropout* após a camada do *max-pooling,* com uma taxa de 0,3. Com esta adição existiu uma ligeira melhoria, porém o modelo ainda não apresentava os valores desejados para a evolução do *validation loss*.

Dada a complexidade da arquitetura *Xception* e a elevada quantidade de camadas, e de forma a tentar melhorar o problema de *overfitting*, decidiu-se como teste reduzir o número de

camadas. Foram retiradas as últimas 60 camadas (aproximadamente metade da rede) e foram adicionadas as 4 camadas mencionadas anteriormente (*max-pooling, flatten, dropout e fully connected*). O valor da *accuracy* reduziu para 0,685, no entanto o *validation loss* melhorou. Com esta experiência conseguiu-se um modelo mais leve, e com uma performance ligeiramente inferior. Assim sendo, deste modo ficou claro que se podia tentar otimizar a estrutura da rede, de forma a obter uma performance semelhante à da rede completa e obter um modelo mais leve.

Testou-se retirar as últimas 50 camadas e, em seguida, as últimas 40 camadas (adicionando sempre as 4 finais anteriormente mencionadas). Ambos apresentaram resultados semelhantes. O desempenho melhorou (*accuracy* superior 4%) relativamente ao teste de remoção de 60 camadas. Assim sendo, foi considerado que retirando as últimas 50 camadas a rede é mais leve, mais rápida e possui bom desempenho. Manteve-se o treino das últimas 50 camadas. Esta última versão possui as seguintes caraterísticas:

Nº camadas	88
Total de parâmetros	9.674.184
Parâmetros a treinar	8.003.649

Tabela 12 - Estrutura da rede *Xception* simplificada

Nos anexos, Anexo A, encontra-se a estrutura final da rede. Face à dimensão da rede somente se representa o seu início e o seu fim. O modelo gerado por esta rede possui uma dimensão aproximada de 150MB.

A evolução das *loss functions* ao longo dos *epochs* no treino com o dataset B (*learning rate* 0.0001), melhorou comparando as figuras 23 e 24. Na figura 24, ambas as funções apresentam um declive descendente, porém ainda se verifica sinais de *overfitting*. O declive da função do *validation loss* é inferior ao declive da função do *training loss*, o que nos leva a concluir que o modelo se está a sobreajustar aos dados de treino.

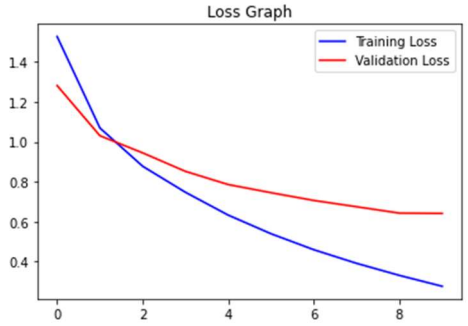


Figura 24 – Curvas de aprendizagem do modelo retificado.

Ao testar com o conjunto de teste, este modelo apresentou uma *accuracy* final de 0,72 e *value loss* de 0,66. A matriz de confusão é apresentada de seguida, na figura 25. Pode-se constatar que o modelo apresenta maiores dificuldades em estimar idades entre os grupos 4 a 7.

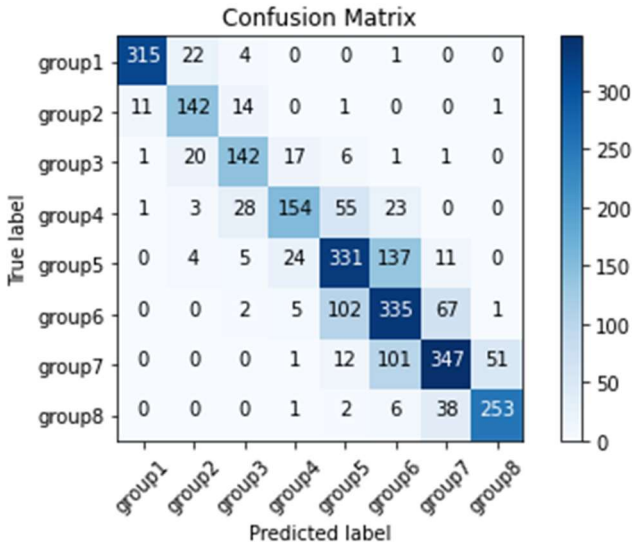


Figura 25 - Matriz de confusão do modelo *Xception.*

Calculando, com base na matriz de confusão, as métricas definidas anteriormente obtemos seguintes os valores:

Accuracy	*Precision*	*Recall*	*F1_score*	MCC
0.72	0.72	0.72	0.72	0.68

Tabela 13 - Métricas do modelo classificação *Xception.*

O valor do coeficiente MCC igual a 0,68 indica que o modelo possui uma forte correlação entre previsões e resultados expectáveis. De salientar que de acordo com o esperado o valor das métricas *precision*, *recall* e *f1 score* (tipo micro) são iguais à *accuracy*. Assim sendo, nos próximos resultados, somente serão apresentados os valores da *accuracy* e MCC.

Também foram realizados alguns testes alterando a *learning rate* e o número de *epochs*. Verificou-se que as taxas ideais são, no treino com o *dataset A* uma taxa de 0.001 e com o *dataset B* 0.0001. Este uso de duas taxas diferentes permite afinar o grau de evolução pretendido nas duas fases. No primeiro treino pretende-se que a rede conheça aspetos genéricos das faces e do processo de envelhecimento. No segundo treino (com *dataset B*) pretende-se que a rede otimize o seu conhecimento e seja capaz de distinguir com maior detalhe diferenças no processo de envelhecimento. Através dos testes foi ainda possível constatar que o aumento da taxa para 0.001 (com o *dataset B*) implicou uma redução na *accuracy para 0,63*. A redução da taxa para 0.00005 (com o *dataset B*) exigiu, para a mesma *accuracy*, uma maior quantidade de iterações. Podemos constatar, portanto, que uma taxa de aprendizagem muito reduzida implica, que o modelo vai convergir de forma muito lenta. Uma taxa de aprendizagem muito elevada pode resultar numa diminuição de desempenho, pois o modelo não consegue convergir numa solução ideal.

Na sequência de um dos trabalhos analisados no segundo capítulo, surge o interesse em testar a função de otimização SGD, em vez da *Adam*. Mantendo a *learning rate* verificou-se que o

modelo apresentava extrema dificuldade em convergir e um *value loss* elevado. Para compensar este facto, aumentou-se o número de *epochs* (aumento de 100%), bem como o valor da *learning rate* (0.01) e, deste modo, os resultados melhoraram e foram semelhantes aos anteriores (*accuracy* de 0,72). Uma vez que os resultados não foram superiores aos anteriores, e que o número de iterações necessário era superior, foi descartado o uso desta função de otimização.

Regressão

Tal como mencionado anteriormente, pretende-se interpretar o presente trabalho quer como um problema de classificação quer como um problema de regressão. O ponto de partida foi a rede, obtida através da biblioteca *TensorFlow*. As últimas 2 camadas da rede foram substituídas. A penúltima, *average-pooling* passou para *max-pooling.* A camada de output foi adaptada para possuir somente 1 nó e recorreu-se à função de ativação *ReLu.*

Nº camadas	134
Total de parâmetros	20.863.529
Parâmetros a treinar	20.809.001

Tabela 14 - Estrutura rede inicial de regressão *Xception.*

O cenário com melhores resultados foi, novamente, através da sequência *dataset A* e *dataset B*. Após o treino do modelo, a avaliação com o conjunto de teste apresentou um RMSE final de 6,24 e *loss* de 38,95.

Analisando o gráfico das *loss functions*, mais especificamente a evolução da *validation loss*, podemos constatar que tem uma evolução bastante irregular e o modelo apresenta sinais de *overfitting*.

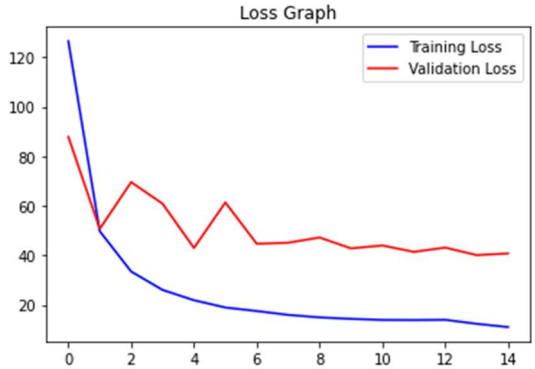


Figura 26- Curvas de aprendizagem iniciais do modelo.

Semelhante ao processo realizado anteriormente, de seguida foram removidas as últimas 50 camadas e adicionadas 4 novas (*max-pooling, flatten, dropout* e *output*), treinando todas as camadas da rede. O desempenho alcançado foi semelhante, no entanto o *validation loss* reduziu de acordo com o pretendido. Analisando a evolução do *validation loss*, podemos constatar que melhorou, apesar de ainda não ser ideal.

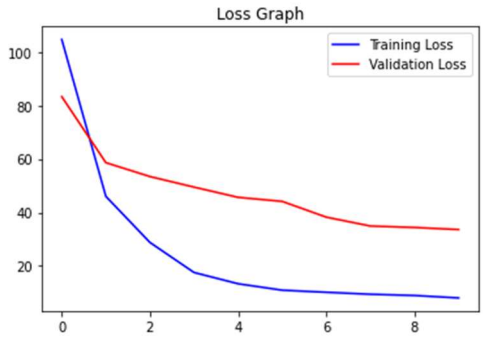


Figura 27 - Curvas de aprendizagem do modelo retificado.

Após este resultado, optou-se por adicionar uma nova *fully connected layer* com 64 nós imediatamente antes da camada de output. Pode-se constatar que esta camada extra implicou um aumento significativo do número de parâmetros do modelo, o que por sua vez implica um modelo mais pesado.

Nº camadas	89
Total de parâmetros	12.976.513
Parâmetros a treinar	12.949.265

Tabela 15 - Estrutura rede simplificada de regressão *Xception*

Esta estrutura apresentou o melhor desempenho de todas as experiências de regressão. A avaliação do modelo resultante com o conjunto de teste originou um RMSE final de 6,1.

RMSE	MAE
6,1	4,1

Tabela 16 - Métricas do modelo regressão Xception

Os resultados para o problema de regressão não são satisfatórios quando comparados com os resultados obtidos para a previsão de faixas etárias (classificação).

5.7.2 VGG-16

As redes criadas com base na arquitetura VGG-16 também foram geradas através da biblioteca *TensorFlow*. Na implementação desta arquitetura foram reaproveitados os pesos do *dataset ImageNet*. Tal como com a arquitetura *Xception*, os melhores resultados foram obtidos treinando a rede numa fase inicial com o *dataset A* e de seguida com o *dataset B*.

Classificação

Na primeira rede baseada nesta arquitetura, somente foi alterada a última camada da rede para se adequar às oito faixas etárias definidas, mantendo a função de ativação *Softmax*. A função de otimização utilizada foi *Adam*. A estrutura da rede consistiu, portanto em:

Nº camadas	23
Total de parâmetros	134.293.320
Parâmetros a treinar	134.293.320

Tabela 17 - Estrutura da rede inicial VGG-16

Ao contrário do que se verificou com os modelos baseados na *Xception*, estes modelos não apresentaram tantos problemas de *overfiting,* portanto não foi necessário simplificar o modelo. A quantidade de camadas desta arquitetura é muito inferior em relação à *Xception* e *Inception-V4,* no entanto o número de parâmetros é superior, sendo por isso designada como uma rede pesada.

Apesar de a rede não apresentar problemas em generalizar o conhecimento para novas imagens, optou-se por testar adicionar uma camada *dropout*, tal como foi realizado para a arquitetura *Xception*. A adição desta camada, entre as *fully connected layers* finais, não acarretou qualquer mais-valia (quer em termos de *accuracy* quer de *value loss*), e por essa mesma razão foi desconsiderada.

Após chegar aos melhores desempenhos, realizando testes com os vários cenários, foi necessário tentar otimizar a quantidade de camadas a treinar. Verificou-se que treinando mais do que as últimas 14 camadas, o desempenho final não era alterado. Desta forma, foi considerado que treinar as últimas 14 camadas seria o ideal.

Nº camadas	23
Total de parâmetros	134.293.320
Parâmetros a treinar	133.147.912

Tabela 18 - Estrutura da rede VGG-16 simplificada

Aplicando a estrutura final definida diretamente no *dataset B* (não utilizando o *dataset A*) obteve-se uma *accuracy* inferior a 0,63 com o conjunto de teste. Na versão final e seguindo a sequência *dataset A* e *dataset B* os resultados foram superiores, tendo obtido uma *accuracy* de 0,70. Um aumento superior a 10%. Este resultado evidencia a importância de realizar o primeiro processo de treino com *dataset A*, através do qual aumentamos a quantidade dos dados utilizados no processo global de treino.

A estrutura final da rede pode ser observada nos anexos, Anexo B. O modelo gerado por esta rede possui uma dimensão aproximada de 1,5 GB.

Através da matriz de confusão verificamos que existe um padrão semelhante à matriz obtida com o modelo baseado na arquitetura *Xception*. O modelo apresenta uma dificuldade significativa a distinguir indivíduos dos grupos 5 e 6.

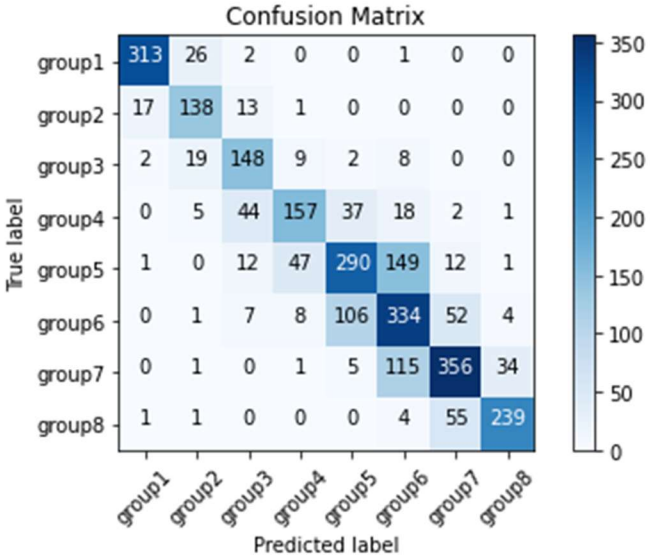


Figura 28 - Matriz de confusão do modelo VGG-16

Com base nos resultados da matriz de confusão foram calculadas as métricas previstas. Analisando o MCC, podemos verificar que existe uma forte correlação entre as previsões do modelo e os resultados expectáveis.

Accuracy	MCC
0,70	0,66

Tabela 19 - Métricas do modelo classificação VGG-16

Através da função *cross entropy* obteve-se a evolução da *validation* e *training loss* ao longo dos vários *epochs*. Podemos verificar o sentido descendente de ambas as funções até aproximadamente a oitava iteração. Esta situação representa que o modelo continuou a aprender, adaptar os seus pesos, e a melhorar as suas previsões quase até ao final. Exibindo sinais de sobreajustamento na fase final.

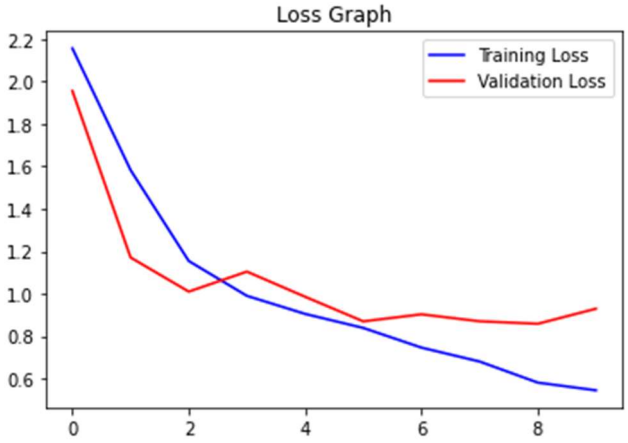


Figura 29 - Curvas de aprendizagem do modelo

De salientar a importância de uma taxa de aprendizagem apropriada, uma vez que um valor inapropriado pode prejudicar os resultados finais. Utilizando uma *learning rate* de 0.001 no treino com o dataset A e 0.0001 no treino com o dataset B o valor da accuracy consiste em 0,70. Alterando a *learning rate*, no treino com o dataset B, para 0.001 implica que a *accuracy* final reduza para 0,66. Tal como mencionado anteriormente, uma taxa de aprendizagem elevada pode originar que o modelo não consiga convergir e melhorar o seu desempenho.

Regressão

Os primeiros testes realizados neste trabalho, com redes neuronais convolucionais foram recorrendo à arquitetura VGG-16. A razão desta escolha prende-se com a sua popularidade e facilidade em obter informação. A primeira interpretação do problema foi como regressão, tentando obter a idade exata do indivíduo em estudo. Nesta fase o único *dataset* utilizado foi o IMDB, sem qualquer tipo de tratamento. Somente foi adaptada a camada de output, recorrendo à função de ativação *ReLu*. Os primeiros resultados foram muito inferiores ao esperado. Esta rede apresentava um RMSE de 29. Ou seja, cada previsão, em média, possuía um erro de 29 anos.

Após todo o processo de obtenção dos restantes *datasets*, limpeza e tratamento dos dados, foram realizados novos testes. A rede que apresentou melhor desempenho, tal como com a anterior arquitetura, possui uma camada extra *fully connected layer* (com 64 nós) antes da camada de output. O modelo gerado com esta rede obteve-se um RMSE de 8,62 e MAE de 6,1.

RMSE	MAE
8,6	6,1

Tabela 20 - Métricas do modelo regressão VGG-16

5.7.3 Inception-V4

A *Inception-V4*, ao contrário das outras duas arquiteturas, não é disponibilizada pela biblioteca *TensorFlow*. No entanto, e de forma a conseguir testar esta rede, foi replicada através da adição individual das várias camadas com um script. Ao contrário dos casos anteriores, não foi possível obter os valores provenientes do treino com o *dataset ImageNet*.

Também não foi possível utilizar a função de pré-processamento de imagens com a classe *ImageDataGenerator.* Neste caso foi necessário criar uma função onde os píxeis das imagens eram normalizados para possuírem valores entre 0 e 1.

Classificação

A última camada da rede foi substituída com o intuito de adaptar a rede aos 8 grupos, e foi utilizada a função de ativação *Softmax*.

Nº camadas	495
Total de parâmetros	53.080.808
Parâmetros a treinar	53.080.808

Tabela 21 - Estrutura da rede inicial *Inception-V4*

Esta rede possui um elevado número de camadas, sendo a mais extensa dos 3 tipos de arquiteturas estudadas. No entanto, não é a rede que possui maior quantidade de parâmetros, sendo essa a VGG-16. Nos anexos, Anexo C, encontra-se a estrutura da rede. Face à sua dimensão somente se representa o seu início e o seu fim. O modelo gerado por esta rede possui uma dimensão aproximada de 170 MB.

Em relação às camadas a treinar, foi definido treinar todas as camadas num primeiro teste e no segundo teste foram treinadas dois terços das camadas, ou seja, durante o processo de treino o primeiro terço de camadas foi congelado.

O melhor desempenho foi, novamente, obtido no cenário onde se treinou primeiro com o *dataset A* e em seguida com o *dataset B*. Para o treino com o *dataset A* o número de *epochs* foi aumentado para 20. Este aumento deve-se ao facto de a rede não possuir os pesos provenientes do treino com o dataset *ImageNet*. O facto de a rede não possuir os pesos derivados deste primeiro pré-treino significa que no seu primeiro processo de treino (com o *dataset A*) foi necessário aprender aspetos básicos como deteção de vértices, deteção de contornos, entre outros.

Nº camadas	495
Total de parâmetros	53.080.808
Parâmetros a treinar	51.026.112

Tabela 22 - Estrutura da rede *Inception-V4* simplificada

Após avaliar o modelo com o conjunto de teste, obteve-se uma *accuracy* final de 0,57. Este modelo apresenta, somente, uma correlação média entre as previsões e os resultados reais. Dada a complexidade da rede, os resultados obtidos, a falta de informação e suporte para a mesma, esta arquitetura foi menos explorada que as predecessoras.

Accuracy	MCC
0,59	0,52

Tabela 23 - Métricas do modelo classificação *Inception-V4*

Regressão

Em relação à abordagem do problema como regressão, a última camada do modelo foi alterada para indicar como output um valor final. Nenhuma camada foi congelada no processo de treino e foi utilizada a sequência primeiro o *dataset A* seguido do *dataset B*. O resultado final foi RMSE de 15,26.

RMSE	MAE
15,3	17,9

Tabela 24 - Métricas do modelo regressão *Inception-V4*

5.8 Avaliação dos modelos

No subcapítulo anterior foram criados diversos modelos, baseados nas arquiteturas VGG-16, *Xception* e *Inception-V4*. Para cada uma das arquiteturas foi definido o melhor modelo de classificação e o melhor modelo de regressão. Assim sendo, nesta fase vamos comparar os 3 modelos de classificação entre si, e de seguida comparar os modelos de regressão.

5.8.1 Definição de hipóteses

Para validar os vários modelos desenvolvidos devemos realizar testes para verificar se os resultados são semelhantes ou se a diferença entre eles é estatisticamente significativa.

O teste de hipóteses é um procedimento estatístico que permite rejeitar ou aceitar uma hipótese com base numa amostra com um certo grau de confiança.

O teste de hipóteses contempla a hipótese nula H_0, que consiste no pressuposto adotado como verdadeiro para a construção dos testes e a hipótese alternativa H_1, que consiste no pressuposto que se pretende concluir que é verdadeiro com base nos dados. Assim sendo, no presente trabalho as hipóteses para os modelos de classificação são:

- H_0 – Os valores do coeficiente MCC são iguais nos modelos.
- H_1 – Os valores do coeficiente MCC são diferentes nos modelos.

Para calcular se a diferença é significativa recorrer-se-á ao teste *t-Student*, caso a amostra possua uma distribuição normal. Para verificar se a amostra apresenta uma distribuição normal realiza-se o teste de *Shapiro*. Caso a amostra não possua uma distribuição normal, em vez de realizar o *t-Student*, realiza-se o *Wilcoxon Test*. Isto é válido tanto para classificação (média MCC) como regressão (média RMSE).

5.8.2 Avaliação de modelos

De forma a obter uma estimativa mais fiável para cada modelo final, pretende-se recorrer ao método *cross-validation.* Este método permite utilizar todos os dados disponíveis para testar o modelo. O *cross-validation* terá somente 5 iterações (*folds*) devido aos recursos necessários para o executar.

Xception

Relativamente aos valores de classificação o melhor resultado anterior para *accuracy* foi 0,72 e MCC foi 0,68. De seguida são apresentadas as métricas, *accuracy* e MCC, obtidas no *cross-validation*. A métrica preferencial para comparação entre modelos, uma vez que se trata de um problema multiclasse com um *dataset* desequilibrado, será o coeficiente MCC devido à sua robustez.

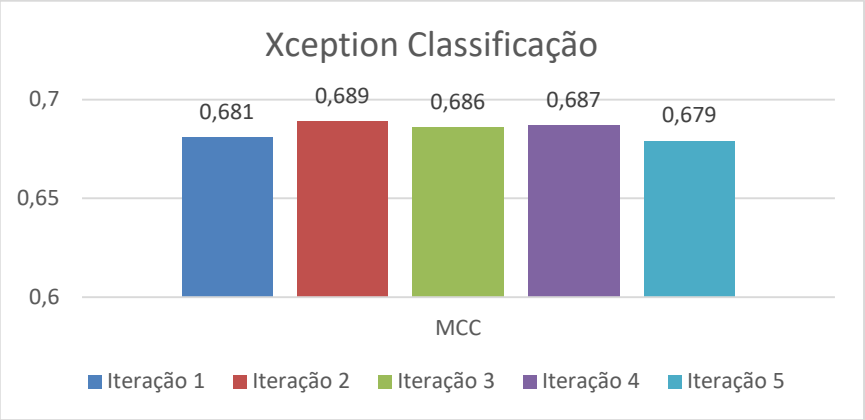


Figura 30 – Resultados MCC - *Xception* classificação

Aplicando o método de *shapiro*, para a amostra de dados MCC, obtemos um *p-value* de 0,53. Uma vez que estes valores são superiores a 0,05 podemos assumir que a amostra segue uma distribuição normal.

Relativamente à regressão aplicando o *cross-validation*, novamente para 5 iterações, obteve-se cinco novos valores de RMSE.

RMSE 1	RMSE 2	RMSE 3	RMSE 4	RMSE 5
6,12	6,08	6,21	6,05	6.18

Tabela 25 - Resultados RMSE - *Xception* regressão

Aplicando novamente o teste de *shapiro* podemos concluir que se trata de uma distribuição normal.

VGG-16

O melhor resultado anterior obteve uma *accuracy* final de 0,70 e um MCC de 0,65. Aplicando o método *cross-validation,* de forma estimar o desempenho do modelo com maior certeza, obtiveram-se novos valores.

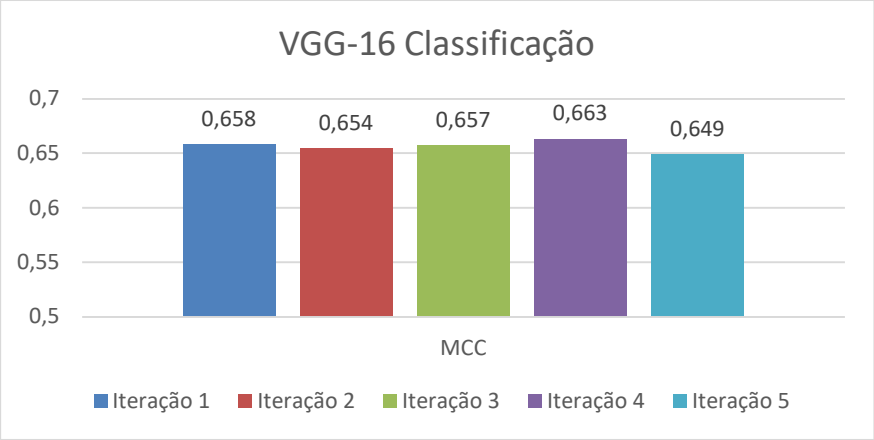


Figura 31 - Resultados MCC – VGG-16 classificação

Face a estes valores referentes ao coeficiente MCC, e através do teste de *shapiro*, confirmamos novamente que a amostra se trata de uma distribuição normal.

Repetindo o processo para o modelo de regressão obteve-se novamente 5 valores de RMSE. Através dos valores resultantes do *cross-validation*, e aplicando o teste *shapiro,* foi confirmado que se trata de uma distribuição normal.

RMSE 1	RMSE 2	RMSE 3	RMSE 4	RMSE 5
8,62	8,54	8,68	8,65	8,57

Tabela 26 - Resultados RMSE – VGG-16 regressão

Inception-V4

O melhor resultado anterior, para estimar a faixa etária, foi uma *accuracy* de 0,59 e um MCC de 0,52. Aplicou-se de seguida o *cross-validation.*

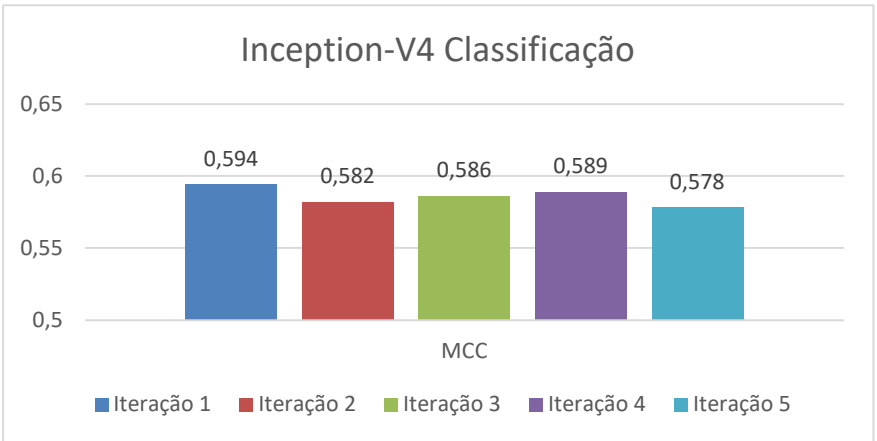


Figura 32 - Resultados MCC - *Inception-V4* classificação

Interpretando o problema como regressão, o melhor valor de RMSE obtido foi 15,3. Aplicando igualmente o *cross-validation* obtiveram-se os valores de RMSE.

RMSE 1	RMSE 2	RMSE 3	RMSE 4	RMSE 5
15,11	15,65	15,27	15,62	15,32

Tabela 27 - Resultados RMSE – *Inception-V4* regressão

Recorrendo ao teste de *shapiro* verificou-se que ambas as amostras de valores (conjunto de MCC e conjunto de RMSE) possuem uma distribuição normal.

5.8.3 Comparação de modelos

De acordo com as hipóteses formuladas, é necessário verificar se as diferenças obtidas nos modelos são estatisticamente significativas. Após verificar que as amostras seguem uma distribuição normal, podemos aplicar o teste *t-Student* para validar as hipóteses.

Serão realizadas 2 comparações para os métodos de classificação e 2 comparações para os métodos de regressão.

Classificação

Comparando o modelo *Xception* com o modelo VGG-16, para averiguar se a diferença de resultados dos coeficientes MCC é significativa, recorrendo ao método *t-Student,* obtemos os valores: valor $t_{calculado}$ 9.45, $t_{tabelado}$ 1,86. Como $t_{calculado} > t_{tabelado}$ então rejeita-se a hipótese Ho, ou seja, os valores do coeficiente MCC diferem estatisticamente.

Aplicando novamente o teste *t-Student* entre os resultados do modelo *Xception* e modelo *Inception-V4*, obtemos os resultados valor $t_{calculado}$ 29.46, $t_{tabelado}$ 1,86. Tal como na anterior comparação podemos afirmar que existe uma diferença estatística nos resultados.

Comparando os valores médios dos coeficientes MCC é facilmente verificado que o modelo derivado da arquitetura *Xception* é o modelo com melhor desempenho (MCC médio - 0,68).

Para além de possuir o melhor desempenho, o modelo derivado da arquitetura *Xception* consiste no modelo mais leve e mais rápido. Ambos estes parâmetros são importantes, uma vez que condicionam os recursos necessários para o servidor que está previsto alojar a aplicação.

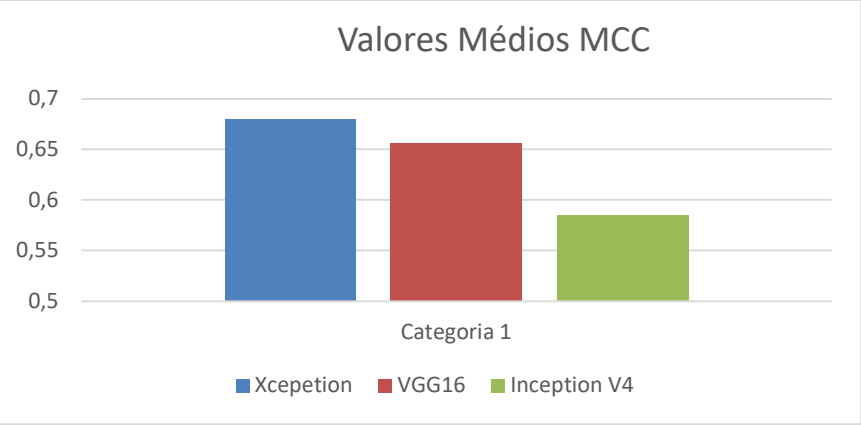


Figura 33 - Valores médios do coeficiente MCC para modelos classificação

Regressão

Aplicando o teste *t-Student* aos resultados dos modelos *Xception* e VGG-16, obtemos um $t_{calculado}$ 63.13, $t_{tabelado}$ 1,86. Podemos, novamente, rejeitar a hipótese H_0 e aceitar H_1.

Replicando o processo para os modelos *Xception* e *Inception-V4* validamos a hipótese H_1.

Tal como no modelo de regressão, o modelo baseado na arquitetura *Xception* apresenta os melhores resultados (RMSE médio - 6,11).

Relativamente aos modelos de regressão foi ainda calculado o RMSE do *baseline value*. Este valor consiste em assumir todas as previsões como o valor médio e calcular o respetivo RMSE. Através deste valor obtemos o erro, caso todas as previsões fossem a média de idades. Neste caso o RMSE tem o valor 19,8. Podemos afirmar que todos os modelos apresentaram valores inferiores.

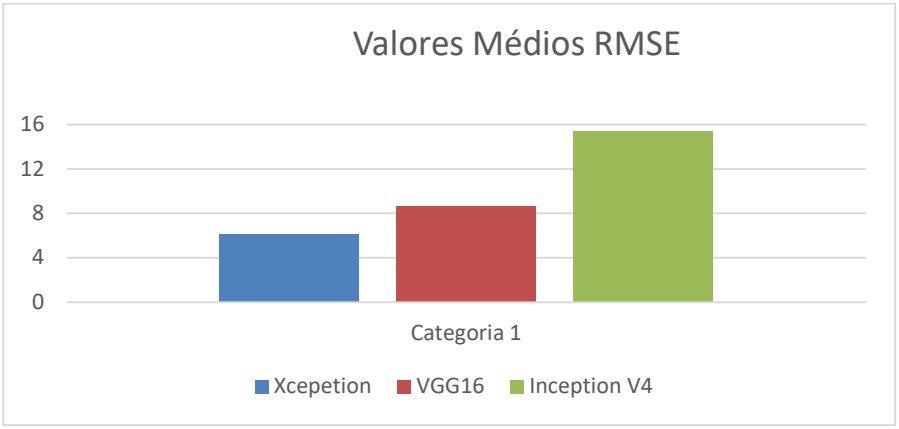


Figura 34 - Valores médios de RMSE para modelos de regressão

O modelo de regressão com melhor desempenho obteve um RMSE médio de 6,11 o que pode ser considerado um bom valor face à dificuldade desta tarefa. Este valor indica que em média estamos aproximadamente 6 anos errados na previsão de idade. Apesar deste valor ser

considerado satisfatório, o modelo de classificação permite melhores resultados e, portanto, será o modelo implantado na aplicação.

5.9 Comparação com trabalhos relacionados

A comparação do modelo de classificação final com os trabalhos relacionados apresentados não pode ser realizada de forma linear, uma vez que existem diferentes parâmetros e condições. No entanto, o trabalho intitulado *Age Estimation from Face Images Based on Deep Learning* apresenta algumas similaridades. Consiste num problema de classificação onde foram definidos 10 grupos etários. Em ambos os casos os modelos com origem na arquitetura *Xception* são os que apresentam melhores resultados, sendo que num trabalho a *accuracy* foi um pouco inferior, porém possui 2 grupos extra.

Trabalho	Nº grupos etários	*Accuracy*
Facial Age Estimation Using CNN	3	86%
Age Estimation from Face Images Based on Deep Learning	10	60%
Atual	8	72%

Tabela 28 – Sumário de trabalhos classificação

5.10 Reestruturação de grupos etários

Apesar dos resultados finais serem satisfatórios, o modelo apresenta alguma dificuldade em diferenciar a idade de faces humanas na fase adulta. Esta situação era previsível uma vez que na idade adulta as alterações faciais devido ao envelhecimento são menos acentuadas. Tal como foi constatado ao analisar as matrizes de confusão, os grupos etários entre 4 e 7 prejudicam significativamente os resultados do modelo. Como experiência final, optou-se por aplicar a última rede desenvolvida (adaptando a camada de output) a uma nova estrutura de grupos. Reduzindo o número de grupos etários de 8 para 4, e deste modo tentando prever crianças, jovens, adultos e idosos. Esta divisão corresponde à sugestão de trabalho futuro do artigo *Facial Age Estimation Using CNN.*

- Grupo 1 – até 5 anos
- Grupo 2 – 6 a 17 anos
- Grupo 3 – 18 a 65 anos
- Grupo 4 – maiores 65

O processo de treino foi semelhante, tendo utilizado novamente os *datasets* A e B, incluindo o todo o trabalho para equilibrar o número de dados por classe. Tal como seria de esperar os resultados melhoraram significativamente. Este modelo apresentou uma *accuracy* de 0,88 e um MCC de 0,84. O que denota que este modelo possui uma correlação muito forte com as idades

expectáveis. Comparando com o trabalho onde foram criados 3 grupos etários, este modelo apresenta um valor de *accuracy* superior e possui um grupo extra.

Trabalho	Nº grupos etários	*Accuracy*
Facial Age Estimation Using CNN	3	86%
Age Estimation from Face Images Based on Deep Learning	10	60%
Atual 1	8	72%
Atual 2	4	88%

Tabela 29 – Sumário retificado dos trabalhos de classificação.

6 Sistema

Após selecionar o modelo de classificação *Xception* como o modelo com melhor desempenho, procedeu-se com o desenvolvimento do sistema. O sistema consiste numa aplicação *web*, denominada "Deteção de Faixa Etária". O acesso à aplicação deve ser realizado através de um browser. Tal como planeado a aplicação possui duas formas de adicionar imagens, realizando upload de uma fotografia (botão "Escolher Ficheiro"), ou usando a webcam (botão "Capturar Imagem").

Figura 35 – Sistemas de *input* da aplicação

Após adicionar uma imagem o utilizador deve somente premir o botão "Submeter", iniciando desta forma o processamento do lado do servidor. O primeiro passo é validar a imagem. Caso o sistema detete que foi introduzida uma imagem inválida, é apresentada uma mensagem ao utilizador, indicando que deve submeter uma imagem de uma face humana. De seguida são submetidas duas imagens inválidas e em ambos os casos o sistema detetou que não se trata de uma face humana.

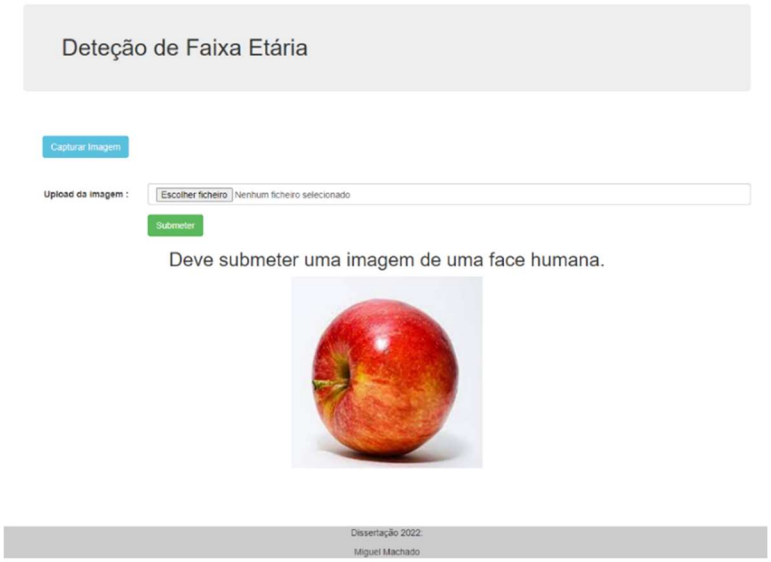


Figura 36 - Sistema de validação da aplicação (maçã)

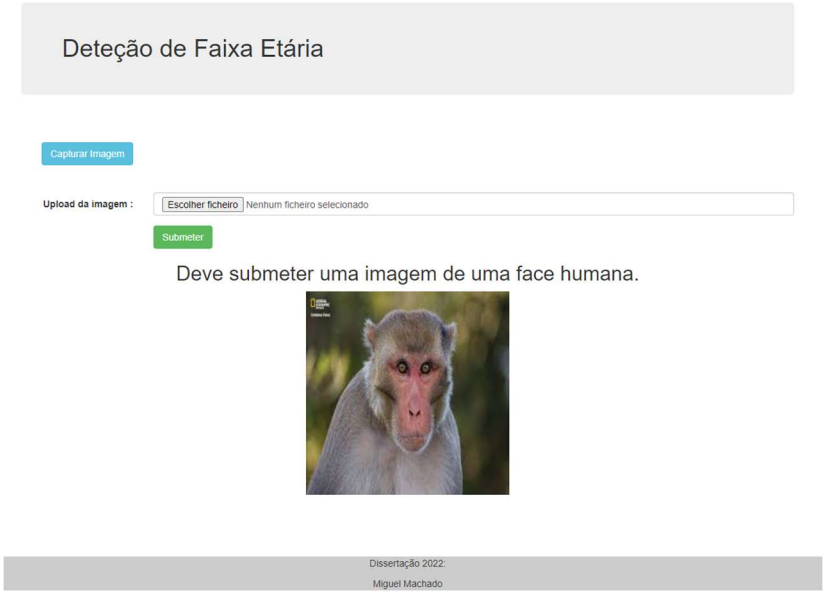


Figura 37 - Sistema de validação da aplicação (macaco)

Caso a imagem seja validada com sucesso, procede-se com a previsão da faixa etária, recorrendo ao modelo de classificação definido. Como output o utilizador consegue visualizar a imagem introduzida, com destaque para o contorno da face, bem como o resultado final da previsão.

Figura 38 - Deteção da faixa etária com aplicação

Recorreu-se ao *dataset* FG-NET para testar a aplicação. Os dados presentes neste conjunto de dados foram reservados para esta fase final do projeto, uma vez que se pretendia realizar testes com dados desconhecidos pelo sistema.

7 Conclusão

Neste último capítulo são apresentadas as conclusões do trabalho. Os objetivos são validados, são apresentadas limitações que surgiram ao longo do trabalho e finalmente são apresentadas possíveis melhorias a desenvolver no futuro.

Para além dos objetivos alcançados, a execução deste trabalho permitiu retirar algumas elações relativamente à conceção de uma rede neuronal convolucional para estimar a idade de um ser humano. Foi possível constatar a importância que os dados possuem no desenvolvimento de uma rede. Um bom desempenho de uma rede neuronal está fortemente relacionado com conjuntos de dados volumosos e com qualidade. O desenvolvimento de uma rede nova, com base em redes pré-treinadas permite obter modelos mais robustos e com melhor performance. Existem inúmeros parâmetros a definir na conceção de uma rede, sendo que a combinação ideal pode variar consoante o tipo de arquitetura utilizado. Consoante a tarefa em questão, uma rede com maior complexidade pode não apresentar resultados superiores, somente consumindo maior quantidade de recursos. Finalmente, de salientar o potencial e capacidade deste tipo de redes, que permitem com base numa imagem facial prever (com uma taxa de acerto de 88%) se diz respeito a uma criança, um jovem, um adulto ou um idoso.

7.1 Objetivos alcançados

O presente trabalho tinha como principal objetivo o desenvolvimento de um sistema capaz de detetar a idade ou faixa etária de indivíduos com base em imagens faciais. Este objetivo foi alcançado com sucesso. Para alcançar esta meta foi necessário realizar diversas tarefas, nomeadamente uma extensa pesquisa bibliográfica, identificar e trabalhar os *datasets* públicos com maior relevância, desenvolver modelos de deteção de idade usando redes neuronais convolucionais, comparar os resultados e criar um sistema final. Todas estas tarefas foram concluídas. Tendo como produto final uma aplicação web capaz de detetar a faixa etária de um ser humano com base numa imagem da sua face.

7.2 Limitações

A principal limitação no presente trabalho foi a dificuldade em obter conjuntos de dados volumosos e de qualidade. Os *datasets* utilizados no presente trabalho foram *datasets* públicos, no entanto os melhores conjuntos de dados para a tarefa de deteção de idade não são públicos. Um exemplo é o *dataset* MORPH que mesmo solicitando-o para fins académicos, não foi disponibilizado gratuitamente.

Outra limitação relevante foram os recursos necessários para desenvolver e treinar uma rede neuronal convolucional. Com o aumento do volume de dados constatou-se que as necessidades de hardware eram significativas. Somente recorrendo ao *Google Colab Pro*, foi possível concluir o presente trabalho e gerar vários modelos.

7.3 Trabalho futuro

Como trabalho futuro existem algumas melhorias interessantes a realizar. Os resultados obtidos para estimar a faixa etária foram satisfatórios, no entanto, não foi possível obter resultados aceitáveis interpretando o problema como regressão. A capacidade de previsão da idade exata foi, tal como nos casos de estudo, insuficiente para permitir uma aplicação comercial adequada da mesma. Assim sendo, seria interessante tentar otimizar os modelos de regressão.

Bibliografia

[1] Angulu, Raphael & Tapamo, Jules-Raymond & Adewumi, Aderemi. (2018). Age estimation via face images: a survey.

[2] Guo, Guodong & Zhang, Na. (2019). A survey on deep learning based face recognition. Computer Vision and Image Understanding.

[3] Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining.

[4] Viola, P., Jones, M.J. Robust Real-Time Face Detection. International Journal of Computer Vision 57 (2004).

[5] Chollet, F. (2017). Deep learning with python. Manning Publications.

[6] PTan, P. N., Steinbach, M., & Kumar, V. (2018) Introduction to data mining, Addison Wesley, 2nd Edition, 2018.

[7] Yoshimura, Yuji & Cai, Bill & Wang, Zhoutong & Ratti, Carlo. (2019). Deep Learning Architect: Classification for Architectural Design Through the Eye of Artificial Intelligence.

[8] Barbosa, Joel. (2020). Deep Learning Approach for UAV Visual Electrical Assets Inspection

[9] Inzamamuzzaman, S. & Al-Sakin, A. & Syeda, Ms & Maitra, S,. (2019). Detection of Early Alzheimer's Disease Applying Multi-layer Neural Networks. 10.13140/RG.2.2.20093.72166.

[10] LeCun, Yann & Bengio, Y. & Hinton, Geoffrey. (2015). Deep Learning. Nature.

[11] Gu, Hao & Wang, Yu & Hong, Sheng & Gui, Guan. (2019). Blind Channel Identification Aided Generalized Automatic Modulation Recognition Based on Deep Learning.

[12] Saraiva, Arata & Ferreira, N. & Sousa, Luciano & Carvalho da Costa, Nator & Santos, D. & Valente, Antonio & Soares, Salviano. (2019). Classification of Images of Childhood Pneumonia using Convolutional Neural Networks.

[13] T M, Navamani. (2019). Deep Learning and Parallel Computing Environment for Bioengineering Systems. Chapter 7 – Efficient Deep Learning Approaches for Health Informatics.

[14] Chollet, F., Allaire, J. (2018). Deep Learning mit R und Keras. Manning Publications.

[15] Zhang, & Wang, & Xu, Dongdong & Chen,. (2019). Research on Scene Classification Method of High-Resolution Remote Sensing Images Based on RFPNet.

[16] Do, Synho & Song, Kyoung & Chung, Joo. (2020). Basics of Deep Learning: A Radiologist's Guide to Understanding Published Radiology Articles on Deep Learning.

[17] Lecun, Yann & Bottou, Leon & Bengio, Y. & Haffner, Patrick. (1998). Gradient-Based Learning Applied to Document Recognition.

[18] Islam, Md & Baek, Joong-Hwan. (2021). Deep Learning Based Real Age and Gender Estimation from Unconstrained Face Image towards Smart Store Customer Relationship Management.

[19] Falqueto, L & Suterio, R & Paes, R & Passaro A. (2019). kNN e Rede Neural Convolucional para o Reconhecimento de Plataformas de Petróleo em Imagens SAR do Sentinel-1.

[20] Pakulich, Dmitry & Yakimov, S. & Alyamkin, S. (2019). Age Recognition from Facial Images using Convolutional Neural Networks.

[21] Szegedy, Christian & Ioffe, Sergey & Vanhoucke, Vincent & Alemi, Alexander. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.

[22] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions.

[23] Ganesh, Mukkesh & Dulam, Sanjana & Venkatasubbu, Pattabiraman. (2022). Diabetic Retinopathy Diagnosis with InceptionResNetV2, Xception, and EfficientNetB3.

[24] Junior, Orlando & Sartori, Andreza (2019). Reconhecimento facial de bugios-ruivos através de redes neuronais convolucionais.

[25] Zhang, Z. et al. “Age Progression/Regression by Conditional Adversarial Autoencoder.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017).

[26] Agustsson, E. & Timofte, R. & Escalera, S. & Baró, Xavier & Guyon, Isabelle & Rothe, Rasmus. (2017). Apparent and Real Age Estimation in Still Images with Deep Residual Regressors on Appa-Real Database.

[27] Ba Alawi, Abdulfattah & Saeed, Ahmed Y. A.. (2021). Facial Age Estimation Using Convolution Neural Networks.

[28] Han, Siyu. (2020). Age Estimation from Face Images Based on Deep Learning.

[29] Agbo-Ajala, Olatunbosun & Viriri, Serestina. (2019). Age Group and Gender Classification of Unconstrained Faces.

[30] Alam, Mahfujul & Talukder, Kamrul. (2019). Age Estimation From Facial Image Using Convolutional Neural Network(CNN).

[31] Liu, G.W. & Shen, Geoffrey. (2001). Extending the horizon of value world – on the 41st annual conference of the Society of American Value Engineers International.

[32] Koen, P. & Ajamian, G. & Burkart, R. & Clamen, A. & Wagner, K. (2001). Providing Clarity and Common Language to the Fuzzy Front End.

[33] Osterwalder, A., & Pigneur, Y. (2010). Business Model Generation. John Wiley and Sons.

[34] Barcelos, João. (2019). Análise Multicritério na Reabilitação de Edifícios: Apoio à decisão na intervenção num edifício público.

[35] Saaty, Thomas. (2008). Decision making with the analytic hierarchy process.

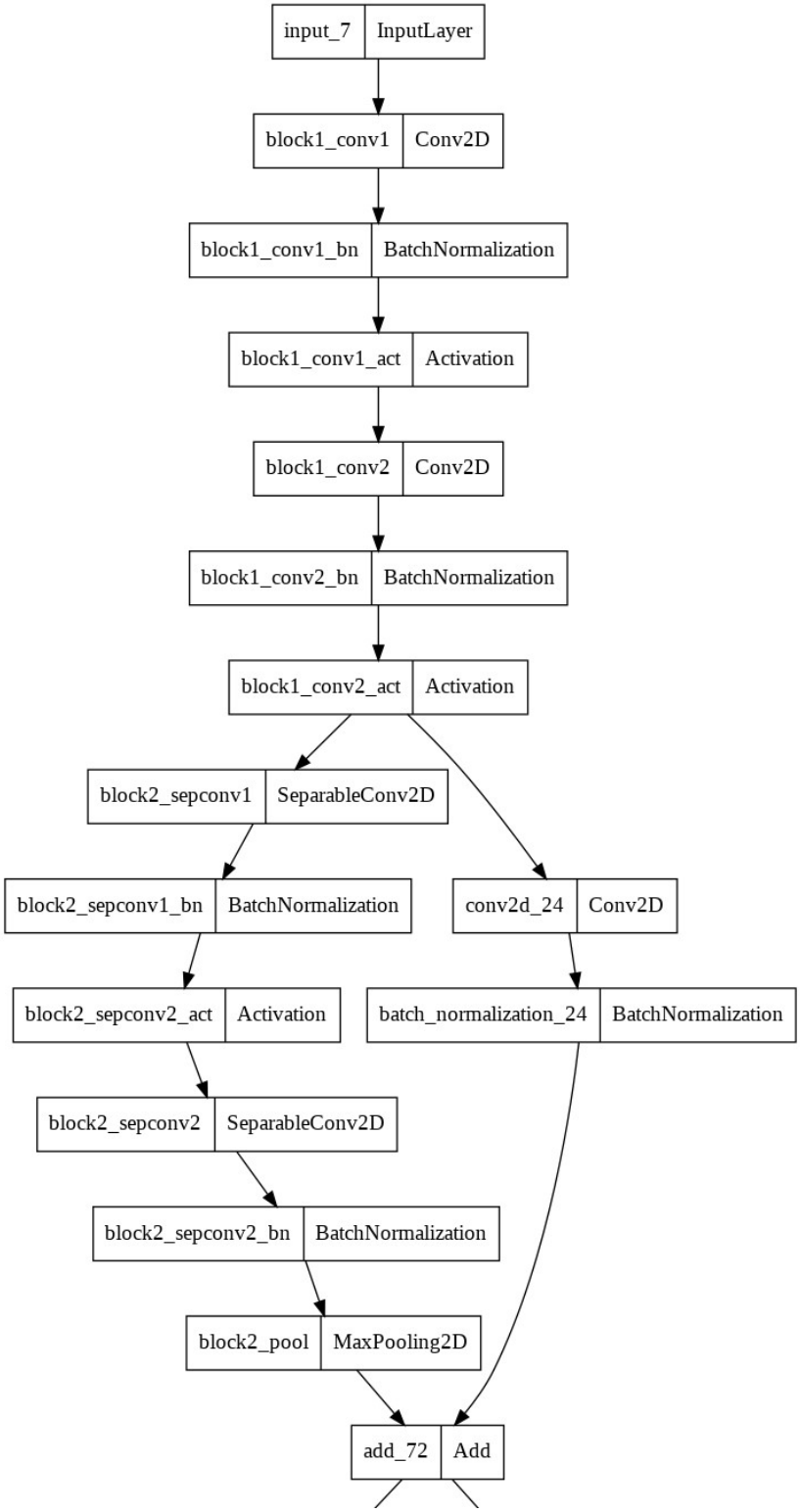
[36] Chicco, D., Jurman, G. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Med Inform Decis (2020)

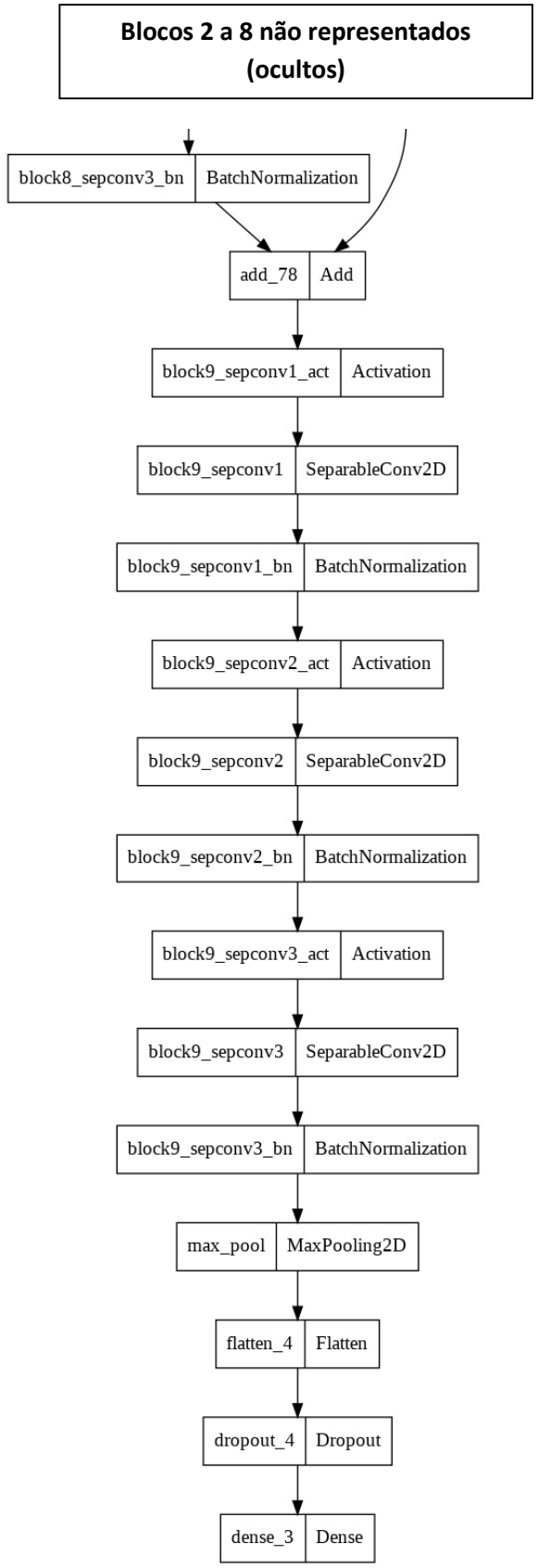
[37] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an Overview.

[38] Powers, D. (2008). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation.

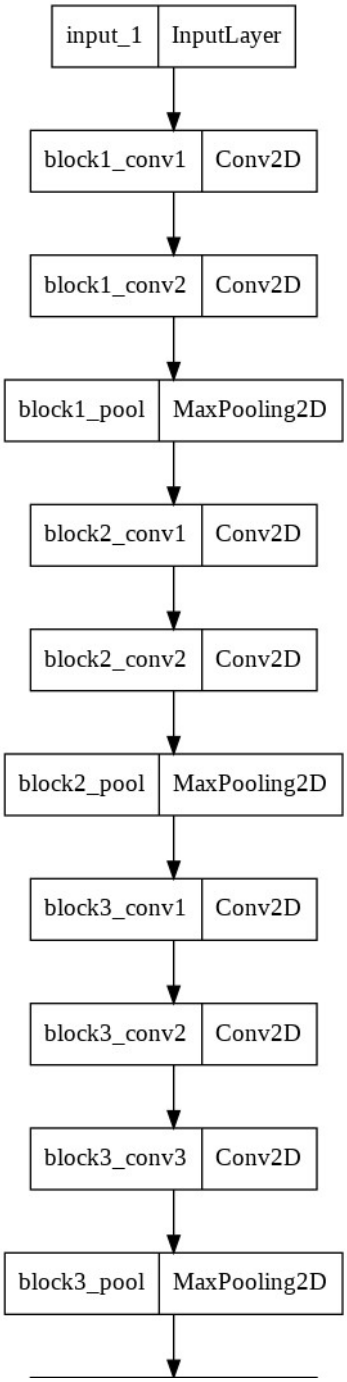
[39] Antoniou, C. Dimitriou, Loukas. Pereira Francisco Mobility Patterns, Big Data and Transport Analytics 2018

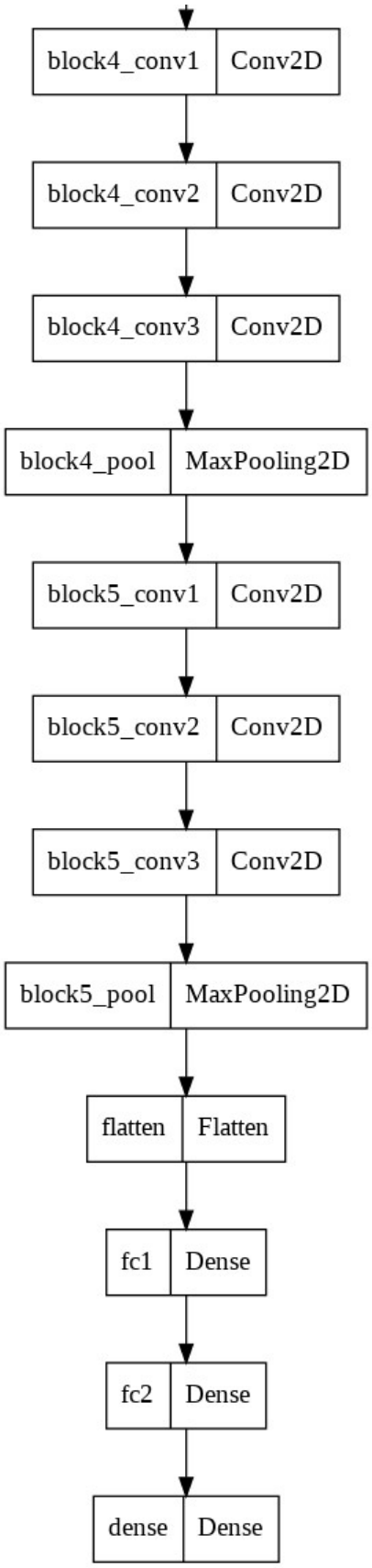
Anexo A


input_7 InputLayer
block1_conv1 Conv2D
block1_conv1_bn BatchNormalization
block1_conv1_act Activation
block1_conv2 Conv2D
block1_conv2_bn BatchNormalization
block1_conv2_act Activation
block2_sepconv1 SeparableConv2D
block2_sepconv1_bn BatchNormalization
conv2d_24 Conv2D
block2_sepconv2_act Activation
batch_normalization_24 BatchNormalization
block2_sepconv2 SeparableConv2D
block2_sepconv2_bn BatchNormalization
block2_pool MaxPooling2D
add_72 Add

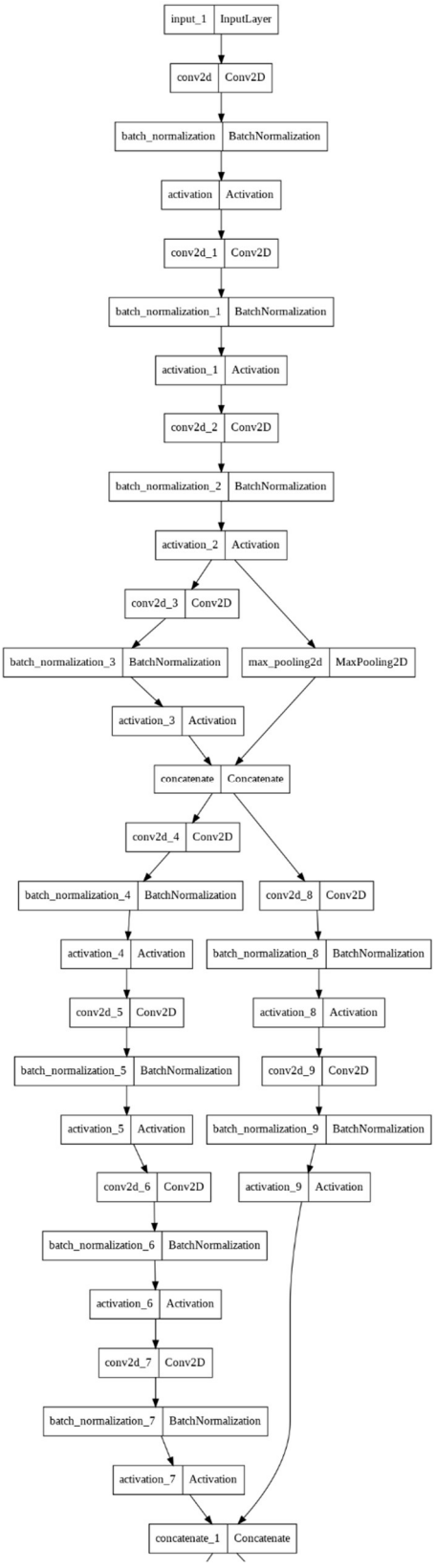

Blocos 2 a 8 não representados (ocultos)
block8_sepconv3_bn
BatchNormalization
add_78
Add
block9_sepconv1_act
Activation
block9_sepconv1
SeparableConv2D
block9_sepconv1_bn
BatchNormalization
block9_sepconv2_act
Activation
block9_sepconv2
SeparableConv2D
block9_sepconv2_bn
BatchNormalization
block9_sepconv3_act
Activation
block9_sepconv3
SeparableConv2D
block9_sepconv3_bn
BatchNormalization
max_pool
MaxPooling2D
flatten_4
Flatten
dropout_4
Dropout
dense_3
Dense

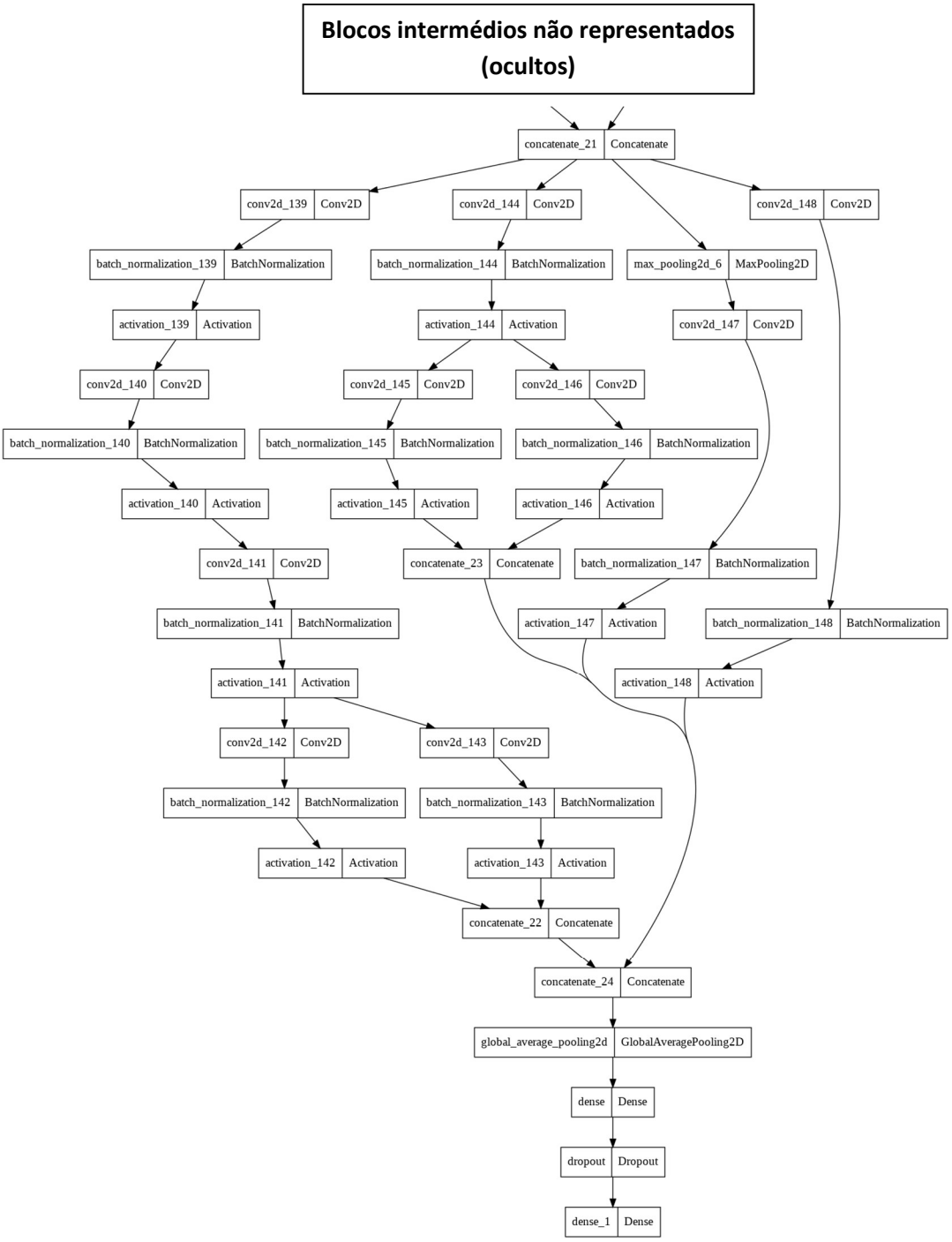
Anexo B


input_1 InputLayer
block1_conv1 Conv2D
block1_conv2 Conv2D
block1_pool MaxPooling2D
block2_conv1 Conv2D
block2_conv2 Conv2D
block2_pool MaxPooling2D
block3_conv1 Conv2D
block3_conv2 Conv2D
block3_conv3 Conv2D
block3_pool MaxPooling2D


block4_conv1
Conv2D
block4_conv2
Conv2D
block4_conv3
Conv2D
block4_pool
MaxPooling2D
block5_conv1
Conv2D
block5_conv2
Conv2D
block5_conv3
Conv2D
block5_pool
MaxPooling2D
flatten
Flatten
fc1
Dense
fc2
Dense
dense
Dense

Anexo C


input_1 InputLayer
conv2d Conv2D
batch_normalization BatchNormalization
activation Activation
conv2d_1 Conv2D
batch_normalization_1 BatchNormalization
activation_1 Activation
conv2d_2 Conv2D
batch_normalization_2 BatchNormalization
activation_2 Activation
conv2d_3 Conv2D
batch_normalization_3 BatchNormalization
max_pooling2d MaxPooling2D
activation_3 Activation
concatenate Concatenate
conv2d_4 Conv2D
batch_normalization_4 BatchNormalization
conv2d_8 Conv2D
activation_4 Activation
batch_normalization_8 BatchNormalization
conv2d_5 Conv2D
activation_8 Activation
batch_normalization_5 BatchNormalization
conv2d_9 Conv2D
activation_5 Activation
batch_normalization_9 BatchNormalization
conv2d_6 Conv2D
activation_9 Activation
batch_normalization_6 BatchNormalization
activation_6 Activation
conv2d_7 Conv2D
batch_normalization_7 BatchNormalization
activation_7 Activation
concatenate_1 Concatenate


Blocos intermédios não representados (ocultos)
concatenate_21 Concatenate
conv2d_139 Conv2D
conv2d_144 Conv2D
conv2d_148 Conv2D
batch_normalization_139 BatchNormalization
batch_normalization_144 BatchNormalization
max_pooling2d_6 MaxPooling2D
activation_139 Activation
activation_144 Activation
conv2d_147 Conv2D
conv2d_140 Conv2D
conv2d_145 Conv2D
conv2d_146 Conv2D
batch_normalization_140 BatchNormalization
batch_normalization_145 BatchNormalization
batch_normalization_146 BatchNormalization
activation_140 Activation
activation_145 Activation
activation_146 Activation
conv2d_141 Conv2D
concatenate_23 Concatenate
batch_normalization_147 BatchNormalization
batch_normalization_141 BatchNormalization
activation_147 Activation
batch_normalization_148 BatchNormalization
activation_141 Activation
activation_148 Activation
conv2d_142 Conv2D
conv2d_143 Conv2D
batch_normalization_142 BatchNormalization
batch_normalization_143 BatchNormalization
activation_142 Activation
activation_143 Activation
concatenate_22 Concatenate
concatenate_24 Concatenate
global_average_pooling2d GlobalAveragePooling2D
dense Dense
dropout Dropout
dense_1 Dense