Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Aplicação de técnicas de aprendizado profundo no problema de estimativa de idade facial

Marcelo Rossini Castro

JUIZ DE FORA
MARÇO, 2021

Aplicação de técnicas de aprendizado profundo no problema de estimativa de idade facial

Marcelo Rossini Castro

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Saulo Moraes Villela

Coorientador: Marcelo Bernardes Vieira

JUIZ DE FORA

MARÇO, 2021

Aplicação de técnicas de aprendizado profundo no problema de estimativa de idade facial

Marcelo Rossini Castro

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Saulo Moraes Villela
Doutor em Engenharia de Sistemas e Computação

Marcelo Bernardes Vieira
Doutor em Ciência da Computação

Rodrigo Luis de Souza da Silva
Doutor em Engenharia Civil

Luiz Maurílio da Silva Maciel
Doutor em Engenharia de Sistemas e Computação

JUIZ DE FORA
15 DE MARÇO, 2021

A todos os meus amigos.

Aos pais, pelo apoio e sustento.

Resumo

O aprendizado de máquina profundo, conhecido como *deep learning*, tem sido cada vez mais utilizado com o intuito de obter soluções em várias áreas da computação moderna, em especial, por sua utilidade e complexidade, a visão computacional. O presente trabalho busca estudar as técnicas existentes de aprendizado profundo conhecidas para realizar a estimativa de idade facial utilizando fotografias faciais. As principais técnicas analisadas consistem na utilização de florestas residuais de árvores de decisão neurais e na utilização de redes neurais convolucionais que produzem previsões classificadas de forma consistente. Aqui, um novo e estratificado conjunto de divisões do conjunto de dados escolhido e a utilização da técnica de fusão tardia entre os modelos analisados são propostos com o objetivo de reduzir o erro médio das previsões.

Palavras-chave: aprendizado profundo, visão computacional, idade facial, fusão tardia.

Abstract

Deep learning techniques has been increasingly used in order to obtain solutions in several modern computer science fields of study, specially, due to it's complexity and utility, computer vision. This work presents a study of the existing deep learning methods to estimate facial age through facial pictures. The main analysed approaches are the use of residual neural deep networks and the use of rank-consistent convolutional neural networks. Here, a new and stratified splits set of the chosen dataset and the use of the late fusion technique between the analyzed models are proposed in order to reduce the average predictions error.

Keywords: deep learning, computer vision, facial age, late fusion.

Agradecimentos

Aos meus pais e amigos, pelo encorajamento e apoio.

Aos professores Saulo e Marcelo pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos, especialmente ao professor Stênio, e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

"Our intelligence is what makes us human, and AI is an extension of that quality".

Yann LeCun

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

CACD	*Cross-Age Celebrity Dataset*
CNN	*Convolutional Neural Network*
CORAL	*Consistent Rank Logits*
CS	*Cumulative Score*
DSM	*Decision Saliency Map*
FC	*Fully Connected*
LBP	*Local Binary Pattern*
MAE	*Mean Absolute Error*
NDF	*Neural Decision Forest*
ReLU	*Rectified Linear Unit*
RGB	*Red, Green, Blue*
RL	*Representation Learning*
RMSE	*Root-mean-square deviation*
RNDF	*Residual Neural Decision Forest*
SGD	*Stochastic Gradient Descent*

1 Introdução

O aprendizado de máquina (*machine learning*) consiste na análise e reconhecimento de padrões em conjuntos de dados de forma que o sistema de inteligência artificial em questão possa utilizar esse aprendizado para tomada de decisões e classificação de informações. Contudo, a sua taxa de acerto depende fortemente da forma como o conjunto de dados é organizado.

Tarefas simples para seres humanos, como reconhecimento de expressões faciais e de ações em imagens e vídeos, mostram-se problemas de alta complexidade para os computadores atuais e estão inseridos no contexto de visão computacional. Isso se deve à dificuldade de se extrair abstrações de alto nível de dados desse tipo utilizando apenas os meios de representação conhecidos para o aprendizado de máquina.

Segundo Shirai (2012), a visão computacional tem o propósito de possibilitar a um computador "entender o seu ambiente através de informações visuais". Assim, trata de campos onde ainda não há técnicas básicas estabelecidas, como aplicações envolvendo objetos em duas ou três dimensões.

O aprendizado profundo (*deep learning*), abordagem derivada do aprendizado de máquina, tem sido cada vez mais utilizado para resolver problemas complexos, como os relacionados à visão computacional (GUO et al., 2016). A estratégia de aprendizado utilizada se baseia em uma hierarquia de conceitos e pode ser definida como um modelo de camadas onde o modelo busca por conceitos simples que formam uma camada e, com base neles, busca novos conceitos, formando novas camadas com maiores níveis de abstração (GOODFELLOW; BENGIO; COURVILLE, 2016).

Problemas na área de visão computacional têm diversas aplicações. Para exemplificar, o reconhecimento de objetos em vídeos é uma forma útil para detectar de forma automática e eficiente o começo de um incêndio e o reconhecimento facial tem aplicação em sistemas de segurança. Em especial, destaca-se o problema de estimativa de idade facial, descrito por Li e Cheng (2019) como "uma tarefa que tem como propósito inferir a idade cronológica através de imagens faciais digitais".

Estimar a idade de indivíduos através de suas imagens faciais tem sua importância ao poder ser utilizada em várias aplicações, como análises demográficas, gerenciamento comercial com base nos usuários e progressão do envelhecimento (NIU et al., 2016), além de poder ser aplicado, por exemplo, como um atributo para refinar a busca de imagens.

Ainda segundo Li e Cheng (2019), trabalhos anteriores em estimativa de idade facial utilizam soluções desde divisão e conquista através de árvores de decisão à combinação de aprendizado profundo de representação com árvores de decisão. Cao, Mirjalili e Raschka (2020) observam que as abordagens de regressão ordinal em redes neurais convolucionais, um dos métodos utilizados em problemas do tipo, ainda lidam com inconsistências do classificador.

1.1 Objetivos

Este trabalho tem como objetivo o estudo e aplicação de técnicas de aprendizado profundo para realizar estimativas de idade facial, focando mais especificamente no estudo de dois modelos propostos recentemente aplicados a um conjunto de imagens faciais de celebridades.

Aqui, é proposta uma nova divisão do conjunto de imagens de forma estratificada, pois assume-se que isso pode contribuir no treinamento dos modelos propostos. Além disso, é proposta também a fusão tardia dos resultados obtidos em ambos os modelos, como tentativa de reduzir o erro médio das previsões.

Posteriormente, é feita uma análise dos resultados obtidos em relação aos publicados na literatura.

1.2 Organização

Este trabalho está dividido da seguinte forma: no Capítulo 2, um breve histórico é apresentado, seguido de alguns conceitos fundamentais para o melhor entendimento; no Capítulo 3 são apresentados alguns dos trabalhos mais recentes que detém o estado da arte em diferentes conjuntos de dados (*datasets*); no Capítulo 4 são apresentados o conjunto de dados utilizado neste trabalho, seus pré-processamentos e divisões, além da nova divisão

estratificada proposta; no Capítulo 5 os métodos estudados são descritos; no Capítulo 6 são apresentados e discutidos os resultados obtidos com a divisão estratificada proposta e com a fusão tardia; e no Capítulo 7 são apresentadas as considerações finais desse estudo e as perspectivas para o problema.

2 Fundamentação teórica

2.1 Histórico

O desejo de criar máquinas que fossem capazes de simular o comportamento da mente humana possibilitou o surgimento da área de Inteligência Artificial. Deste então, estudos foram feitos na área com o objetivo de resolver vários tipos de problemas.

Segundo Goodfellow, Bengio e Courville (2016), as primeiras soluções funcionais criadas na área se propunham a solucionar problemas que seres humanos apresentam dificuldades para resolver (em tempo ou complexidade). Isso se devia ao formato de representação desses problemas, que eram baseadas em regras matemáticas formais.

Entretanto, tarefas simples para os seres humanos se mostraram difíceis de solucionar com abordagens computacionais devido à dificuldade de representação desses problemas. Assim, uma abordagem utilizada para tentar solucionar problemas desse tipo era a de base de conhecimento, onde os pesquisadores tentavam representar situações do mundo real em forma de regras matemáticas da forma mais precisa possível, o que é algo difícil.

Assim, segundo os mesmos autores, se tornou interessante que os sistemas de inteligência artificial possuíssem a capacidade de, através de dados em representação simples, aprender sobre esses dados através de reconhecimento e extração de padrões, o que caracteriza o aprendizado de máquina. Essa capacidade permitiu aos computadores a habilidade de resolver vários problemas, mas a performance dos algoritmos depende fortemente da representação dos dados que recebem.

Extrair e representar dados de uma aplicação do mundo real pode ser uma tarefa complexa e de alto nível de abstração. Dessa forma, surge a abordagem de aprendizado profundo, que é definida por Goodfellow, Bengio e Courville (2016) como “uma solução que permita aos computadores aprender com a experiência e compreender o mundo em termos de uma hierarquia de conceitos, com cada conceito definido em termos da sua relação com conceitos mais simples”.

Assim, o aprendizado profundo permite que o computador obtenha conhecimentos e extraia padrões dos dados de forma mais eficiente, dado que a hierarquia de conceitos pode ser vista como uma composição de funções, onde cada camada criada é uma solução da função mais interna da composição a ser solucionada.

O conceito de aprendizado profundo não é algo novo, considerando que redes neurais artificiais, que é um dos muitos nomes dados a esse conceito, foram concebidas com o intuito de modelar o aprendizado biológico (GOODFELLOW; BENGIO; COURVILLE, 2016) e são definidas por Schmidhuber (2015) como um conjunto de "vários processadores simples e conectados chamados neurônios, com cada um produzindo ativações com valor real".

Ainda segundo o mesmo autor, há modelos desenvolvidos com sucessivas camadas não lineares que datam da década de 1960. Atualmente, ainda são uma das principais técnicas utilizadas para aprendizado profundo.

2.2 Conceitos básicos

Os métodos de aprendizado profundo para estimativa de idade facial estudados neste trabalho abordam conceitos conhecidos como redes neurais convolucionais (*convolutional neural networks* – CNNs), métricas de desempenho, além do conceito de florestas de árvores de decisão neurais, todos apresentados a seguir.

2.2.1 Redes neurais convolucionais

Um dos principais conceitos utilizados no problema analisado neste trabalho é o de redes neurais convolucionais. Goodfellow, Bengio e Courville (2016) destacam a especialidade desse tipo de rede neural para processar dados com topologia em grade, como imagens. Ainda segundo os autores, redes convolucionais são "simplesmente redes neurais que usam convolução em vez da tradicional multiplicação de matrizes em pelo menos uma de suas camadas".

Muito do avanço obtido após a possibilidade de utilização de redes neurais convolucionais vem da necessidade de aplicação em problemas de visão computacional. O

aumento de poder computacional experimentado nos últimos anos proporcionou o uso mais frequente dessa técnica, que utiliza de vários conceitos e técnicas para ter desempenho efetivo.

Mesmo nos dias atuais, uma rede neural profunda totalmente conectada (*fully connected* - FC), ou seja, todos os nós de uma camada se conectam a todos os nós da camada seguinte, possui alto custo computacional.

Uma rede neural que recebe como entrada uma imagem colorida de tamanho 120×120 píxeis (*pixels*), por exemplo, possuirá 43.200 nós, ou neurônios, em sua camada de entrada, dado que a imagem seria representada através de três matrizes de mesma dimensão, cada uma representando um canal de cor no padrão RGB. A camada de entrada da rede terá então 43.200 neurônios, um para cada posição das matrizes.

Assim, tornou-se comum a utilização de operações de convolução e agrupamento (*pooling*) nesses casos. A operação de convolução, ilustrada na Figura 2.1, utiliza um núcleo (*kernel*), que é uma matriz de pesos (ou parâmetros) de tamanho definido e é sobreposta na matriz que representa um canal da imagem, realizando a soma dos produtos dos valores que se sobrepõem, resultando em um valor que compõe uma matriz denominada mapa de características (*feature map*) do canal. Como há mais de um canal, tem-se um núcleo para cada canal, compondo um filtro.

Especificamente, o núcleo tem dimensão menor que o canal da imagem e caminha neste, horizontal e verticalmente, de cima para baixo para compor o mapa de características com um passo (*stride*), que é o número de linhas e colunas que o núcleo se desloca, até que o processo seja concluído, completando o mapa de características, onde é aplicada uma função de ativação.

Essas funções são responsáveis por introduzir a não-linearidade, característica da maioria dos problemas no mundo real, após a operação de convolução, que é linear. Entre as mais utilizadas, estão a função sigmoide e a unidade linear retificada (*Rectified Linear Unit* - ReLU), esta última consistindo em substituir todos os valores negativos no mapa de características por zero.

Em seguida, a este mapa de características é aplicada a operação de agrupamento, onde um núcleo pequeno sem pesos caminha por ele e assume os valores das posições

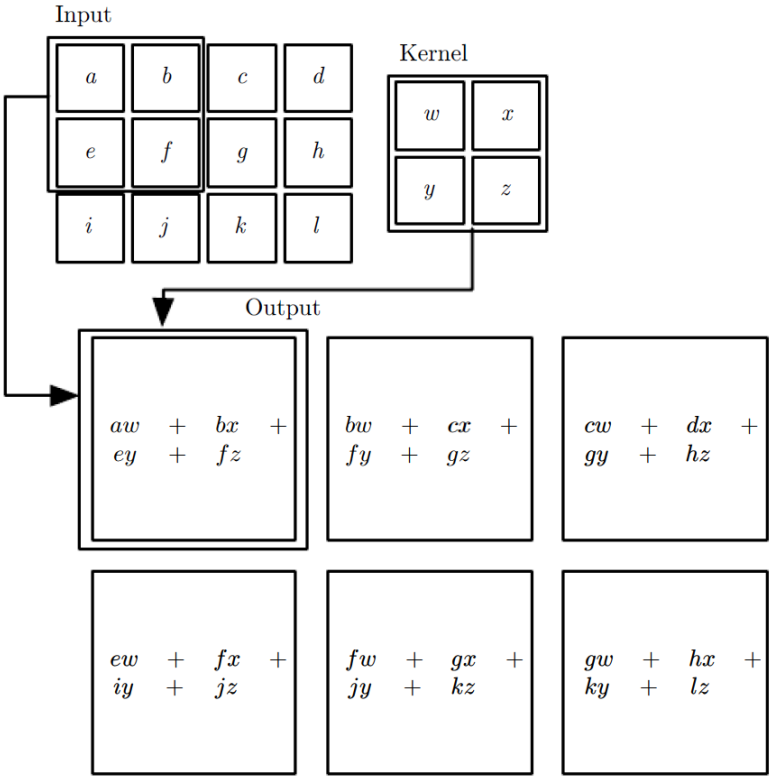


Figura 2.1: Ilustração de uma operação de convolução 2D com passo 1. A entrada (*input*) é uma matriz, que pode representar uma imagem em escala de cinza, por exemplo. Os seis valores gerados na saída (*output*) compõem o mapa de características obtido através da operação. Fonte: Goodfellow, Bengio e Courville (2016).

correspondentes. Com esses valores, realiza uma operação, como a máxima (*max pooling*), que retorna o maior dentre os valores, ou a média (*average pooling*), que calcula a média deles, atribuindo o valor resultante a um mapa de características menor e mais abstrato.

Após as operações de convolução e agrupamento, o mapa de características obtido é uma representação da imagem com dimensão reduzida que pode passar por novas operações nas camadas seguintes da rede, reduzindo, filtrando e extraindo cada vez mais características (*features*) relevantes da imagem.

O mapa de características gerado após as convoluções e agrupamentos é então linearizado e utilizado como entrada para camadas totalmente conectadas da rede, que são responsáveis por processar os dados e retornar um resultado.

Para exemplificar, em uma rede de classificação, a última camada totalmente conectada tipicamente contém um nó para cada classe e o resultado seria dado em probabilidades da imagem pertencer a cada uma das classes após uma operação, denominada *softmax*, que converte os valores retornados por cada nó da última camada.

Durante o processo de treino, ao passar pela rede, a resposta obtida é comparada com a resposta esperada para obter um erro, processo que pode ser feito através de funções de perda (*loss functions*), como a entropia cruzada.

Esse erro é utilizado para atualizar os pesos dos filtros por toda a rede em um processo denominado retropropagação (*backpropagation*). Para minimizar o erro, a ele é aplicada a operação de gradiente descendente, para que os pesos sejam atualizados na direção do menor erro utilizando o valor obtido da operação, que é tipicamente ajustado por uma taxa de aprendizado (*learning rate*), parâmetro que atua no impacto dessas atualizações.

Esse processo é realizado várias vezes com todos os exemplos presentes no conjunto de treino, compondo as denominadas épocas (*epochs*). Os exemplos são, em geral, divididos em grupos menores, denominados lotes (*batches*). Quanto todos os *batches* passam pela rede, uma época é completada.

Ao longo do processo, o objetivo é que a rede consiga generalizar para novas entradas que não estão no conjunto de treino, isto é, que não ocorra o superajuste (*overfitting*) da rede aos dados utilizados no treino, cenário ilustrado na Figura 2.2. Para realizar esse teste, um subconjunto de imagens extraído do conjunto de dados é tipicamente utilizado para validação.

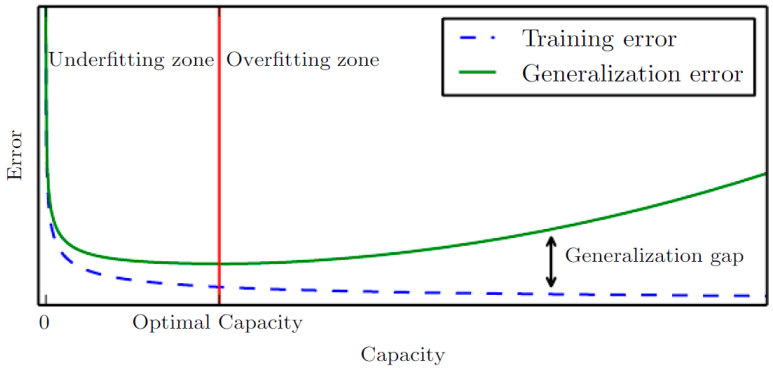


Figura 2.2: Relação entre capacidade e erro. À esquerda da linha vermelha, que representa a capacidade ótima do modelo de aprender com os dados, está a zona de subajuste (*underfitting*), onde o modelo ainda não consegue obter erros suficientemente baixos no conjunto de treino. À direita da linha, está a zona de *overfitting*, onde, à medida que tenta-se aumentar a capacidade, o erro no treino diminui, mas o erro de generalização aumenta. Fonte: Goodfellow, Bengio e Courville (2016).

Caso ocorra o *overfitting*, algumas estratégias podem ser utilizadas, como o ajuste

da taxa de aprendizado, o acréscimo de dados (*data augmentation*), que consiste em efetuar transformações nos dados de entrada de forma aleatória para criar variedade, ou até mesmo a adição de mais camadas na rede, que permitiria extrair padrões mais complexos.

Entretanto, adicionar muita profundidade à rede pode fazer com que o gradiente descendente se torne cada vez menor, efeito conhecido como desaparecimento do gradiente (*vanishing gradient*), tornando o aprendizado mais lento e causando um erro maior no treino.

A solução proposta e utilizada em vários trabalhos é o uso de Aprendizado Residual. A técnica, proposta por He et al. (2016) e nomeada *ResNet*, consiste em deixar camadas se encaixarem em um mapeamento residual em vez de esperar que algumas camadas a mais se encaixem diretamente em um mapeamento subjacente desejado.

O autor assume a hipótese de que é mais fácil otimizar o novo mapeamento. Essa técnica, ilustrada na Figura 2.3, pode ser implementada por *feedfoward*, o processo de passar adiante a saída de uma camada como entrada para a próxima, e conexões de atalho, sendo que estas apenas fazem mapeamento de identidade e suas saídas são adicionadas às saídas das camadas empilhadas, o que não adiciona complexidade.

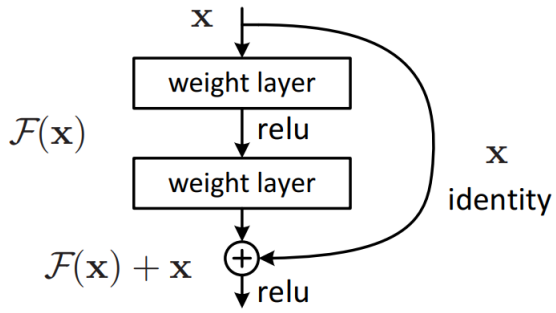


Figura 2.3: Esquema de um bloco de construção (*building block*), estrutura básica de uma rede, geralmente composta de camadas convolucionais. Aqui, é composta de duas camadas convolucionais com função de ativação ReLU. A conexão de atalho faz o mapeamento de identidade. Fonte: He et al. (2016).

Ao final do processo de treino, o desempenho da rede pode ser avaliado utilizando um conjunto de entradas de teste. Os valores estimados pela rede para cada uma das N entradas, representados por $\hat{y}$, e os valores reais, representados por y, podem ser utilizados em métricas para avaliar o desempenho médio.

2.2.2 Métricas de desempenho

Dentre as métricas de desempenho utilizadas nas redes analisadas neste trabalho, temos duas qualitativas e uma quantitativa.

A Raiz do Erro Quadrático Médio (*Root-mean-square deviation* - RMSE) avalia o desvio médio das previsões de forma qualitativa, penalizando os valores estimados distantes dos reais (*outliers*). É expressa por

$$\sqrt{\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$

O Erro Médio Absoluto (*Mean Absolute Error* - MAE) também avalia o desvio médio das previsões de forma qualitativa, mas considera os desvios de forma linear. É expresso por

$$\frac{1}{N} \cdot \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Já o Escore Cumulativo (*Cumulative Score* - CS) avalia o desvio médio das previsões de forma quantitativa, para observar a proporção das entradas cuja distância do valor estimado para o valor real é menor ou igual a um limite T. É expresso por

$$\frac{1}{N} \cdot \sum_{i=1}^{N} \mathbb{1}\{|y_i - \hat{y}_i| \leq T\}$$

2.2.3 Florestas de árvores de decisão neurais

Com o intuito de unir propriedades do Aprendizado por Representação (*Representation Learning* - RL) no contexto de aprendizado profundo com o princípio de divisão e conquista das árvores de decisão, Kontschieder et al. (2015) propõem o conceito de florestas de árvores de decisão neurais.

Segundo os autores, tratam-se de árvores de decisão estocásticas, diferenciáveis e, portanto, compatíveis com retropropagação, guiando o aprendizado por representação nas camadas inferiores de CNNs profundas.

Dessa forma, a tarefa do aprendizado por representação é reduzir a incerteza das

decisões de roteamento de amostras em nós de decisão, de forma a minimizar uma função de perda globalmente definida.

A abordagem permite obter, no momento do teste, decisões ótimas para uma amostra terminando nas folhas, em relação a todos os dados de treinamento e ao estado atual da rede.

Para definir o conceito formalmente, os autores propõem um problema de classificação com entrada $\mathcal{X}$ e saída finita $\mathcal{Y}$. Uma árvore de decisão é um classificador em formato de uma árvore composta por nós de decisão, ou divisão, e de previsão, ou folhas.

Cada nó folha $l \in \mathcal{L}$ contém uma distribuição de probabilidade $\boldsymbol{\pi}_l$ sobre $\mathcal{Y}$ e a cada nó de decisão $n \in \mathcal{N}$ é atribuída uma função de decisão $d_n(\cdot;\Theta) : \mathcal{X} \to [0,1]$ parametrizada por Θ, que é responsável por rotear as amostras pela árvore.

Quando uma amostra $\boldsymbol{x} \in \mathcal{X}$ atinge um nó de decisão n, sua rota é definida probabilisticamente pela saída de uma variável aleatória Bernoulli com média $d_n(\boldsymbol{x};\Theta)$.

Ao atingir um nó folha l, a previsão relacionada à árvore é dada pela distribuição $\boldsymbol{\pi}_l$ do rótulo (*label*) da classe. Considerando que as rotas são estocásticas, a previsão de uma folha será estimada pela probabilidade de se chegar até ela.

A predição final para uma amostra $\boldsymbol{x}$ de uma árvore T com nós de decisão parametrizados por Θ é dada por

$$\mathbb{P}_T[y|\boldsymbol{x},\Theta,\boldsymbol{\pi}] = \sum_{l\in\mathcal{L}} \pi_{ly}\mu_l(\boldsymbol{x}|\Theta), \tag{2.1}$$

onde $\boldsymbol{\pi} = (\boldsymbol{\pi}_l)_{l\in\mathcal{L}}$ e π_{ly} denota a probabilidade de uma amostra atingir uma folha l para assumir a classe y e $\mu_l(\boldsymbol{x}|\Theta)$ é considerado como a função de rota, fornecendo a probabilidade da amostra $\boldsymbol{x}$ atingir a folha l. Portanto, $\sum_{l\in\mathcal{L}} \mu_l(\boldsymbol{x}|\Theta) = 1, \forall\ \boldsymbol{x} \in \mathcal{X}$.

Sabendo que $\overline{d}_n(\boldsymbol{x};\Theta) = 1 - d_n(\boldsymbol{x};\Theta)$ e que $l \swarrow n$ e $l \searrow n$ são relações binárias que indicam se l pertence, respectivamente, à sub-árvore esquerda ou direita de n, tem-se

$$\mu_l(\boldsymbol{x}|\Theta) = \prod_{n\in\mathcal{N}} d_n(\boldsymbol{x};\Theta)^{\mathbb{1}_{l\swarrow n}}\ \overline{d}_n(\boldsymbol{x};\Theta)^{\mathbb{1}_{l\searrow n}}, \tag{2.2}$$

onde $\mathbb{1}_P$ é uma função indicadora condicionada pelo argumento P. Dessa forma, apenas os nós que estão no caminho de computação contribuirão para μ_l, como ilustrado na Figura

2.4.

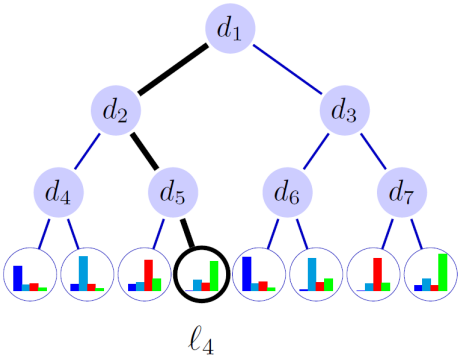


Figura 2.4: O traço destacado é um exemplo de rota para uma amostra $\boldsymbol{x}$ por uma árvore para atingir a folha l_4, que tem probabilidade $d_1(\boldsymbol{x})\overline{d}_2(\boldsymbol{x})\overline{d}_5(\boldsymbol{x})$. Fonte: Kontschieder et al. (2015).

Para a definição formal apresentada, os autores consideram que as funções de decisão retornam rotas estocásticas e são da forma

$$d_n(\boldsymbol{x};\Theta) = \sigma(f_n(\boldsymbol{x};\Theta)), \tag{2.3}$$

onde $\sigma(\boldsymbol{x}) = (1+e^{-\boldsymbol{x}})^{-1}$ é a função sigmoide e $f_n(\cdot;\Theta) : \mathcal{X} \to \mathbb{R}$ é uma função real que depende da amostra e da parametrização Θ.

Assim, os autores definem uma floresta como um conjunto $\mathcal{F} = \{T_1, ..., T_k\}$ de árvores de decisão que retornam uma predição para uma amostra $\boldsymbol{x}$ ao calcular a saída de cada árvore, como na expressão

$$\mathbb{P}_{\mathcal{F}}[y|\boldsymbol{x}] = \frac{1}{k}\sum_{h=1}^{k}\mathbb{P}_{\mathbb{T}_h}[y|\boldsymbol{x}] \tag{2.4}$$

O aprendizado ocorre, segundo os autores, inicializando os parâmetros Θ dos nós de decisão de forma aleatória e iterando o procedimento de aprendizado por um número de épocas, dado um conjunto de treino $\mathcal{T}$.

Em cada época, inicialmente obtém-se uma estimativa dos parâmetros $\boldsymbol{\pi}$ dos nós de previsão, dado o atual valor de Θ, através de uma minimização por otimização convexa.

O conjunto de treino é então dividido em uma sequência aleatória de mini-*batches*, onde uma atualização de Θ é feita por Gradiente Descendente Estocástico (*Stochastic Gradient Descent* - SGD).

Ainda segundo os autores, ao fim de cada época, a taxa de aprendizado pode, eventualmente, ser alterada, de acordo com planejamento pré-determinado. A Figura 2.5 ilustra uma floresta de árvores de decisão neurais.

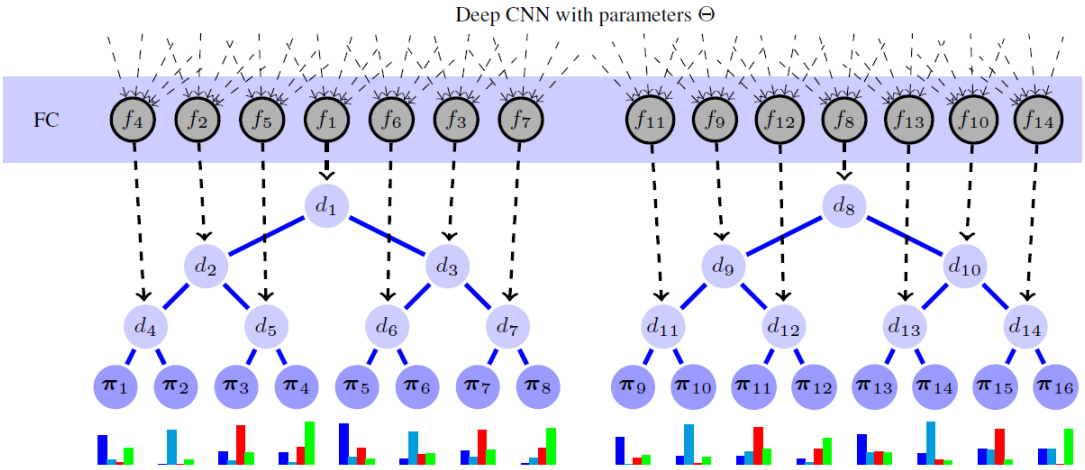


Figura 2.5: Topo: CNN profunda com número variável de camadas integradas pelos parâmetros Θ. Bloco FC: camada totalmente conectada usada para fornecer funções f_n (Eq. 2.3). Cada saída de f_n é associada, em ordem que pode ser arbitrária, com um nó de decisão na árvore, produzindo, eventualmente, a decisão de rota $d_n(\boldsymbol{x}) = \sigma(f_n(\boldsymbol{x}))$. Os círculos na parte inferior correspondem aos nós folha, contendo distribuições de probabilidade $\boldsymbol{\pi}_l$ como resultado ao realizar a otimização convexa. Fonte: Kontschieder et al. (2015).

2.2.4 Fusão tardia

Em algumas situações, pode ser interessante realizar uma combinação de diferentes modelos em relação às previsões obtidas em teste. Essa técnica de fusão tardia (*late fusion*) é descrita como um tipo de aprendizado por agregação (*ensemble learning*) que tem por objetivo melhorar a precisão (ZHANG et al., 2019b).

A fusão tardia pode ser utilizada manipulando-se diretamente os valores obtidos nos nós das camadas de saída ou com base nas previsões obtidas para cada entrada nos diferentes modelos (*score-based*).

3 Trabalhos relacionados

Existem na literatura vários trabalhos que lidam, direta ou indiretamente, com o problema de estimar a idade facial através de fotografias utilizando técnicas de aprendizado profundo. Os trabalhos utilizam de conjuntos de dados, ou *datasets*, de imagens faciais para realizar experimentos, com diferentes trabalhos atingindo o estado da arte em diferentes conjuntos de dados.

Entre os mais recentes, há a utilização de vários métodos, tais como regressão ordinal em CNNs com consistência do classificador e floresta residual de árvores de decisão neurais, estudados neste trabalho utilizando o *Cross-Age Celebrity Dataset* (CACD), descrito adiante.

Cao, Mirjalili e Raschka (2020) apresentam uma técnica de aprendizado profundo que propõe o modelo *Consistent Rank Logits* (CORAL) para regressão ordinal em suas CNNs de forma que as tarefas binárias produzam previsões classificadas de forma consistente. A consistência é garantida teoricamente pelos autores.

Li e Cheng (2019) apresentam uma técnica de aprendizado profundo que combina a utilização de redes neurais profundas e árvores de decisão, criando florestas de árvores de decisão neurais profundas (*Neural Decision Forest* - NDF).

Além disso, os autores assumem a hipótese de que a utilização de aprendizado residual pode contribuir com o aprendizado das funções de recomendação. Dessa forma, a NDF torna-se o que os autores classificam como uma floresta de árvores de decisão neurais residuais (*Residual Neural Decision Forest* - RNDF). O método detém o estado-da-arte no *Cross-Age Celebrity Dataset*.

Há ainda na literatura outras técnicas utilizando CNNs para estimativa de idade facial, entre elas, as propostas por Zhang et al. (2019a) e Berg, Oskarsson e O'Connor (2020).

Zhang et al. (2019a) propõem a utilização de um método de inferência de idade baseado em contexto. A proposta é justificada pelo propósito de utilização de modelos de CNNs em dispositivos móveis, que apresentam, em alguns casos, condições limitadas de

armazenamento.

Assim, os autores propõem um modelo compacto para imagens de baixa escala utilizando convoluções padrão, onde o número de parâmetros é menor por conta do tamanho das imagens. O trabalho detém o estado da arte no conjunto de dados FG-NET, que contém 1.002 imagens faciais de 82 indivíduos, com idades variando entre 0 e 69 anos.

Berg, Oskarsson e O'Connor (2020) observaram que métodos de regressão via classificação precisam lidar com a ambiguidade em como as classes discretas devem ser criadas a partir da distribuição da variável dependente, o que geralmente é contornado ao criar os chamados *bins*, que podem ser vistos como caixas, de larguras iguais cobrindo o intervalo de saída desejado.

Os autores propõem um método para criar diversidade nos rótulos utilizando sobreposição entre os *bins* para melhorar a acurácia da estimativa sem aumentar a complexidade computacional em relação a um classificador comum e o aplicam a alguns problemas. Em estimativa de idade facial, os autores conseguiram o estado da arte no conjunto de dados UTKFace, que contém, mais de 20.000 imagens faciais de indivíduos com idades entre 0 e 116 anos.

4 Conjunto de dados

Proposto por Chen, Chen e Hsu (2014), o *Cross-Age Celebrity Dataset* é um conjunto de dados com 163.446 imagens faciais de 2.000 celebridades nascidas entre os anos de 1951 e 1990 em diferentes idades, variando entre 14 e 62 anos. As imagens possuem tamanho 250×250 píxeis. A Figura 4.1 apresenta algumas imagens contidas no conjunto de dados.

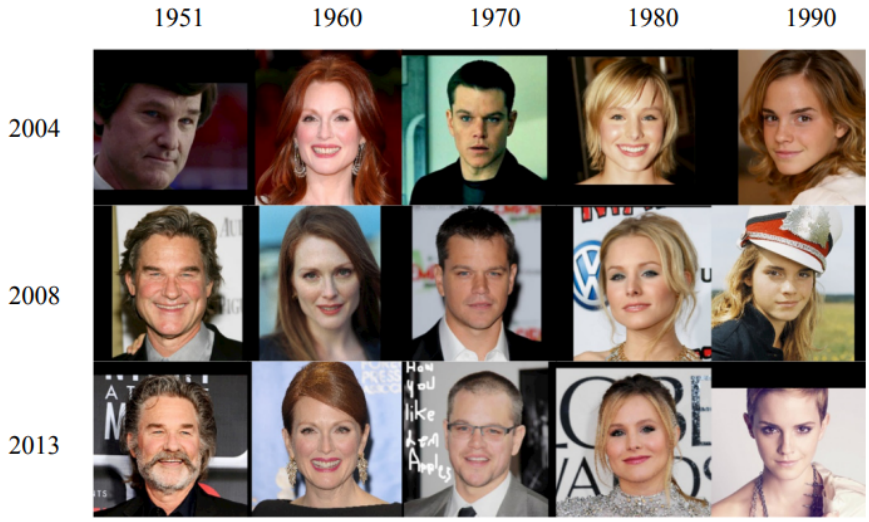


Figura 4.1: Exemplos de imagens contidas no conjunto de dados. No topo estão os anos de nascimento das celebridades e à esquerda os anos em que as imagens foram feitas. Cada coluna tem diferentes imagens da mesma celebridade. Fonte: Chen, Chen e Hsu (2014).

Juntamente com o conjunto de dados, são fornecidos meta-dados à ele associados como nome, idade, ano da foto, estimativa da idade na foto, padrões locais binários (*Local Binary Pattern* – LBP) de dimensão 75.520 extraídos de 16 pontos de referência faciais, entre outros.

O método de coleta das imagens foi através de buscas pelo nome da celebridade seguido de um determinado ano. Assim, a idade do artista foi estimada subtraindo-se o ano de seu nascimento do ano em que a fotografia foi feita.

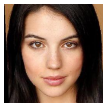
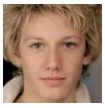
A escolha por esse conjunto de dados se deve ao seu domínio público e gratuito, e ao fato de ser utilizado em ambos os métodos estudados neste trabalho. O conjunto de dados MORPH, também utilizado em ambos, não foi utilizado pois este não está mais disponível gratuitamente.

4.1 Pré-processamentos

Devido à natureza do conjunto de dados, é esperado que grande parte das imagens contenha não apenas a face da celebridade, mas também o ambiente em que ela estava quando a fotografia foi feita ou até mesmo outras pessoas.

Para que o treino seja mais efetivo, é importante que o conjunto de dados seja pré-processado para tratar casos como os citados. Os autores dos modelos estudados neste trabalho implementam diferentes formas de pré-processamento, visualizadas na Tabela 4.1, e divisões, ou *splits*, em subconjuntos de treino, teste e validação, apresentadas na Seção 4.2.

Tabela 4.1: Comparativo utilizando duas imagens do conjunto de dados CACD em suas formas originais e após aplicação do pré-processamento utilizado nas abordagens estudadas.

Pré-processamento	**Imagem**		**Tamanho (píxeis)**
	14 Adelaide Kane 0001	**14 Alex Pettyfer 0007**	
Nenhum			250 px
CORAL			120 px
RNDF			256 px

4.1.1 *Consistent Rank Logits*

A abordagem proposta por Cao, Mirjalili e Raschka (2020) pré-processa as imagens utilizando o detector de faces frontais da biblioteca Dlib. Caso o número de faces detectadas seja diferente de 1, a imagem é descartada.

As imagens onde apenas uma face foi detectada são, então, realinhadas para que ocupem toda a extensão da imagem, de forma que a ponta do nariz fique ao centro da imagem. Logo após esse processo, a imagem é redimensionada para 120×120 píxeis.

4.1.2 Florestas de decisão neurais residuais

A abordagem proposta por Li e Cheng (2019) pré-processa as imagens utilizando pontos faciais (*facial landmarks*) fornecidos pelo conjunto de dados para encontrar a região da face e eliminar rotações. As imagens são, então, redimensionadas para 256×256 píxeis e normalizadas com base na média e desvio padrão dos três canais de cores.

Há, portanto, maior atenção ao pré-processamento nessa estratégia, padronizando o aspecto das imagens do conjunto de dados.

4.2 *Splits*

Para que seja possível realizar uma comparação justa entre os modelos estudados, é necessário que os *splits* sejam os mesmos. Dessa forma, ambos os modelos serão treinados e testados com os mesmos subconjuntos de imagens.

Entretanto, como não há um conjunto de *splits* definido utilizado na literatura para esse conjunto de dados, ambos os autores definiram diferentes proporções.

Cao, Mirjalili e Raschka (2020) dividem o conjunto de dados aleatoriamente, de forma que os conjuntos de treino, teste e validação compreendam, respectivamente, 72%, 20% e 8% das 159.449 imagens selecionadas no pré-processamento. Assim, 3.997 imagens são descartadas pelos autores.

Já Li e Cheng (2019) realizam a divisão também de forma aleatória, porém com diferentes proporções: 89%, 6,5% e 4,5% das imagens do conjunto de dados para os conjuntos de treino, teste e validação, respectivamente. Nenhum descarte é realizado pelos autores.

A aleatoriedade na escolha das imagens para cada um dos *splits* em ambas as abordagens abre margem para a falta de diversidade de imagens, de forma que o conjunto de teste possa conter muitas imagens de indivíduos de uma determinada idade, enquanto o conjunto de treino tenha poucas, o que pode resultar em uma possível queda na eficácia.

Neste trabalho, os experimentos são conduzidos utilizando um mesmo conjunto de *splits* aqui proposto, onde os conjuntos de treino, teste e validação compreendem, respectivamente, 63%, 30% e 7% das imagens, definidos com base na proporção 70%-

30%, amplamente utilizadas em outros trabalhos em aprendizado profundo (AKARSH et al., 2019; ALSHAWWA et al., 2020; HUANG et al., 2020; TORRES et al., 2018). É importante ressaltar que o conjunto de validação é utilizado durante o treino, sendo por este motivo compreendido na fatia dos 70% da proporção citada.

O processo de criação dos *splits* aqui propostos não realiza descartes e é de forma estratificada. Assim, a proporção definida para os *splits* se mantém para cada uma das idades representadas no conjunto de dados. Em outras palavras, cada subconjunto de imagens c_i, onde i é a idade dos indivíduos nestas, é dividido de forma que 63%, 30% e 7% das suas imagens estejam presentes nos conjuntos de treino, teste e validação, respectivamente.

Essa divisão é implementada agrupando-se as imagens por idade real e calculando as proporções para cada. Em seguida, são selecionadas aleatoriamente imagens de cada subconjunto de imagens c_i de forma a preencher os *splits* obedecendo as proporções. Ao fim, o resultado dessa operação pode ser visualizado através da Figura 4.2.

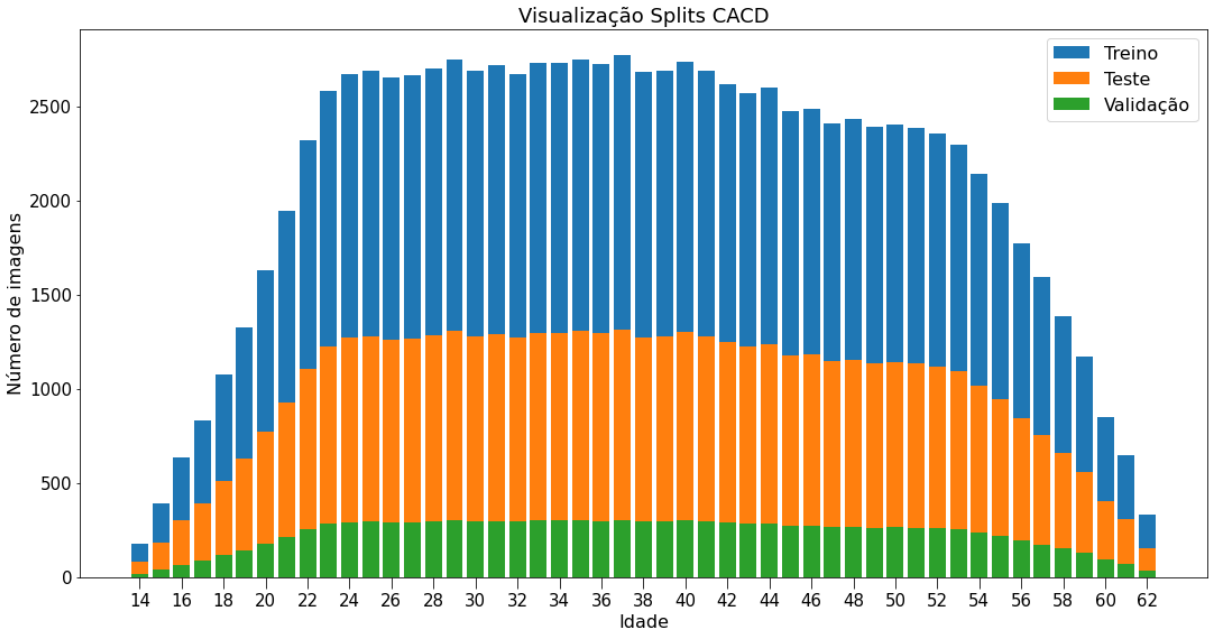


Figura 4.2: Visualização do conjunto de dados CACD após a divisão em *splits* estratificados. Cada coluna apresenta os totais de imagens para a idade associada, sendo a base de treino representada em azul, a de teste em laranja e a de validação em verde.

É importante ressaltar que no método CORAL, Cao, Mirjalili e Raschka (2020) realizam descartes, como mencionado anteriormente. Para que os experimentos pudessem ser conduzidos utilizando todas as imagens do conjunto de dados, foram feitas as

modificações necessárias.

Assim, quando mais de uma face é detectada, a primeira detecção é escolhida. Caso nenhuma face seja detectada, a imagem não sofre qualquer modificação além do redimensionamento.

5 Modelos para estimativa de idade

Os modelos para estimativa de idade facial estudados neste trabalho são apresentados em mais detalhes a seguir.

5.1 *Consistent Rank Logits*

Cao, Mirjalili e Raschka (2020) definem o *Consistent Rank Logits* como um *framework*, ou modelo, para regressão ordinal com garantia teórica da consistência do classificador sem aumento na complexidade do treino.

Segundo os autores, regressão ordinal descreve a tarefa de prever rótulos, ou *labels*, em uma escala ordinal. Assim, um classificador, ou *ranking rule*, h mapeia cada objeto $\boldsymbol{x}_i \in \mathcal{X}$ em um conjunto ordenado $h : \mathcal{X} \rightarrow \mathcal{Y}$, onde $\mathcal{Y} = \{r_1 \prec \cdots \prec r_k\}$. Além disso, a diferença entre os valores dos rótulos é arbitrária.

A abordagem de classificação binária estendida, proposta por Li e Lin (2007), é descrita por Cao, Mirjalili e Raschka (2020) como a base para várias implementações de regressão ordinal. Contudo, os autores pontuam que implementações de redes neurais de regressão ordinal apresentam inconsistências do classificador entre *rankings* binários.

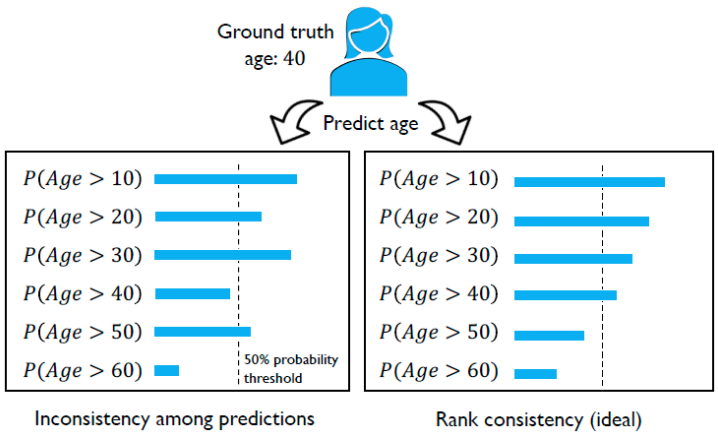


Figura 5.1: Ilustração das inconsistências que podem ocorrer entre classificadores binários individuais. À esquerda um modelo *rank*-inconsistente e à direita, *rank*-consistente. Fonte: Cao, Mirjalili e Raschka (2020).

É ilustrado, através da Figura 5.1, o problema da inconsistência entre as previsões

de classificadores binários individuais em modelos genéricos de redução de regressão ordinal para classificação binária.

Os autores mencionam que Niu et al. (2016) adotam a ideia de Li e Lin (2007) para propor sua abordagem, denominada *Ordinal Regression CNN*. Nela, um problema de regressão ordinal com K *ranks* é transformado em $K-1$ problemas de classificação binária, com a k-ésima tarefa prevendo se o rótulo da idade de um exemplo ultrapassa ou não o *rank* r_k, com $k = 1, \ldots, K-1$.

Todas as $K-1$ tarefas compartilham as mesmas camadas intermediárias, mas são atribuídos diferentes pesos na camada de saída.

Ainda assim, a abordagem não garante a consistência entre os classificadores, algo que é reconhecido pelos autores. Entretanto, observam que garantir a consistência dos $K-1$ classificadores causaria aumento substancial da complexidade do treinamento.

O modelo proposto por Cao, Mirjalili e Raschka (2020) aborda o problema da inconsistência no classificador na abordagem proposta por Niu et al. (2016).

Para definir formalmente, dada a base de treino $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, onde $\boldsymbol{x}_i$ é a i-ésima imagem, um *rank* y_i é primeiramente estendido em $K-1$ rótulos binários $y_i^{(1)}, \ldots, y_i^{(K-1)}$ de tal modo que $y_i^{(K-1)} \in \{0, 1\}$ indica y_i se ultrapassa ou não o *rank* r_k, ou seja, se $y_i > r_k$ ou não.

Com os rótulos binários estendidos como entrada durante o treino do modelo, os autores treinam uma CNN única com $K-1$ classificadores binários na camada de saída

Com base no retorno das tarefas binárias, o *rank* previsto para uma entrada $\boldsymbol{x}_i$ é obtido por

$$h(x_i) = r_q, \qquad q = 1 + \sum_{k=1}^{K-1} f_k(\boldsymbol{x}_i) \tag{5.1}$$

onde $f_k(x_i) \in \{0, 1\}$ é a previção do k-ésimo classificador binário na camada de saída.

Segundo os autores, é necessário que $\{f_k(\boldsymbol{x}_i)\}_{k=1}^{K-1}$ reflita a informação ordinal e seja monótono ($f_1(\boldsymbol{x}_i) \geq f_2(\boldsymbol{x}_i) \geq \cdots \geq f_{K-1}(\boldsymbol{x}_i)$), o que garante que as previsões sejam consistentes. Para alcançar esse objetivo e garantir a consistência do classificador binário, as $K-1$ tarefas binárias compartilham os mesmos parâmetros, mas possuem unidades de *bias* independentes.

O processo é ilustrado através da Figura 5.2.

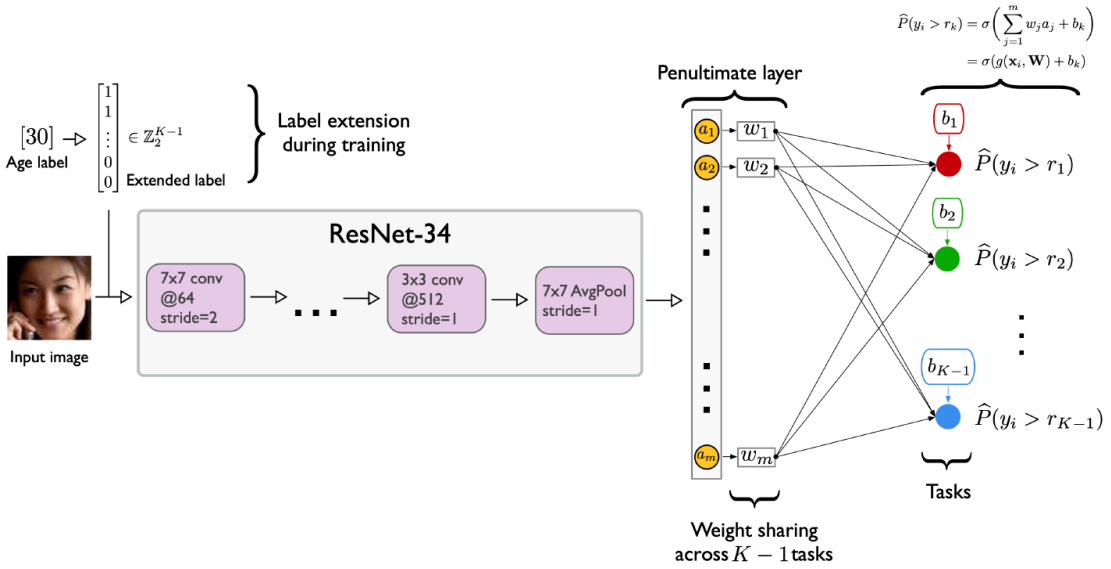


Figura 5.2: Ilustração do *Consistent Rank Logits*. Para os valores de probabilidade estimados, os rótulos (*labels*) binários são obtidos através da Equação (5.3) e convertidos para o rótulo da idade através da Equação (5.1). Fonte: Cao, Mirjalili e Raschka (2020).

Para a função de perda, os autores definem como $\boldsymbol{W}$ os parâmetros da rede excluindo-se as unidades de *bias* na última camada. A penúltima camada, que tem saída representada por $g(\boldsymbol{x}_i, \boldsymbol{W})$, compartilha um único peso com todos os nós na camada de saída final. $K-1$ unidades de *bias* independentes são então adicionadas a $g(\boldsymbol{x}_i, \boldsymbol{W})$ de tal modo que $\{g(\boldsymbol{x}_i, \boldsymbol{W}) + b_k\}_{k=1}^{K-1}$ são as entradas para os classificadores binários correspondentes na camada final.

Minimiza-se então, para o treino do modelo, a função de perda $L(\boldsymbol{W}, \boldsymbol{b})$, que é a entropia cruzada ponderada dos $K-1$ classificadores binários, expressa por

$$L(\boldsymbol{W}, \boldsymbol{b}) = -\sum_{i=1}^{N}\sum_{k=1}^{K-1} \lambda^{(k)}[\log(\widehat{P}_{i,k})y_i^{(k)} + \log(1-\widehat{P}_{i,k})(1-y_i^{(k)})], \tag{5.2}$$

onde $\widehat{P}_{i,k}$ denota a probabilidade empírica prevista para a tarefa k, definida por $\widehat{P}(y_i^{(k)} = 1) = \sigma(g(\boldsymbol{x}_i, \boldsymbol{W}) + b_k)$; $\lambda^{(k)}$ denota o parâmetro de importância para a tarefa k, que é o peso da perda associada ao k-ésimo classificador ($\lambda^{(k)} > 0$); e $\sigma(z) = 1/(1+exp(-z))$ é a função logística sigmoide.

Para a previsão do *rank*, os rótulos binários são obtidos através da equação

$$f_k(\boldsymbol{x}_i) = \mathbb{1}\{\widehat{P}(y_i^{(k)} = 1) > 0.5\}. \tag{5.3}$$

Os autores provaram teoricamente que ao minimizar a função de perda, definida na Equação (5.2), as unidades de *bias* aprendidas da camada de saída são não-crescentes de forma que $b_1 \geq b_2 \geq \cdots \geq b_{K-1}$.

Assim, os escores de confiança, ou estimativas de probabilidade, das $K-1$ tarefas são não-crescentes, garantindo a consistência, como ilustrado na expressão

$$\widehat{P}(y_i^{(1)} = 1) \geq \widehat{P}(y_i^{(2)} = 1) \geq \cdots \geq \widehat{P}(y_i^{(K-1)} = 1), \quad \forall i. \tag{5.4}$$

A implementação da rede é baseada na arquitetura *ResNet-34*, proposta por He et al. (2016), onde uma modificação é feita substituindo-se a última camada pelas tarefas binárias correspondentes.

5.2 Florestas de decisão neurais residuais

Li e Cheng (2019) propõem o conceito de florestas de árvores de decisão neurais residuais (RNDF), que é a utilização de aprendizado residual, proposto por He et al. (2016), em florestas de árvores de decisão neurais, propostas por Kontschieder et al. (2015), para realizar estimativa de idade facial. Essas técnicas foram descritas no Capítulo 2 desse trabalho. Adicionalmente, propõem a aplicação de uma técnica de visualização do processo de inferência da RNDF.

Nesse conceito, um nó de divisão $\mathcal{S}_i$ é associado à uma função de recomendação $\mathcal{R}_i$ que extrai características profundas da entrada $\boldsymbol{x}$ e retorna o escore de recomendação $s_i = \mathcal{R}_i(\boldsymbol{x})$ que a entrada obtém para a sua sub-árvore esquerda.

Cada nó folha guarda um vetor de previsão $\boldsymbol{p}_i$ de valores reais que representa a resposta por ele dada, como definem os autores. Para obter a previsão final $\boldsymbol{P}$, cada nó folha contribui com seu vetor de previsão ponderado de acordo com a probabilidade de seguir seu caminho de computação, calculada através da Equação

$$\boldsymbol{P} = \sum_{i \in \mathcal{N}_l} w_i \boldsymbol{p}_i, \tag{5.5}$$

onde $\mathcal{N}$ é o conjunto de todos os nós folha.

Ainda segundo os autores, o peso é obtido multiplicando-se todos os escores de

recomendação dados pelos nós de divisão ao longo do caminho de computação, ou seja, o caminho único $\mathcal{P}_i$ do nó raiz até a folha $\mathcal{L}_i$ passando por uma sequência de q nós de divisão s, como descrito na Equação

$$w_i = \prod_{m=1}^{q} (s_{i_m})^{\mathbb{1}(j_m=0)} \ (1 - s_{i_m})^{\mathbb{1}(j_m=1)}, \tag{5.6}$$

onde o valor de j_m é 0 caso a entrada seja roteada para a sub-árvore esquerda ou 1, caso seja roteada para a sub-árvore direita.

A previsão final é uma combinação convexa de todos os vetores de previsão dos nós folhas e os autores assumem que as funções de recomendação descritas são diferenciáveis e parametrizadas por Θ_i no nó i. Assim, a previsão final é diferenciável e, portanto, a função de perda pode ser minimizada por gradiente descendente.

Uma CNN profunda é então usada para extrair características da entrada e atribuir cada nó de divisão a um neurônio da última camada totalmente conectada, onde a função sigmoide é utilizada para computar os escores de recomendação finais.

Assim, os autores assumem a hipótese de que o aprendizado residual pode contribuir no aprendizado das funções de recomendação, utilizando então a NDF incorporada com o aprendizado residual. A função de perda utilizada é definida pela Equação

$$L(\mathbb{D}) = \frac{1}{2} \sum_{i=1}^{N} ||\boldsymbol{P}_i - \boldsymbol{y}_i||^2, \tag{5.7}$$

onde $\mathbb{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$ é um conjunto de dados com N entradas com rótulos associados.

Os parâmetros da rede são otimizados por gradiente descendente mantendo as previsões dos nós folha fixos, já que nestes, uma matriz de covariância é usada para especificar a incerteza da previsão e uma distribuição Gaussiana é assumida, com o vetor de previsão sendo usado como a média. As atualizações dos parâmetros da rede e das previsões das folhas são feitas de forma alternada.

A implementação da rede, ilustrada na Figura 5.3, é baseada na arquitetura *ResNet-50*, proposta por He et al. (2016), com duas camadas totalmente conectadas tendo suas saídas ativadas pela função sigmoide e enviadas para a floresta para obter as previsões finais.

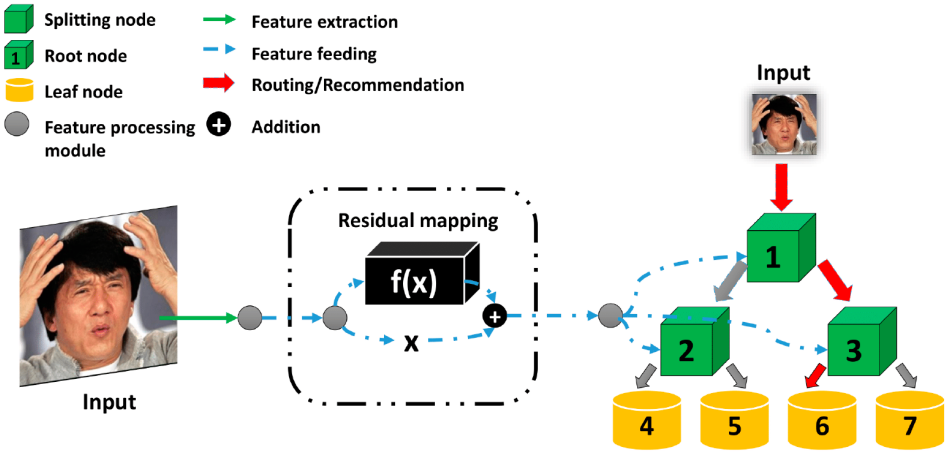


Figura 5.3: Ilustração de uma RNDF. As imagens de entrada são roteadas pelos nós de divisão e chegam à previsão dada pelos nós folha (setas vermelhas). Características são extraídas da entrada e enviadas para cada nó de divisão para tomada de decisão (setas azuis). O aprendizado residual é incorporado ao processo de extração de características para contribuir com a otimização das funções de decisão. Fonte: Li e Cheng (2019).

Os autores especificam que a visualização da influência da entrada no processo de tomada de decisão do modelo, denominado mapa de saliência de decisão (*Decision Saliency Map* - DSM) é feito utilizando o gradiente da probabilidade de roteamento em relação à entrada, como descrito na Equação

$$DSM = \frac{\partial s_i}{\partial \boldsymbol{x}} \tag{5.8}$$

Os autores afirmam ainda que o mapa de saliência é computado para a NDF e regressão de idade, diferentemente das abordagens anteriores desse tipo de visualização, focadas em CNNs tradicionais e classificação de imagens.

6 Experimentos

Para verificar a eficácia das abordagens estudadas em relação ao conjunto de dados analisado neste trabalho, foram conduzidos experimentos com ambas as implementações discutidas no capítulo anterior.

A Seção 6.1 descreve os parâmetros utilizados pelos autores nas duas abordagens, além dos resultados publicados por cada um. Já a Seção 6.2 apresenta os resultados obtidos ao realizar experimentos utilizando os mesmos parâmetros que os respectivos autores em cada abordagem, mas com os *splits* propostos no Capítulo 4 deste trabalho. Além disso, apresenta os resultados obtidos ao realizar a fusão tardia dos resultados obtidos durante o teste.

6.1 Especificações

6.1.1 *Consistent Rank Logits*

Cao, Mirjalili e Raschka (2020) utilizam como métricas para avaliar o desempenho do modelo CORAL o Erro Médio Absoluto (MAE) e a Raiz do Erro Quadrático Médio (RMSE).

Os autores afirmam que foram treinados três modelos com diferentes sementes aleatórias (0, 1 e 2) para a inicialização dos pesos do modelo. O treino é feito com 200 épocas e com *batch* de tamanho 256.

Além disso, é escolhido que a ponderação das tarefas seja uniforme para a entropia cruzada dos $K-1$ classificadores binários. Em outras palavras, os autores definem que, para todo k, $\lambda^{(k)} = 1$ na Equação (5.2).

A taxa de aprendizado foi fixada em 5×10^{-5} após um processo de ajuste (*tuning*) dos hiperparâmetros no conjunto de validação.

Ainda segundo os autores, ao final das 200 épocas, o melhor modelo é selecionado com base na performance do MAE no conjunto de validação e avaliado com o conjunto

de teste.

Todas as imagens passam por um processo de acréscimo de dados que consiste no redimensionamento para 128×128 píxeis seguido de corte randômico para 120×120 píxeis. Durante a avaliação do modelo, as imagens são cortadas de forma central para o tamanho de 120×120 píxeis para entrada do modelo.

Os resultados publicados pelos autores são apresentados na Tabela 6.1.

Tabela 6.1: Resultados apresentados pelos autores do CORAL utilizando o conjunto de dados CACD. Os resultados são referentes ao conjunto de testes utilizado por eles. Fonte: Cao, Mirjalili e Raschka (2020).

Semente	**MAE**	**RMSE**
0	5,25	7,41
1	5,25	7,50
2	5,24	7,52
	5,25 ± 0,01	7,48 ± 0,06

6.1.2 Florestas de decisão neurais residuais

Li e Cheng (2019) utilizam como métricas para avaliar o desempenho da RNDF o Erro Médio Absoluto (MAE) e Escore Cumulativo (CS).

Um único treino é feito com 8 épocas e com *batch* de tamanho 50. A floresta é composta por 5 árvores, cada uma tendo profundidade 6. Ainda de acordo com os autores, os vetores de previsão nos nós folha são atualizados a cada 50 *batches*. A taxa de aprendizado inicial é de 5×10^{-1}, sendo dividida pela metade quando o treino do modelo atinge o platô.

Os dados de treino já pré-processados passam por um processo de acréscimo de dados que consiste no giro horizontal das imagens de forma randômica com probabilidade 0,5 e no corte aleatório para que o tamanho espacial final das imagens seja de 224×224 píxeis. No conjunto de teste, são utilizados apenas cortes de forma central.

As métricas de desempenho são utilizadas junto ao conjunto de teste para avaliar a performance. Especificamente, um limite de 5 anos é utilizado no Escore Cumulativo. Entretanto, o Escore Cumulativo não é publicado pelos autores para o conjunto de dados

utilizado. Assim, o MAE informado é de **4,595**.

6.2 Resultados obtidos

6.2.1 *Splits* estratificados

Como mencionado no Capítulo 4 desse trabalho, *splits* foram gerados para que a comparação seja feita de forma justa, efetuando alterações apenas no pré-processamento do CORAL para que imagens não fossem descartadas, possibilitando a condução dos experimentos com os *splits* gerados.

Os modelos foram treinados com os mesmos parâmetros especificados pelos autores em suas respectivas publicações, utilizando código-fonte disponibilizado por ambos em repositório.

Os treinos ocorreram nas duas abordagens utilizando, em cada uma, ambas as versões pre-processadas do conjunto de dados.

Todos os experimentos foram realizados utilizando Python 3.6.10 e PyTorch 1.4.0. A placa gráfica utilizada foi o modelo NVIDIA GeForce GTX 1070 SC, do fabricante EVGA.

Os resultados obtidos nos experimentos com o CORAL são apresentados nas Tabelas 6.2 e 6.3.

Os resultados obtidos nos experimentos com o RNDF são apresentados nas Tabelas 6.4 e 6.5. Além disso, a visualização por DSMs é apresentada na Figura 6.1.

Experimentos extras foram feitos gerando novos conjuntos de *splits* com as mesmas proporções utilizadas por cada autor para verificar a influência dessas proporções. Foi utilizado o pré-processamento correspondente à abordagem. A abordagem CORAL obteve MAE = 5,00 $\pm$ 0,03 e a abordagem RNDF, MAE = 4,769.

Os resultados apresentados nas Tabelas 6.2 e 6.4 demonstram, a julgar pelo erro médio absoluto e pela raiz do erro quadrático médio, que a abordagem RNDF mantém desempenho superior à abordagem CORAL para o conjunto de dados utilizado com respectivo pré-processamento e com os *splits* propostos no Capítulo 4 desse trabalho.

Entretanto, a abordagem RNDF apresenta queda de desempenho em relação ao

Tabela 6.2: Resultados obtidos nos experimentos com a implementação CORAL com cada um dos conjuntos de dados pré-processados. Uma comparação desses pré-processamentos pode ser vista na Tabela 4.1.

Pré-processamento	**Semente**	**MAE**	**RMSE**	**CS**
CORAL	0	5,15	7,46	0,68
	1	5,14	7,50	0,68
	2	5,13	7,39	0,68
		5,14 ± 0,01	7,45 ± 0,05	
RNDF	0	5,19	7,54	0,68
	1	5,19	7,48	0,67
	2	5,15	7,46	0,68
		5,18 ± 0,02	7,49 ± 0,03	

Tabela 6.3: Resultados obtidos nos experimentos com a implementação CORAL para duas das imagens do conjunto de dados escolhidas aleatoriamente com o pré-processamento da própria implementação.

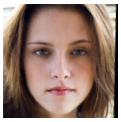
Imagem	**Idade Real**	**Idade Prevista**	**Semente**
	19	22	0
		21	1
		22	2
	47	47	0
		48	1
		47	2

Tabela 6.4: Resultados obtidos nos experimentos com a implementação RNDF com cada um dos conjuntos de dados pré-processados. Uma comparação desses pré-processamentos pode ser vista na Tabela 4.1.

Pré-processamento	**MAE**	**RMSE**	**CS**
CORAL	5,23	7,18	0,60
RNDF	4,81	6,80	0,65

que é publicado pelos seus autores, fator que pode ser atribuído à mudança na divisão dos *splits*. Enquanto os autores reservam mais de 90% das imagens para o treino em seus

Tabela 6.5: Resultados obtidos nos experimentos com a implementação RNDF para duas das imagens do conjunto de dados escolhidas aleatoriamente com o pré-processamento da própria implementação.

Imagem	**Idade Real**	**Idade Prevista**
	19	24,80
	47	51,40

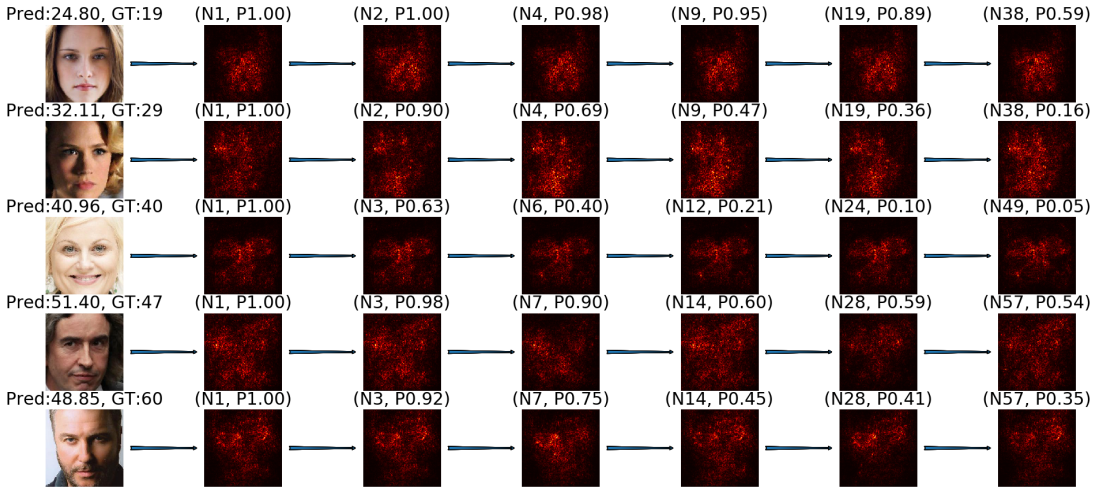


Figura 6.1: Visualização dos DSMs de algumas imagens aleatórias do conjunto de dados ao longo do caminho de computação da entrada no RNDF. As regiões em vermelho são as consideradas pela floresta ao fazer a análise. Acima das entradas, são indicadas a idade prevista (Pred) e a idade real (GT). Os pares (Nα, Pβ) indicam que a entrada passa pelo nó de divisão α com probabilidade β durante o processo tomada de decisão.

splits, nos experimentos realizados para este trabalho são utilizadas 70% das imagens, como descrito no Capítulo 4.

Já a abordagem CORAL, em relação ao publicado pelos seus autores, apresenta, em erro médio absoluto, eficácia levemente superior, enquanto apresenta, do ponto de vista da raiz do erro quadrático médio, resultados próximos dos publicados.

Nota-se que o Escore Cumulativo é 3% maior em relação à abordagem RNDF. Entretanto, é válido observar que os autores da abordagem CORAL realizam alguns descartes de imagens do conjunto de dados, que não são realizados neste trabalho.

Com isso, observa-se que, mesmo com os autores do CORAL reservando 80% das imagens do conjunto de dados para o treino, uma fatia ainda superior à utilizada nos *splits* utilizados no experimento, o desempenho não sofreu queda. Elege-se como possível fator para isso o fato de os *splits* serem estratificados, dado que, mesmo com a não realização de descartes, o número de imagens utilizado durante o experimento no treino é próximo ao número utilizado pelos autores. Ressalta-se ainda a natureza empírica dos experimentos nessa abordagem.

Adicionalmente, a utilização do pré-processamento do conjunto de dados feito pelos autores do RNDF para treinar o modelo CORAL não impactou o desempenho do modelo, enquanto a utilização do pré-processamento do conjunto de dados feito pelos autores do CORAL para treinar o modelo RNDF resultou em piora do desempenho do modelo.

Para verificar o comportamento das previsões de ambas as abordagens, é feita uma análise sobre quantas imagens correspondem e quantas foram associadas pelos modelos à cada idade dentro dos limites do conjunto de dados, utilizando o *split* de teste proposto e os modelos treinados com seus respectivos pré-processamentos.

Ressalta-se que, mesmo que o número de imagens correspondentes e associadas pelos modelos seja o mesmo para uma determinada idade, não é possível afirmar que todas as previsões para esta idade foram corretas.

As Figuras 6.2, 6.3 e 6.4 apresentam o comportamento dos modelos treinados na abordagem CORAL para cada semente. Através delas, observa-se que as previsões são distribuídas por todas as idades possíveis, com leve queda do número de previsões associadas às idades iniciais e finais. O comportamento da média simples entre as previsões das sementes, apresentado na Figura 6.5, mantém comportamento parecido, como esperado.

Já o comportamento do modelo treinado na abordagem RNDF, apresentado na Figura 6.6, concentra suas previsões aproximadamente entre 20 e 57 anos, demonstrando maior dificuldade do modelo com as idades iniciais e finais, mas ainda assim obtendo desempenho médio superior.

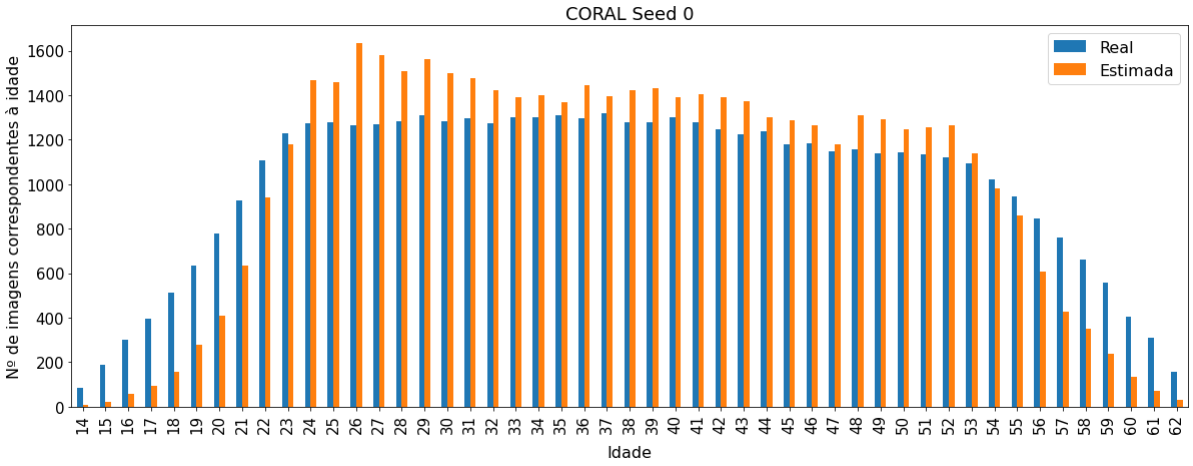


Figura 6.2: Gráfico ilustrando as respectivas quantidades reais e previstas de imagens de pessoas para cada idade na abordagem CORAL com semente 0.

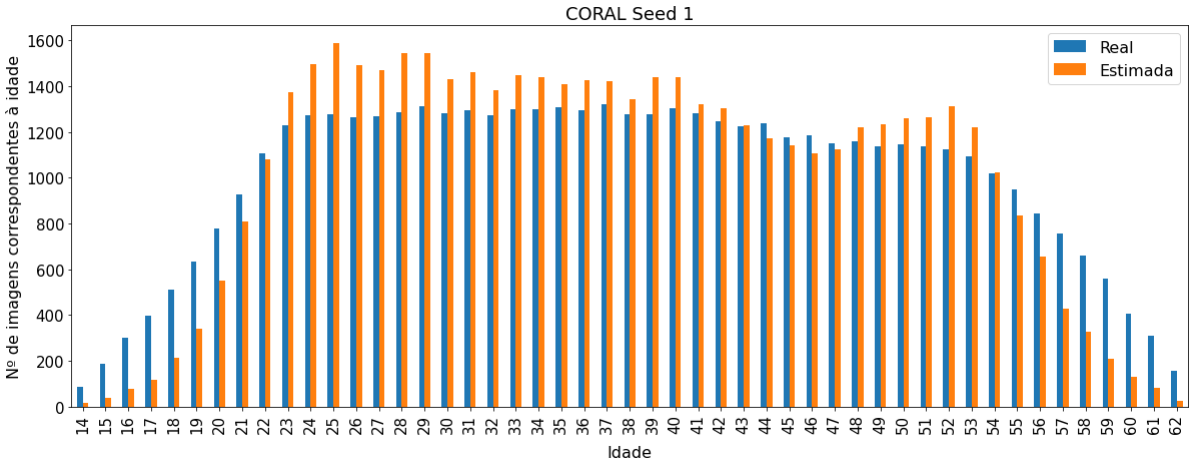


Figura 6.3: Gráfico ilustrando as respectivas quantidades reais e previstas de imagens de pessoas para cada idade na abordagem CORAL com semente 1.

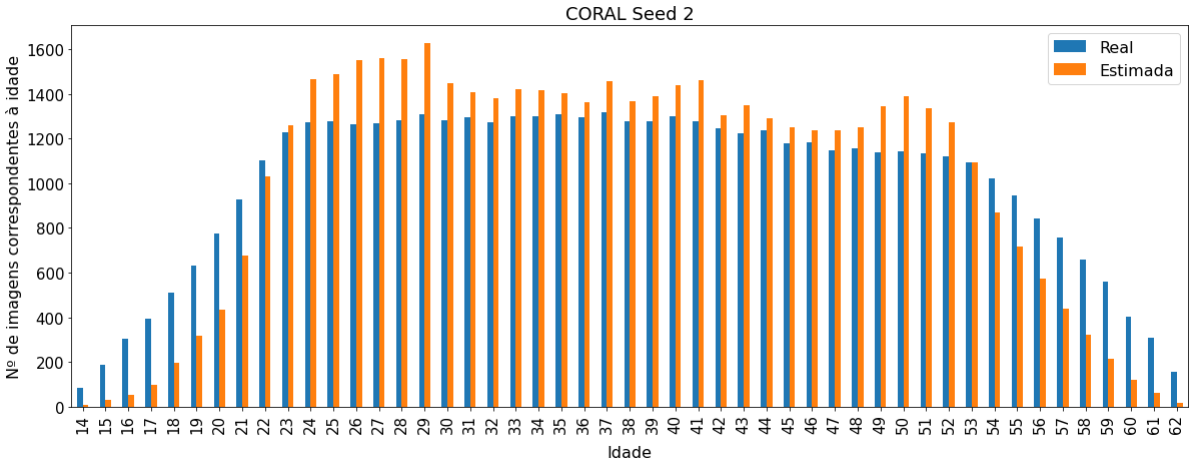


Figura 6.4: Gráfico ilustrando as respectivas quantidades reais e previstas de imagens de pessoas para cada idade na abordagem CORAL com semente 2.

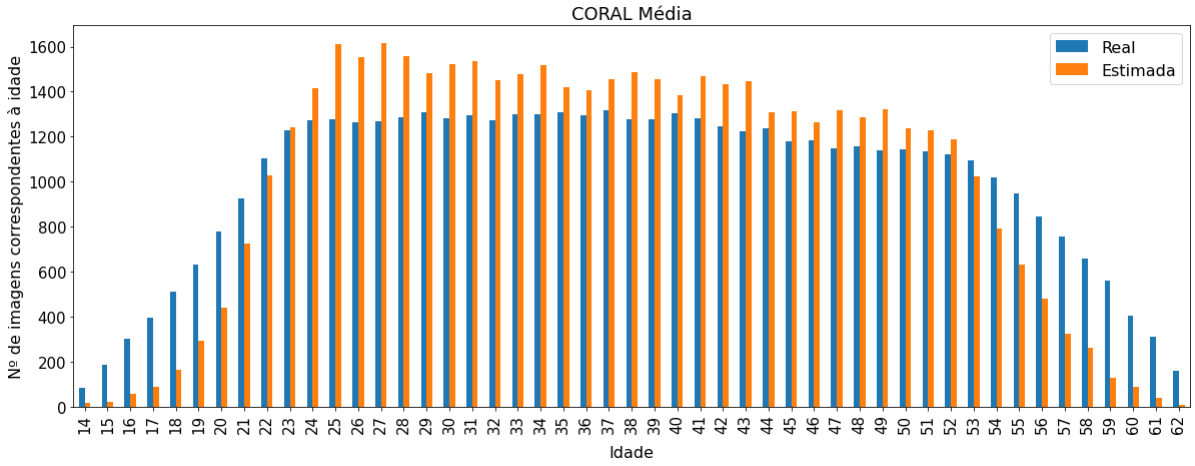


Figura 6.5: Gráfico ilustrando as respectivas quantidades reais e previstas de imagens de pessoas para cada idade utilizando a média entre as previsões da abordagem CORAL para cada semente.

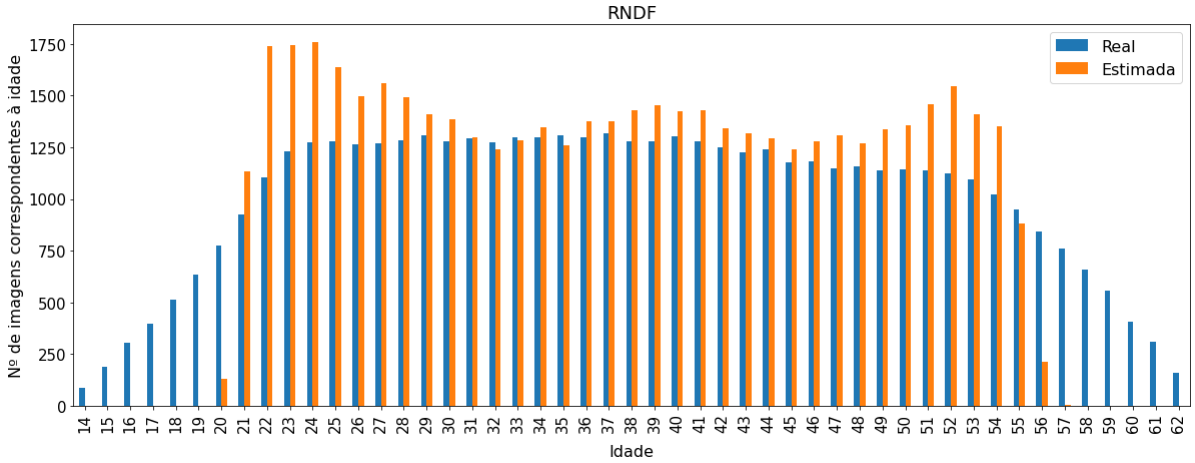


Figura 6.6: Gráfico ilustrando as respectivas quantidades reais e previstas de imagens de pessoas para cada idade na abordagem RNDF.

6.2.2 Fusão tardia

Neste trabalho, assume-se a hipótese de que é possível obter melhor eficácia aplicando a técnica de fusão tardia ao combinar as previsões realizando ponderações.

Para tanto, são utilizados os valores previstos pela abordagem RNDF com seu respectivo pré-processamento, a média aritmética simples dos valores previstos nos modelos CORAL com seu respectivo pré-processamento, além de ambas as abordagens trocando-se os pré-processamentos.

Assim, experimentos com diversas ponderações entre os quatro modelos treinados foram realizados, com alguns desses resultados apresentados na Tabela 6.6. Os modelos treinados com pré-processamento trocado apresentam o sufixo “-INV”.

Tabela 6.6: Resultados obtidos ao realizar experimentos de fusão das abordagens a partir de média ponderada entre elas. Os resultados apresentados para o CORAL correspondem à média aritmética simples dos modelos de diferentes sementes.

Ponderação				**MAE**	**RMSE**	**CS**
CORAL	**RNDF**	**CORAL-INV**	**RNDF-INV**			
25	75	0	0	4,57	6,53	0,68
40	60	0	0	4,49	6,46	0,69
50	50	0	0	4,48	6,46	0,69
60	40	0	0	4,48	6,49	0,69
75	25	0	0	4,55	6,60	0,69
0	0	25	75	4,89	6,82	0,64
0	0	40	60	4,76	6,70	0,65
0	0	50	50	4,70	6,66	0,66
0	0	60	40	4,66	6,65	0,67
0	0	75	25	4,66	6,71	0,68
40	30	20	10	4,45	6,44	0,69
30	30	20	20	4,47	6,43	0,69
25	25	25	25	4,49	6,45	0,69
20	20	30	30	4,52	6,47	0,68
10	20	30	40	4,58	6,51	0,67
30	40	10	20	4,49	6,44	0,69
20	10	40	30	4,55	6,52	0,68
20	30	40	10	4,46	6,44	0,69
20	30	30	20	4,47	6,43	0,69
30	20	20	30	4,52	6,47	0,68
30	20	10	40	4,58	6,52	0,67

Nota-se, pelos resultados dos experimentos de fusão tardia, que o melhor caso é ponderando da seguinte forma: CORAL = 40; RNDF = 30; CORAL-INV = 20; RNDF-INV = 10. Ao considerar apenas os modelos das duas abordagens com seus respectivos pré-processamentos, o melhor resultado obtido é ponderando igualmente as previsões das duas abordagens.

O comparativo de resultados em relação ao obtido com cada abordagem, apresen-

tado na Tabela 6.7, demonstra que a fusão entre os quatro modelos com as ponderações mencionadas manteve erro médio absoluto inferior em relação aos resultados obtidos com cada abordagem separadamente.

Resultados melhores também são observados na raiz do erro quadrático médio, o que indica uma redução média da distância dos valores previstos para os valores reais, além de pequeno aumento no escore cumulativo, que indica que 69% das imagens tem previsões que distanciam em no máximo 5 anos, para mais ou para menos, dos valores reais.

Tabela 6.7: Comparativo entre os resultados das abordagens utilizadas e o melhor resultado obtido através da fusão tardia.

Modelo	**MAE**	**RMSE**	**CS**
Média CORAL	4,79	6,95	0,68
RNDF	4,81	6,80	0,65
Fusão tardia	4,45	6,44	0,69

O comportamento das previsões utilizando a melhor combinação obtida em fusão tardia, apresentado na Figura 6.7, demonstra um aumento das previsões em idades iniciais e finais, possivelmente por influência do modelo CORAL. Ainda que o número de previsões nessas idades se mantenha baixo, a eficácia média é superior, como mencionado anteriormente.

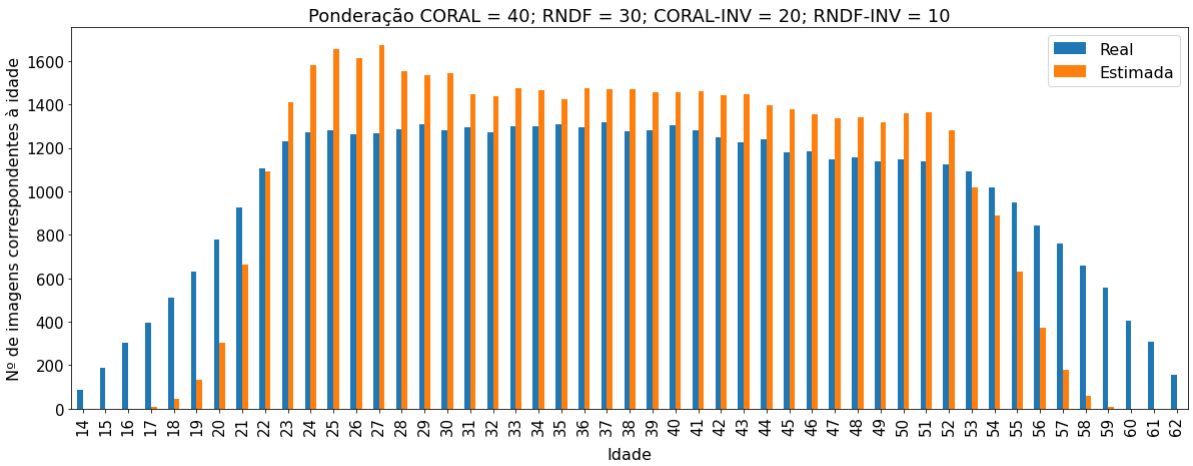


Figura 6.7: Gráfico ilustrando as respectivas quantidades reais e previstas de imagens de pessoas para cada idade utilizando a fusão tardia entre as abordagens analisadas com a melhor ponderação encontrada, especificada no título do gráfico.

7 Considerações finais

As abordagens aqui estudadas utilizam diferentes métodos para tratar de um mesmo problema, cada uma propondo meios, segundo os autores, não vistos até então, como a garantia teórica da consistência do classificador ou o uso de aprendizado residual associado a florestas de árvores de decisão neurais.

O conjunto de dados utilizado para a análise mostra-se desafiador pela variedade de imagens em diversas condições de iluminação, posição e cor, tornando imprescindível o seu pré-processamento para obter um melhor desempenho.

Os autores do conjunto de dados não disponibilizam *splits* definidos, o que leva os autores das abordagens a gerarem seus próprios, com diferentes proporções. Isso, em conjunto com a necessidade vista de se levar em conta a estratificação ao gerar os *splits*, levou à criação deles, proposta no Capítulo 4 desse trabalho.

Através dos resultados apresentados na Tabela 6.4, observa-se que, mesmo que o RNDF tenha apresentado uma queda de desempenho, manteve-se melhor em relação ao CORAL utilizando seu próprio pré-processamento das imagens.

Já quanto ao CORAL, com resultados apresentados na Tabela 6.2, observa-se que a utilização dos *splits* gerados junto ao seu próprio pré-processamento do conjunto de dados gerou resultado próximo ao publicado pelos autores.

Nota-se ainda que a utilização dos conjuntos de dados pré-processados de uma abordagem no modelo da outra gerou pouca diferença no CORAL. Entretanto, uma diferença razoável pode ser vista no RNDF.

Além disso, observa-se, ao comparar os resultados dos experimentos extras com os principais, que a proporção utilizada pelos autores do CORAL associada à estratégia utilizada para geração dos *splits* gerou resultado superior, enquanto que com o RNDF, gerou resultado próximo ao obtido no experimento principal.

A aplicação proposta neste trabalho da técnica de fusão tardia gerou resultados melhores com base nas métricas utilizadas, como pode ser visto na Tabela 6.7, onde observa-se leve melhora no resultado obtido em relação ao publicado pelos autores do

RNDF, que detém o estado-da-arte no conjunto de dados utilizado.

Existem algumas estratégias que podem ser aplicadas em trabalhos futuros. Uma delas é melhorar a fusão tardia utilizando os pesos na última camada das redes utilizadas ou efetuando diferentes ponderações.

Algumas ideias propostas pelos autores dos modelos também podem ser utilizadas, como a utilização de árvores com mais de duas sub-árvores no RNDF. Adicionalmente, experimentos com outros conjuntos de dados podem ser realizados com o objetivo de verificar de forma mais aprofundada a influência da geração de *splits* na eficácia dos modelos.

Bibliografia

AKARSH, S. et al. Deep learning framework and visualization for malware classification. In: IEEE. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. [S.l.], 2019. p. 1059–1063.

ALSHAWWA, I. A. et al. Analyzing types of cherry using deep learning. 2020.

BERG, A.; OSKARSSON, M.; O'CONNOR, M. Deep ordinal regression with label diversity. *arXiv preprint arXiv:2006.15864*, 2020.

CAO, W.; MIRJALILI, V.; RASCHKA, S. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, Elsevier BV, v. 140, p. 325–331, Dec 2020. ISSN 0167-8655. Disponível em: ⟨http://dx.doi.org/10.1016/j.patrec.2020.11.008⟩.

CHEN, B.-C.; CHEN, C.-S.; HSU, W. H. Cross-age reference coding for age-invariant face recognition and retrieval. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2014.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. ⟨http://www.deeplearningbook.org⟩.

GUO, Y. et al. Deep learning for visual understanding: A review. *Neurocomputing*, Elsevier, v. 187, p. 27–48, 2016.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

HUANG, F. et al. A deep learning algorithm using a fully connected sparse autoencoder neural network for landslide susceptibility prediction. *Landslides*, Springer, v. 17, n. 1, p. 217–229, 2020.

KONTSCHIEDER, P. et al. Deep neural decision forests. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1467–1475.

LI, L.; LIN, H.-T. Ordinal regression by extended binary classification. MIT press, 2007.

LI, S.; CHENG, K.-T. Facial age estimation by deep residual decision making. *arXiv preprint arXiv:1908.10737*, 2019.

NIU, Z. et al. Ordinal regression with multiple output cnn for age estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 4920–4928.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015.

SHIRAI, Y. *Three-dimensional computer vision*. [S.l.]: Springer Science & Business Media, 2012.

TORRES, J. F. et al. A scalable approach based on deep learning for big data time series forecasting. *Integrated Computer-Aided Engineering*, IOS press, v. 25, n. 4, p. 335–348, 2018.

ZHANG, C. et al. C3ae: Exploring the limits of compact model for age estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 12587–12596.

ZHANG, Y. et al. A late fusion cnn for digital matting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 7469–7478.