Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Avaliação de Imagens com Aprendizado de Máquina

Ian Silva Galvão

Monografia Final

mac 499 — Trabalho de Formatura Supervisionado

Supervisor: Prof. Alfredo Goldman
Cossupervisor: Renato Cordeiro

São Paulo
2017

Agradecimentos

Agradeço, primeiramente, ao Goldman e ao Renato, meus supervisores, o apoio conhecimento e incentivo que me deram ao longo desta pesquisa. Agradeço as reuniões contantes, que não deixaram a pesquisa estagnar. Por fim, agradeço carinho que os dois tem pelo ensino e pelos alunos.

Agradeço imensamente a todos àqueles que mantiveram o IME funcionando durante a difícil pandemia que passamos. Manter a cabeça ocupada com educação foi essencial.

Agradeço à membros da minha família por todo apoio durante minha graduação, sobretudo às minhas avós, Efigênia e Adelaide e ao meu irmão, Teo.

Os meus pais merecem um troféu por apoiar a educação do filho por quase 30 anos, sem nunca desistir. Muito obrigado por estarem sempre ao meu lado! Sempre estarei ao lado de vocês.

À minha companheira Ludmila, obrigado por ser minha 'tutora não oficial', me ensinando tanta coisa do IME e da computação. Agradeço todos os dias você estar comigo todos os dias. Vou deixar até um S2 aqui!

Resumo

Ian Silva Galvão. **Avaliação de Imagens com Aprendizado de Máquina**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2017.

Essa pesquisa realiza um estudo avaliação de qualidade de imagens com aprendizado de máquina supervisionado. Foi selecionado o banco de dados KonIQ-10k com imagens avaliadas por *crowdsourcing*. Em seguida, foram extraídas características representativas dos seguintes aspectos estéticos das imagens: luminosidade, paleta de cores, desfoque, e distribuição de detalhes. A partir das características extraídas e avaliações das imagens, foram treinados cinco modelos com técnicas distintas de aprendizado de máquina supervisionado: regressão linear, k-vizinhos, árvore de decisão, floresta aleatória e rede neural. O treinamento contou com a otimização de hiperparâmetros e validação cruzada. Por fim, foi realizada uma extração de características e alternativa, com *deep learning* e transferência de aprendizado. A modelagem com base nessas características extraídas com deep learning apresentou melhor desempenho que as demais. Os resultados foram analisados e comparados com um modelo da literatura treinado no mesmo banco de dados.

Palavras-chave: Aprendizado de Máquina. Processamento de Imagens. Avaliação de Qualidade de Imagens.

Abstract

Ian Silva Galvão. **Image Quality Assesment with Machince Learning**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2017.

This research performs a study of image quality assesment with supervised machine learning methods. The KonIQ-10k database, with images qualities evaluated by crowdsourcing, was selected. Then, a set of characteristics were extracted, that are representative of the following aesthetic aspects: luminosity, color palet, blur and details distribution. From the extracted features and image evaluations, five models were trained with different supervised machine learning techniques: linear regression, k-neighbors, decision tree, random forest and neural network. The training included hyperparameter optimization and cross-validation. The training included hyperparameter optimization and cross-validation. Finally, it was develeped an alternative feature extraction, using deep learning and transfer learning. The model based on these features extracted with deep learning performed better than the previous ones. The results were analyzed and compared with a model from the literature trained on the same database.

Keywords: Machine Learning. Image Processing. Image Quality Assesment.

Lista de Figuras

Lista de Tabelas

Lista de Programas

Sumário

Capítulo 1

Introdução

A avaliação de qualidade de imagens é uma área interdisciplinar que busca predizer qualidade estéticas de imagens, e que encontra diversas aplicações (Wang, 2011). A qualidade de uma imagem pode ser definida de diferentes formas, e mesmo considerando uma definição subjetiva de qualidade, podem ser utilizados diferentes métodos para metrificar o problema (Datta, Li *et al.*, 2008; Ma *et al.*, 2017). A partir de métricas objetivas, uma das abordagens utilizadas é o uso de técnicas de aprendizado de máquina supervisionado (Datta, Joshi *et al.*, 2006; Hosu *et al.*, 2019).

Com o objetivo de estudar os fundamentos e técnicas da área, esta pesquisa realiza um estudo de avaliação de qualidade de imagens com aprendizado de máquina supervisionado. O processo foi divido em em três etapas:

1. seleção de um banco de dados de imagens, que contenha avaliações estéticas dessas imagens;
2. extração de características de cada imagem, que representem informações relativas a estética das imagens; e
3. desenvolvimento de um modelo de regressão que utiliza essas características para predizer a avaliação dos usuários.

1.1 Metodologia

A busca pelos dados do projeto levou em consideração bancos de dados utilizados em pesquisas de avaliação de qualidade de imagens.

A etapa de extração foi feita seguindo as descrições dos algoritmos apresentadas nos artigos de referência, e a seleção dos métodos implementados está descrita no capítulo 4. A extração das características selecionadas foi realizada em todas as imagens do banco antes de prosseguir com a etapa de modelagem, de forma a simplificar o processo.

A modelagem foi feita de modo incremental e experimental, com o uso de *Jupyter notebooks* para fornecer rapidez na visualização dos resultados e documentação das análises exploratórias. Optou-se por iniciar o processo com o treinamento de um modelo de

regressão linear e aumentar a complexidade dos modelos juntamente com as ferramentas de análise e suporte.

1.2 Ambiente

O hardware utilizado foi um laptop Asus com processador Intel i5 quadcore, placa gráfica Mesa Intel HD Graphics 620 e 16GB de memória RAM. O sistema operacional utilizado foi Ubuntu.

O projeto foi desenvolvido em Python, com Docker e Poetry para fornecer reprodutibilidade para o código, mantido no Github. A maior parte do desenvolvimento foi feito em Jupyter Notebooks.

Capítulo 2

Fundamentos

Os principais conceitos necessários para o entendimento do projeto desenvolvido são das áreas de processamento de imagens e de aprendizado de máquina. O processamento das imagens ocorre na primeira etapa do projeto, com a extração de um vetor de características numéricas para cada imagem. A segunda etapa utiliza os conceitos de aprendizado de máquina para produzir um modelo com capacidade de predizer a avaliação dos usuários para uma imagem dada, a partir das características extraídas na primeira etapa.

Dessa forma, os conceitos de processamento de imagem, apresentados na próxima seção desse capítulo, cobrem os requisitos para o entendimento do processo de extração de características, descrito no capítulo 4, e os conceitos de aprendizado de máquina cobrem a etapa de modelagem vistas no capítulo 5.

2.1 Processamento de Imagens

Nesta seção, são apresentados conceitos de representação digital de imagens e de espaços de cores. São descritos dois algoritmos utilizados nesta pesquisa: o filtro laplaciano e a transformada de Fourier em duas dimensões. Finalmente, é fornecido uma breve explicação da conexão entre filtros gaussianos e a transformada de Fourier.

2.1.1 Representação Digital de uma Imagem

Uma imagem monocromática pode ser vista como uma função $f(x, y) : \mathbb{R}^2 \implies \mathbb{R}$. Imagens geradas analogicamente assumem valores contínuos tanto na sua imagem quando no seu domínio. Para a representação digital de uma imagem é necessário um processo de discretização e amostragem. O resultado desse processo é um conjunto discreto de ***pixels***, usualmente representado por uma matriz (Gonzalez e Woods, 2008).

Para imagens coloridas são utilizadas múltiplas matrizes, chamadas de **canais**. Para o restante do capítulo, quando não especificado de outra forma, considera-se que uma imagem se refere à uma imagem digital de um único canal. Convenciona-se que o pixel na posição (x, y) de uma imagem I é denotado por I(x,y), e que seu valor pertence aos reais positivos.

2.1.2 Espaços de Cores

Esta pesquisa utiliza transformações entre espaços de cores para extrair características das imagens relativas ao seu aspecto estético. Em particular, as transformações entre espaços de cor utilizadas partem do espaço RGB para um dos dois espaços descritos abaixo:

- O espaço Lab representa as cores de uma imagem em um canal de luminosidade (L) e dois de crominância (a e b). O canal de L será utilizado para extrair características relativas à luminosidade da imagem.
- O espaço HSV representa a imagem em crominância, saturação e luminosidade. A crominância dá o valor do pixel relativo à posição da cor representada no espectro cromático. A saturação oferece o valor do quão presente essa cor é no *pixel*, em relação à um pixel cinza de mesma luminosidade.

  Os canais H e S foram utilizados para a extração de características relacionadas às cores das imagens. O canal V não foi utilizado pois o espaço de cor Lab utiliza um método mais preciso para calcular a luminosidade.

2.1.3 Filtro de Convolução

Filtros de convolução são transformações em imagens que levam em conta a vizinhança de cada pixel e uma matriz com a mesma dimensão dessa vizinhança, chamada de máscara de convolução. Essa matriz é convolucionada com a vizinhança de cada pixel, e o resultado mapeado em uma nova imagem (Gonzalez e Woods, 2008). A convolução discreta C de uma máscara M com uma imagem I, ambas de tamanho N por M, é dada pela equação:

$$C = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} M(i,j)I(i,j) \tag{2.1}$$

2.1.4 Filtro Laplaciano

O filtro laplaciano é um filtro de convolução com a seguinte máscara:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Essa máscara pode ser escrita como a soma das duas máscaras abaixo:

$$L_x(x,y) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \; L_y(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.2}$$

Os filtros com máscaras L_x e L_y são versões discretizadas das derivadas parciais de segunda ordem da imagem no eixo x e y, respectivamente. Isso pode ser observado notando-se que cada um deles apresenta características desejadas para uma derivada de segunda ordem de

valor discreto: a convolução resulta em 0 em vizinhanças de valor ou variação constante, e em valor diferente de 0 quando há mudança na taxa de variação. O filtro laplaciano, dessa maneira, é sensível a mudanças de segunda ordem nos valores da imagem. Por isso, pode ser utilizado para localizar saliências(Ke *et al.*, 2006).

2.1.5 Transformada de Fourier

A transformada de Fourier converte um sinal no domínio do tempo ou espaço para o domínio das frequências. De modo intuitivo, ela decompõe a função inicial em uma soma de senoides de diferentes frequências, de forma que temos, após a transformada, as amplitudes das senoides em função da frequência, ao invés do valor do sinal em função de uma variável no domínio original (Gonzalez e Woods, 2008).

Este método pode ser aplicado em casos contínuos ou discretos. Esta pesquisa utiliza a transformada discreta de Fourier em duas dimensões. Formalmente, a transformada de Fourier Y(I) de uma matriz $I(N, M)$, – correspondente a um canal de uma imagem – é uma bijeção dada por:

$$Y(x, y) = \frac{1}{NM} \sum_{j=0}^{N-1} \sum_{k=0}^{M-1} e^{-2\pi i(\frac{x*j}{N} + \frac{y*k}{M})} I(i, j). \tag{2.3}$$

Sua inversa é:

$$Y^{-1}(x, y) = \frac{1}{NM} \sum_{j=0}^{N-1} \sum_{k=0}^{M-1} e^{2\pi i(\frac{x*j}{N} + \frac{y*k}{M})} I(i, j). \tag{2.4}$$

´ Para este trabalho, é relevante o modo como os valores da imagem se comportam em relação aos valores da transformada. O valor da transformada na origem do domínio das frequências, $Y(0, 0)$, corresponde ao nível de cinza geral da imagem. Aumentando-se esse valor no domínio das frequências aumenta-se por igual a luminosidade de toda imagem no domínio espacial. Próximo da origem $Y(0, 0)$, mudanças no valor da transformada correspondem a variações de baixa frequência na imagem. Afastando-se da origem, têm se variações de maior frequência.

Com isso, variações em pequena escala na imagem (nível de detalhe) correspondem a maiores valores na faixa de altas frequências na transformada. Essa ideia é explorada na próxima seção, que analisa o efeito de um filtro gaussiano aplicado a uma imagem no espectro de sua transformada.

Filtro Gaussiano

Filtros podem ser aplicados tanto no domínio espacial quanto no domínio das frequências. Isto é, um filtro em um desses domínios pode ser convertido para um filtro no outro domínio com a aplicação da transformada de Fourier ou sua inversa (Gonzalez e Woods, 2008).

Dessa forma, tem-se a seguinte relação entre a aplicação de um filtro α no domínio

espacial e de um filtro correspondente α' no domínio das frequências, considerando a transformada de Fourier Y em uma imagem I:

$$Y(\alpha(I)) = \alpha'(Y(I)). \tag{2.5}$$

Dessa forma, a transformada de uma imagem filtrada é igual ao filtro correspondente aplicado na transformada dessa imagem.

Para um filtro Gaussiano no domínio espacial, o seu correspondente no domínio das frequências é um filtro passa-baixo. Quanto maior a amplitude do filtro espacial, mais estreito é o filtro no domínio das frequências. Logo, a amplitude das frequências altas será reduzida com sua aplicação.

Essa fato será utilizado no capítulo 4 para estimar a quantidade de desfoque em uma imagem. Para tanto, usa-se a contagem de frequências com amplitudes acima de certo limiar na transformada de *Fourier* de uma imagem.

2.2 Aprendizado de Máquina Supervisionado

Técnicas de aprendizado de máquina supervisionado se baseiam em dados para solucionar problemas computacionais. No aprendizado supervisionado, os dados são N **exemplos rotulados** $(\mathbf{x_i}, y_i)$ com $1 \leq i \leq N$, onde $\mathbf{x_i}$ é um vetor chamado de **vetor de características**, cujos elementos têm o papel de descrever alguma característica do exemplo rotulado ao qual eles se referem, e y_i é o **rótulo** do i-ésimo exemplo. O objetivo de técnicas de aprendizado supervisionado é gerar um modelo que receba um vetor de característica e consiga prever qual o rótulo se associa a este vetor (Burkov, 2019). Para isso, o modelo passa por um processo de treino com os exemplos.

A partir dessa descrição, podemos dividir os problemas de aprendizado de máquina como de classificação ou regressão. Na classificação, procura-se estimar um rótulo y_i que pode assumir valores discretos. Na regressão, faz-se o mesmo com rótulos pertencentes a um conjunto contínuo. Este trabalho utiliza o método de **aprendizado supervisionado com regressão**, que será abordado na próxima seção.

2.3 Modelos Aplicados no Projeto

Existem diferentes algoritmos de regressão para o treinamento supervisionado de um modelo. A escolha entre eles deve ser feita considerando uma série de fatores, incluindo a quantidade de dados disponíveis, sua distribuição, a explicabilidade do algoritmo, e o tempo de treinamento e predição.

Nesta pesquisa foram empregados seis algoritmos de aprendizado de máquina supervisionado: regressão linear, k-vizinhos, árvore de decisão, floresta aleatória, rede neural de regressão, e um modelo baseado em transferência de conhecimento baseado um modelo pré-treinado de rede neural de convolução.

Abaixo, segue uma descrição breve do funcionamento dos modelos utilizados no projeto.

Considere um conjunto de N exemplos rotulados $(\mathbf{x_i}, y_i)$ com $1 \leq i \leq N$, cada um com um vetor de características de tamanho M.

2.3.1 Regressão Linear

A regressão linear é um algoritmo de aprendizado supervisionado que modela o alvo como uma combinação linear das características (Burkov, 2019). Quer-se encontrar um vetor $\mathbf{w}$, pertencente a $\mathbb{R}^M$, e um número real b, que minimizem a função:

$$\sum_{i=1}^{N}(\mathbf{x_i} * \mathbf{w} + b - y_i)^2$$

Com $(\mathbf{x_i} * \mathbf{w} + b - y_i)^2$ sendo o erro quadrático do í-ésimo exemplo, que é a função de perda do modelo. O erro quadrático é utilizado por desconsiderar o sinal resultante do erro e por ser derivável. A fórmula acima representa a perda média do erro quadrático. Esta função, chamada de função de custo, é minimizada pelo algoritmo de treinamento.

2.3.2 Modelo K-Vizinhos

O método dos k-vizinhos pode ser utilizado tanto no aprendizado supervisionado quanto no aprendizado não-supervisionado (Pedregosa *et al.*, 2011). Nesta pesquisa, foi utilizado o algoritmo de aprendizado supervisionado de regressão.

O modelo devolve, para um vetor de características de entrada $\mathbf{x}$, a média dos valores y_i dos k pontos do conjunto de treinamento que estejam mais próximos desse vetor. Isto é, dado o conjunto de treinamento, o algoritmo encontra os k vetores de características com menor distância em relação ao vetor $\mathbf{x}$ e retorna a média dos valores associados(Burkov, 2019).

Como hiperparâmetro, o modelo recebe o número k de pontos próximos para se efetuar o cálculo da média e a função de distância a ser utilizada na busca por esses pontos.

Durante o treino, é construída uma árvore que permite a busca eficiente dos vizinhos mais próximos. Diferentemente dos nos demais modelos vistos, não é feito um processo de otimização de uma função de custo.

2.3.3 Árvore de Decisão

Árvores de decisão são grafos acíclicos usados para realizar decisões (Burkov, 2019). Cada nó interno da árvore possui um índice de uma das características e um limiar. Uma entrada que seja analisada nesse nó seguirá na subárvore à esquerda se o valor da sua característica indexada for menor que o limiar, e seguirá na subárvore à direita caso contrário. Em cada nó, estão agrupados os exemplos rotulados de treinamento, e a previsão é feita nas folhas. A previsão de uma entrada é a média dos rótulos dos exemplos rotulados agrupados na folha em que a entrada acaba de percorrer a árvore. Dessa forma, uma árvore é uma aproximação constante em trechos do domínio.

A construção da árvore ocorre incrementalmente na etapa de treinamento. Na primeira iteração, todos os dados são agrupados em um único nó, e é computada a média e impureza desse agrupamento. Em seguida, é escolhido o índice de uma característica e um limiar. São, então, criados dois nós, um com os exemplos que possuem o valor da característica indexada menor ou igual ao limiar, e o outro com o restante dos exemplos. As escolhas do índice e do limiar são feitas de forma a minimizar a soma das impurezas dos conjuntos resultantes.

2.3.4 Floresta Aleatória

Florestas aleatórias de regressão são estimadores construídos ao se combinar a previsão de múltiplas árvores de decisão. Na biblioteca *SKLearn* o algoritmo introduz aleatoriedade na construção das árvores para criar um conjunto diverso de regressores. A predição é feita tomando-se a média de cada árvore (Pedregosa *et al.*, 2011).

2.3.5 Rede Neural

Uma rede neural de regressão, tal qual outros modelos vistos, é uma função real no domínio do espaço de características:

$$y = f_{NN}(\mathbf{x}). \tag{2.6}$$

Na rede neural, em particular, essa função assume uma forma composta por funções vetoriais:

$$f_{NN}(\mathbf{x}) = F_k(\dots F_{i+1}(F_i(\mathbf{x})\dots) = F_1 \circ F_2 \cdots \circ F_k(\mathbf{x}). \tag{2.7}$$

Por sua vez, as funções vetoriais são da forma:

$$F_i(\mathbf{x}) = G(W_i * \mathbf{x} + b_i) \tag{2.8}$$

Onde $G(\mathbf{x}) : R^N \implies R^N$ é uma função não linear, W_i é uma matriz quadrada e $\mathbf{b_i}$ é um vetor, ambos de dimensão N. Redes neurais são utilizadas para aproximar funções, e cada composição de uma função $F_i(\mathbf{x})$ corresponde a uma camada da rede. A função não linear $G(\mathbf{x})$ permite que a rede neural seja utilizada para aproximar funções não lineares, pois sem ela teria-se apenas a composição de funções lineares que, por sua vez, seria linear.

O número de camadas e a função $G(\mathbf{x})$ devem ser passadas à priori. Esses hiperparâmetros serão otimizados com processo descrito no capítulo 4. Outras aspectos podem ser modelados, como as conexões entre as camadas ou sistemas de *feedback*, resultando em variantes desse algoritmo. O método descrito brevemente acima, utilizado nesta pesquisa, é o perceptron multi-camadas para regressão, que é totalmente conexo entre camadas sucessivas e não utiliza sistemas de feedback.

2.3.6 Rede Neural de Convolução

Redes neurais de convolução (*convolution neural netwroks*, CNNs) são redes neurais desenvolvidas no estudo de problemas de visão computacional (Burkov, 2019). Elas adap-

tam o conceito de redes neurais para lidar com a estrutura e tamanho típicos de imagens digitais. O principal conceito para o entendimento de seu funcionamento é o de filtro de convolução.

Dessa forma, o tamanho de uma imagem é reduzido após a aplicação do filtro, pois os *pixels* na borda da imagem não são utilizados como centro da janela de convolução. Alternativamente pode se usar zeros para completar a janela nas posições em que não existem *pixels*, em uma técnica chamada de *padding*.

Os parâmetros dos filtros são estimados na etapa de treinamento. À priori, é definida a estrutura da rede, com a quantidade de camadas e de filtros em cada camada, bem como as dimensões e parâmetros desses filtros (como o *padding*, por exemplo).

Em uma CNN, a informação é reduzida a imagens cada vez menores. No final do processo, pode-se ter escalares, vetores, ou mesmo um conjunto de imagens de dimensão menor, a depender da estrutura definida para o problema.

2.4 Métricas de Desempenho

Existem várias métricas para a avaliação do desempenho de modelos. Sua utilização depende do problema em questão. Elas variam primeiramente em relação à estrutura do problema, como sendo de regressão, de classificação, ou de ranqueamento. Ao mesmo tempo, variam quanto ao que se quer priorizar no desempenho do modelo, como poder de previsão ou de generalização.

As métricas utilizadas no projeto foram o erro absoluto médio, a raiz do erro quadrático médio e coeficiente de determinação.

As definições a seguir, consideram um vetor $\mathbf{p}$ com as N previsões de um modelo para um conjunto de teste, e um vetor $\mathbf{y}$ de valores observados.

2.4.1 Erro Absoluto Médio

O erro absoluto médio é definido como:

$$E_{mae}(\mathbf{p}) = \frac{1}{N}\sum_{k=1}^{N} |y_k - p_k|. \tag{2.9}$$

2.4.2 Raiz do Erro Absoluto Médio

A raiz do erro quadrático médio é definida como:

$$E_{rmse}(P) = \frac{1}{N}\sum_{i=k}^{N} \sqrt{(y_k - p_k)^2}. \tag{2.10}$$

2.4.3 Coeficiente de Determinação

O coeficiente de determinação R^2 é uma métrica de pontuação. Ao contrário dos erros, aumenta conforme o desempenho do modelo. É calculado com a equação:

$$R^2 = 1 - \frac{\sum_{k=1}^{N}(y_k - p_k)^2}{\sum_{k=1}^{N}(y_k - \bar{Y})^2}. \tag{2.11}$$

Onde $\bar{Y}$ denota a média do vetor $\mathbf{y}$. Pode assumir valores no intervalo $(-\infty, 1]$. O coeficiente de determinação assume valor 1 no caso de *fitting* perfeito dos dados de teste,e assume valor 0 para um modelo com previsão constante igual a média.

2.5 Transferência de Aprendizado

A transferência de aprendizado se baseia na capacidade de redes de *Deep Learning*, tais como *CNNs*, desempenharem múltiplas tarefas de forma que camadas inferiores da rede (mais próximas às entradas) são compartilhas para realização de tarefas diferentes. (Goodfellow *et al.*, 2016). Desse modo, foi utilizado um modelo pré-treinado para a tarefa de classificação de conteúdo de imagens, treinado utilizando uma rede neural de convolução (Simonyan e Zisserman, 2014). Os detalhes desse processo, como o modelo escolhido e a modificação na estrutura de camadas, estão descritos no capítulo 4.

2.6 Análise de Componentes Principais

Análise de componentes principais (PCA) é um método de redução de dimensionalidade que, tem por objetivo reduzir o ruído em dados de muitas dimensões Burkov, 2019. É um método que utiliza princípios de álgebra linear para definir um novo conjunto de n bases para um espaço de dados de N dimensões, com o número $n \leq N$ de bases passado como hiperparâmetro do algoritmo.

O método PCA constrói o conjunto de bases iterativamente, com a primeira base definida como um vetor unitário na direção de maior variabilidade dos dados. A segunda base é restrita a ser ortogonal à primeira e tem a sua direção na maior variabilidade dos dados permitida pela restrição imposta. O processo se repete iterativamente com as bases seguintes, até o completar um conjunto de n bases ortonormais.

Capítulo 3

Dados

Esta pesquisa necessita de acesso à um banco de dados de imagens com avaliações de usuários. Foram levantadas opções como sites especializados em fotografia, bancos de dados públicos para pesquisa, além de uma opção de API de site de comércio eletrônico.

3.1 Bancos de Dados Pesquisados

Os bancos de dados pesquisados são descritos abaixo. Os sites de fotografia foram agrupados por possuírem características similares.

1. **Sites de fotografia**: foram pesquisados os sites photo.net e DPChallenge, utilizados em pesquisas de avaliações de qualidade de imagens (Datta, Li *et al.*, 2008). As fotos são carregadas e avaliadas por usuários do site.

2. **Photo.net Dataset e o DPChallenge.com Dataset**: esses bancos contém imagens selecionadas de sites citados acima para o uso em avaliação de qualidade de imagens (Ke *et al.*, 2006). São apresentadas as avaliações de usuários utilizadas na pesquisa, mas as imagens, contudo, não são diretamente fornecidas. A pesquisa foi feita no ano de 2008, e parte das imagens não estão mais presentes nos sites originais.

3. **KonIQ-10k**: contém 10047 imagens obtidas do YFCC100M e avaliações de usuários feitas por *crowdsourcing* (Hosu *et al.*, 2019; Thomee *et al.*, 2015). Em particular, esse banco utiliza métodos para garantir a variabilidade estética nas imagens amostradas, além de fornecer o desempenho de diversos modelos da literatura aplicados no banco proposto, incluindo um modelo próprio, desenvolvido na pesquisa.

4. **Waterloo IAA**: contém 4744 selecionadas dos site photo.net. As imagens foram selecionadas por serem consideradas de alta qualidade e sem distorções como compressão ou desfoque. A metodologia para as avaliações das imagens é baseada em distorções aplicadas nas imagens selecionadas, e não conta com avaliações de usuários (Ma *et al.*, 2017).

5. *API de e-commerce*: foi considerado o uso da API da Etsy para o carregamento de imagens de itens de venda, junto com métricas, como cliques, visualizações e vendas, que podem ser utilizadas como alvo da regressão proposta (Zakrewsky *et al.*, 2016).

3.1.1 Seleção do Banco

A escolha do banco levou em conta o número de imagens, sua variedade estética, o tipo de licença, e a disponibilidade das imagens no banco de dados. O número de imagens precisa ser grande o suficiente para produzir resultados de qualidade na etapa de modelagem. Em contra partida, precisa ser limitado adequadamente para o hardware disponível. Além disso, as licenças das imagens devem ser *Creatives Commons*.

Nesse sentido, os sites citados e a API da Etsy, todos com milhões de imagens disponíveis, necessitam de uma seleção prévia para um processamento e armazenamento compatíveis, além de ser necessário conferir as licenças de cada imagem. Para os sites, cada imagem deve ser baixada em separado.

Os bancos *Photo.net Dataset* e o *DPChallenge.com Dataset* fazem a seleção prévia, mas não fornecem as imagens diretamente, o que leva ao problemas já citados.

Os bancos Waterloo IAA e KonIQ-10k, além de realizar a seleção prévia, disponibilizam as imagens diretamente em arquivos ZIP. No entanto, a medidas de avaliação do banco Waterloo difere da pretendida nesta pesquisa, pois considera apenas distorções aplicadas artificialmente nas imagens.

Entre os bancos pesquisados, o KonIQ-10k foi o único que apresentou uma metodologia baseada em parâmetros estéticos para selecionar imagens suficientemente variadas para o desenvolvimento de sistemas de avaliação de qualidade de imagens. Pela referência do desempenho de modelos treinados neste banco de dados, considera-se que o número de imagens é adequado para modelagem (Hosu *et al.*, 2019). Todo conjunto de imagens é disponibilizado em formato *.zip* e todas as imagens possuem licença *creative commons* livres para modificação e redistribuição.

Como o banco KonIQ-10k atendeu a a todos os requisitos, ele foi selecionado para a utilização nesta pesquisa.

Banco de Dados	Número de Imagens	Variação Estética	Avaliações de Usuários	Licença Creative Commons	Aquisição
photo.net	1.000.000+	Sim/Selecionar	Sim	Sim/Selecionar	Baixar cada Imagem
dcchallenge.com	1.000.000+	Sim/Selecionar	Sim	Sim/Selecionar	Baixar cada Imagem
Photo.net Database	20278*	Sim	Sim	Sim	Baixar cada Imagem
DCChallenge Database	16509*	Sim	Sim	Sim	Baixar cada Imagem
KonIQ-10k	10047	Sim**	Sim	Sim***	Arquivo .zip
Waterloo	4744	Não	Não	Sim	Arquivo .zip
Etsy	1.000.000+	Sim/Selecionar	Sim	Sim/Selecionar	Baixar imagens em lotes

Figura 3.1: ***Comparações dos requisitos apresentados pelos bancos de dados pesquisados****: número de imagens, variação estética (se há disponibilidade e se é necessário pré-seleção), avaliações de usuários (disponíveis ou não), licença creative commons (disponibilidade e necessidade de seleção) e método de aquisição das imagens (individual, em grupo ou em formato comprimido). * Há imagens indisponíveis. ** O banco foi desenvolvido com o foco em variedade estética para avaliação de qualidade de imagens. ** As licenças permitem livre distribuição e modificação.*

3.2 KonIQ-10k

Na Tabela 3.1 são apresentadas estatísticas para descrever as avaliações de usuários do banco KonIQ-10k, com o histograma na Figura 3.2. Na Figura 3.3 são apresentadas algumas

imagens do banco de dados.

número de avaliações	10073
média	58.72
desvio padrão	15.43
mínimo	3.91
25%	49.24
50%	62.35
75%	70.71500
máximo	88.38

Tabela 3.1: ***Estatísticas das avaliações de usuários no banco KonIQ-10k***

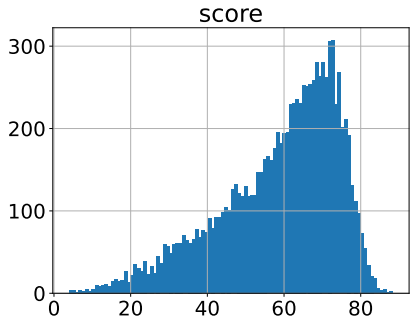


Figura 3.2: ***Histograma das avaliações de usuários no banco KonIQ-10k***

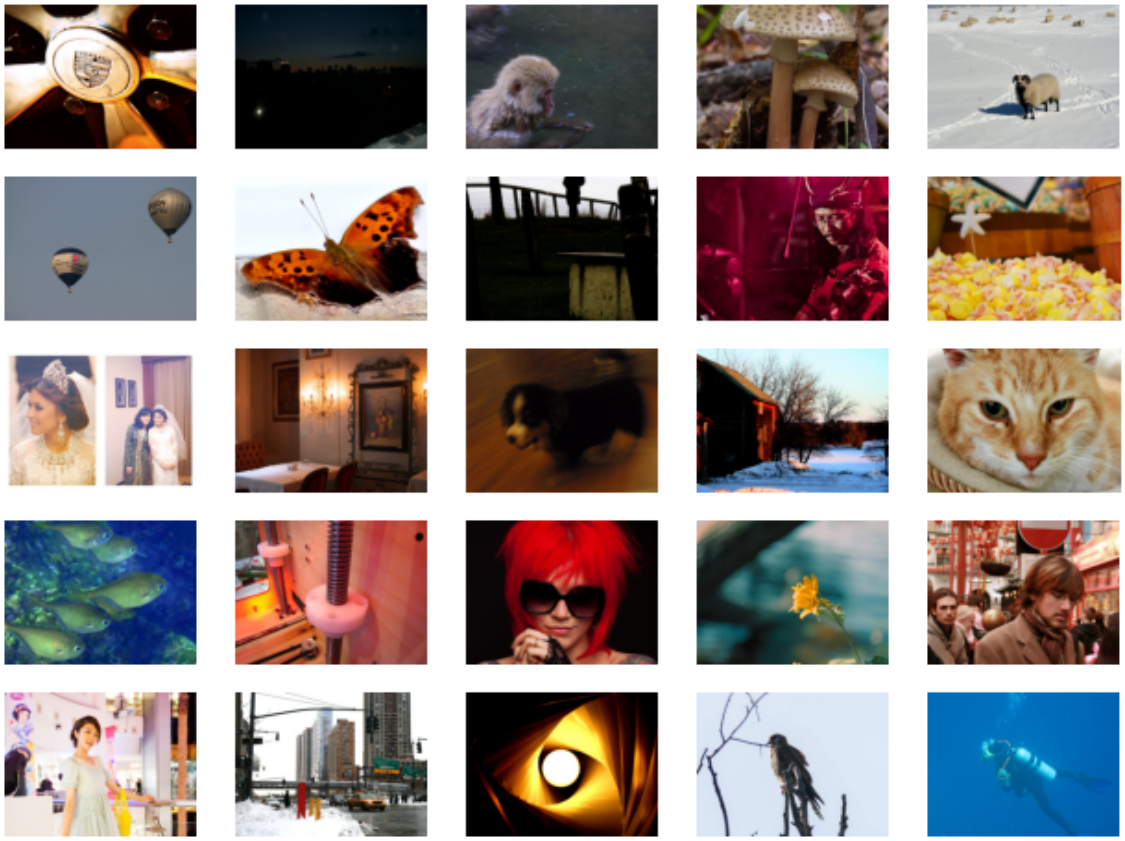


Figura 3.3: ***Amostra de imagens do banco KonIQ-10k****: as imagens retratam diferentes temas, com técnicas variadas. Há, por exemplo imagens tanto dentro como fora de foco (como a imagem desfocada ao centro ou a borboleta em foco na fileira acima), assim como imagens com alta exposição (como a primeira foto da terceira fileira) e com baixa exposição (como a segunda foto da primeira fileira)*

Capítulo 4

Extração de Características

A extração de características foi feita a partir de métodos que buscam metrificar aspectos estéticos das imagens. Este capítulo explica o processo de seleção dos algoritmos de extração de características, descreve os métodos selecionados, e relata a fase de implementação, apresentando os resultados obtidos a partir dos dados usados na pesquisa.

O código foi escrito em *Jupyter Notebooks* e o processamento foi feito em *batch*, com o uso da linha de comando. A análise foi feita com as bibliotecas *Pandas*, *Seaborn* e *Matplotlib*.

4.1 Critério de Seleção

Os algoritmos utilizados para a extração de características foram selecionados a partir de métodos publicados na literatura (Zakrewsky *et al.*, 2016; Ke *et al.*, 2006). A escolha de quais algoritmos implementar nesta pesquisa foi feita para simplificar a execução e análise. Desse modo, decidiu-se não utilizar métodos que resultassem em características de dimensionalidade maior que 1. Além disso, métodos que utilizam a transformada *wavelets* foram evitados, por falta de familiaridade com a técnica.

4.2 Descrição dos Métodos

A descrição de cada método de extração de características é apresentada a seguir em três partes: um breve resumo do aspecto que se pretende medir, um relato do algoritmo que realiza essa função, e um comentário sobre o que seria esperado observar em fotos consideradas boas ou ruins. Esse comportamento esperado não é necessariamente observado pelo modelo, mas é apresentado na característica extraída em avaliar as imagens quanto à qualidade.

O conjunto de técnicas de extração utilizadas parte da premissa de que há um objeto a ser retratado, e que as qualidades estéticas da imagem final devem realçar o objeto retratado.

4.2.1 Luminosidade

Foram utilizadas três características: a luminosidade média da imagem, o contraste máximo, e a largura do histograma da escala de cinzas.

A luminosidade média é o valor médio do canal de luminosidade no espaço *lab*. Para boas fotos, espera-se que haja uma faixa de valores ótimas, que permita uma boa visualização dos objetos da imagem, com áreas áreas claras e escuras.

O contraste máximo é a razão entre o maior e omenor valor de luminosidade da imagem. É esperado um valor alto em boas fotos, pois recomenda-se ajustar a configuração da câmera (abertura, exposição) para que a cena ocupe todo o espectro do histograma.

A largura do histograma é a largura de 95% do centro de massa do histograma no canal de luminosidade. Idealmente, deve possuir um valor alto, para que se tire maior proveito da cena fotografada.

4.2.2 Simplicidade de Cores

Esta característica mede a quantidade de cores fortemente presentes na foto.

Para realizar este processo, a imagem é transformada para o espaço de cor *Hue-Luminosity-Saturation*, com a analise do canal *Hue*. Seus valores são agregados em 20 intervalos, e, em seguida, é calculada a porcentagem desses intervalos que possuem um quantidade de *pixels* maior do que um certo limiar.

Espera-se que um valor baixo indique imagens com melhores avaliações, pois uma paleta de cores reduzida pode ajudar a destacar os objetos retratados.

4.2.3 Simplicidade de Desfoque

Considera-se que boas fotos contém um objeto em foco e um plano de fundo desfocado (para destacar o objeto) O algoritmo realiza uma transformada de Fourier bi-dimensional em cada canal da imagem. Em seguida, calcula a proporção das frequências resultantes totais cujo módulo é maior que uma constante α, obtida experimentalmente. Esse método considera um desfoque gaussiano, que age na imagem na base de Fourier atenuando as frequências mais altas, como explicado no Capítulo 2.

4.2.4 Simplicidade de Saliências

É esperado que uma imagem possua regiões de maior e menor contraste nas bordas entre os *pixels*. Considera-se que a maior densidade de informação (e, portanto, a área com maior variação de contraste) está na área do objeto retratado, que ocupa o centro da imagem.

O algoritmo começa com a aplicação do filtro laplaciano, visto no Capítulo 2 para gerar uma imagem das intensidades das saliências entre os *pixels* de cada canal de cor. Em seguida, as três imagens são combinadas, com o uso da média. A característica resultante é o centro de massa do histograma dessa nova imagem.

4.3 VGG16

Além das características descritas anteriormente, foi utilizado um método de extração baseado em *Deep Learning*. Foi utilizado o modelo *VGG16*, selecionado pela facilidade de implementação (ele está incorporado na biblioteca Tensorflow) e pela disponibilidade de recursos *online* para guiar o processo.

Foram removidas as duas últimas camadas da rede, resultando em 4096 caraterísticas para cada imagem.

4.4 Resultados

Foi realizada uma análise preliminar das características, de forma independente da modelagem, com o intuito de entender a distribuição das características nos dados e sua correlação entre si.

Foram analisadas as seguintes estatísticas de posição das distribuições de cada característica: média, variância, e quartis. Além disso, foi criada uma visualização do *box-plot* e histograma de cada uma. As características também foram analisadas par a par e em conjunto com a nota de avaliação de cada imagem, com o uso de distribuições conjuntas.

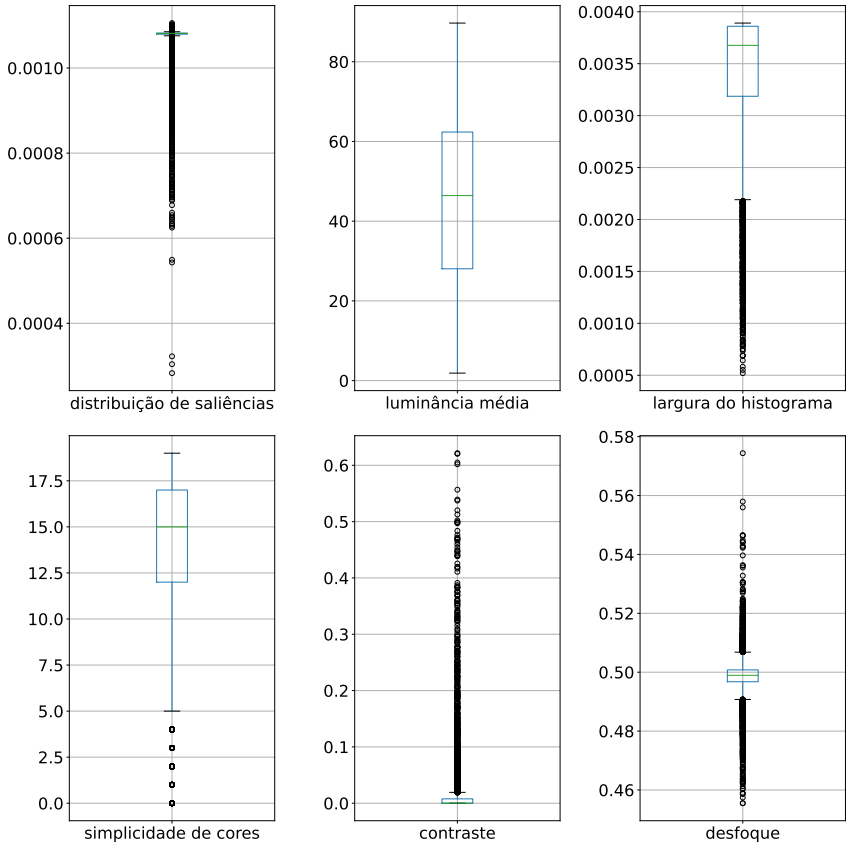


Figura 4.1: ***Box-plots das características***

4.4.1 Distribuições Conjuntas

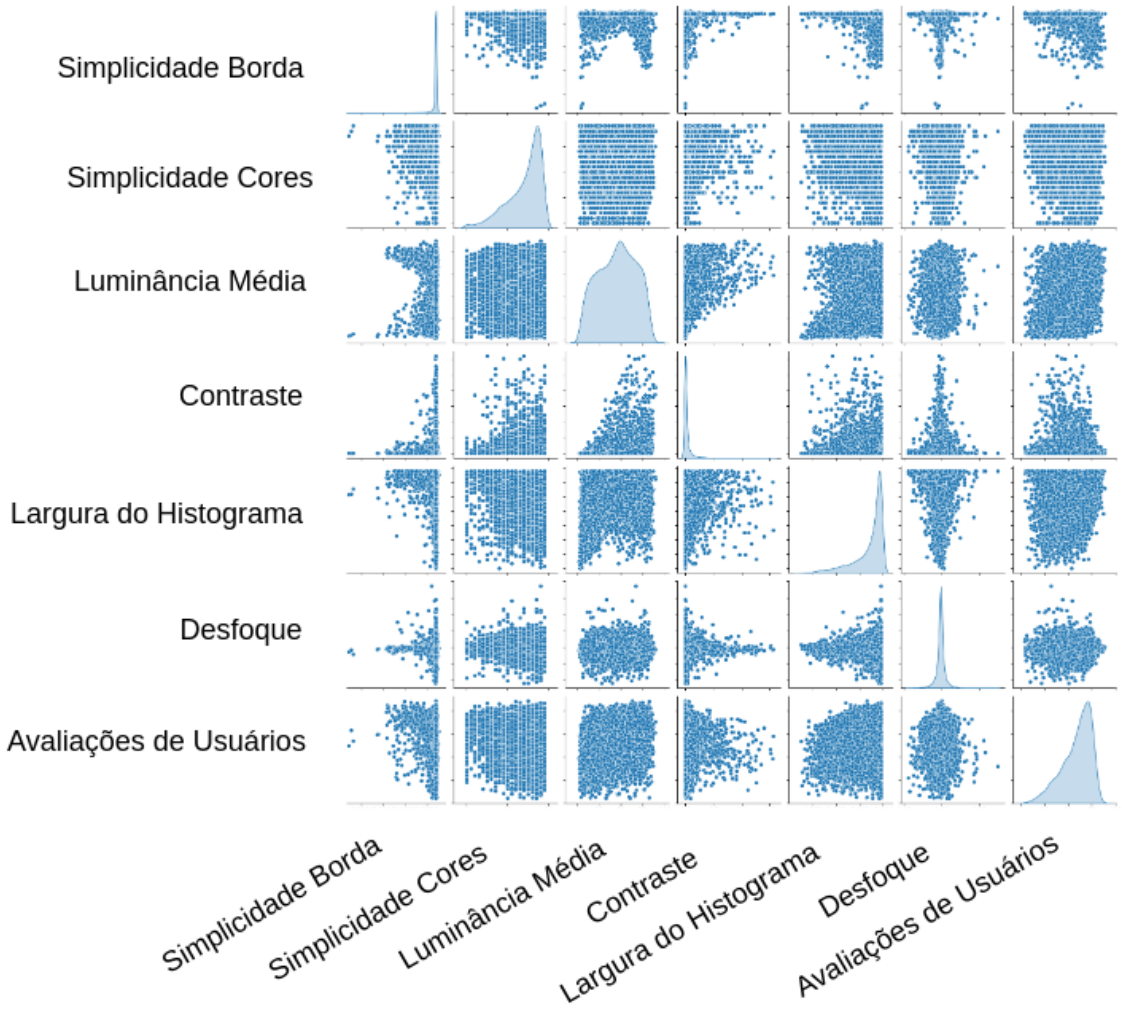


Figura 4.2: ***Distribuições Conjuntas e Histogramas da Extração****: cada linha da matriz de imagens está a distribuição conjunta de uma das características com as demais e com a avaliação dos usuários, na última imagem da linha. As imagens localizadas na diagonal da matriz são histogramas da distribuição de cada características. Na última observa-se a distribuição conjunta entre a avaliação de usuários e cada característica.*

Na Figura 4.2, os histogramas das características extraídas mostram que nenhuma delas possui uma correlação evidente em relação à avaliação dos usuários. Além disso, três características (simplicidade de bordas, contraste e desfoque) apresentam uma distribuição concentrada dos seus valores.

A característica que mais parece se correlacionar com o alvo são é a luminosidade média, embora essa correlação seja sutil e apareça como uma leve inclinação positiva nos valores.

Como esperado, imagens com menor simplicidade de saliências tiveram melhores avaliações, apesar do ruído visto na distribuição conjunta (que pode ser vista na coluna 1, linha 7 da Figura 4.2). De forma similar, a simplicidade de cores e luminosidade média tam-

bém apresentaram o comportamento esperado. Nas outras três características (contraste, largura do histograma e desfoque) não se observou uma relação direta com as avaliações dos usuários.

Essa análise indica que a modelagem deve priorizar técnicas de aprendizado que consigam lidar com dados com ruído, e que as características devem ser analisadas em conjunto para atingir uma modelagem que consiga predizer as avaliações de usuários.

Capítulo 5

Modelagem

Este capítulo aborda a etapa de modelagem, com a descrição da seleção dos modelos testados, das métricas utilizadas, e dos processos de treinamento. Ao final, são apresentados os resultados obtidos. Os modelos treinados com a características extraídas na etapa anterior foram:

1. regressão linear,
2. arvore de decisão,
3. floresta aleatória,
4. k-vizinhos, e
5. rede neural.

A modelagem realizada a partir das características extraídas com o *VGG16* foi feita de forma mais simples, utilizando um modelo de regressão linear após uma transformação de análise das componentes principais, que reduziu o conjunto de 4096 para 300 características.

A descrição de cada modelo pode ser encontrada no capítulo 2.

5.1 Seleção dos Modelos

A Seção 4.4 mostrou que as características extraídas tem uma correlação individual fraca com o alvo (avaliações de usuários), e que há ruído nos dados de forma significativa. Os modelos utilizados foram selecionados por desempenharem bem neste cenário (Pedregosa *et al.*, 2011). A exceção é a regressão linear, que foi utilizada como modelo base para comparações, por sua simplicidade de implementação, e por possuir uma tendência de *underfitting*, ao contrário dos demais modelos.

Foi utilizado apenas a regressão linear para modelar as características extraídas com o *VGG16*. Não foram testados outros modelos devido ao enfoque nas características estéticas selecionadas da literatura e às dificuldades apresentadas pela dimensionalidade mais alta desse conjunto de características.

5.2 Métricas Utilizadas

Foram utilizadas as seguintes métricas de regressão: erro absoluto médio, raiz do erro quadrático médio, e coeficiente de determinação. Esse último foi utilizado como pontuação no processo de otimização dos hiperparâmetros.

Durante o processo de treinamento, além das métricas citadas, foram utilizadas outras ferramentas para acompanhar o desempenho dos modelos, tais como como gráficos das previsões em função dos valores verdade, e gráficos de coordenadas paralelas na otimização dos hiperparâmetros.

5.3 Treinamento

Os algoritmos utilizados dependem de hiperparâmetros passados na inicialização. Inicialmente, foram utilizados os valores padrão da biblioteca Scikit Learn. Com o desenvolvimento da pesquisa, foi feita uma busca no espaço de configuração desses hiperparâmetros, de modo a otimizar os modelos em relação à sua capacidade preditiva.

Para comparar os resultados, foi realizada uma validação cruzada em todo o treinamento. Esse processo permite utilizar todo o conjunto de dados para treinamento e validação, além de fornecer a média e desvio padrão das métricas na validação do modelo (Pedregosa *et al.*, 2011).

O processo foi feito de modo incremental, com a regressão linear como primeiro modelo treinado, para servir de modelo base, seguido da aplicação da validação cruzada para cada modelo. Por fim, feoi feita a otimização dos hiperparâmetros de cada modelo selecionado.

5.4 Otimização de Hiperparâmetros

Para a otimização de hiperparametros foram considerados dois processos: a busca em grade e a busca aleatorizada. Ambos foram testados e comparados, usando a otimização dos hiperparâmetros da árvore de decisão como base. Para a busca em grade, foram testadas 4000 combinações de híperparâmetros. Para a busca aleatorizada, foram geradas 100 combinações. O resultado foi similar entre os dois métodos. Sendo assim, para diminuir a carga de processamento, os demais modelos foram otimizados apenas com o método de busca aleatorizada.

Os intervalos para realização de cada busca foi definido de acordo com boas práticas apontadas pela documentação da biblioteca *Scikit Learn*. Em seguida, foram reajustados a partir da visualização dos resultados.

5.5 Resultados

A seguir, são relatados os melhores parâmetros encontrados para os modelos testados. Em seguida, são apresentados o desempenho dos modelos treinados com os hiperparâme-

tros ótimos segundo as métricas estabelecidas. Por fim, são compilados os gráficos dos valores previstos em relação aos observados, que permitem visualizar a distribuição do erro de predição.

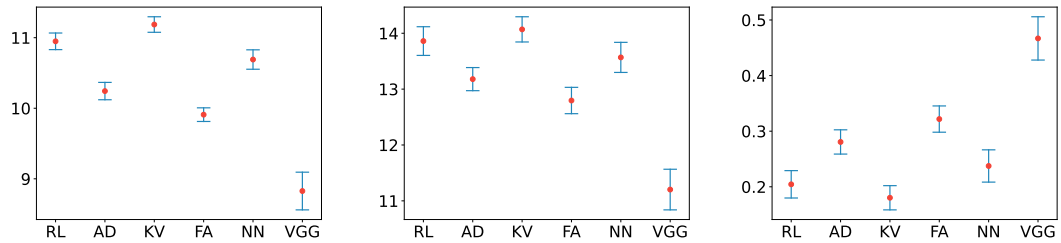
5.5.1 Hiperparâmetros Ótimos Encontrados

No modelo de k-vizinhos, o número de vizinhos encontrado na otimização dos hiperparâmetros foi de 95. Para a rede neural, os melhores hiperparâmetros foram uma estrutura de rede com 20 camadas ocultas de tamanho 15, e α com valor 0.01. A árvore de decisão ótima encontrada foi de tamanho 5, treinada com número máximo de folhas de 35 e número mínimo de amostras para divisão de um nó interno igual a 125. A floresta aleatória teve, para os mesmo híperparâmetros, combinação ótima com valores de 10, 550 e 10, respectivamente, e com o uso de 150 estimadores.

5.5.2 Desempenho

O desempenho dos modelos treinados segundo as métricas utilizadas pode ser observado na Figura 5.1 e na Tabela 5.1. Primeiramente, note-se que o modelo com melhor desempenho foi o modelo treinado no conjunto de características extraídas com o VGG16. Além disso, o modelo de k-vizinhos e o de rede neural tiveram desempenho similar ao de regressão linear, escolhido como modelo base. Por fim, a árvore de decisão e a floresta aleatória tiveram desempenhos melhores que os demais modelos treinados com o mesmo conjunto de características, com destaque para o modelo de floresta aleatória.

Foi utilizado como referência o modelo *KonCept512*, apresentado juntamente com o banco de dados KonIQ-10k, utilizado nesta pesquisa (Hosu *et al.*, 2019). Para a comparação, foram utilizadas as mesmas métricas validadas no KonCept512: coeficiente de correlação de ranqueamento de *Spearman*, e coeficiente de correlação linear de *Pearson*. Ambos são métricas par a par, que levam a ordenação relativa de cada par de imagens em consideração na sua pontuação. Os resultados são apresentados na Tabela 5.2



(a) *Erro Absoluto Médio* **(b)** *Raiz do Erro Quadrático Médio* **(c)** *Coeficiente de Determinação*

Figura 5.1: ***Desempenhos dos modelos treinados****: regressão linear(RL), árvore de decisão (AD), k-vizinhos (KV), floresta aleatória (FA), rede neural (NN) e regressão linear nas características extraídas com VGG16 (VGG). São apresentados intervalos de confiança de* 95%

5.5.3 Previsões

Na Figura 5.2 estão as previsões dos modelos em relação aos valores observados.

Modelo	mae	rmse	r2
Regressão Linear	10.94	13.86	0.204
Árvore de Decisão	10.24	13.17	0.280
K-Vizinhos	11.18	14.06	0.180
Floresta Aleatória	9.90	12.79	0.312
VGG16	8.82	11.20	0.47

Tabela 5.1: ***Desempenho dos Modelos nas Métricas Escolhidas: erro absoluto médio (mae), raíz do erro quadrático médio (rmse) e coeficiente de determinação (r2)***

Modelo	SROCC	PLCC
Regressão Linear	0.410	0.430
Árvore de Decisão	0.505	0.512
K-Vizinhos	0.336	0.403
Floresta Aleatória	0.542	0.550
VGG16	0.678	0.697
KonIQ	0.921	0.937

Tabela 5.2: ***Comparação entre os modelos produzidos nesta pesquisa e modelo de referência*** *: os modelos produzidos são comparados com o modelo KonCept512 desenvolvido pelo grupo que apresentou o banco de dados usado nesta pesquisa.A tabela mostra o coeficiente de correlação de ranqueamento de Spearman)SROCC) e o coeficiente de correlação linear de Pearson (PLCC).*

Vemos que os modelos treinados têm suas previsões enviesadas para a média das avaliações, de forma que os valores mais baixos são sobrestimados e os mais altos subestimados. Isso sugere que o modelo não está capturando corretamente os padrões existentes nos dados. O modelo baseado nas características extraídas com o *VGG16* obteve uma correlação melhor com os valores observados nas suas previsões.

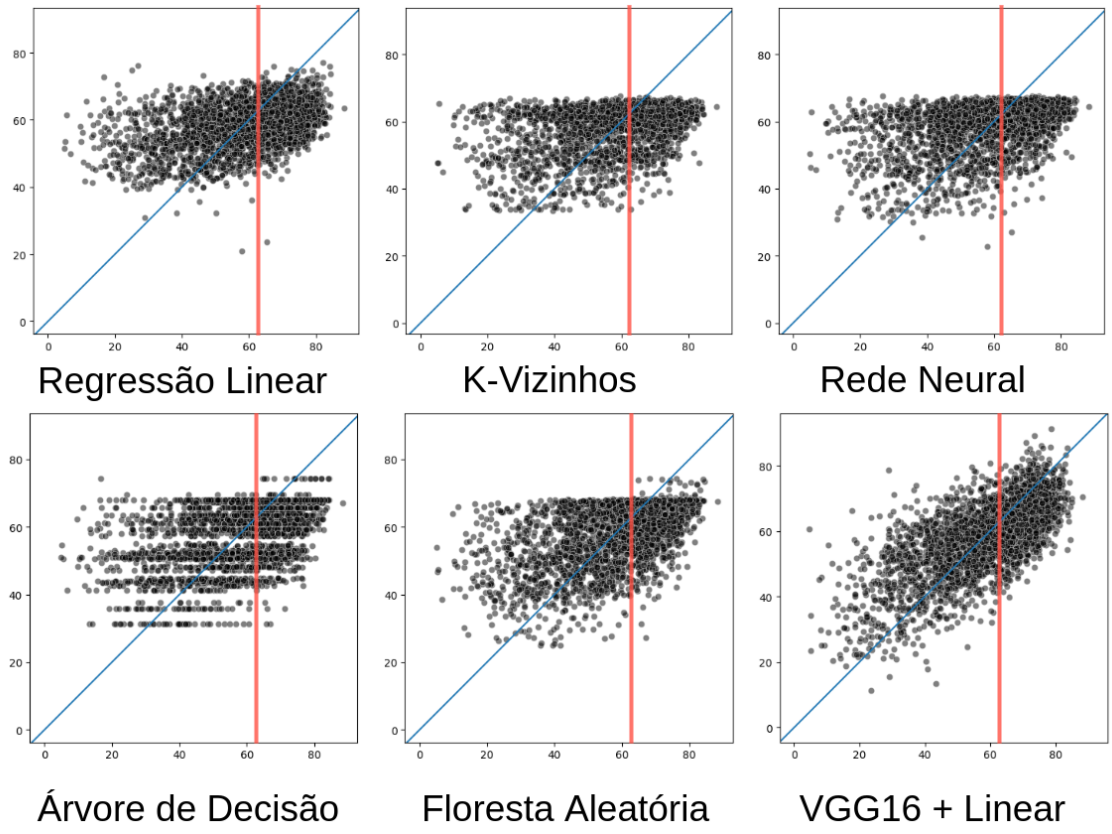


Figura 5.2: ***Previsões vs Valores Observados****: No eixo x das imagens estão os valores observados. No eixo y, os valores das previsões dos modelos. As previsões de um modelo perfeito ficariam n reta $x = y$. O comportamento do modelo treinado com as características extraídas com o VGG16 se aproximou mais dessa reta do que os demais.*

Capítulo 6

Conclusões

6.1 Análise dos Resultados

Conforme apresentado nos capítulos anteriores, o processo de extração e modelagem proposto teve um desempenho baixo em relação aos modelos treinados no mesmo banco , com correlação fraca entre as características extraídas e as avaliações de usuários (Hosu *et al.*, 2019). O modelo nas características extraídas com o VGG16 mostrou o melhor desempenho e melhor correlação com as avaliações de usuários. Além disso, esse método poupou esforço, uma vez que a etapa de extração de caraterísticas é feita automaticamente com a transferência de aprendizado a partir de um modelo pré-treinado. Adicionalmente, é necessário considerar que a modelagem feita com essas caraterísticas foi mais simples que as demais, sem teste e otimização de modelos. Isso sugere que é possível alcançar um desempenho ainda melhor com esse método de extração de características.

Entre os modelos treinados com algoritmos de extração de características da literatura, o desempenho foi consideravelmente mais baixo. O modelo que apresentou melhores resultados foi o de floresta aleatória, de acordo com as métricas utilizadas. Apesar do pior desempenho, essas técnicas de extração de caraterísticas oferecem maior explicabilidade em relação às características obtidas de modelos de *deep learning* pré-treinados, pois cada caraterística tem uma interpretação bem definida.

Apesar dos desempenhos aquém do esperado dos modelos, considera-se que a pesquisa alcançou o objetivo de estudar os conceitos e técnicas de avaliação de imagens com aprendizado de máquina.

6.2 Próximos Passos

Como visto, os modelos treinados com as características extraídas tiveram desempenho pior em relação ao modelo treinado com extração por rede de convolução pré-treinada. Contudo, apenas um pequeno conjunto de características da literatura foi testado nos dados. É possível continuar o projeto a partir do aumento do conjunto de características, com o desempenho do modelo baseado no *VGG16* como referência. Além disso, podem ser

testados modelos baseados em *stacking*, como *AdaBoost* ou *XGBoost*, que mostram bom desempenho em trabalhos publicados.

De modo complementar, uma diretriz para a continuação do projeto é explorar os modelos de rede convolução pré-treinados, com teste de outros modelos a partir das características extraídas ou com outras técnicas de transferência de aprendizado. Esse método é mais promissor e tem sido mais utilizado na área de pesquisa de avaliação de qualidade de imagens.

Outros ponto do projeto sugerido para contribuição é a utilização de métricas mais adequadas ao problema de ranqueamento de imagens, como em modelos por pares ou lista (*ranknet*, *lambdanet* ou similares). Isso pode aproximar o modelo de um eventual uso para sugestão de imagens no ambiente *web*, como proposto pelo artigo que inspirou esta pesquisa.

Referências

[Burkov 2019] A. Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. isbn: 9781999579517. url: https://books.google.com.br/books?id=0jbxwQEACAAJ (citado nas pgs. 6–8, 10).

[Datta, Joshi *et al.* 2006] Ritendra Datta, Dhiraj Joshi, Jia Li e James Z. Wang. "Studying aesthetics in photographic images using a computational approach". Em: *Computer Vision – ECCV 2006*. Ed. por Aleš Leonardis, Horst Bischof e Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pgs. 288–301. isbn: 978-3-540-33837-6 (citado na pg. 1).

[Datta, Li *et al.* 2008] Ritendra Datta, Jia Li e James Wang. "Algorithmic inferencing of aesthetics and emotion in natural images: an exposition". Em: nov. de 2008, pgs. 105–108. doi: 10.1109/ICIP.2008.4711702 (citado nas pgs. 1, 11).

[Gonzalez e Woods 2008] Rafael C. Gonzalez e Richard E. Woods. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. isbn: 9780131687288 013168728X 9780135052679 013505267X. url: http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X (citado nas pgs. 3–5).

[Goodfellow *et al.* 2016] Ian J. Goodfellow, Yoshua Bengio e Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press, 2016 (citado na pg. 10).

[Hosu *et al.* 2019] Vlad Hosu, Hanhe Lin, Tamás Szirányi e Dietmar Saupe. "Koniq-10k: an ecologically valid database for deep learning of blind image quality assessment". Em: *CoRR* abs/1910.06180 (2019). arXiv: 1910.06180. url: http://arxiv.org/abs/1910.06180 (citado nas pgs. 1, 11, 12, 23, 27).

[Ke *et al.* 2006] Yan Ke, Xiaoou Tang e Feng Jing. "The design of high-level features for photo quality assessment". Em: vol. 1. Jul. de 2006, pgs. 419–426. isbn: 0-7695-2597-0. doi: 10.1109/CVPR.2006.303 (citado nas pgs. 5, 11, 15).

[Ma *et al.* 2017] Kede Ma *et al.* "Waterloo Exploration Database: new challenges for image quality assessment models". Em: *IEEE Transactions on Image Processing* 26.2 (fev. de 2017), pgs. 1004–1016 (citado nas pgs. 1, 11).

[Pedregosa *et al.* 2011] F. Pedregosa *et al.* "Scikit-learn: machine learning in Python". Em: *Journal of Machine Learning Research* 12 (2011), pgs. 2825–2830 (citado nas pgs. 7, 8, 21, 22).

[Simonyan e Zisserman 2014] Karen Simonyan e Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. doi: 10.48550/ARXIV.1409.1556. url: https://arxiv.org/abs/1409.1556 (citado na pg. 10).

[Thomee *et al.* 2015] Bart Thomee *et al.* "The new data and new challenges in multimedia research". Em: *CoRR* abs/1503.01817 (2015). arXiv: 1503.01817. url: http://arxiv.org/abs/1503.01817 (citado na pg. 11).

[Wang 2011] Zhou Wang. "Applications of objective image quality assessment methods". Em: 2011 (citado na pg. 1).

[Zakrewsky *et al.* 2016] Stephen Zakrewsky, Kamelia Aryafar e Ali Shokoufandeh. "Item popularity prediction in e-commerce using image quality feature vectors". Em: (mai. de 2016). url: http://arxiv.org/abs/1605.03663 (citado nas pgs. 11, 15).