
Geração de imagens artificiais e quantização
aplicadas a problemas de classificação

Gabriela Salvador Thumé

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Gabriela Salvador Thumé

**Geração de imagens artificiais e quantização aplicadas a
problemas de classificação**

Dissertação apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Mestra em Ciências – Ciências de Computação
e Matemática Computacional. *EXEMPLAR DE
DEFESA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientador: Prof. Dr. Moacir Antonelli Ponti
Coorientador: Prof. Dr. João do Espírito Santo
Batista Neto

USP – São Carlos
Março de 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

T532g Thumé, Gabriela Salvador
Geração de imagens artificiais e quantização
aplicadas a problemas de classificação / Gabriela
Salvador Thumé; orientador Moacir Antonelli Ponti;
coorientador João do Espírito Santo Batista Neto. -
São Carlos - SP, 2016.
91 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática Computacional)
- Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2016.

1. Processamento de imagens. 2. Bases de
dados desbalanceados. 3. Geração de imagens.
4. Quantização. 5. Classificação de imagens. I.
Ponti, Moacir Antonelli, orient. II. Neto, João do
Espírito Santo Batista, coorient. III. Título.

Gabriela Salvador Thumé

Artificial images generation and quantization applied to classification problems

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Moacir Antonelli Ponti
Co-advisor: Prof. Dr. João do Espírito Santo Batista Neto

USP – São Carlos
March 2016

AGRADECIMENTOS

Em primeiro lugar agradeço ao professor Dr. Moacir Ponti, por todo o apoio e orientação. Muito obrigada pela oportunidade, principalmente por possibilitar todo o conhecimento adquirido durante esses dois anos de mestrado.

À minha família. Tivemos poucos dias próximos fisicamente, mas vocês sempre estiveram muito perto. À vocês tenho que agradecer por todo amor e companheirismo. Pai e mãe, serei sempre grata por tudo! Maninha, sei que compartilharemos muitos outros momentos nessa vida e obrigada por me deixar ser sua inspiração! Sogros Vilson e Salete piscadinha, obrigada pelos bolinhos de arroz e melancias geladinhas!

Ao amor da minha vida, Vilson Vieira, pelo apoio incondicional. Você é a minha fortaleza! Sei que posso contar com você para alcançar todos os meus sonhos.

Deixo também meu agradecimento à todos os meus amigos maravilhosos. Aos que compartilham os dias de pós-graduação comigo e aos que sempre estiveram presentes ainda que distantes.

À todos os mestrandos e doutorandos do VICG. A pesquisa pode ser um tanto solitária, mas vocês estiveram sempre presentes! Especialmente à galera do Moacir: Vô, Tiagão e Welinton. Tanto pelas conversas no bandejão sobre o mestrado e ideias para a pesquisa, quanto as noites de jogatina. Vô, lembre-se que a vida é curta! Se não aproveitar, a gelatina já vai ter acabado (vide Tiago). Venham jogar *pirates and buccaneers!* ;-)

Agradeço à Deus, por todas as experiências vividas e por todas as pessoas maravilhosas que colocaste em minha vida.

Ao CNPq, processo nº 130098/2015-9, pelo auxílio financeiro.

“Being a scientist requires having faith in uncertainty, finding pleasure in mystery, and learning to cultivate doubt. There is no surer way to screw up an experiment than to be certain of its outcome.”

(Stuart Firestein)

“No one saves us but ourselves. No one can and no one may. We ourselves must walk the path.”

(Paul Carus)

“Wisest is she who knows she does not know.”

(Jostein Gaarder, Sophie’s World)

“What I cannot create, I do not understand.”

(Richard Feynman)

RESUMO

THUMÉ, G. S.. **Geração de imagens artificiais e quantização aplicadas a problemas de classificação.** 2016. 91 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Cada imagem pode ser representada como uma combinação de diversas características, como por exemplo o histograma de intensidades de cor ou propriedades de textura da imagem. Essas características compõem um vetor multidimensional que representa a imagem. É comum esse vetor ser dado como entrada para um método de classificação de padrões que, após aprender por meio de diversos exemplos, pode gerar um modelo de decisão. Estudos sugerem evidências de que a preparação das imagens — por meio da especificação cuidadosa da aquisição, pré-processamento e segmentação — pode impactar significativamente a classificação. Além da falta de tratamento das imagens antes da extração de características, o desbalanceamento de classes também se apresenta como um obstáculo para que a classificação seja satisfatória. Imagens possuem características que podem ser exploradas para melhorar a descrição dos objetos de interesse e, portanto, sua classificação. Entre as possibilidades de melhorias estão: a redução do número de cores das imagens antes da extração de características ao invés de métodos de quantização no vetor já extraído; e a geração de imagens a partir das originais, de forma a promover o balanceamento de bases de dados cujo número de exemplos de cada classe é desbalanceado. Portanto, a proposta desta dissertação é melhorar a classificação de imagens utilizando métodos de processamento de imagens antes da extração de características. Especificamente, busca analisar a influência do balanceamento de bases de dados e da quantização na classificação. Este estudo analisa ainda a visualização do espaço de características após os métodos de geração artificial de imagens e de interpolação das características extraídas das imagens originais (SMOTE), comparando com o espaço original. A ênfase dessa visualização se dá na observação da importância do rebalanceamento das classes. Os resultados obtidos indicam que a quantização simplifica as imagens antes da extração de características e posterior redução de dimensionalidade, produzindo vetores mais compactos; e que o rebalanceamento de imagens através da geração de imagens artificiais pode melhorar a classificação da base de imagens, em relação à classificação original e ao uso de métodos no espaço de características já extraídas.

Palavras-chave: Processamento de imagens, Bases de dados desbalanceados, Geração de imagens, Quantização, Classificação de imagens.

ABSTRACT

THUMÉ, G. S.. **Geração de imagens artificiais e quantização aplicadas a problemas de classificação.** 2016. 91 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Each image can be represented by a combination of several features like color frequency and texture properties. Those features compose a multidimensional vector, which represents the original image. Commonly this vector is given as an input to a classification method that can learn from examples and build a decision model. The literature suggests that image preparation steps like accute acquisition, preprocessing and segmentation can positively impact such classification. Besides that, class unbalancing is also a barrier to achieve good classification accuracy. Some features and methods can be explored to improve objects' description, thus their classification. Possible suggestions include: reducing colors number before feature extraction instead of applying quantization methods to raw vectors already extracted; and generating synthetic images from original ones, to balance the number of samples in an uneven dataset. We propose to improve image classification using image processing methods before feature extraction. Specifically we want to analyse the influence of both balancing and quantization methods while applied to datasets in a classification routine. This research also analyses the visualization of feature space after the artificial image generation and feature interpolation (SMOTE), against to original space. Such visualization is used because it allows us to know how important is the rebalacing method. The results show that quantization simplifies images by producing compacted vectors before feature extraction and dimensionality reduction; and that using artificial generation to rebalance image datasets can improve classification, when compared to the original one and to applying methods on the already extracted feature vectors.

Key-words: Image processing, Unbalanced datasets, Image generation, Image quantization, Image classification.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas canônicas do reconhecimento de padrões desde a aquisição da imagem até sua posterior classificação. As etapas de pré-processamento e segmentação — apresentadas em destaque — são normalmente pouco exploradas, quando comparadas com as etapas posteriores. O enfoque deste estudo é dar maior atenção à etapa de pré-processamento.	3
Figura 2 – Visualização pixelizada de uma imagem da base Corel-1000. Cada pixel $I(x,y)$ corresponde a uma tripla (r,g,b) de números, com $0 \leq r \leq 255$, $0 \leq g \leq 255$ e $0 \leq b \leq 255$, onde 0 é a intensidade de cor mais escura e 255 a mais clara.	8
Figura 3 – Exemplo de pré-processamento de imagens, caracterizado por receber uma imagem como entrada e fornecer uma imagem de saída. Sobre a imagem RGB de entrada (à esquerda) foram realizadas operações de <i>borramento</i> , realce e de equalização de histograma. A imagem à direita é resultante dessas operações. É possível notar que a imagem resultante apresenta melhor contraste e bordas realçadas.	9
Figura 4 – Exemplo de filtragem Gaussiana como operação de pré-processamento. Resulta no borramento e redução de ruído, a fim de remover detalhes da imagem.	11
Figura 5 – Exemplo de realce de imagens. A imagem original, já em escala de cinza, foi realçada utilizando o método <i>unsharp masking</i> . É possível notar como detalhes de borda foram enfatizados pelo método.	12
Figura 6 – Conversão para a escala de cinza com os métodos utilizados nessa pesquisa. Os métodos resultam em uma imagem com 8 bits (256 cores).	14
Figura 7 – Método SMOTE: interpolação entre dois exemplos vizinhos no espaço de características. A diferença entre o vetor de características de um elemento e do seu vizinho mais próximo é multiplicada por um valor x , sendo $0 \leq x \leq 1$. Esse valor é adicionado ao vetor original, criando um novo elemento entre eles.	19
Figura 8 – O pipeline de reconhecimento de imagens pode envolver uma etapa de conversão de imagens coloridas em imagens em escala de cinza, obtendo uma imagem quantizada que pode ser então processada por métodos de extração de características. O vetor com essas características é então dado como entrada a algum método de classificação.	26

- Figura 9 – Plano no espaço RGB, computado pelo método de conversão para escala de cinza *Intensidade*, quando um dos canais de cor (vermelho, verde ou azul) possui valor 255. O resultado é o mapeamento de características cromáticas bem diferentes em valores de intensidade similares. 27
- Figura 10 – Resultado da aplicação de métodos de quantização. A imagem original (a) resultou em versões de um canal de cor com 232 intensidades únicas para o método (e) MSB e 184 intensidades para os demais métodos. Ao analisar-se as barras de gradiente, assim como as paletas de cores, observa-se que os métodos *Luminância*’ e MSB conseguiram uma melhor discriminação entre cores. 27
- Figura 11 – Exemplo de duas imagens da base de dados *Caltech101-600* com variações no parâmetro de cor utilizando o método MSB. Da esquerda para a direita: imagem original 24-bits e suas versões quantizadas com: 256, 64, 32, 16 e 8 cores. 28
- Figura 12 – Fluxo das operações e os métodos utilizados nos experimentos. Após a aquisição da imagem, ela é convertida para escala de cinza e seus níveis de cor são reduzidos de acordo com um parâmetro da quantização (i.e. número de cores). Dependendo do método, a correção *gamma* é realizada. A imagem quantizada serve então como entrada para um método de extração de características e posteriormente é classificada com *SVM*. Uma das etapas de experimentos prevê também a concatenação de todos os vetores extraídos e a seleção das características com *LPP* antes da classificação. 29
- Figura 13 – Bases de imagens Corel-1000, Caltech101-600 e Produce. Elas são utilizadas nos experimentos de quantização. 30
- Figura 14 – Resultados para as bases Corel-1000 (a), Produce (b) e Caltech101-600 (c), utilizando todos os métodos de quantização. Para cada método de extração de características a acurácia é resultante da sua aplicação utilizando 256, 128, 64, 32, 16 e 8 cores, da esquerda para a direita. 32
- Figura 15 – Resultados de acurácia média da classificação utilizando o método de quantização MSB considerando 256, 128, 64 e 32 cores com o método de extração de características BIC. Os *boxplots* em cinza correspondem às significâncias estatísticas com $p\text{-value} < 0.01$ quando comparado à acurácia de 256 cores. 33
- Figura 16 – Acurácia média da classificação após a utilização do método de quantização *Luminância*’ considerando 256, 128, 64 e 32 cores com o descritor Haralick. Os *boxplots* em cinza correspondem às significâncias estatísticas com $p\text{-value} < 0.01$ quando comparado à acurácia de 256 cores. 33

Figura 17 – Resultados de acurácia para os método MSB (quantização), LPP (redução de dimensionalidade) e BIC (extração de características). A comparação do LPP versus MSB foi realizada com a mesma dimensionalidade. Os boxplots em cinza correspondem às significâncias estatísticas com $p < 0.01$ quando comparado a acurácia de 256 cores.	34
Figura 18 – Comparação da acurácia alcançada com diferentes métodos de quantização: <i>Gleam</i> , <i>Intensidade</i> ', <i>Luminância</i> ' e MSB. Inicialmente as imagens foram convertidas para escala de cinza com esses quatro métodos e foram dadas como entrada para todos os métodos de extração. O vetor de características resultante com $D = 2310$ sofreu então redução da dimensionalidade com o método LPP para $d = 1160, 582, 294$ e 150	35
Figura 19 – Comparação da acurácia com o uso da projeção LPP e o método MSB para quantização das imagens com o objetivo de redução de dimensionalidade.	35
Figura 20 – Resultados para a projeção do LPP sobre o espaço de características produzido pelo método de quantização MSB utilizando 256 ($d = 2310$) e 64 cores ($d = 582$)	36
Figura 21 – Geração artificial da classe minoritária para rebalancear a base de imagens. Para cada imagem necessária para igualar o número de imagens da base, $1 \leq n \leq 16$ imagens originais são dadas como entrada para uma operação de geração artificial. A nova imagem é utilizada como treinamento da base.	38
Figura 22 – Geração artificial utilizando <i>borramento</i> com filtro bilateral. A imagem (b) possui detalhes borrados, porém preservando as bordas.	40
Figura 23 – Geração artificial utilizando <i>unsharp masking</i> . A imagem resultante (b) apresenta saliência nas transições de intensidade.	41
Figura 24 – Geração artificial utilizando <i>adição de ruído</i> de Poisson. Regiões claras de (b) apresentam mais ruído que as regiões escuras.	42
Figura 25 – Geração artificial utilizando o método SMOTE no espaço visual. É possível notar a sobreposição de uma “sombra” da Figura (b) em (a).	44
Figura 26 – Geração artificial utilizando uma <i>mistura ponderada</i> de duas imagens. A imagem (c) representa a mistura de (a) e (b).	44
Figura 27 – Geração artificial utilizando uma <i>mistura limiarizada</i> de duas imagens. A imagem resultante (c) é uma composição do <i>foreground</i> da primeira imagem sobre o <i>background</i> da segunda.	45
Figura 28 – Geração artificial utilizando a <i>mistura saliente</i> de duas imagens. A imagem resultante (c) apresenta a região saliente de (b) sobreposta em (a).	46
Figura 29 – Geração artificial utilizando uma <i>composição</i> de imagens. Várias imagens, dispostas em um mosaico, formam a imagem resultante. Cada célula do mosaico sofre uma operação, sorteada no momento da geração da imagem.	48

Figura 30 – Fluxo de operações para obtenção dos resultados do rebalanceamento de classes. O mesmo protocolo de conversão para escala de cinza, extração de características e classificação foi seguido para três sub-experimentos: base desbalanceada; base rebalanceada com interpolação dos vetores de características (método SMOTE); e base rebalanceada com a geração artificial de imagens.	50
Figura 31 – Ilustração de como os experimentos de geração de imagens artificiais foram realizados. Primeiramente as imagens são separadas de forma aleatória em $k = 5$ folds em cada classe. Em seguida, as duas classes compõem 40 configurações, consistindo em todas as possibilidades de: um fold para teste e os outros como treino para a classe que permanecerá balanceada; e um de teste e apenas um de treino para a classe que os métodos de processamento irão rebalancear. Tal validação é repetida para todas as classes, ou seja, cada classe tem a possibilidade de ser a minoritária.	51
Figura 32 – Classes <i>Horse</i> e <i>Elephant</i> utilizadas no experimento. São duas classes bem discriminadas com 100 imagens cada, originalmente da base de imagens Corel-1000.	52
Figura 33 – Exemplo da geração artificial de imagens com o método de <i>mistura</i> para as classes <i>Elephant</i> e <i>Horse</i> da base Corel-1000. A imagem resultante (c) é composta pela mistura de (a) e (b).	53
Figura 34 – À esquerda a projeção dos dois componentes principais obtidos com a aplicação de PCA nas classes <i>Elephant</i> — em azul — e <i>Horse</i> — em verde. À direita, as mesmas classes após a remoção de 50% das imagens de treino da classe <i>Horse</i> . A diferença dos marcadores consiste na definição de imagens para treino e teste não existente nas classes originais.	54
Figura 35 – Comparação dos exemplos de treinamento da geração com SMOTE e no campo visual. Em laranja estão representados os novos exemplos, projetados no plano da base original balanceada.	55
Figura 36 – Resultado do teste da classificação com K-NN com $K = 1$ após o treinamento realizado com as bases rebalanceadas. A cor no interior dos marcadores quadrados representa a classe real dos exemplos e a borda representa a classe predita pelo classificador.	56
Figura 37 – Região de decisão com K-NN ($K = 1$). Pode ser observado que em ambas técnicas a região da classe minoritária apresenta-se melhor representada. Além disso, é possível verificar que o SMOTE ocasionou uma certa invasão do espaço de características da classe majoritária.	57

Figura 38 – Melhores subespaços encontrados após a geração de novos exemplos para o SMOTE, para a geração artificial de imagens, e após a remoção de imagens para a projeção dos dados desbalanceados. Pode-se notar que a geração de imagens artificiais proporciona a criação de um subespaço que melhor discretaiza as classes, quando comparado com SMOTE ou com a base desbalanceada.	58
Figura 39 – Visualização do impacto do método de extração de características na separação entre classes. Possível verificar que o BIC utiliza as intensidades como principal representação de uma imagem.	59
Figura 40 – Conversão em escala de cinza com <i>Gleam</i> e ACC como método de extração de características. Nota-se que o método de geração baseado em Composição 4 obteve maior valor de <i>F1-Score</i> .	60
Figura 41 – A imagem gerada apresenta uma <i>composição</i> de quatro imagens da classe <i>Elephant</i> .	61
Figura 42 – Conversão em escala de cinza com MSB e HOG como método de extração de características. Essa combinação de métodos obteve a maior variância de <i>F1-Score</i> . Nota-se que o método de <i>adição de ruído</i> apresentou-se como o melhor.	61
Figura 43 – Imagem gerada utilizando <i>adição de ruído</i> em imagens da classe <i>Horse</i> . Regiões mais escuras apresentam maior ruído na imagem resultante (b).	62
Figura 44 – Imagens representativas das classes <i>Beach</i> e <i>Mountain</i> da base de imagens Corel-1000. Tais classes apresentam sobreposição de características por ambas apresentarem paisagens naturais. O objetivo desse experimento é verificar o desempenho da geração artificial em um problema de difícil classificação.	63
Figura 45 – Geração artificial utilizando o método de <i>saliência</i> em duas imagens da classe <i>Beach</i> da base de imagens Corel-1000. Detalhes salientes da imagem (b) aparecem sobrepostos à imagem (a) na imagem resultante (c).	64
Figura 46 – Base de imagens Corel-1000. Essa base é composta por fotografias que representam 10 classes: tribos africanas, praia, construções, ônibus, dinossauros, flores, elefantes, cavalos, montanhas e tipos de comidas.	65
Figura 47 – Geração artificial de <i>mistura</i> para imagens da base Corel-1000. A imagem resultante (c) apresenta uma mistura das imagens (a) e (b).	66
Figura 48 – Conjunto de seis classes balanceadas: aviões, bonsais, candelabros, tartarugas, motocicletas e relógios da base de imagens Caltech101-600.	67
Figura 49 – Exemplo de uma geração artificial utilizando o método de <i>mistura limiarizada</i> para a base Caltech101-600.	68
Figura 50 – Base de imagens Produce, composta por imagens de vegetais e frutas tropicais. A base possui classes com fundo similar, porém com variações na iluminação, número de objetos, cor e escala.	69

Figura 51 – Exemplo de geração artificial utilizando a <i>mistura</i> de duas imagens para a base Produce.	70
Figura 52 – Imagens das classes naturalmente desbalanceadas <i>Eiffel Tower</i> , com 1607 imagens, e <i>Rome antica</i> , com apenas 125 imagens.	71
Figura 53 – Exemplo de geração artificial utilizando a <i>mistura</i> de duas imagens para a base Produce.	72
Figura 54 – Imagens representativas das classes <i>Trafalgar Square</i> , <i>Madeleine Church</i> e <i>Pantheon</i> . Elas contêm tanto imagens parecidas (difícil classificação), quanto imagens bem divergentes (fácil diferenciação).	73
Figura 55 – Exemplo de geração artificial realizada com a <i>composição</i> de quatro imagens para a classe <i>Pantheon</i>	74
Figura 56 – Imagens representativas das classes <i>Deer</i> e <i>Ship</i> . São imagens em baixa resolução (32x32 pixels), o que dificulta o seu processamento e posterior clasificação.	75
Figura 57 – Imagens das classes <i>Shark</i> e <i>Fish</i> . Essas classes apresentam fundo similar.	77
Figura 58 – Exemplo de geração artificial utilizando a <i>mistura limiarizada</i> de duas imagens para a classe <i>Fish</i> . Por se tratarem de imagens com <i>foreground</i> facilmente destacado, a combinação de limiares oferece uma mistura interessante dos objetos.	77

LISTA DE ALGORITMOS

Algoritmo 1 – SMOTE: método para rebalancear classes	19
Algoritmo 2 – K-NN: método de classificação supervisionada	22
Algoritmo 3 – Geração artificial: <i>borramento</i> com filtro bilateral	39
Algoritmo 4 – Geração artificial: <i>aguçamento</i>	40
Algoritmo 5 – Geração artificial: <i>adição de ruído</i> de Poisson	42
Algoritmo 6 – Geração artificial: <i>SMOTE visual</i>	43
Algoritmo 7 – Geração artificial: <i>mistura ponderada</i>	44
Algoritmo 8 – Geração artificial: <i>mistura limiarizada</i>	46
Algoritmo 9 – Geração artificial: <i>mistura saliente</i>	47
Algoritmo 10 – Geração artificial: <i>composição</i>	48

LISTA DE TABELAS

Tabela 1 – Resultados de <i>F1-Score</i> para as classes <i>Horse</i> e <i>Elephant</i> , utilizando <i>Gleam</i> como método para conversão em escala de cinza e ACC para extração de características. Nota-se que o método de geração baseado em Composição 4 obteve maior valor de <i>F1-Score</i>	60
Tabela 2 – Resultados de <i>F1-Score</i> para as classes <i>Horse</i> e <i>Elephant</i> , utilizando MSB como método para conversão em escala de cinza e HOG para extração de características. O método de <i>adição de ruído</i> foi aquele que obteve melhor valor de <i>F1-Score</i>	62
Tabela 3 – Resultados de <i>F1-Score</i> para as classes <i>Beach</i> e <i>Mountain</i> , utilizando <i>Luma</i> como método para conversão em escala de cinza e CCV para extração de características. As gerações com aguçamento e saliência obtiveram os melhores resultados.	64
Tabela 4 – Resultados de <i>F1-Score</i> para as 10 classes da Corel-1000, utilizando <i>Gleam</i> como método para conversão em escala de cinza e LBP para extração de características. A geração artificial com o método <i>mistura</i> obteve <i>F1-Score</i> similar ao SMOTE.	66
Tabela 5 – Resultados do experimento com a base Caltech101-600. É possível notar que tanto o SMOTE quanto a geração artificial utilizando todos os métodos obtiveram um <i>F1-Score</i> melhor que a versão desbalanceada.	68
Tabela 6 – Resultados de <i>F1-Scores</i> para a base de imagens Produce. O método <i>mistura</i> apresentou-se como o melhor.	70
Tabela 7 – Resultados de <i>F1-Score</i> para as classes <i>Eiffel Tower</i> e <i>Rome antica</i> , naturalmente desbalanceadas. O método <i>mistura</i> destacou-se como o melhor método para rebalanceamento dessas classes.	72
Tabela 8 – Resultados utilizando os métodos HOG e <i>Gleam</i> para as classes <i>Trafalgar Square</i> , <i>Madeleine Church</i> e <i>Pantheon</i> . O método SMOTE piorou significativamente a classificação, mas a geração artificial ficou muito similar a classificação sem gerar imagem nenhuma.	74
Tabela 9 – Resultados de <i>F1-Score</i> para as classes <i>Deer</i> e <i>Ship</i> , utilizando <i>Gleam</i> como método para conversão em escala de cinza e HOG para extração de características. As gerações de imagens com <i>aguçamento</i> , <i>mistura limiarizada</i> e <i>mistura ponderada</i> obtiveram os melhores resultados.	76

Tabela 10 – Resultados de <i>F1-Score</i> para as classes <i>Shark</i> e <i>Fish</i> . A geração artificial com a mistura limiarizada apresenta-se como o melhor método para rebalanceá-las.	78
Tabela 11 – <i>Ranking</i> dos métodos de rebalanceamento ao acumular os resultados de todos os experimentos. Esse valor é dado pela soma da posição de cada método em relação ao <i>F1-Score</i> , em ordem ascendente.	79
Tabela 12 – Apresenta a média dos <i>F1-Scores</i> para cada método, ordenada pela coluna de todos os experimentos. É possível verificar que, no cenário multiclasse, apesar de ter sido o melhor método em relação à sua posição, o SMOTE apresentou piores resultados mesmo comparando com as bases desbalanceadas.	79

LISTA DE ABREVIATURAS E SIGLAS

- ACC *Auto Color Correlogram*
BIC *Border-Interior Classification*
CCV *Color Coherence Vector*
CNN *Convolutional Neural Network*
EM-ML .. *Expectation-Maximization Maximum Likelihood*
GCH *Global Color Histogram*
HOG *Histogram of Oriented Gradients*
HSD *Honest Significant Difference*
K-NN *K-Nearest Neighbors*
LBP *Local Binary Patterns*
LPP *Locality Preserving Projections*
MSB *Most Significant Bits*
PCA *Principal Component Analysis*
RBM *Restricted Boltzmann Machine*
SLIC *Simple Linear Iterative Clustering*
SMOTE .. *Synthetic Minority Over-sampling Technique*

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Hipóteses	4
1.2	Contribuições	4
1.3	Estrutura do documento	5
2	FUNDAMENTOS	7
2.1	Considerações iniciais	7
2.2	Pré-processamento de imagens	7
2.2.1	<i>Filtragem espacial e convolução</i>	10
2.2.2	<i>Realce de imagens</i>	11
2.2.3	<i>Quantização</i>	13
2.3	Extração de características	14
2.4	Desbalanceamento de classes	17
2.4.1	<i>Sobreamostragem</i>	18
2.4.2	<i>Subamostragem</i>	19
2.5	Redução de dimensionalidade	20
2.5.1	<i>Análise de componentes principais</i>	20
2.5.2	<i>Locality Preserving Projections</i>	21
2.6	Classificador de padrões	22
2.6.1	<i>K-Nearest Neighbors</i>	22
2.6.2	<i>Support Vector Machines</i>	22
2.7	Considerações finais	23
3	QUANTIZAÇÃO DE IMAGENS	25
3.1	Considerações iniciais	25
3.2	Quantização de imagens	25
3.3	Experimentos	28
3.3.1	<i>Base de Imagens</i>	28
3.3.2	<i>Protocolo</i>	29
3.4	Resultados e Discussão	31
3.5	Considerações finais	36
4	GERAÇÃO DE IMAGENS ARTIFICIAIS	37
4.1	Considerações iniciais	37

4.2	Métodos para geração de imagens artificiais	38
4.2.1	<i>Borramento</i>	38
4.2.2	<i>Aguçamento</i>	40
4.2.3	<i>Adição de ruído</i>	41
4.2.4	<i>SMOTE visual</i>	43
4.2.5	<i>Mistura ponderada</i>	44
4.2.6	<i>Mistura limiarizada</i>	45
4.2.7	<i>Mistura saliente</i>	46
4.2.8	<i>Composição</i>	47
4.3	Experimentos	49
4.3.1	<i>Experimento 1: duas classes discriminadas</i>	52
4.3.2	<i>Experimento 2: duas classes sobrepostas</i>	62
4.3.3	<i>Experimento 3: multiclasses</i>	65
4.3.3.1	<i>Base de imagens Corel</i>	65
4.3.3.2	<i>Base de imagens Caltech101-600</i>	67
4.3.3.3	<i>Base de imagens Produce</i>	69
4.3.4	<i>Experimento 4: classes naturalmente desbalanceadas</i>	71
4.3.4.1	<i>Duas classes desbalanceadas: Eiffel Tower e Rome antica</i>	71
4.3.4.2	<i>Multiclasses desbalanceadas: Trafalgar Square, Madeleine Church e Pantheon</i>	73
4.3.5	<i>Experimento 5: classes com muitas imagens</i>	75
4.3.5.1	<i>Classes balanceadas, distintas e com grande número de imagens</i>	75
4.3.5.2	<i>Classes balanceadas, similares e com grande número de imagens</i>	76
4.4	Considerações finais	78
5	CONCLUSÕES	81
5.1	Publicações	82
5.2	Trabalhos Futuros	82
Referências		85



INTRODUÇÃO

A tarefa de classificação de imagens consiste em predizer corretamente uma imagem como pertencente a uma classe previamente determinada. Um exemplo prático é a classificação da imagem de um *oceano* como parte de uma classe denominada *praia*. Uma forma de definir que certa imagem pertence à uma classe é especificar todas as regras que a caracterizam. Porém, para a maioria dos casos isso é impossível. Considere imagens coloridas, com três canais de cores e de tamanho 256×256 pixels onde cada um desses 65536 pixels pode ser representado por 256^3 combinações discretas de cores. Essa complexidade pode ser reduzida ao utilizar métodos de extração de características. Tais métodos visam representar uma imagem com um número significativamente menor de valores vetoriais. Utilizando-se tal representação, pode-se desenvolver métodos computacionais que consigam definir e identificar a qual classe pertence a imagem — sem a necessidade de se codificar todas as regras possíveis — por meio de algoritmos de Aprendizado de Máquina. Esses algoritmos possuem capacidade de generalização, crucial para classificar novos exemplos não contidos na base de imagens originalmente utilizada para o seu treinamento. Assim, “aprendem” a determinar a classe correta para as imagens de entrada. Em uma etapa posterior pode-se validar esse aprendizado, aplicando o algoritmo a novos exemplos não contidos no treinamento.

O reconhecimento de padrões em imagens possui aspectos particulares para cada aplicação. Apesar da grande variedade de extratores de características disponíveis, nem sempre é possível representar as imagens de maneira satisfatória. Isso porque existem conjuntos de características que dificultam a diferenciação entre as classes. Um dos objetivos da engenharia de atributos é encontrar quais são essas características que melhor discriminam as classes e, dessa forma, obter melhores resultados na etapa de reconhecimento. Para lidar com a deficiência da extração dessas características, é comum concentrar o maior esforço dessa tarefa no espaço de características já extraídas, utilizando transformações do espaço ou sistemas de classificação complexos. No entanto, imagens obtidas de diferentes fontes, como imagens naturais, de microscopia, telescopia e tomografia, possuem características que podem ser exploradas além

dos métodos clássicos. Por isso é importante investigar métodos de processamento e preparação de imagens antes da etapa extração, ao invés de lidar com a má representação das imagens. O uso desses métodos pode revelar características latentes, não visíveis nas imagens originais. Tais características podem melhor descrever certas classes, pois melhoram o conjunto de representações de imagens fornecidas à etapa de classificação. Em dois estudos relacionados, é possível identificar a diferença do desempenho em problemas de classificação de imagens após o uso do pré-processamento. Em [Rocha et al. \(2010\)](#), os autores atingem acurácia acima de 98% na classificação de frutas após investigar alterações nos parâmetros de aquisição, realizar o pré-processamento e obter a segmentação. Já em [Kanan e Cottrell \(2012\)](#), os autores indicam que o método utilizado para obter a imagem em escala de cinza pode impactar significativamente a classificação final de diversas bases de imagens.

Considerando que é comum realizar a extração de características a partir da imagem original, sem preocupação com a preparação da imagem, o enfoque desta pesquisa é na etapa de pré-processamento, destacada na Figura 1. Nela, ilustra-se as etapas canônicas do reconhecimento de padrões, desde a aquisição da imagem até sua posterior classificação. As etapas de pré-processamento e segmentação — apresentadas em destaque no *pipeline* — são normalmente pouco exploradas, quando comparadas com as etapas posteriores.

Primeiro, propõe-se a redução da complexidade do problema no início do processo do reconhecimento, ao quantizar as imagens antes da extração de características. Embora a quantização normalmente faça parte do *pipeline*, faltam estudos que descrevam o método de quantização e seus parâmetros. Ao negligenciar essa etapa, perde-se a oportunidade de redução da dimensionalidade do vetor de características e/ou do tempo de execução das etapas posteriores.

O desbalanceamento de classes também se apresenta como um obstáculo para que a classificação de imagens seja satisfatória. Muitos métodos de transformação do espaço de características e de classificação assumem que as classes da base estão平衡adas, o que nem sempre é verdade. Esse problema é caracterizado pela diferença entre o número de exemplos disponíveis para cada classe da base de imagens. Em bases médicas, por exemplo, a quantidade de imagens relacionadas com uma doença rara é menor do que as imagens de pacientes sem a doença. Nessas situações, em que as imagens representam eventos importantes porém menos frequentes, o sistema de classificação pode ter problemas para lidar com a classe minoritária. Normalmente esses sistemas dão preferência à predição da classe majoritária, prejudicando a classificação da minoritária.

Com o objetivo de promover o rebalanceamento de classes, algumas pesquisas sobre os efeitos da sobreamostragem em dados de aprendizado de máquina já foram realizadas ([KUN-CHEVA, 2004; CHAWLA; HALL; BOWYER, 2002](#)). O método mais divulgado na literatura é conhecido como *Synthetic Minority Over-sampling Technique* (SMOTE). Este método propõe a geração de exemplos artificiais a partir dos vetores de características originais das classes minoritárias. Por outro lado, não há registro conhecido de um estudo dessas técnicas em dados

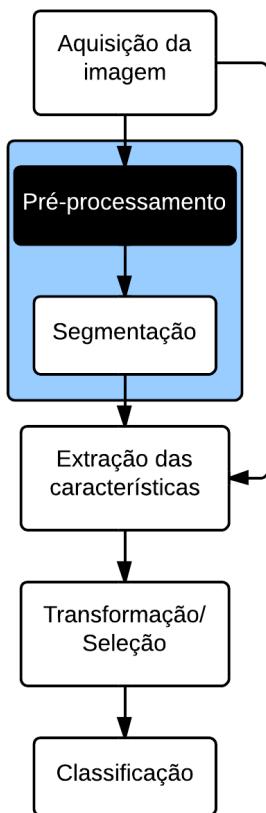


Figura 1 – Etapas canônicas do reconhecimento de padrões desde a aquisição da imagem até sua posterior classificação. As etapas de pré-processamento e segmentação — apresentadas em destaque — são normalmente pouco exploradas, quando comparadas com as etapas posteriores. O enfoque deste estudo é dar maior atenção à etapa de pré-processamento.

Fonte: Elaborado pela autora.

de informação *visual* para o rebalanceamento de classes. A geração de imagens artificiais (ou sintéticas) é uma tarefa comum, mas não para o rebalanceamento de classes. Em [Xie et al. \(2016\)](#), os autores utilizam um modelo gerativo de uma *Convolutional Neural Network* (CNN) para sintetizar imagens realistas. Essas imagens foram geradas, mas não utilizadas como entrada para um algoritmo de classificação ou no treinamento da própria rede. Propõe-se, portanto, a geração de imagens artificiais a partir do processamento das imagens originais, com o objetivo de rebalancear a base de imagens e consequentemente o modelo criado para a classificação. A fim de validar o método proposto, este é então comparado com o SMOTE.

De maneira sumária, **esta pesquisa busca melhorar a classificação de imagens, com foco no rebalanceamento de classes e na quantização de imagens, ambos aplicados antes da extração de características.** Os resultados obtidos, posteriormente apresentados nos Capítulos [3](#) e [4](#), demonstram o potencial do processamento de imagens antes da extração de características.

1.1 Hipóteses

Conforme anteriormente mencionado, muitos aspectos influenciam o desempenho da classificação de coleções de imagens. É comum encontrar bases cuja extração de características é considerada difícil. Ou seja, nas quais algoritmos canônicos de extração não conseguem extrair características que diferenciem bem as classes, prejudicando sua posterior classificação. Nessa situação, normalmente tenta-se lidar com as particularidades das características extraídas através de transformações no espaço de atributos ou mesmo projetando classificadores mais elaborados. Acredita-se que, ao invés disso, é importante investigar métodos de processamento e preparação de imagens antes da extração das características.

A hipótese é que explorar etapas do processamento de imagens melhora a discriminação entre classes de uma coleção de imagens. Dois cenários são especificamente abordados:

- **Utilizar um número reduzido de cores — juntamente com um método de quantização apropriado, antes da extração de características — pode permitir obter vetores de características mais compactos e com maior capacidade de discriminação entre classes.** Melhorando, assim, a classificação e diminuindo a complexidade do sistema. Além disso, analisa-se como diferentes métodos de quantização afetam ambas extração de características e posterior redução de dimensionalidade.
- **A geração de imagens artificiais como preparação para a extração de características pode melhorar a acurácia da classificação, quando comparada à geração de exemplos artificiais no espaço de atributos.** Ou seja, gerar novas imagens artificiais — que serão posteriormente reduzidas a atributos — pode apresentar melhores resultados para a classificação do que o *bootstrap* de atributos artificiais.

1.2 Contribuições

Contribuição geral

Investigar os métodos de pré-processamento de imagens de forma a preparar uma coleção de imagens para a extração de características. Observa-se o efeito da quantização de imagens e do balanceamento do número de instâncias de diferentes classes na classificação.

Contribuições específicas

- Demonstrar que é possível obter vetores de características compactos e efetivos ao extrair características de imagens com níveis reduzidos de intensidade a um custo computacional baixo. Reduzindo assim, o tamanho do vetor de características de métodos de descrição de cor após a quantização, e possibilitando a redução do tempo de processamento para os métodos de descrição de textura;

- Demonstrar que a geração de imagens artificiais utilizando métodos de processamento — como *borramento*, *mistura* e *composição* de imagens — pode contribuir com o balanceamento entre classes (em se tratando de problemas de classes desbalanceadas). Melhorando, dessa forma, o *F1-Score* resultante de algoritmos de classificação, quando comparado à geração de exemplos artificiais no espaço de atributos (SMOTE) e à classificação da base original.

Contribuições em código e reproduzibilidade

O código fonte desenvolvido durante esta pesquisa, no que tange a geração artificial, está disponibilizado em <<https://github.com/GabiThume/msc-src>>. Já a implementação da quantização está disponível em <<http://dx.doi.org/10.5281/zenodo.15932>>. Esses repositórios correspondem à última versão do código para possível reprodução da pesquisa.

1.3 Estrutura do documento

Esta dissertação inicia apresentando os fundamentos necessários para a definição dos métodos utilizados nos experimentos. Esses métodos são então discutidos e os resultados da quantização e da geração artificial são mostrados. O conteúdo está estruturado como segue.

Capítulo 2: são conceituados os principais fundamentos necessários para o desenvolvimento desta pesquisa: pré-processamento de imagens, extração de características, desbalanceamento de classes, redução de dimensionalidade e classificação de imagens.

Capítulo 3: a redução do número de intensidades de cor utilizando métodos de quantização antes da etapa de extração de características é descrita. Além disso, os resultados obtidos com a quantização de imagens são apresentados e discutidos.

Capítulo 4: descreve os métodos de processamento utilizados para a geração artificial de imagens — sobre as imagens originais — com o objetivo de rebalancear classes. Os experimentos são apresentados e seus resultados discutidos.

Capítulo 5: sumariza as contribuições deste trabalho e apresenta os trabalhos futuros.



FUNDAMENTOS

2.1 Considerações iniciais

Neste capítulo são destacados os tópicos mais relevantes para a compreensão da metodologia e análise dos resultados deste trabalho, com foco na preparação das imagens para a extração de características. O problema do desbalanceamento de classes e seu efeito na classificação também são abordados, assim como a extração de características para compreender as propriedades extraídas das imagens. Trabalhos relacionados são utilizados como exemplo, com o objetivo de elucidar tais tópicos.

Inicialmente, a Seção 2.2 apresenta alguns dos principais métodos utilizados para o pré-processamento de imagens (i.e. remoção de ruído, realce de imagens e quantização), relevantes para o desenvolvimento desta pesquisa. Após, os métodos para extração de características são definidos na Seção 2.3. Tendo como proposta melhor compreender o problema do desbalanceamento de classes, a Seção 2.4 exemplifica as operações utilizadas para o rebalanceamento. Nesta seção, além de caracterizar o problema, são apresentadas duas vertentes para solucioná-lo: sobreamostragem e subamostragem. A seção seguinte (2.5) aborda dois métodos de redução de dimensionalidade usados nos experimentos e a Seção 2.6 apresenta os classificadores de padrões utilizados.

2.2 Pré-processamento de imagens

Uma imagem I pode ser definida como uma função $f(x, y)$, onde x e y são as coordenadas espaciais de um determinado ponto e f a intensidade da imagem naquele ponto. Quando ambos x , y e os valores de intensidade são finitos e discretos, trata-se de uma imagem digital. Essa imagem é composta por finitos elementos chamados de pixels que podem ser diretamente acessados através de sua posição x e y . Digitalmente, uma imagem é representada por uma matriz de valores com M linhas e N colunas onde cada elemento representa a sua intensidade. Uma imagem

colorida I do sistema RGB possui três canais de cores representantes das cores vermelho, verde e azul. Cada elemento $I(x,y)$ corresponde a uma tripla (r,g,b) de números, com $0 \leq r \leq 255$, $0 \leq g \leq 255$ e $0 \leq b \leq 255$, onde 0 é a intensidade de cor mais escura e 255 a mais clara. A combinação dessas intensidades resulta na cor do pixel (GONZALEZ; WOODS, 2007). A Figura 2 mostra uma visualização ampliada dos pixels de uma imagem digital.



Figura 2 – Visualização pixelizada de uma imagem da base Corel-1000 (WANG; WIEDERHOLD, 2001). Cada pixel $I(x,y)$ corresponde a uma tripla (r,g,b) de números, com $0 \leq r \leq 255$, $0 \leq g \leq 255$ e $0 \leq b \leq 255$, onde 0 é a intensidade de cor mais escura e 255 a mais clara.

Fonte: Elaborado pela autora.

O processo de aquisição por um sistema de imageamento pode causar diversas imperfeições nas imagens, como pixels ruidosos, brilho inadequado e outras degradações. O pré-processamento de imagens é caracterizado por receber uma imagem de entrada e fornecer uma imagem de saída. Nessa etapa, efeitos indesejáveis podem ser eliminados e certas características realçadas (Figura 3). Considera-se que, um determinado critério utilizado para uma imagem pode não ser o mais eficiente para outra, dependendo da área de aplicação.

Assim, técnicas de pré-processamento tornam os dados mais adequados para posterior análise, ao eliminar ou reduzir problemas como ruídos e imperfeições. Em Ponti (2010), o autor relata que a utilização de métodos de restauração na etapa de pré-processamento da imagem, antes da segmentação, proporcionou qualidade superior em todos os testes, com valores reduzidos de erro e desvio padrão. No referido estudo, métodos de realce causaram perda de informação e por isso não são indicados para uso em imagens obtidas por microscópio, por exemplo. O método indicado para evitar a amplificação de ruído nessas imagens é o algoritmo iterativo Richardson-Lucy (PONTI *et al.*, 2011). Esse método de restauração utiliza um processo iterativo

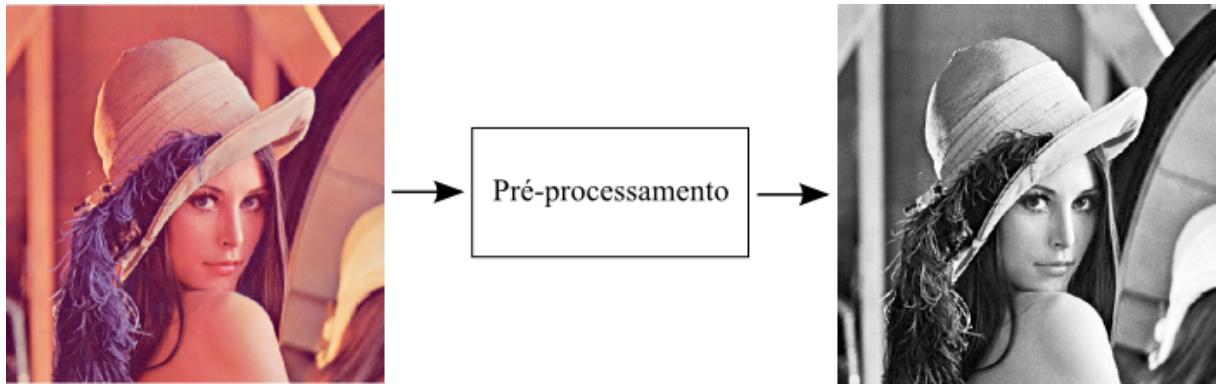


Figura 3 – Exemplo de pré-processamento de imagens, caracterizado por receber uma imagem como entrada e fornecer uma imagem de saída. Sobre a imagem RGB de entrada (à esquerda) foram realizadas operações de *borramento*, realce e de equalização de histograma. A imagem à direita é resultante dessas operações.

É possível notar que a imagem resultante apresenta melhor contraste e bordas realçadas.

Fonte: Elaborado pela autora.

para recuperar uma imagem degradada que foi borrada por algum processo conhecido. Utiliza uma metodologia probabilística, baseada em *Expectation-Maximization Maximum Likelihood* (EM-ML), para encontrar uma imagem que maximize a probabilidade de se visualizar a imagem original sem degradação, considerando um modelo de ruído de Poisson. Algoritmos iterativos como o Richardson-Lucy tem a vantagem de permitir soluções parciais, evitando amplificação de ruído.

Em contrapartida, Ahonen, Hadid e Pietikäinen (2006) propuseram uma representação para imagens faciais baseada em características de textura, sem utilizar pré-processamento. Este aparece somente como sugestão de trabalho futuro, como possível correção de problemas do sistema de captura (i.e. suavização causada pela captura fora de foco). O que implica que, apesar dos bons resultados, a melhoria com a utilização de pré-processamento não foi investigada. Pode-se imaginar, portanto, que o uso de pré-processamento pode melhorar os resultados já obtidos, através do realce de textura e eliminação de imperfeições nas imagens.

Como exemplo do uso de métodos de pré-processamento, considere imagens de algas verdes obtidas por um microscópio de alta resolução. Essas algas são mergulhadas em um líquido que normalmente causa problemas de ruído e pouco contraste. Para a preparação dessas imagens antes da extração de características, Borges *et al.* (2013) cita algumas etapas comuns em processamento de imagens digitais:

1. As imagens — originalmente em RGB — são convertidas para escala de cinza;
2. A dimensão da imagem é reduzida para diminuir o tempo de execução dos passos subsequentes de processamento;
3. O contraste é “ajustado”, para aumentar a diferença das intensidades dos pixels da imagem e corrigir o brilho;

4. A imagem é filtrada, removendo ruídos causados pelo processo de captura;
5. O contorno é realçado, pois a forma é uma das características mais importantes para discriminar algas (e outros objetos);
6. Por fim, o histograma é equalizado.

Xu et al. (2016) propuseram um método de pré-processamento de imagens de faces com o objetivo de gerar imagens sintéticas para posterior reconhecimento de padrões. Inicialmente, o método separa uma imagem em metade-esquerda e metade-direita e espelham a metade-direita. Após, um algoritmo de gradiente descendente iterativo é utilizado para atualizar os vetores que representam cada metade, otimizando-os. Por fim, esses vetores são concatenados para compor uma nova face do mesmo tamanho da imagem original. Os resultados são apresentados como estado da arte para o pré-processamento de imagens de face para a tarefa de reconhecimento.

Esses estudos evidenciam a importância da etapa de pré-processamento de imagens, indicando que o tratamento das imagens antes da extração de características pode melhorar os resultados obtidos.

2.2.1 *Filtragem espacial e convolução*

Um filtro espacial, também conhecido como *kernel*, máscara ou janela, consiste em uma matriz de vizinhanças e uma operação a ser realizada nos pixels de uma imagem. A filtragem cria um novo pixel com as mesmas coordenadas do centro da vizinhança, contendo o valor resultante da filtragem. Dessa forma, a imagem filtrada contém os pixels resultantes da passagem do centro do filtro espacial por todos os pixels da imagem original. O processo de percorrer a imagem com um filtro espacial é chamado de correlação. A convolução, que pode ser definida como o operador $*$ em

$$\text{mapa de características} = \text{imagem de entrada} * \text{filtro},$$

trata-se do mesmo processo, mas com o filtro previamente rotacionado em 180° ([GONZALEZ; WOODS, 2007](#)).

Os métodos de filtragem possuem como objetivo aperfeiçoar certos aspectos da imagem de entrada. Essa filtragem pode ser realizada no domínio da frequência ou no domínio espacial. Um filtro de suavização típico no domínio espacial é o de Gaussiana, que resulta no *borramento* e redução de ruído, a fim de remover detalhes da imagem (Figura 4). Esse filtro utiliza uma função Gaussiana para calcular a transformação a ser realizada em cada pixel. A equação que representa a função Gaussiana em duas dimensões é definida por

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

onde x, y são coordenadas de um determinado pixel da imagem e σ o desvio padrão que determina o raio da distribuição Gaussiana aplicada. Valores altos de variância (σ^2) fazem com que o resultado da função se aproxime da média.



Figura 4 – Exemplo de filtragem Gaussiana como operação de pré-processamento. Resulta no borramento e redução de ruído, a fim de remover detalhes da imagem.

Fonte: Elaborado pela autora.

2.2.2 Realce de imagens

O realce de imagens é o processo de modificar uma imagem para que se torne mais adequada para uma aplicação específica do que na sua forma original. Diferentemente da restauração — que leva em consideração o processo de formação da imagem — é subjetivo, porque depende do sujeito que está analisando a imagem discernir a qualidade desse realce (GONZALEZ; WOODS, 2007).

Na Figura 5 está demonstrado o efeito do algoritmo de *unsharp masking*, utilizando como borramento um filtro de média. Com o objetivo de realçar imagens, os passos deste método são:

1. Borramento da imagem original;
2. Cálculo da diferença entre a imagem suavizada e a original;
3. Adição dessa diferença à imagem original.

Um exemplo clássico de utilização de realce, é para compensar a variação de iluminação em diversas imagens. Em Gross e Brajovic (2003), os autores propuseram um algoritmo para o

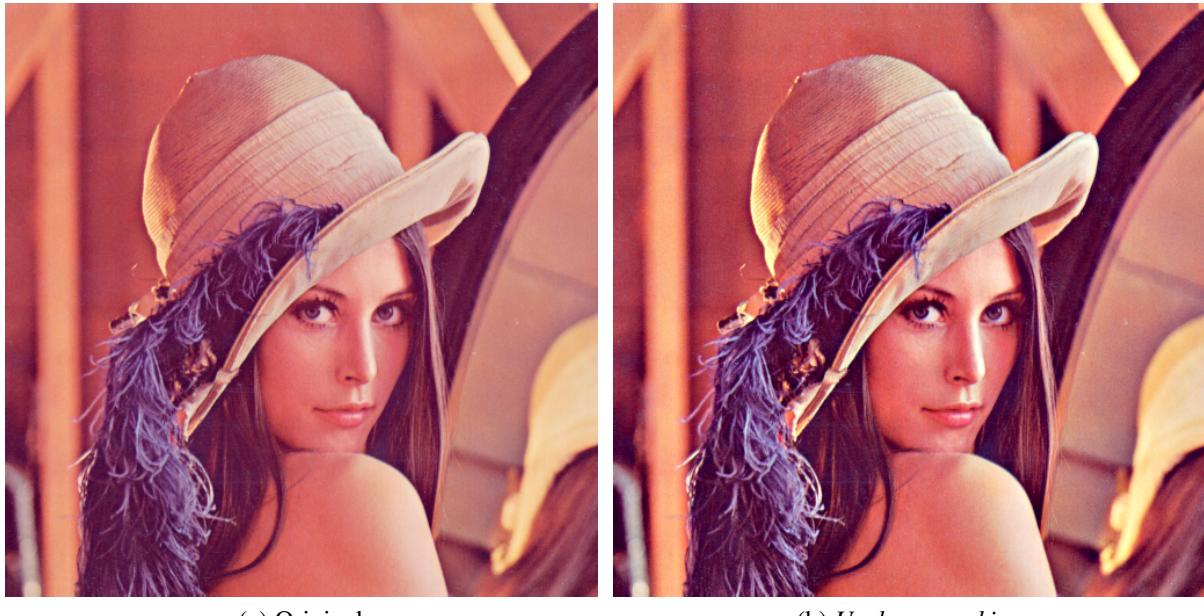


Figura 5 – Exemplo de realce de imagens. A imagem original, já em escala de cinza, foi realçada utilizando o método *unsharp masking*. É possível notar como detalhes de borda foram enfatizados pelo método.

Fonte: Elaborado pela autora.

reconhecimento de faces invariante à iluminação. Eles ressaltam que, desconsiderando a variação da posição, a iluminação é o fator de maior impacto na aparência das faces. A luz varia durante o dia, entre um dia e outro, e entre diferentes ambientes. Isso afeta o conjunto de imagens a ser analisado, que passa a conter imagens com diferentes contrastes, o que pode acentuar ou diminuir certas características faciais.

O contraste é a diferença de intensidade entre os níveis de maior e menor intensidade na imagem. Imagens com baixa resolução podem ser geradas a partir de uma iluminação pobre ou outras complicações durante a captura. Dessa forma, o processo de “esticar” o contraste expande os níveis de intensidade da imagem (GONZALEZ; WOODS, 2007).

É possível aumentar o contraste de uma imagem ao manipular o seu histograma h , que pode ser definido como

$$h(i_k) = n_k,$$

onde k é o índice do pixel e n_k é o número de pixels de intensidade i_k . Ou seja, o histograma é uma representação da frequência de cada intensidade na imagem. Ao observar os histogramas de diferentes imagens, é possível notar que imagens com alto contraste possuem um histograma com níveis próximos a uma distribuição uniforme. Isso permite que certas operações, como a equalização de histograma, obtenham o melhor contraste de uma imagem. Essa operação é caracterizada pela máxima variância do histograma e pode ser definida como

$$s_k = T(i_k) = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad (2.1)$$

onde L é o número de intensidades e MN as dimensões da imagem. A imagem de saída é obtida ao mapear cada pixel de intensidade i_k em um nível s_k , com i entre $[0, L - 1]$, sendo $i = 0$ um pixel preto e $i = L - 1$, branco (GONZALEZ; WOODS, 2007).

2.2.3 Quantização

Muitos métodos de extração de características são preparados para receber imagens em escala de cinza ou em apenas um canal de cor. Se existir a necessidade de utilizar a imagem RGB, as características são extraídas para cada canal de cor separadamente e posteriormente são concatenadas. Isso ocorre porque a complexidade de lidar com um pixel representado em três dimensões é muito maior do que em apenas uma dimensão. Assim, os métodos de quantização visam, de alguma forma, reduzir os canais de cores ($2^3 \times 3 = 24$ bits) em apenas um ($2^3 \times 1 = 8$ bits). Kanan e Cottrell (2012) demonstraram que os métodos para a conversão de uma imagem colorida para escala de cinza influenciam o desempenho do reconhecimento de imagens. Eles também salientam que o método utilizado deveria estar claramente descrito nas publicações da área. Os métodos de conversão para escala de cinza utilizados nesta dissertação foram escolhidos com base em Kanan e Cottrell (2012): *Luma*, *Gleam* e as versões corrigidas por *gamma* de *Intensidade* e *Luminância*: *Intensidade'* e *Luminância'*. A operação *gamma* utilizada $z' = z^{1/2}$ é o padrão. Os métodos de conversão para escala de cinza, utilizados nesta pesquisa, são descritos a seguir.

Intensidade: consiste em computar a média entre os canais RGB da imagem a partir de

$$Q_{\text{Intensidade}} = \frac{1}{3}(R + G + B), \quad (2.2)$$

e então realizar a correção por *gamma*, obtendo assim o método *Intensidade'*.

Gleam: ao corrigir por *gamma* cada canal antes de realizar a combinação linear, tem-se o método

$$Q_{\text{Gleam}} = \frac{1}{3}(R' + G' + B'), \quad (2.3)$$

onde R' , G' e B' são os canais R, G e B corrigidos por *gamma*.

Luminância: computa a soma ponderada dos canais de cor. Esse método foi desenvolvido para levar em conta a percepção visual humana. O olho humano percebe verde melhor que vermelho, e vermelho melhor que azul:

$$Q_{\text{Luminância}} = 0.299R + 0.587G + 0.114B, \quad (2.4)$$

e então realizar a correção por *gamma*, obtendo assim o método *Luminância'*.

Luma: similar ao anterior, utilizado nas televisões de alta definição:

$$Q_{\text{Luma}} = 0.2126R' + 0.7152G' + 0.0722B', \quad (2.5)$$

onde R' , G' e B' são os canais R, G e B corrigidos por *gamma*.

Most Significant Bits (MSB): ao invés de realizar uma combinação linear dos canais de cores, ordena os bits dos canais coloridos em um único canal. Após, calcula quantos bits de cada canal de cor contribuem para a imagem final. Por fim, extrai os bits do código binário dos canais originais (PONTI; ESCOBAR, 2013).

A Figura 6 apresenta a conversão em escala de cinza obtida com o uso desses métodos. A análise específica da utilização de cada método é discutida no Capítulo 3.

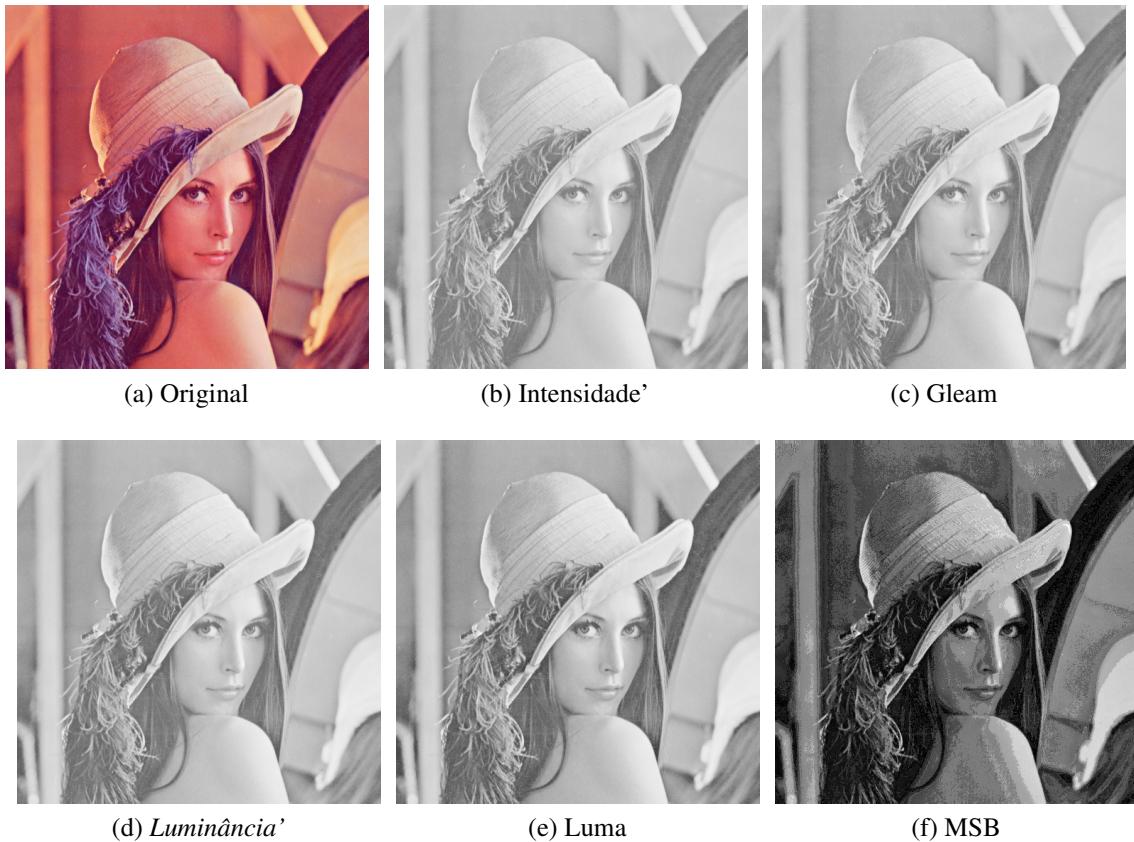


Figura 6 – Conversão para a escala de cinza com os métodos utilizados nessa pesquisa. Os métodos resultam em uma imagem com 8 bits (256 cores).

Fonte: Elaborado pela autora.

2.3 Extração de características

O objetivo da extração de características é descrever as informações visuais relevantes em um vetor de características. Esse vetor pode ser utilizado como entrada para um algoritmo de classificação de padrões. Por exemplo, em aplicações que envolvem a classificação de algas, uma informação muito importante para a discriminação entre classes é a forma (BORGES *et al.*, 2013). Características, como essa, devem salientar as diferenças entre imagens de classes distintas e suavizar possíveis diferenças de imagens da mesma classe. Algumas características, segundo Gonzalez e Woods (2007), são:

Textura: na sua descrição estatística, possui propriedades como: suavidade, aspereza e uniformidade. Um exemplo de medida para descrever a textura é a entropia.

Forma: representa os objetos em termos de suas características externas, como por exemplo, a medida da curvatura.

Cor: considera a distribuição espacial de cores na imagem. O histograma de uma imagem pode descrever essa configuração de forma global.

Exemplos de métodos conhecidos capazes de descrever outras características são: histogramas de orientação de gradiente (WANG; HAN; YAN, 2009), descritores de Fourier, métodos baseados na detecção de SUSAN (SMITH; BRADY, 1997), Harris-Affine (HARRIS; STEPHENS, 1998) e diferença de Gaussianas (LOWE, 2004). Os descritores utilizados no desenvolvimento desta pesquisa para a obtenção dos resultados dos Capítulos 3 e 4 estão abaixo descritos.

Global Color Histogram (GCH): calcula o histograma global dos níveis de intensidade da imagem. É a alternativa mais simples para representar as informações de uma imagem (GONZALEZ; WOODS, 2007). Produz um vetor de N dimensões, sendo N o número de intensidades.

Color Coherence Vector (CCV): captura informações sobre como as cores são organizadas em regiões conectadas, de acordo com um *threshold* (foi utilizado $T = 25$, sugerido pelos autores do método). Classifica os pixels da imagem em coerentes e incoerentes, computa os respectivos histogramas e os concatena (PASS; ZABIH; MILLER, 1996). Dessa forma, o vetor de características produzido possui $2N$ dimensões.

O vetor de coerência de cor armazena o número de pixels coerentes e de incoerentes para cada cor. Pixels coerentes fazem parte de uma região contígua enquanto incoerentes não. Seu funcionamento pode ser resumido nos seguintes passos:

1. Suaviza a imagem com filtro de média 3x3;
2. Discretiza o espaço de cor de forma que a imagem contenha apenas $n \leq 256$ cores distintas;
3. Classifica os pixels entre coerentes e incoerentes: se o tamanho do seu componente conectado for maior ou igual a um *threshold*, é coerente; caso contrário, incoerente;
4. Computa dois histogramas:
 - Histograma de pixels coerentes;
 - Histograma de pixels incoerentes.
5. Concatena os histogramas.

Border-Interior Classification (BIC): computa dois histogramas, um para pixels definidos como borda e outro como interior. Se um pixel possuir a mesma cor que seus vizinhos, é pixel de interior; caso contrário, será pixel de borda. Os histogramas são concatenados, gerando um vetor de $2N$ dimensões (STEHLING; NASCIMENTO; FALCÃO, 2002). Para computar tal vetor, as operações realizadas são:

1. Classifica os pixeis entre borda e interior:
 - *Borda:* se está na borda da imagem ou se ao menos um dos seus quatro vizinhos tem uma cor diferente do que o próprio pixel;
 - *Interior:* se seus quatro vizinhos possuem a mesma cor.
2. Computa dois histogramas:
 - Histograma dos pixels classificados como borda;
 - Histograma dos pixels classificados como interior.
3. Concatena os histogramas.

Auto Color Correlogram (ACC): captura a correlação espacial entre cores idênticas. Para tal, computa a probabilidade de encontrar dois pixels com a mesma cor, a uma distância d um do outro (HUANG *et al.*, 1997). O vector resultante consiste na concatenação dos auto-correlogramas, um para cada distância. Neste estudo, são consideradas quatro distâncias para d : 1, 3, 5 e 7; resultando em um vetor com $4N$ características.

Haralick-6: descreve a textura das imagens, ou seja, diferenças locais em níveis de intensidade (HARALICK; SHANMUGAM; DINSTEIN, 1973). O estudo original sugere 14 características, porém apenas seis delas são utilizadas nesta pesquisa. Isso porque essas características possuem independência entre si. O vetor resultante possui seis dimensões que representam as seguintes características:

- *Probabilidade máxima:* maior resposta na matriz de co-ocorrência. Intervalo: $[0, 1]$;
- *Correlação:* descreve as correlações entre as linhas e colunas da matriz. Intervalo: $[-1, 1]$;
- *Contraste:* mede as variações locais dos níveis de cinza da matriz. Intervalo: $[0, (\text{intensidades} - 1)^2]$;
- *Uniformidade:* soma dos elementos quadrados. Também conhecido como energia ou segundo momento angular. Intervalo: $[0, 1]$;
- *Homogeneidade:* mede a proximidade da distribuição dos elementos em relação à diagonal. Intervalo: $[0, 1]$;
- *Entropia:* descreve a aleatoriedade. Intervalo: $[0, 2 * \log_2 \text{intensidades}]$.

Histogram of Oriented Gradients (HOG): calcula a frequência da ocorrência da orientação dos gradientes em janelas da imagem ([DALAL; TRIGGS, 2005](#)):

1. Divide a janela da imagem em regiões chamadas células;
2. Computa os gradientes de primeira ordem para cada célula. O método mais comum para tal é aplicar a máscara da derivada 1D centralizada em cada pixel, nas direções horizontal e vertical;
3. Para cada célula, computa o histograma das orientações dos gradientes de todos os pixels da célula;
4. Normaliza o contraste de “blocos” da janela, que se sobrepõem. Dessa forma, cada célula é normalizada em relação a diferentes blocos;
5. Concatena os histogramas de todos os blocos.

O método HOG resulta em um vetor de características com o número de dimensões sendo

$$N = \text{blocos por imagem} \times \text{tamanho do bloco} \times \text{número de bins},$$

onde o *número de bins* é o tamanho do histograma de uma célula. Esses parâmetros são dados como entrada para o algoritmo.

Local Binary Patterns (LBP) utilizando padrões uniformes: baseia-se em reconhecer que padrões de textura uniformes são propriedades fundamentais da textura local da imagem. O histograma da sua ocorrência provou-se ser um bom extrator de características. Computa um histograma de ocorrência dos padrões locais binários em uma vizinhança da imagem, detectando micro-estruturas cuja distribuição é estimada pelo histograma ([OJALA; PIETIKAINEN; MAENPAA, 2002](#)):

1. Divide a imagem em células;
2. Compara cada pixel em uma célula com seus oito vizinhos. Quando o pixel do centro tiver intensidade maior que o vizinho, escreve 0; caso contrário, 1. Esse passo resulta em um código binário de oito dígitos;
3. Computa o histograma da célula. Cada padrão uniforme possui um *bin* e todos os padrões não uniformes constituem um único *bin*;
4. Normaliza o histograma;
5. Concatena os histogramas de todas as células. Resulta em um vetor de características para a imagem.

Considerando que são 58 padrões uniformes, o tamanho do vetor de cada célula é de 59 dimensões. Assim, o vetor resultante do método LBP é $N = \text{número de células} \times 59$.

2.4 Desbalanceamento de classes

Nesta seção é definido o problema do desbalanceamento de classes e apresentados os trabalhos relacionados que possuem duas diferentes abordagens: sobreamostragem (*over-sampling*) e subamostragem (*under-sampling*).

Em conjuntos de dados desbalanceados, determinadas classes possuem um número muito maior de instâncias do que outras. As classes com mais elementos são chamadas de classes majoritárias, e as com menos elementos, de minoritárias. O desempenho de algoritmos de Aprendizado de Máquina é prejudicado quando tratam de bases de dados desbalanceadas. Esses algoritmos tendem a favorecer a classificação de um novo objeto à classe majoritária, pois esta fica muito melhor representada após o treinamento do que a minoritária. Considera-se, então, que esse problema é um obstáculo para a classificação satisfatória. Porém, como apontado por [Batista, Prati e Monard \(2004\)](#), o desbalanceamento não é o único responsável por reduzir o desempenho de algoritmos de aprendizagem. Eles sugerem que é possível haver uma ótima classificação mesmo contendo alto desbalanceamento na base de dados. Assim, a motivação do estudo de vários algoritmos para rebalanceamento não é apenas balancear os dados de treinamento, mas obter uma melhor diferenciação entre as classes. Isso porque o desbalanceamento por si só pode não ser um problema, mas em conjunto com a sobreposição de classes pode diminuir significativamente a acurácia da classificação da classe minoritária.

[Castro e Braga \(2011\)](#) destacam que duas abordagens têm sido utilizadas para solucionar esse problema: pré-processar os dados de forma a rebalancear as distribuições das classes, ao reamostrar os dados; ou então modificar métodos de aprendizado — e.g. através da adição de melhores funções de custo na classificação. Em geral, são reportados melhores resultados obtidos por algoritmos de *over-sampling*, os quais consistem em reamostrar os dados, aumentando o número de elementos da classe minoritária ([BATISTA; PRATI; MONARD, 2004](#)).

2.4.1 Sobreamostragem

Realizar uma sobreamostragem (*over-sampling*) em um determinado conjunto de dados significa aumentar — utilizando alguma estratégia — o número de elementos desse conjunto. Em [Chawla, Hall e Bowyer \(2002\)](#), a simples replicação de exemplos pertencentes à classe minoritária não melhorou a classificação. Isso se deve ao reconhecimento de regiões muito específicas, causando *overfitting*.

O *Synthetic Minority Over-sampling Technique* (SMOTE) é um método desenvolvido por [Chawla, Hall e Bowyer \(2002\)](#) para rebalancear classes ao gerar artificialmente novos elementos, ao invés de apenas replicá-los. O Algoritmo 1 é aplicado sobre os vetores de características previamente extraídos, com operações de perturbação dos dados de treino no espaço de características, e não no espaço dos dados. A diferença entre o vetor de características de um elemento e do seu vizinho mais próximo é multiplicada por um valor x , sendo $0 \leq x \leq 1$. Esse valor é adicionado

ao vetor original, criando um novo elemento.

Algoritmo 1: SMOTE: método para rebalancear classes

Entrada: Imagem colorida I em formato RGB

Saída: Exemplos sintéticos S

```

1  $N \leftarrow \text{vizinhos}(\text{classe minoritária});$ 
2 para cada exemplo da classe minoritária faça
3    $nn \leftarrow \text{vizinho aleatório de } N;$ 
4    $\text{novo\_elemento} \leftarrow \emptyset;$ 
5   para cada característica  $(x, y)$  do exemplo faça
6      $\text{diferença} \leftarrow nn(x, y) - \text{exemplo}(x, y);$ 
7      $\text{gap} \leftarrow \text{número aleatório entre } 0 \text{ e } 1;$ 
8      $\text{novo\_elemento}(x, y) \leftarrow \text{exemplo}(x, y) + \text{gap} * \text{diferença};$ 
9   fim
10   $S \leftarrow S \cup \text{novo\_elemento};$ 
11 fim

```

Como pode ser visualizado na Figura 7, essa abordagem provoca a geração de um elemento resultante da interpolação dos dois vetores originais. Os exemplos sintéticos forçam uma região de decisão maior e mais geral para serem aprendidas como exemplos da classe minoritária. Dessa forma, o SMOTE provê mais elementos para o classificador aprender, ao contrário da replicação de dados. Como trabalhos futuros, os autores apontam que diferentes estratégias para criar esses exemplos sintéticos podem melhorar o desempenho da classificação. Inclusive salientando exemplos que foram erroneamente classificados.

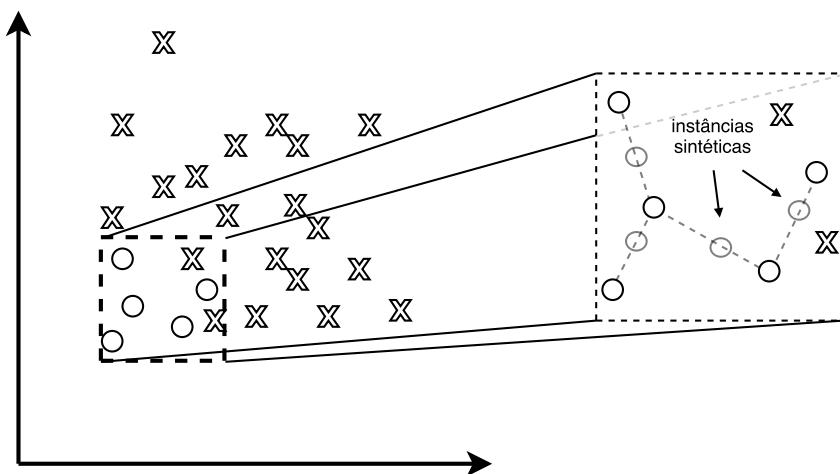


Figura 7 – Método SMOTE: interpolação entre dois exemplos vizinhos no espaço de características. A diferença entre o vetor de características de um elemento e do seu vizinho mais próximo é multiplicada por um valor x , sendo $0 \leq x \leq 1$. Esse valor é adicionado ao vetor original, criando um novo elemento entre eles.

Fonte: Elaborado pela autora.

Uma variação desse algoritmo, denominada Borderline-SMOTE1 (HAN; WANG; MAO, 2005), considera que elementos fora da linha de borda de cada classe pouco contribuem para a

classificação. Por isso, propõe a geração de elementos sintéticos utilizando apenas elementos de borda. Considera que se os vizinhos mais próximos são da classe majoritária, o exemplo é ruído, e se há mais vizinhos da classe majoritária do que da minoritária, considera esse elemento como sendo de borda. Como trabalho futuro, destacam a necessidade de considerar diferentes estratégias para definir quais elementos utilizar para realizar o *over-sampling*.

2.4.2 Subamostragem

Ao contrário da sobreamostragem, a subamostragem visa diminuir o número de elementos de um determinado conjunto. A ideia é eliminar elementos da classe majoritária que estão distantes da fronteira de decisão, isso porque eles são considerados menos relevantes para a aprendizagem.

Métodos para remoção de exemplos da classe majoritária normalmente apresentam resultados piores do que métodos de sobreamostragem, conforme relatado por [Batista, Prati e Monard \(2004\)](#) e [Japkowicz e Stephen \(2002\)](#). Um dos motivos pela preferência natural à sobreamostragem é o fato de que ao realizar subamostragem pode-se remover informações essenciais dos dados originais. Mas não há uma estratégia única que funcione melhor para todos os cenários.

2.5 Redução de dimensionalidade

Após a etapa de extração de características, é importante obter-se um espaço de características reduzido, ou seja, com menor dimensionalidade. Muitos métodos de aprendizagem de máquina possuem complexidade computacional proporcional ao número de imagens e de dimensões. Além disso, devido a *maldição da dimensionalidade*, um espaço com muitas dimensões pode dificultar a classificação. Ou seja, há um número máximo de características do qual o desempenho do classificador será comprometido. Uma solução é reduzir a dimensão ao utilizar métodos de seleção de atributos ou métodos de redução de dimensionalidade.

Em cenários de reconhecimento de imagens, um vetor de características extraído com extratores comuns pode possuir, por exemplo, 6 (e.g. Haralick), 512 (e.g. BIC) ou mais características. Caso seja requerida a visualização dessas amostras, é necessário mapear esses valores em apenas duas dimensões. Para isso, a redução de dimensionalidade mapeia os vetores de N dimensões para um espaço 1D ou 2D. A partir desses novos vetores, pode então ser criada alguma representação visual que tente manter a relação de distância entre os novos e os originais ([PAULOVICH; OLIVEIRA; MINGHIM, 2007](#)).

2.5.1 Análise de componentes principais

Principal Component Analysis (PCA) é uma técnica não supervisionada que pode ser utilizada para reduzir a dimensionalidade dos dados com a máxima variância possível. Cada imagem, originalmente representada por um vetor com N características, pode então ser representada por apenas um ou mais valores. O objetivo é representar os dados como um conjunto de variáveis ortogonais chamadas de componentes principais. Para isso, encontra-se uma outra base: uma combinação linear da base original, que melhor representa os dados ao assumir que as direções das maiores variâncias são as mais importantes. Ou seja, a variância associada com cada direção quantifica o quanto principal é aquela direção (ABDI; WILLIAMS, 2010). Pode-se, portanto, enumerar os passos necessários para o PCA como:

1. Centraliza todos os atributos na origem ao subtrair a média de cada dimensão;
2. Calcula a matriz de covariância C_x dada por

$$C_x = XX^T, \quad (2.6)$$

onde X é a matriz de dados original e X^T sua transposta;

3. Encontra os autovalores e autovetores de C_x . Um autovetor \vec{u} de uma matriz A pode ser definido por $A\vec{u} = \lambda\vec{u}$, onde λ é um autovalor escalar associado ao autovetor. Um vetor \vec{u} é um autovetor da matriz A se o tamanho do vetor — e não sua direção — é modificado quando multiplicado por A . Os autovalores podem ser representados na diagonal de uma matriz λ (com outros valores como zero) e o conjunto dos autovetores de A em uma matriz U . Assim, $A = U\lambda U^T$;
4. Então, os autovetores são ordenados de forma decrescente de acordo com seus autovalores correspondentes e escolhe-se os k principais autovetores (i.e. maiores autovalores) para formar uma matriz P de dimensão $n \times k$, onde cada coluna representa um autovetor. O valor k será a quantidade de dimensões do novo espaço de atributos;
5. O novo subespaço pode ser encontrado multiplicando essa matriz P pela matriz original, de acordo com a equação $Y = PX$, onde X representa o conjunto de dados original, Y é uma nova representação desses dados e P a matriz ortonormal que transforma X em Y . As linhas de P são os componentes principais de X .

2.5.2 Locality Preserving Projections

Locality Preserving Projections (LPP) é um algoritmo linear de redução de dimensionalidade com propriedades de preservação da estrutura local dos dados. Não apresenta a dificuldade dos algoritmos tradicionais (como PCA) de manter o *manifold* não linear dos dados originais

(ZHUO; CHENG; ZHANG, 2014). Embora o método mais utilizado para redução da dimensionalidade de forma não-supervisionada seja o PCA, métodos como esse produzem melhores projeções em termos de separação das classes. Em Zhuo, Cheng e Zhang (2014) o LPP obteve a melhor relação entre complexidade computacional e a redução da dimensionalidade, enquanto manteve a acurácia. Seu algoritmo segue três passos principais (HE; NIYOGI, 2004):

1. Constrói um grafo de adjacências. Os nós i e j possuem uma aresta entre si se fazem parte do conjunto de k -vizinhos mais próximos de cada nó (sendo k um parâmetro do algoritmo);
2. Encontra os pesos $W_{ij} = 1$ se os vértices i e j estão próximos, ou seja, conectados por uma aresta, e $W_{ij} = 0$ caso contrário;
3. Computa os autovalores e autovetores

$$XLX^T \vec{a} = \lambda XDX^T \vec{a}, \quad (2.7)$$

onde D é a matriz diagonal na qual seus elementos são as somas das colunas de W e L é a matriz Laplaciana $L = D - W$. X é a matriz original dos dados e \vec{a} é o vetor solução (matriz de projeção), ordenado pelos autovalores λ .

2.6 Classificador de padrões

A tarefa de classificação de imagens consiste em tentar predizer a classe de uma determinada imagem. Na etapa de treinamento o método recebe como entrada um conjunto de imagens rotulado com suas respectivas classes. Com o modelo treinado, é possível realizar a classificação de exemplos de rótulo desconhecido. Assim, em um experimento são preditas as classes de um conjunto de imagens de teste.

2.6.1 K-Nearest Neighbors

O classificador *K-Nearest Neighbors* (K-NN) considera a proximidade entre os dados para realizar as previsões. Baseia-se na premissa de que os objetos do mesmo conceito são semelhantes. O seu funcionamento está descrito no Algoritmo 2. Na fase de treinamento, apenas armazena os exemplos rotulados do conjunto de dados de treinamento. Quando um novo exemplo deve ser classificado, calcula a distância entre os vetores de características do novo exemplo e aqueles já rotulados. O novo exemplo é então classificado como sendo da classe do exemplo de treinamento com menor distância (BOIMAN; SHECHTMAN; IRANI, 2008). Com $K = 1$, a previsão da classe corresponde ao exemplo mais próximo. Trata-se de um classificador que utiliza todas as instâncias do modelo.

Algoritmo 2: K-NN: método de classificação supervisionada

Entrada: Conjunto de exemplos S_{treino} e S_{teste}
Saída: Classes C dos exemplos de teste preditas

```

1  $C \leftarrow \emptyset;$ 
2 para cada  $teste \in S_{teste}$  faz
3    $N \leftarrow k$  vizinhos mais próximos( $teste, S_{treino}$ );
4    $D \leftarrow \emptyset;$ 
5   para cada vizinho  $n \in N$  faz
6      $D \leftarrow D \cup$  distância( $n, teste$ );
7   fim
8    $C \leftarrow C \cup$  menor( $D$ );
9 fim
```

2.6.2 Support Vector Machines

Uma SVM encontra um hiperplano separador no espaço de características otimizando a máxima distância para o exemplo de treinamento mais próximo de cada classe. A versão atual do classificador, que permite sobreposição entre classes (também conhecida como versão *soft-margin*), foi proposta por [Cortes e Vapnik \(1995\)](#). O algoritmo utiliza exemplos de treinamento, resolvendo um problema de otimização formulado por meio de multiplicadores de Lagrange e um sistema de equações *Karush Kuhn Tucker* (KKT).

Os exemplos de treinamento utilizados para computar o hiperplano são chamados vetores de suporte. Por isso, após o treinamento, todos os dados podem ser descartados, exceto os vetores de suporte que são utilizados para a etapa de classificação ([MARSLAND, 2015](#)).

Comumente, dados em um espaço de características de entrada não são linearmente separáveis. Nesses casos pode-se mapear o espaço original em um espaço de dimensão maior no qual os dados são presumivelmente separáveis por um hiperplano. Esse mapeamento é comumente projetado de forma que produtos internos possam ser computados nas variáveis do espaço original definindo-as em termos de uma função *kernel*. Ao lidar com características obtidas a partir de imagens, a dimensão do espaço original é normalmente de alta dimensão (> 1000) se comparada com o tamanho do conjunto de treinamento. Por isso, os dados de treinamento são geralmente linearmente separáveis ([CHAPELLE; HAFFNER; VAPNIK, 1999](#)).

2.7 Considerações finais

Deu-se destaque à discussão das etapas de pré-processamento para a quantização de imagens, além das abordagens existentes para o rebalanceamento de classes. Este capítulo apresentou métodos para exemplificação, além de trabalhos relacionados. A extração de características foi abordada, apresentando os principais descritores utilizados nesta pesquisa. A lacuna destacada é que existem características não passíveis de extração por descritores convencionais. Esses fundamentos permitem compreender o contexto no qual esta dissertação está inserida.



QUANTIZAÇÃO DE IMAGENS

3.1 Considerações iniciais

Sistemas de reconhecimento de imagens comumente utilizam uma imagem em níveis de cinza ($2^3 = 8$ bits, $2^8 = 256$ intensidades) para as etapas subsequentes à extração de características. Dá-se o nome de quantização à etapa responsável por esta redução no nível de cores em uma imagem. Ao aplicar a quantização na etapa de pré-processamento, é esperada a redução da complexidade do vetor de características logo no início do processo, beneficiando todos os passos subsequentes.

Com o objetivo de analisar o impacto do uso da quantização, foram utilizados diferentes parâmetros de quantização, combinados com quatro métodos de extração de cor e um de textura. Esses métodos foram escolhidos de acordo com os resultados apresentados por [Penatti, Valle e Torres \(2012\)](#) e [Wang, Han e Yan \(2009\)](#). Este capítulo aborda a quantização de imagens antes da etapa de extração de características, assim como os métodos utilizados para tal, já apresentados na Seção 2.2.3. Após, apresenta os resultados encontrados ao aplicar os métodos de quantização de imagens no *pipeline* de reconhecimento. Para cada experimento realizado são descritos: a base de imagens; o protocolo utilizado; os resultados encontrados e a sua relevância. Os resultados devem refletir melhorias nas etapas subsequentes, como uma melhor acurácia na etapa de classificação ou a redução do tempo de processamento.

3.2 Quantização de imagens

O *pipeline* de reconhecimento de imagens comumente envolve um passo de conversão de imagens coloridas em imagens com apenas um canal de cor. Obtém-se, assim, uma imagem quantizada, que pode ser então processada por métodos de extração de características. Dessa forma, cada imagem — originalmente no espaço de cor RGB — é convertida a um único canal

com C níveis de intensidade. Após, são utilizados os métodos de extração de características para obter um vetor representativo da imagem. A Figura 8 ilustra esses passos, desde a aquisição até a classificação das imagens.

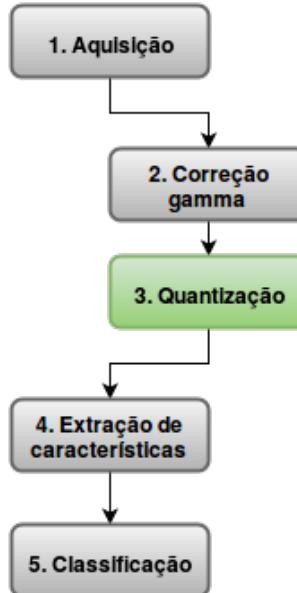


Figura 8 – O pipeline de reconhecimento de imagens pode envolver uma etapa de conversão imagens coloridas em imagens em escala de cinza, obtendo uma imagem quantizada que pode ser então processada por métodos de extração de características. O vetor com essas características é então dado como entrada a um método posterior de classificação.

Fonte: Elaborado pela autora.

Cada método de quantização se comporta diferentemente para uma dada imagem RGB. O método *Intensidade* (Equação 2.2), por exemplo, mapeia todas as permutações dos mesmos valores de RGB para a mesma cor de cinza. Dessa forma, produz um plano no espaço RGB conforme mostrado na Figura 9. O efeito do método *Gleam* (Equação 2.3) é similar, mas dada a natureza da função *gamma* (i.e. transformação não linear que define a relação entre o valor do pixel e sua real luminância), cobre uma superfície curva. Tal resultado também é alcançado utilizando o método *Intensidade'* (método *Intensidade* corrigido por *gamma*). Independente do método utilizado, o resultado é o mapeamento de características cromáticas bem diferentes em valores de intensidades similares. Ou seja, cores distintas podem ser convertidas em cores próximas, podendo aumentar a confusão entre objetos. Pode-se imaginar esse efeito em uma imagem natural que retrata céu e grama. Considere a confusão das intensidades resultantes da região de céu e grama (e.g RGB(0, 0, 255) e RGB(0, 255, 0), respectivamente): apesar de terem intensidades bem distintas em RGB, os valores em apenas um canal de cor podem estar relativamente próximos. Os métodos *Luminância'* (Equação 2.4) e *Luma* (Equação 2.5) procuram aprimorar essa quantização ao ponderar a combinação linear dos canais. Esses métodos são normalmente considerados melhores por se aproximarem ao modelo visual humano, que pondera as cores através do número de cones sensíveis às cores vermelho, verde e azul. O método MSB também tenta enfatizar as diferenças cromáticas, ao ordenar os bits de cores em um único canal.

Para mais detalhes sobre esses métodos, consulte Seção 2.2.3.

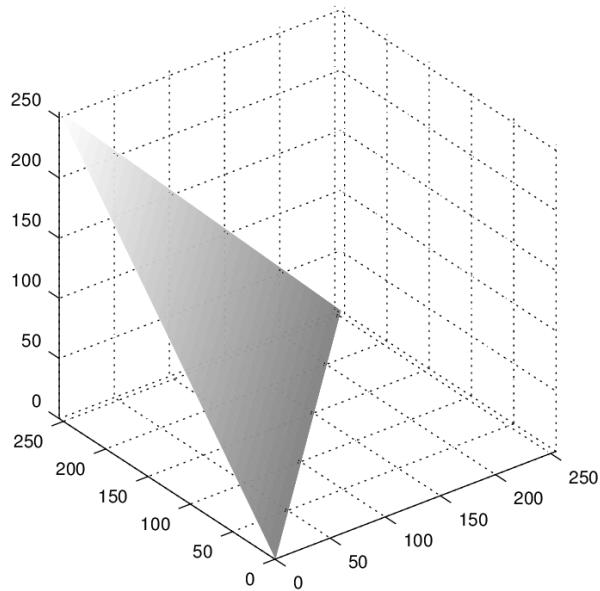


Figura 9 – Plano no espaço RGB, computado pelo método de conversão para escala de cinza *Intensidade*, quando um dos canais de cor (vermelho, verde ou azul) possui valor 255. O resultado é o mapeamento de características cromáticas bem diferentes em valores de intensidade similares.

Fonte: Ponti, Nazaré e Thumé (2016).

Exemplos de imagens obtidas após os métodos de quantização apresentados anteriormente podem ser vistos na Figura 10. A barra de gradientes abaixo da imagem dos pincéis demonstra como os métodos de quantização se comportam dada a variação da cor. É possível notar que os métodos *Luminância'* e MSB conseguiram melhor discriminar as cores. Além disso, o mapa de cores MSB obteve um maior número de intensidades únicas, quando comparado aos demais métodos.

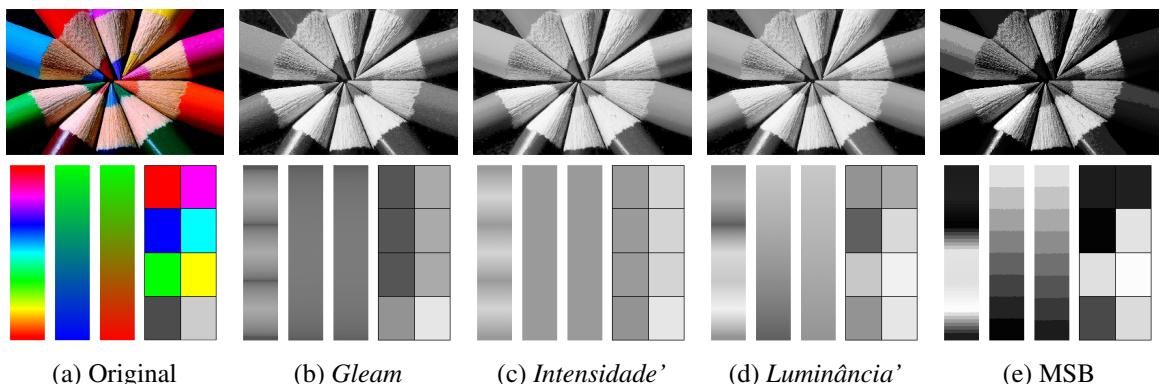


Figura 10 – Resultado da aplicação de métodos de quantização. A imagem original (a) resultou em versões de um canal de cor com 232 intensidades únicas para o método (e) MSB e 184 intensidades para os demais métodos. Ao analisar-se as barras de gradiente, assim como as paletas de cores, observa-se que os métodos *Luminância'* e MSB conseguiram uma melhor discriminação entre cores.

Fonte: Ponti, Nazaré e Thumé (2016).

Complementando, a Figura 11 apresenta um exemplo de redução de cores utilizando o método MSB para um par de imagens da base de dados *Caltech101-600* ([FEI-FEI; FERGUS; PERONA, 2007](#)). É possível notar que há uma certa preservação das cores, especialmente quando utilizados entre 64 e 256 níveis. Com apenas 32 cores, as imagens ainda lembram a sua versão original, mas há perda considerável de informação, principalmente em regiões da imagem com pouco contraste.

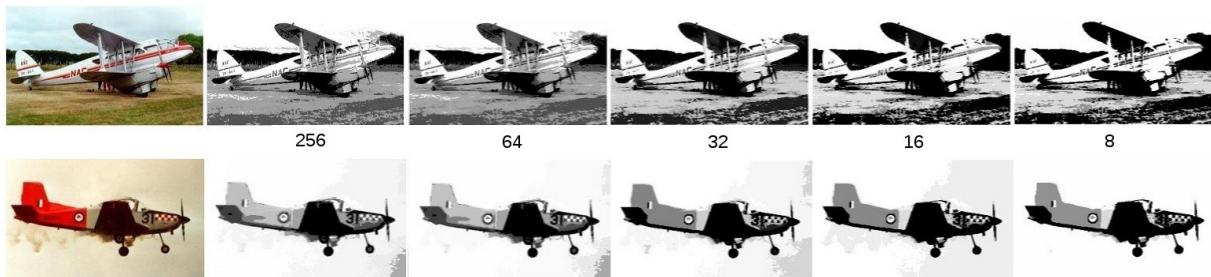


Figura 11 – Exemplo de duas imagens da base de dados *Caltech101-600* com variações no parâmetro de cor utilizando o método MSB. Da esquerda para a direita: imagem original 24-bits e suas versões quantizadas com: 256, 64, 32, 16 e 8 cores.

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

3.3 Experimentos

O objetivo desta seção é mostrar os efeitos da etapa de quantização e como ela pode ser utilizada para reduzir a dimensionalidade do espaço de características ou a complexidade em etapas posteriores do *pipeline* de classificação. A Figura 12 demonstra o fluxo das operações, juntamente com os métodos utilizados nos experimentos. Inicialmente, as imagens foram quantizadas em 256, 128, 64, 32 e 16 cores. Dependendo do método de conversão para a escala de cinza, a correção *gamma* é realizada (ver Seção 2.2.3). Após, suas características são extraídas e duas etapas distintas de experimentos são realizadas:

1. Experimentos utilizando um método de extração de características seguido pela classificação (sem posterior seleção de características);
2. Experimentos utilizando o vetor resultante da concatenação de todos os métodos de extração, seguido pela classificação com e sem a seleção de características.

3.3.1 Base de Imagens

Três bases de imagens, exemplificadas na Figura 13, foram utilizadas nestes experimentos de quantização:



Figura 12 – Fluxo das operações e os métodos utilizados nos experimentos. Após a aquisição da imagem, ela é convertida para escala de cinza e seus níveis de cor são reduzidos de acordo com um parâmetro da quantização (i.e. número de cores). Dependendo do método, a correção *gamma* é realizada. A imagem quantizada serve então como entrada para um método de extração de características e posteriormente é classificada com *SVM*. Uma das etapas de experimentos prevê também a concatenação de todos os vetores extraídos e a seleção das características com *LPP* antes da classificação.

Fonte: Elaborado pela autora.

Corel-1000 ([WANG; WIEDERHOLD, 2001](#)): consiste em dez classes balanceadas de imagens naturais, com algumas classes bem definidas e algumas não;

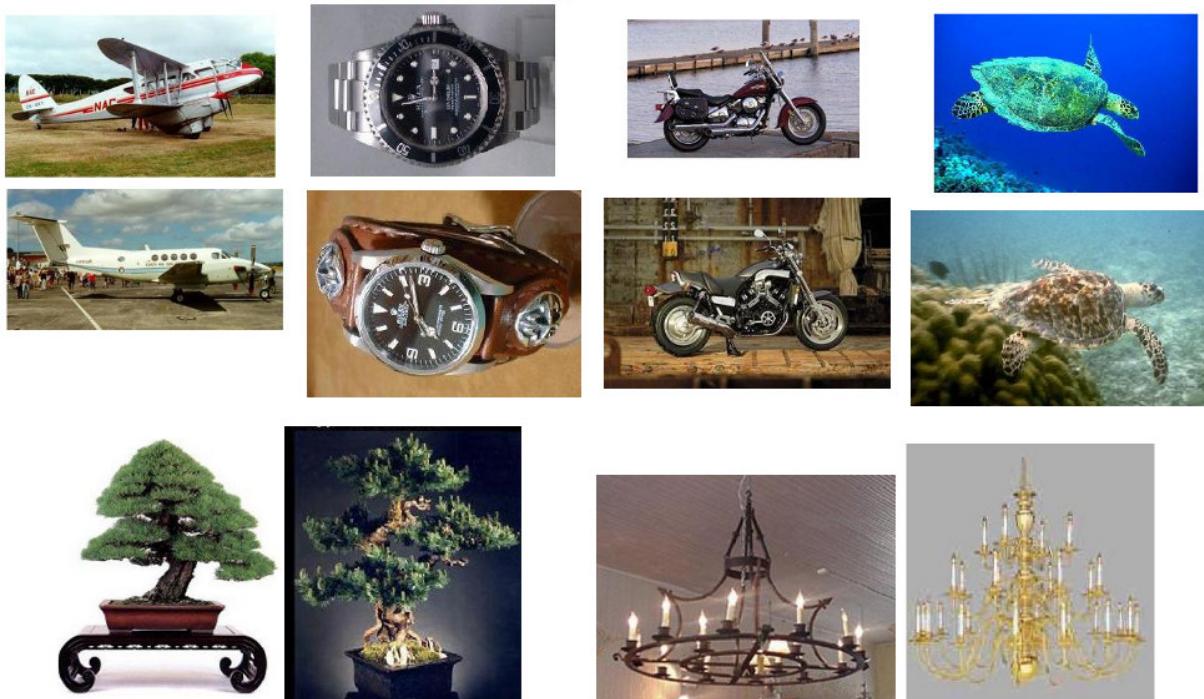
Caltech101-600 ([FEI-FEI; FERGUS; PERONA, 2007](#)): contém fotos e desenhos. Dessa base, foi utilizado um conjunto de seis classes balanceadas: aviões, bonsais, candelabros, tartarugas, motocicletas e relógios;

Produce ([ROCHA *et al.*, 2010](#)): também conhecido como base de vegetais e frutas tropicais. Composta por imagens com um fundo similar mas mudanças representativas na iluminação, no número de objetos e na escala. Apesar da oclusão parcial de objetos ser observada, essa classe possui dados bem comportados.

Considerando que estes experimentos possuem foco na redução na dimensionalidade, para evitar o problema do desbalanceamento, as bases Produce e Caltech101-600 foram modificadas. Dessa forma, as classes disponíveis foram balanceadas ao remover imagens das classes majoritárias.



(a) Base de imagens Corel-1000



(b) Base de imagens Caltech101-600



(c) Base de imagens Produce

Figura 13 – Bases de imagens Corel-1000, Caltech101-600 e Produce. Elas são utilizadas nos experimentos de quantização.

Fonte: Ponti, Nazaré e Thumé (2016).

3.3.2 Protocolo

Os experimentos foram realizados com uma validação cruzada de *10-fold*. Considerando que as bases estão balanceadas e que a seleção de exemplos para a validação cruzada é estratificada, a medida estatística de *acurácia* foi utilizada para avaliar o desempenho da classificação. O seguinte protocolo foi seguido para a obtenção dos resultados:

1. **Quantização:** com os métodos *Intensidade'*, *Gleam*, *Luminância'* e MSB.
2. **Extração de características:** utilizando os métodos — e parâmetros escolhidos com base nas recomendações dos artigos que proporam tais métodos — a seguir:
 - *Auto Color Correlogram* (ACC): a métrica de distância utilizada entre os pixels $p(x,y)$ e $q(s,t)$ é a tabuleiro de xadrez $D_8(p,q) = \text{Max}(|x-s|+d, |y-t|+d)$ para quatro distâncias $d = 1, 3, 5$ e 7 ;
 - *Border-Interior Classification* (BIC): com uma vizinhança de quatro pixels;
 - *Color Coherence Vector* (CCV): adotando um valor de $\text{threshold} = 25$ para a classificação dos pixels entre coerentes e incoerentes;
 - Haralick-6: o pixel vizinho para o qual iniciar a computar a matriz de co-ocorrência foi o pixel à direita.
3. **Redução da dimensionalidade:** a projeção utilizando *Locality Preserving Projections* (LPP) foi realizada com o parâmetro $k = 128, 64, 32$ e 16 dimensões e 10 vizinhos. Esse parâmetro foi determinado empiricamente e não influencia consideravelmente a acurácia.
4. **Classificação:** realizada com o classificador *Support Vector Machines* (SVM). Os parâmetros para essa etapa foram encontrados utilizando uma *grid search* no conjunto de treinamento.

3.4 Resultados e Discussão

A Figura 14 ilustra a acurácia média para o primeiro conjunto de experimentos: para cada combinação de base de dados e método de extração, são demonstrados seis resultados de acurácia correspondentes à quantização para $256, 128, 64, 32, 16$ e 8 cores. Com base nessa figura é possível identificar que o método para obter a imagem quantizada tem um impacto significativo na acurácia da classificação. Além disso, a redução de 256 para um menor número de cores normalmente manteve as acurárias e em alguns casos resultou em uma ligeira melhora, especialmente para os níveis de 128 e 64 .

Considerando que a utilização de apenas 16 e 8 cores resultou em uma acurácia muito inferior, o restante dos resultados utilizam $256, 128, 64$ e 32 cores. A partir dessa análise geral, uma análise mais específica foi realizada com a combinação dos métodos BIC e MSB; e

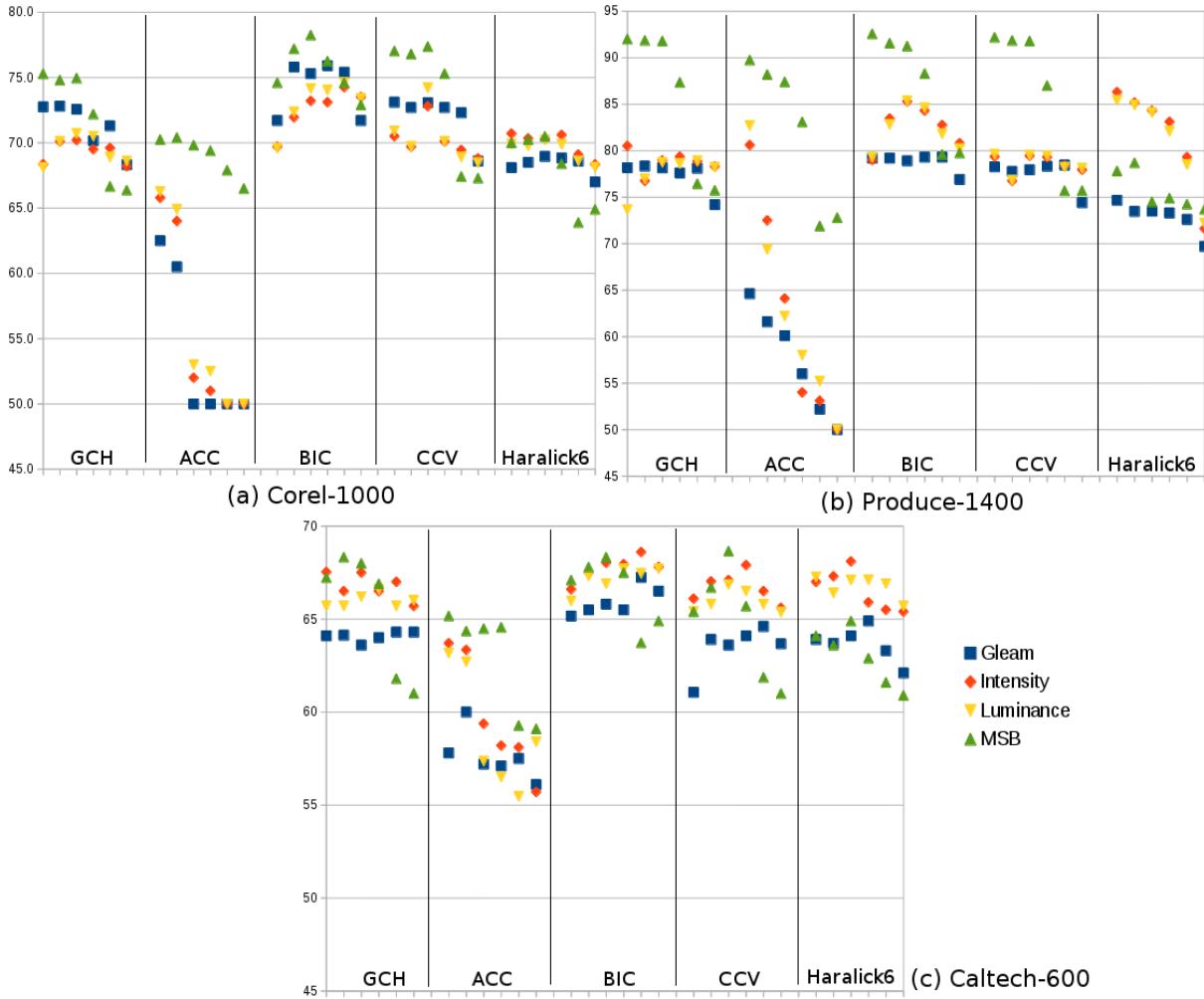


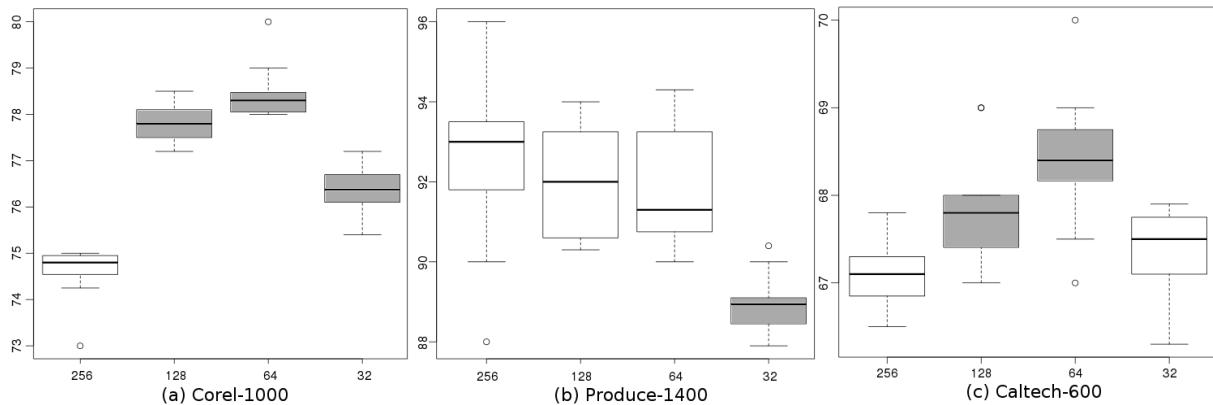
Figura 14 – Resultados para as bases Corel-1000 (a), Produce (b) e Caltech101-600 (c), utilizando todos os métodos de quantização. Para cada método de extração de características a acurácia é resultante da sua aplicação utilizando 256, 128, 64, 32, 16 e 8 cores, da esquerda para a direita.

Fonte: Ponti, Nazaré e Thumé (2016).

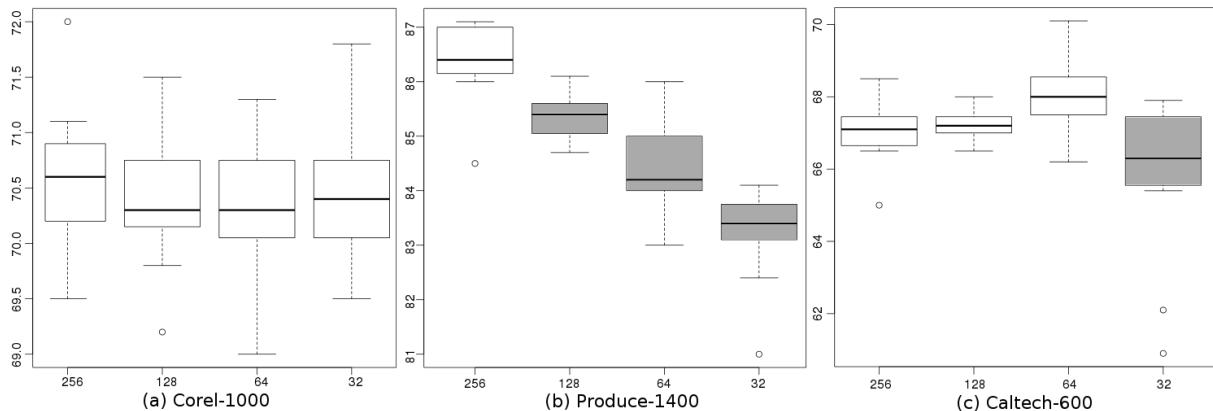
Haralick e *Luminância*'. O teste estatístico ANOVA foi realizado para comparar as acuráciais dos experimentos das Figuras 15 e 16. Para identificar se algum método obteve uma diferença significativa no valor de acurácia, foi utilizado o teste *Honest Significant Difference* (HSD) de Tukey. Um nível de significância de $\alpha = 0.01$ foi utilizado. Por conta disso, os boxplots em cinza correspondem aos dados com diferença estatística relevante quando comparados com a acurácia de 256 cores, obtendo um *p-value* < 0.01.

De acordo com o teste estatístico representado na Figura 15, utilizar características de cor (extraídas com o método BIC) e níveis de quantização providos pelo método MSB demonstrou resultados melhores do que o *baseline* de 256 cores para as bases Corel-1000 (128, 64 e 32 cores) e Caltech101-600 (64 cores). O único resultado que piorou significativamente foi para 32 cores da base de imagens Produce. Portanto, converter as imagens para a escala de cinza e reduzir os 256 possíveis valores para apenas 64 provou uma boa escolha de processamento anterior a extração de características. Ou seja, $d \approx D/4$. Menores valores podem degradar os resultados

em características de textura, como mostrado na Figura 16.



Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).



Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

Uma outra comparação importante é entre a redução de dimensionalidade obtida com métodos de quantização versus o método LPP. A redução de dimensionalidade obtida com os métodos MSB, BIC e LPP está ilustrada na Figura 17. A imagem de entrada foi convertida para escala de cinza com o método MSB em 256 cores. Essa imagem foi dada como entrada para o método de extração de características BIC, que resultou em um vetor dado como entrada para o LPP. Esse último passo teve o objetivo de produzir versões reduzidas desse vetor para 256, 128 e 64 dimensões. As acurárias obtidas foram então comparadas com a classificação dos vetores reduzidos apenas pela quantização. Como a comparação foi feita em pares, foram realizados testes t de Student sobre a suposição de dois exemplos independentes com variâncias desiguais e um nível de significância de 0.01. O método de quantização obteve valores de acurácia menores

à utilização do LPP em três experimentos: 256 cores com a base Corel-1000 e com 256 e 64 na base Produce. Para a base Caltech101-600 a quantização foi melhor com 256 e 128 dimensões. O restante dos experimentos não apresentaram diferença estatística relevante. Apesar da perda de acurácia em alguns casos, é importante notar que — se utilizado um número de cores correto — é possível manter ou até mesmo melhorar as acuráncias após a redução da dimensionalidade. Isso pode ser observado na Figura 17 referente à base de dados Caltech101-600.

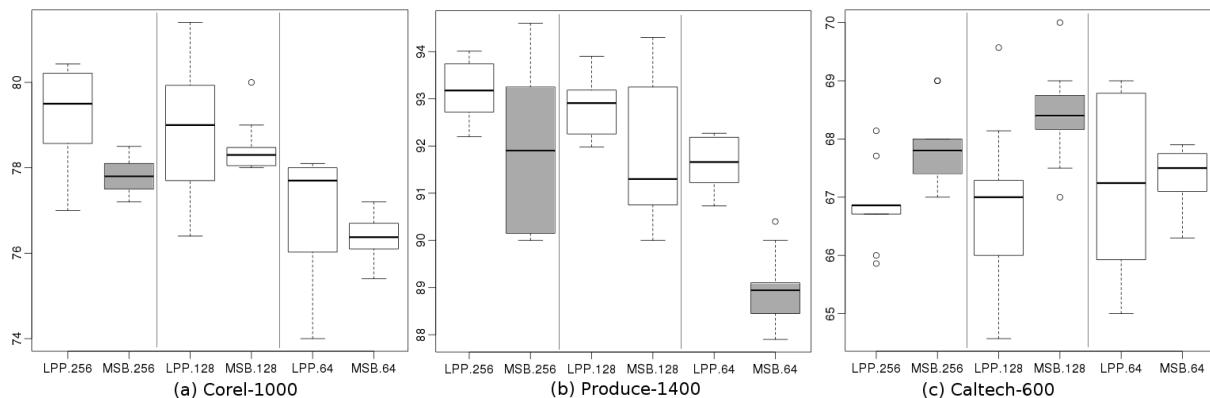


Figura 17 – Resultados de acurácia para os método MSB (quantização), LPP (redução de dimensionalidade) e BIC (extração de características). A comparação do LPP versus MSB foi realizada com a mesma dimensionalidade. Os boxplots em cinza correspondem às significâncias estatísticas com $p < 0.01$ quando comparado a acurácia de 256 cores.

Fonte: Ponti, Nazaré e Thumé (2016).

O número de dimensões de um vetor resultante de apenas um método de extração de características pode ser considerado baixo. É comum a extração de diversos descritores para uma determinada situação, considerando que normalmente não é claro qual método deveria ser utilizado em cada caso. Por conta disso, os próximos experimentos foram realizados a partir da concatenação de tais características. O objetivo destes experimentos é verificar se a concatenação de todos os descritores pode melhorar os resultados de acurácia. Além disso, comparar os resultados com os experimentos anteriores, afim de verificar se a quantização pode ser uma alternativa à redução da dimensionalidade com métodos convencionais (LPP, neste caso). A melhor configuração encontrada com os experimentos anteriores, entre tamanho do vetor e acurácia, foi utilizando 128 e 64 cores.

A redução do número de cores influencia a dimensionalidade original D . O número de características em relação ao número de cores, concatenando todos os vetores resultantes dos métodos de extração de características, é: 256 cores — 2310 características; 128 cores — 1160 características; 64 cores — 582 características; 32 cores — 294 características; e 16 cores — 150 características.

Primeiramente, as imagens foram convertidas para escala de cinza e mantidas com 256 cores. Essas imagens foram descritas por todos os métodos de extração e suas características concatenadas em um vetor com dimensão original $D = 2310$. A redução de dimensionalidade com LPP foi realizada para $d = 1160, 582, 294$ e 150 . Ou seja, produzindo vetores com o mesmo

tamanho dos obtidos apenas com a quantização como método de redução da dimensionalidade. A Figura 18 mostra os resultados utilizando LPP para as três bases de imagens. Note que o método de quantização MSB resultou em acuráncias melhores que os outros métodos.

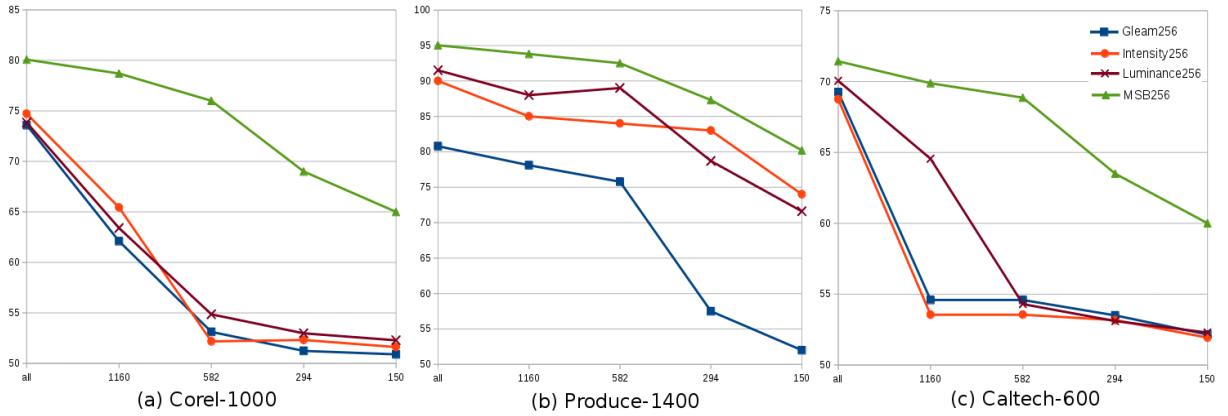


Figura 18 – Comparação da acurácia alcançada com diferentes métodos de quantização: *Gleam*, *Intensidade*, *Luminância* e MSB. Inicialmente as imagens foram convertidas para escala de cinza com esses quatro métodos e foram dadas como entrada para todos os métodos de extração. O vetor de características resultante com $D = 2310$ sofreu então redução da dimensionalidade com o método LPP para $d = 1160$, 582, 294 e 150.

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

A utilização de todos os vetores concatenados melhorou a acurácia em relação ao melhor descritor individual. A Figura 19 apresenta a comparação do espaço original com LPP e MSB para redução da dimensionalidade. Os testes estatísticos HSD de Tukey foram realizados utilizando $\alpha = 0.01$ como nível de significância. Os resultados que não mudaram significativamente as acuráncias foram: MSB com 582 características para a base de dados *Corel*; e MSB com 1160 para as três bases. O resultado de piora significativa foi para 32 cores com a base de imagens *Produce*. Dados tais resultados, utilizar 64 cores é indicado como uma boa escolha do parâmetro de quantização.

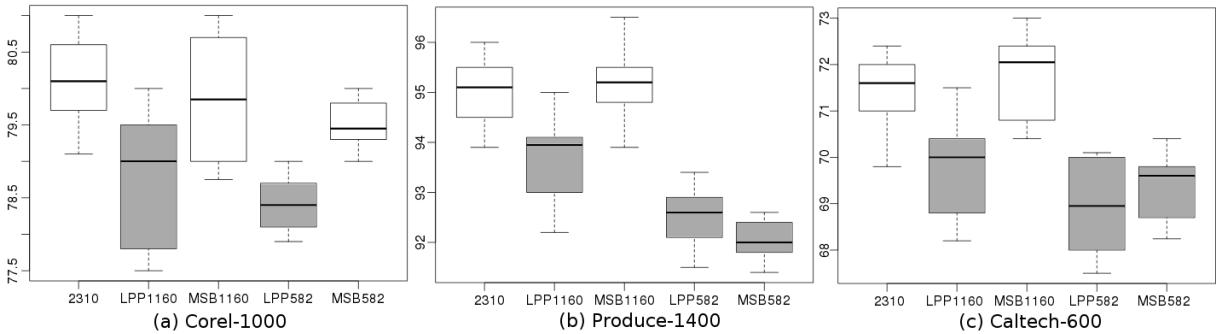


Figura 19 – Comparação da acurácia com o uso da projeção LPP e o método MSB para quantização das imagens com o objetivo de redução de dimensionalidade.

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

Os resultados indicam que a quantização pode ser utilizada como redução da dimensão de dados visuais, especialmente utilizando 128 e 64 cores. Como outro experimento, a Figura 20

mostra as acuráncias resultantes da aplicação do LPP sob o vetor obtido após a quantização com MSB utilizando 256 e 64 cores ($d = 2310$ e $d = 582$, respectivamente). É interessante notar que as projeções LPP em geral foram melhores com as imagens quantizadas em 64 cores com MSB ao invés da original em 256. A razão para isso deve estar no fato da quantização remover informações confusas: ela simplifica as imagens de forma que as cores restantes possam melhor descrever uma certa classe.

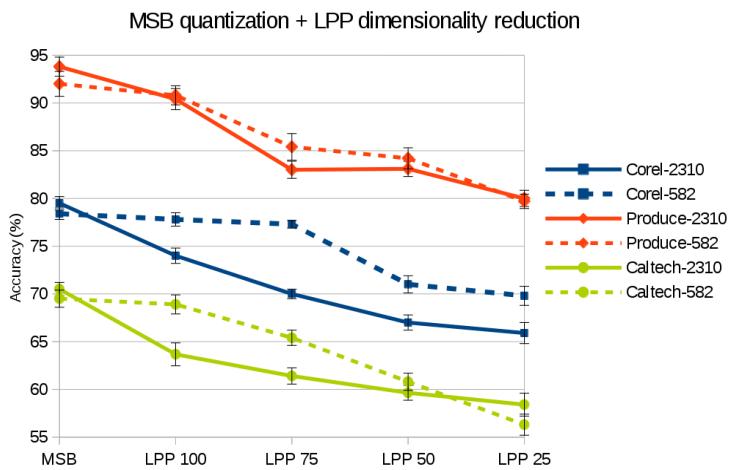


Figura 20 – Resultados para a projeção do LPP sobre o espaço de características produzido pelo método de quantização MSB utilizando 256 ($d = 2310$) e 64 cores ($d = 582$)

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

3.5 Considerações finais

Ao aplicar a quantização na etapa de pré-processamento, obteve-se a redução da dimensionalidade do vetor de características no início do *pipeline*, beneficiando todas as etapas posteriores. Utilizar um número reduzido de cores reduziu significativamente a dimensionalidade, enquanto melhorou ou manteve a classificação do sistema. O método MSB para a quantização obteve o melhor desempenho, dada a preservação das cores observada na Figura 11.

Ao comparar o uso da quantização com a utilização de métodos mais complexos para a redução da dimensionalidade, esse processamento permite uma redução significante, enquanto normalmente preserva ou melhora a acurácia do sistema. Independente da utilização de um método de seleção de características, ao escolher um método de quantização apropriado e seus parâmetros é possível reduzir a dimensionalidade e acelerar computacionalmente as etapas que precedem o reconhecimento de imagens. O vetor concatenado com todos os descritores possui $9C + 6$ dimensões, onde C é o número de cores da imagem de entrada. O tempo de execução para a extração de todas as características é $f(N) = 42N + 6C^2$, onde N é o número de pixels. Para cada imagem são necessárias $D^2 + kD + d^2$ operações para computar o vetor reduzido com LPP, onde D é o tamanho do vetor original, d o tamanho do vetor de saída e k é o número de vizinhos utilizados no algoritmo. Considere o seguinte exemplo: 100 imagens com 256 cores demandam

231.6 milhões de instruções para extrair as características e reduzir o vetor utilizando o método LPP (com $k = 10$ e $d = 50$). Se, ao invés disso, fossem utilizadas 64 cores, esse número cairia para 58.7 milhões, o que corresponde a uma redução de 74,6%.



GERAÇÃO DE IMAGENS ARTIFICIAIS

4.1 Considerações iniciais

O processo de manipular imagens para que elas se tornem mais satisfatórias para um determinado objetivo depende do domínio de aplicação. Ou seja, não existe uma teoria geral para melhorar qualquer imagem (GONZALEZ; WOODS, 2007); um método que processa melhor uma imagem bem definida por suas cores difere do processamento de imagens texturizadas, às quais um processamento sobre a intensidade dos pixels da imagem — como uma operação de borrimento — pode ocasionar perda da textura. Assim, justifica-se a exploração de um vasto número de métodos de processamento de imagens e bases.

Neste capítulo oito métodos de processamento de imagens são utilizados com o fim de obter imagens artificiais para subamostrar classes minoritárias em problemas de classificação, rebalanceando as bases de imagens antes da extração de características. Isso é realizado com dois propósitos: 1) para compensar a baixa disponibilidade de exemplos de uma determinada classe, e 2) permitir a extração de informações antes não disponíveis nas imagens originais por meio da combinação ou perturbação das imagens de entrada.

Dada a quantidade de imagens necessárias para rebalancear a base original, são geradas novas imagens a partir de um conjunto de métodos. Como demonstrado na Figura 21, dado o conjunto de treinamento da classe (ou classes) com menor número total de imagens, é realizado o rebalanceamento ao aplicar os métodos descritos neste capítulo. Posteriormente, essas imagens resultantes têm suas características extraídas e são utilizadas no treinamento da classe minoritária.

Os resultados encontrados ao rebalancear classes a partir da geração de imagens artificiais são também apresentados neste capítulo. Para cada experimento realizado, são descritos: o protocolo utilizado (i.e. base de imagens, os métodos de conversão para escala de cinza e extração de características), os resultados encontrados e a discussão da relevância de tais resultados.

Foram realizados diversos experimentos direcionados a explorar o rebalanceamento

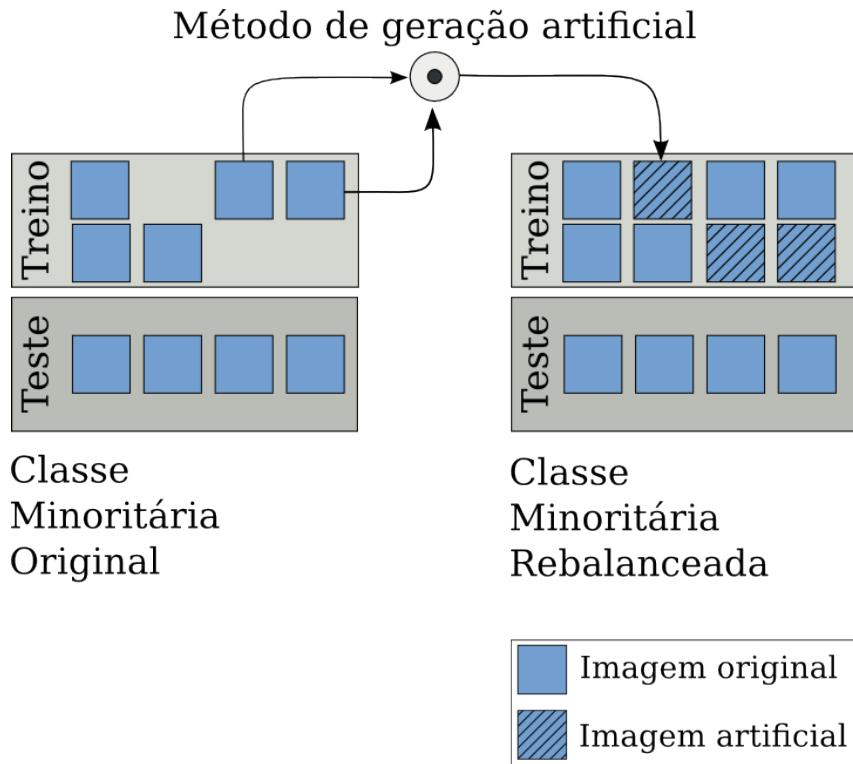


Figura 21 – Geração artificial da classe minoritária para rebalancear a base de imagens. Para cada imagem necessária para igualar o número de imagens da base, $1 \leq n \leq 16$ imagens originais são dadas como entrada para uma operação de geração artificial. A nova imagem é utilizada como treinamento da base.

Fonte: Elaborado pela autora.

com os métodos de processamento, com o objetivo de melhorar a acurácia da classificação de bases de imagens. Como entrada são utilizadas imagens provenientes de diversas coleções disponíveis na literatura. Como resultado, são calculadas medidas estatísticas da classificação dessas bases de imagens após o rebalanceamento destas. Tal processamento é realizado antes da extração de características, e portanto no campo visual. Espera-se que os resultados encontrados proporcionem melhorias na etapa subsequente de classificação.

4.2 Métodos para geração de imagens artificiais

Os métodos de geração artificial para o rebalanceamento de classes de imagens são descritos nesta seção. Os experimentos, posteriormente destacados na Seção 4.3, foram realizados utilizando as operações de: *borramento*; *mistura ponderada*; *aguçamento*; *composição*; *mistura limiarizada*; *mistura saliente*; *SMOTE visual*; e *adição de ruído*.

4.2.1 *Borramento*

Também conhecido como suavização, o *borramento* é uma operação de processamento comumente utilizada com o objetivo de filtrar uma imagem, removendo ruídos e detalhes não relevantes. Normalmente esse tipo de filtragem provoca também um certo borramento das bordas,

como pôde ser observado anteriormente na Figura 4. Esse comportamento não é esperado quando queremos gerar novas imagens, pois informações relevantes podem ser removidas. Com o intuito de preservar as bordas, a operação de borramento **filtro bilateral** pode ser utilizada. Ela substitui o valor do pixel $I(x,y)$ pela média dos pixels vizinhos de intensidade similar (TOMASI; MANDUCHI, 1998). Ou seja, é uma média ponderada das intensidades que considera a diferença dos valores entre vizinhos para preservar bordas. Assim, para um pixel influenciar outro, deve estar próximo no espaço de coordenadas e possuir intensidade similar. O Algoritmo 3 descreve os passos desse filtro na sua versão mais simples, de força bruta. Considere W o termo de normalização, e G o filtro de Gaussianas. A Figura 22 exemplifica o seu funcionamento: à esquerda está demonstrada a imagem original e à direita a imagem borrada.

Algoritmo 3: Geração artificial: *borramento com filtro bilateral*

Entrada:

Imagen colorida $I_{original}$ em formato RGB

Sigma do espaço de cor σ_{range}

Sigma do espaço de coordenadas $\sigma_{spatial}$

Saída:

Imagen borrada $I_{borrada}$

```

1 para cada pixel  $(x,y)$  faça
2    $I_{borrada}(x,y) \leftarrow 0;$ 
3    $W(x,y) \leftarrow 0;$ 
4   para cada pixel  $(i,j)$  faça
5      $w \leftarrow G_{\sigma_{spatial}}(\|(x,y) - (i,j)\|)G_{\sigma_{range}}(|I_{original}(x,y) - I_{original}(i,j)|);$ 
6      $I_{borrada}(x,y) \leftarrow I_{borrada}(x,y) + wI_{original}(i,j);$ 
7      $W(x,y) \leftarrow W(x,y) + w;$ 
8   fim
9    $I_{borrada}(x,y) \leftarrow I_{borrada}(x,y)/W(x,y);$ 
10 fim

```

Parâmetros e suas variações Conforme descrito no Algoritmo 3, os parâmetros para essa geração são: o σ_{range} do espaço de cor e o $\sigma_{spatial}$ do espaço de coordenadas. Esses parâmetros dependem das propriedades das imagens e dos resultados pretendidos. Quanto maior o σ_{range} , mais próximo da convolução Gaussiana e assim ocorre o borramento de intensidades mais distintas. Já o $\sigma_{spatial}$ controla o tamanho da vizinhança. Dessa forma, esses valores são escolhidos arbitrariamente para cada aplicação específica (TOMASI; MANDUCHI, 1998). Como o nosso objetivo com a geração das imagens não foi especializar no comportamento de uma classe de imagens, um valor foi escolhido aleatoriamente, e a partir dele os parâmetros de entrada foram definidos.



Figura 22 – Geração artificial utilizando *borramento* com filtro bilateral. A imagem (b) possui detalhes borrados, porém preservando as bordas.

Fonte: Elaborado pela autora.

Limitações Esse filtro tende a remover texturas e criar novos contornos. Dependendo dos valores, pode gerar uma imagem “cartoonizada”.

Métodos relacionados São diversos os métodos de borramento descritos na literatura, como a filtragem Gaussiana e as filtragens de medianas e médias.

4.2.2 Aguçamento

Diferente da suavização, o processamento de *aguçamento* procura enfatizar as transições de intensidade. Um método bem conhecido para atingir tal objetivo é o *unsharp mask*. Ele borra a imagem, subtrai a imagem borrada da original e adiciona essa diferença na imagem original, dado um peso k (ver Algoritmo 4). A imagem resultante, ilustrada na Figura 23, é uma versão realçada da imagem original. Isso porque adiciona à imagem justamente o que é removido com um filtro de suavização.

Algoritmo 4: Geração artificial: *aguçamento*

Entrada:

Imagen colorida $I_{original}$ em formato RGB

Saída:

Imagen aguçada $I_{aguçada}$

- 1 $I_{borrada} \leftarrow$ filtro de suavização($I_{original}$)
 - 2 **para cada pixel** (x, y) em $I_{original}$ **faça**
 - 3 $I_{diferença} \leftarrow I_{original}(x, y) - I_{borrada}(x, y);$
 - 4 $I_{aguçada}(x, y) \leftarrow I_{original}(x, y) + k \times I_{diferença};$
 - 5 **fim**
-



Figura 23 – Geração artificial utilizando *unsharp masking*. A imagem resultante (b) apresenta saliência nas transições de intensidade.

Fonte: Elaborado pela autora.

Parâmetros e suas variações Pode-se variar o parâmetro k de forma a ponderar a soma dessa diferença. Para a geração das imagens da classe minoritária, foi utilizado $k = 1$.

Limitações É possível que existam pixels com valor negativo no resultado final. Isso pode causar o aparecimento de uma áurea em volta das bordas, efeito não desejado (GONZALEZ; WOODS, 2007).

Métodos relacionados Outros algoritmos conhecidos para o aguçamento de imagens são: utilizar primeira derivada (gradiente), ou a segunda derivada da imagem (Laplaciano).

4.2.3 Adição de ruído

O ruído de Poisson ocorre na contagem de fótons de dispositivos ópticos. Ele segue a distribuição de Poisson, que representa o número de ocorrências de um evento em um dado instante de tempo (KNUTH, 1997). O efeito da *adição de ruído* pode ser visto na Figura 24.

Uma possível implementação para encontrar os valores de Poisson foi desenvolvida por Knuth (1997) e pode ser vista no Algoritmo 5. A diferença do cálculo da distribuição de Poisson para sua adição em uma imagem, é que para calcular o valor de ruído em um pixel, esse pixel é considerado a média dessa distribuição. Ou seja, $\mu \equiv I_{original}(x, y)$. Para intensidades próximas a zero, o limite será $L \approx 1$ e, portanto, a probabilidade p estará muito próxima do limite e o contador k terá um valor baixo. Dessa forma, para intensidades escuras haverá pouco ruído. Por outro lado, em intensidades próximas a 255, $p \approx 0$. Assim, são necessárias muitas interações até $p > L$ e maior será a intensidade resultante $k - 1$.

Parâmetros e suas variações Para a adição desse ruído em uma imagem, não é fornecido nenhum parâmetro. O ruído é calculado para cada pixel.



Figura 24 – Geração artificial utilizando *adição de ruído* de Poisson. Regiões claras de (b) apresentam mais ruído que as regiões escuras.

Fonte: Elaborado pela autora.

Algoritmo 5: Geração artificial: *adição de ruído* de Poisson

Entrada:

Imagen colorida $I_{original}$ em formato RGB

Saída:

Imagen gerada I_{gerada}

```

1 para cada canal de cor faça
2   para cada pixel  $(x,y)$  faça
3      $L \leftarrow \exp(-I_{original}(x,y));$ 
4      $p \leftarrow 1;$ 
5      $k \leftarrow 0;$ 
6     faça
7        $k \leftarrow k + 1;$ 
8        $p \leftarrow p \times \text{número aleatório uniforme entre } 0 \text{ e } 1;$ 
9     enquanto  $p > L;$ 
10     $I_{ruidosa}(x,y) \leftarrow k - 1;$ 
11  fim
12 fim

```

Limitações A adição de ruído é normalmente indesejável. Porém, a utilizamos para englobar um processamento de imagens que, de certa forma, se contrapõe ao *borramento*.

Métodos relacionados Esse método está relacionado com diversos outros ruídos, como o sal e pimenta, por exemplo.

4.2.4 SMOTE visual

Conforme visto na Seção 2.4.1, o SMOTE é um método de rebalanceamento aplicado após a extração de características. É proposta uma alternativa, chamada de *SMOTE visual*, onde imita-se o funcionamento do SMOTE, porém no nível de pixels. A diferença é que não é feito entre as imagens mais próximas, mas sim entre duas imagens escolhidas de forma aleatória do conjunto de treinamento da classe minoritária.

Para cada pixel é calculado a diferença entre as duas imagens. Essa diferença é então multiplicada por um número aleatório no intervalo $[0 - 1]$ e adicionada na imagem original (ver Algoritmo 6). O efeito que esse processamento causa na imagem pode ser visualizado na Figura 25.

Algoritmo 6: Geração artificial: *SMOTE visual*

Entrada:

Imagen colorida $I_{original}$ em formato RGB

Imagen colorida $I_{segunda_original}$ em formato RGB

Saída:

Imagen gerada I_{gerada}

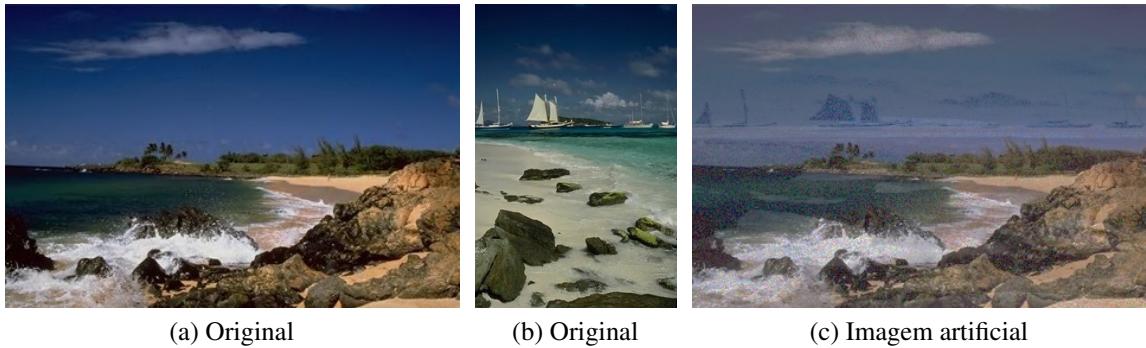
```

1 para cada pixel  $(x,y)$  faça
2   |   diferença  $\leftarrow I_{original}(x,y) - I_{segunda\_original}(x,y);$ 
3   |   gap  $\leftarrow$  número aleatório entre 0 e 1;
4   |    $I_{gerada}(x,y) \leftarrow I_{original}(x,y) + gap \times diferença;$ 
5 fim
6 mínimo  $\leftarrow$  menor valor( $I_{gerada}$ );
7 máximo  $\leftarrow$  maior valor( $I_{gerada}$ );
8 para cada pixel  $(x,y)$  faça
9   |    $I_{gerada}(x,y) \leftarrow I_{gerada}(x,y) - mínimo;$ 
10 fim
11 para cada pixel  $(x,y)$  faça
12   |    $I_{gerada}(x,y) \leftarrow I_{gerada}(x,y) \times (255/(máximo - mínimo));$ 
13 fim

```

Limitações Esse método adiciona texturas e bordas que não estavam originalmente nas imagens.

Métodos relacionados Esse método é visualmente parecido com o de *mistura ponderada*, apresentado na próxima seção.



(a) Original

(b) Original

(c) Imagem artificial

Figura 25 – Geração artificial utilizando o método SMOTE no espaço visual. É possível notar a sobreposição de uma “sombra” da Figura (b) em (a).

Fonte: Elaborado pela autora.

4.2.5 Mistura ponderada

Essa geração calcula a soma ponderada de duas imagens, de acordo com o Algoritmo 7. O efeito dessa mistura pode ser visto na Figura 26, onde dadas duas imagens como entrada, a imagem de saída corresponde à soma delas.

Algoritmo 7: Geração artificial: *mistura ponderada*

Entrada:

Primeira imagem colorida I em formato RGB

Segunda imagem colorida I_2 em formato RGB

Saída: Imagem gerada I_{gerada}

- 1 $\alpha \leftarrow$ número aleatório entre 10 e 90;
 - 2 $\beta \leftarrow 100 - \alpha;$
 - 3 para cada pixel (x,y) faça
 - 4 | $I_{gerada}(x,y) \leftarrow \beta I(x,y) + \alpha I_2(x,y);$
 - 5 fim
-



(a) Original

(b) Original

(c) Imagem artificial

Figura 26 – Geração artificial utilizando uma *mistura ponderada* de duas imagens. A imagem (c) representa a mistura de (a) e (b).

Fonte: Elaborado pela autora.

Parâmetros e suas variações Os parâmetros α e β são escolhidos de forma aleatória. Um valor entre 10% e 90% é escolhido para α ; e β equivale ao valor restante para completar 100%.

Limitações Assim como todas as gerações artificiais que envolvem a mistura de imagens, efeitos são adicionados às imagens originais. Dependendo da combinação dos métodos de descrição, quantização e classificação, isso pode piorar a acurácia da classificação.

Métodos relacionados É um método de combinação de imagens primitivo. Algoritmos similares são muito mais complexos, como os de limiares e saliência descritos a seguir.

4.2.6 Mistura limiarizada

A *mistura limiarizada* é uma composição do fundo (*background*) de uma imagem e do objeto da cena (*foreground*) de outra imagem. A Figura 27 mostra a mistura dos *thresholds* de duas imagens originais para compor uma nova imagem. O Algoritmo 8 descreve as operações necessárias para realizar tal processamento, que utiliza OTSU para encontrar o *foreground* da primeira imagem. Após, operações morfológicas são realizadas para melhorar a definição dos limiares.



Figura 27 – Geração artificial utilizando uma *mistura limiarizada* de duas imagens. A imagem resultante (c) é uma composição do *foreground* da primeira imagem sobre o *background* da segunda.

Fonte: Elaborado pela autora.

Parâmetros e suas variações No âmbito desta pesquisa, os parâmetros estão fixos, mas é possível modificar o tamanho dos elementos estruturantes que fazem as operações de abertura e dilatação para remover pequenas regiões.

Limitações Dependendo da quantidade de informações da imagem, o *threshold* de OTSU pode não conseguir extrair nenhuma informação relevante ou mesmo indicar a imagem inteira.

Métodos relacionados Essa geração está fortemente correlacionada com a *mistura* a partir da saliência da imagem, apresentada a seguir.

Algoritmo 8: Geração artificial: *mistura limiarizada*

Entrada:

Imagen colorida I em formato RGB

Imagen colorida I_2 em formato RGB

Saída: Imagem gerada I_{gerada}

- 1 $I_{cinza} \leftarrow escala\ de\ cinza(I);$
 - 2 $I_{threshold} \leftarrow OTSU(I_{cinza});$
 - 3 $I_{morfologica} \leftarrow abertura\ e\ dilatação(I_{threshold});$
 - 4 $I_{foreground} \leftarrow aplica\ máscara(I_{morfologica}, I);$
 - 5 $I_{morfologica} \leftarrow oposto(I_{morfologica});$
 - 6 $I_{background} \leftarrow aplica\ máscara(I_{morfologica}, I_2);$
 - 7 $I_{gerada} \leftarrow I_{background} + I_{foreground};$
-

4.2.7 Mistura saliente

A combinação de regiões salientes é muito similar com o método anterior de *mistura limiarizada*, porém, utiliza um algoritmo mais rebuscado que detecta o mapa de saliência da imagem baseado no método *Graph-Based Manifold Ranking* (YANG *et al.*, 2013). A Figura 28 mostra a combinação da região saliente da imagem original à esquerda com a imagem central, resultando na imagem combinada à direita.



Figura 28 – Geração artificial utilizando a *mistura saliente* de duas imagens. A imagem resultante (c) apresenta a região saliente de (b) sobreposta em (a).

Fonte: Elaborado pela autora.

As operações aplicadas na imagem para extrair a região mais saliente são: segmentação pelo método *Simple Linear Iterative Clustering* (SLIC); rotulação por conectividade; *threshold*

de OTSU; e operações morfológicas. O Algoritmo 9 apresenta os passos para o cálculo do *background* e *foreground*.

Algoritmo 9: Geração artificial: *mistura saliente*

Entrada:

Imagen colorida I em formato RGB

Imagen colorida I_2 em formato RGB

Saída: Imagen gerada I_{gerada}

- 1 $I_{rotulada \ por \ segmento} \leftarrow SLIC(I);$
 - 2 $I_{mapa \ de \ saliência} \leftarrow rotulação \ por \ conectividade(I_{rotulada \ por \ segmento});$
 - 3 $I_{threshold} \leftarrow OTSU(I_{mapa \ de \ saliência});$
 - 4 $I_{morfologica} \leftarrow abertura \ e \ dilatação(I_{threshold});$
 - 5 $I_{foreground} \leftarrow aplica \ máscara(I_{morfologica}, I);$
 - 6 $I_{morfologica} \leftarrow oposto(I_{morfologica});$
 - 7 $I_{background} \leftarrow aplica \ máscara(I_{morfologica}, I_2);$
 - 8 $I_{gerada} \leftarrow I_{background} + I_{foreground};$
-

Parâmetros e suas variações Assim como no método anterior, os parâmetros são relacionados ao tamanho do elemento estruturante para a abertura e dilatação e são fixos para os experimentos desta pesquisa.

Limitações Não é garantido que o algoritmo de saliência consiga extrair a melhor região, ou mesmo que sempre haja uma região saliente.

Métodos relacionados Assemelha-se à mistura por *thresholds*.

4.2.8 Composição

Essa geração pretende compor informações de diversas imagens em uma única. Assim é feito um mosaico com várias imagens, conforme pode ser visto na Figura 29. Para cada quadrado a ser preenchido, sorteia uma imagem do conjunto de treinamento; realiza uma operação de *borramento*, *aguçamento*, *mistura ponderada* ou *SMOTE visual*; e adiciona essa imagem no quadrado respectivo. Os passos para tal *composição* estão descritos no Algoritmo 10.

Parâmetros e suas variações O parâmetro q controla quantos quadrados são criados na imagem resultante. Nesta pesquisa, foram realizados experimentos com 4 e 16 quadrados.

Limitações O término brusco de uma imagem para início da outra, ao formar a grade de imagens, tem efeitos colaterais de inserção de textura que, em alguns casos, não excedam a vantagem de compor uma mesma imagem com várias cores, texturas e formas das imagens originais.



Figura 29 – Geração artificial utilizando uma *composição* de imagens. Várias imagens, dispostas em um mosaico, formam a imagem resultante. Cada célula do mosaico sofre uma operação, sorteada no momento da geração da imagem.

Fonte: Elaborado pela autora.

Algoritmo 10: Geração artificial: *composição*

Saída: Imagem gerada I_{gerada}

- 1 **enquanto** $total < \text{número de quadrados } q$ **faça**
- 2 $I \leftarrow \text{imagem aleatória do conjunto de treinamento};$
- 3 $\text{operação} \leftarrow \text{número aleatório entre } 1 \text{ e } 4;$
- 4 **seleçãone** operação **faça**
- 5 **caso** 1
- 6 $I \leftarrow \text{borramento}(I);$
- 7 **caso** 2
- 8 $I \leftarrow \text{mistura ponderada}(I);$
- 9 **caso** 3
- 10 $I \leftarrow \text{aguçamento}(I);$
- 11 **caso** 4
- 12 $I \leftarrow \text{SMOTE visual}(I);$
- 13 **fim**
- 14 $x \leftarrow \text{posição aleatória em } x \text{ de } I;$
- 15 $y \leftarrow \text{posição aleatória em } y \text{ de } I;$
- 16 $qx \leftarrow \text{posição atual para o quadrado em } x \text{ de } I_{gerada};$
- 17 $qy \leftarrow \text{posição atual para o quadrado em } y \text{ de } I_{gerada};$
- 18 $I_{gerada}(qx, qy) \leftarrow I(x, y);$
- 19 $total ++;$
- 20 **fim**

Métodos relacionados Fazer uma composição de imagens em quadrantes pode estar relacionado com a composição ao utilizar saliência.

4.3 Experimentos

Esta seção descreve os resultados encontrados ao rebalancear as classes de imagens aplicando os processamentos — descritos na seção anterior — nas imagens originais. As imagens geradas são então utilizadas como treinamento da classe minoritária. A Figura 30 destaca o fluxo de operações realizadas para a análise do impacto da geração de imagens no rebalanceamento de classes. O mesmo protocolo de conversão para escala de cinza, extração de características e classificação foi seguido para três sub-experimentos: base desbalanceada; base rebalanceada com interpolação dos vetores de características (método SMOTE); e base rebalanceada com a geração artificial de imagens. Alguns experimentos foram realizados com bases de imagens originalmente balanceadas. Para tais casos, foi necessário desbalancear a base para testar o rebalanceamento.

Procurando estabilidade dos resultados obtidos com a geração das imagens artificiais, foi identificada a necessidade de controlar a remoção de imagens da base no momento da criação da base desbalanceada. Assim, os resultados foram obtidos a partir de uma forma de validação *k-fold* com o objetivo de prover mais robustez ao sistema. A Figura 31 ilustra como tal validação é realizada, utilizando como exemplo uma base com duas classes de imagens. Primeiramente as imagens são separadas de forma aleatória em $k = 5$ folds em cada classe. Em seguida, as duas classes compõem 40 configurações, consistindo em todas as possibilidades de: um fold para teste e os outros como treino para a classe que permanecerá balanceada; e um de teste e apenas um de treino para a classe que os métodos de processamento irão rebalancear. Tal validação é repetida para todas as classes, ou seja, cada classe tem a possibilidade de ser a minoritária. Se originalmente a base é naturalmente desbalanceada, um fold é utilizado para teste e os demais como treino para todas as classes.

A acurácia pode não refletir propriamente os resultados em um cenário de bases desbalanceadas. Isso se deve ao fato de que se a classe minoritária não obtiver nenhum resultado correto e a classe majoritária tiver 100% de acertos, tal acurácia poderá ser muito alta, mesmo considerando que nenhuma imagem da classe minoritária foi corretamente classificada. Dessa forma, considera que os erros são igualmente importantes. Pode-se estender essa medida obtendo-se a acurácia balanceada: medida de acerto baseada na divisão do conjunto de objetos em teste e treinamento, realizando a repetição dos experimentos n vezes e obtendo a média e o desvio padrão. A acurácia de cada experimento é obtida por

$$Acc = 1 - \frac{\sum_{i=1}^c E(i)}{2c},$$

que considera problemas de desbalanceamento de classes, onde c é o número de classes e

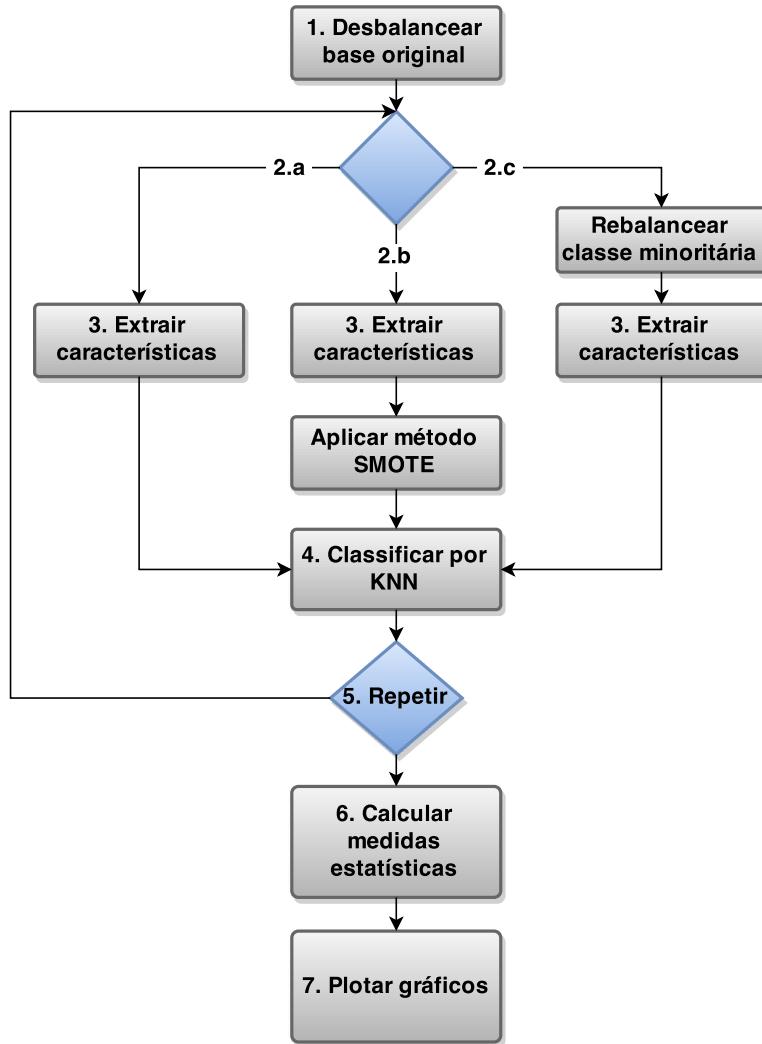


Figura 30 – Fluxo de operações para obtenção dos resultados do rebalanceamento de classes. O mesmo protocolo de conversão para escala de cinza, extração de características e classificação foi seguido para três sub-experimentos: base desbalanceada; base rebalanceada com interpolação dos vetores de características (método SMOTE); e base rebalanceada com a geração artificial de imagens.

Fonte: Elaborado pela autora.

$E(i) = e_{i,1} + e_{i,2}$ é o erro relativo a c , calculado por

$$e_{i,1} = \frac{FP(i)}{N - N(i)} \quad \text{e} \quad e_{i,2} = \frac{FN(i)}{N(i)}, \quad i = 1, \dots, c,$$

onde $FN(i)$ são os exemplos pertencentes à classe i e incorretamente classificados (falsos negativos), e $FP(i)$ são os exemplos erroneamente rotulados como i (falsos positivos). Os experimentos consideraram a classe minoritária como positiva e a minoritária como negativa.

Uma outra medida, que pode efetivamente avaliar o desempenho da classificação em cenários desbalanceados, é o *F1-Score*:

$$F1 = 2 \frac{PR}{P + R}.$$

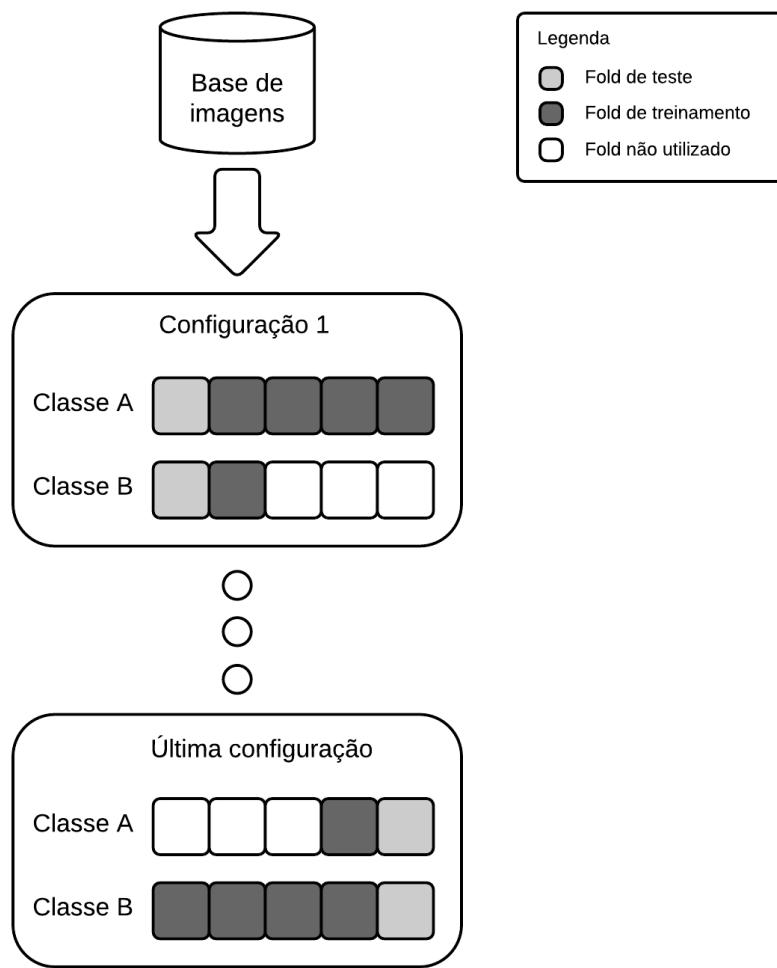


Figura 31 – Ilustração de como os experimentos de geração de imagens artificiais foram realizados. Primeiramente as imagens são separadas de forma aleatória em $k = 5$ folds em cada classe. Em seguida, as duas classes compõem 40 configurações, consistindo em todas as possibilidades de: um fold para teste e os outros como treino para a classe que permanecerá balanceada; e um de teste e apenas um de treino para a classe que os métodos de processamento irão rebalancear. Tal validação é repetida para todas as classes, ou seja, cada classe tem a possibilidade de ser a minoritária.

Fonte: Elaborado pela autora.

Essa medida combina precisão e revocação como medida de efetividade da classificação ([GARCIA, 2009](#)). A precisão é a medida da exatidão:

$$P = \frac{VP}{VP + FP},$$

onde VP são os exemplos positivos corretamente classificados. Dos exemplos classificados como positivos, essa medida indica quantos realmente são. Ao mesmo tempo, a revocação é a medida de completude. Essa métrica indica quantos exemplos positivos foram corretamente classificados, sendo determinada por:

$$R = \frac{VP}{VP+FN}.$$

A partir dessas medidas estatísticas, o teste *Honest Significant Difference* (HSD) de Tukey pode ser utilizado para determinar se há diferença significativa em uma amostra de resultados gerados. A hipótese nula a ser testada por estes experimentos é que não há diferença estatística relevante entre as observações de *F1-Scores*. Para analisar se o teste da hipótese é significativo, pode ser utilizado o *p-value*, que indica o quanto estatisticamente significante o resultado é: quanto menor o seu valor, maior a evidência contra a hipótese nula (utiliza-se um limiar de 0,05).

A seguir, para cada experimento realizado são descritos: a base de imagens utilizada; o protocolo e parâmetros adotados; e por fim os resultados obtidos a partir de seu uso são mostrados e discutidos.

4.3.1 Experimento 1: duas classes discriminadas

Neste experimento foram utilizadas duas classes de fácil diferenciação, porém com alguma sobreposição. Um sub-experimento de visualização foi realizado para análise do espaço de características.

Protocolo

1. **Imagens originais:** classes *Horse* e *Elephant* da base de imagens Corel-1000, exemplificadas na Figura 32 ([WANG; WIEDERHOLD, 2001](#)). A principal característica dessas imagens é a diferença de cores, contendo pequeno grau de sobreposição.



Figura 32 – Classes *Horse* e *Elephant* utilizadas no experimento. São duas classes bem discriminadas com 100 imagens cada, originalmente da base de imagens Corel-1000.

Fonte: [Hu e Collomosse \(2013\)](#).

2. **Desbalanceamento:** para o sub-experimento de visualização, cada classe foi dividida em 50% para treino e 50% para teste, de maneira aleatória. Após, a classe *Horse* sofreu

remoção de 50% do seu conjunto de treino, tornando-a desbalanceada. Já para a análise estatística do experimento, todas as 40 configurações de *folds* com $k = 5$ foram realizadas (padronização anteriormente descrita na Figura 31).

3. **Método para geração artificial:** para a visualização do espaço de características foi utilizado o método de *mistura* de duas imagens originais, exemplificado na Figura 33. Para a análise do *boxplot* de *F1-Scores*, todas as gerações foram testadas e os resultados são reportados a seguir.



Figura 33 – Exemplo da geração artificial de imagens com o método de *mistura* para as classes *Elephant* e *Horse* da base Corel-1000. A imagem resultante (c) é composta pela mistura de (a) e (b).

Fonte: Elaborado pela autora.

4. **Conversão em escala de cinza:** método *Intensidade'* para a visualização. Todas as combinações de extração e conversão em escala de cinza foram testadas, portanto todos os métodos de conversão foram utilizados.
5. **Extração de características:** classificação de pixels de borda e interior (BIC) para a visualização. Todos os métodos de extração foram testados para a análise estatística.
6. **Classificação:** o classificador supervisionado KNN com $K = 1$ (para mais detalhes ver Seção 2.6.1) foi utilizado.
7. **Projeção multidimensional para visualização:** dois componentes principais encontrados ao aplicar PCA (Seção 2.5.1) nos vetores de características para redução de dimensionalidade foram projetados.

Visualização

A visualização do espaço de características obtido após a geração artificial de imagens pode ajudar a verificar se as novas imagens melhoram a definição da classe minoritária em relação ao espaço original. Ou seja, se o método utilizado revelou características latentes. Dessa forma, ao projetar os novos vetores no espaço das imagens originais, é possível analisar qual método — SMOTE ou geração de imagens no campo visual — mais se assemelha à distribuição original dos dados.

As classes *Elephant* e *Horse* possuem 100 imagens cada. O primeiro passo foi remover imagens de uma das classes, tornando a base desbalanceada. Na Figura 34 está ilustrada a remoção de 50% das imagens de treino da classe *Horse*, originalmente balanceada. Essa e as próximas projeções desta seção foram obtidas com a técnica para redução de dimensionalidade PCA, descrita na Seção 2.5.1, e são referentes aos dois componentes principais com maiores autovalores.

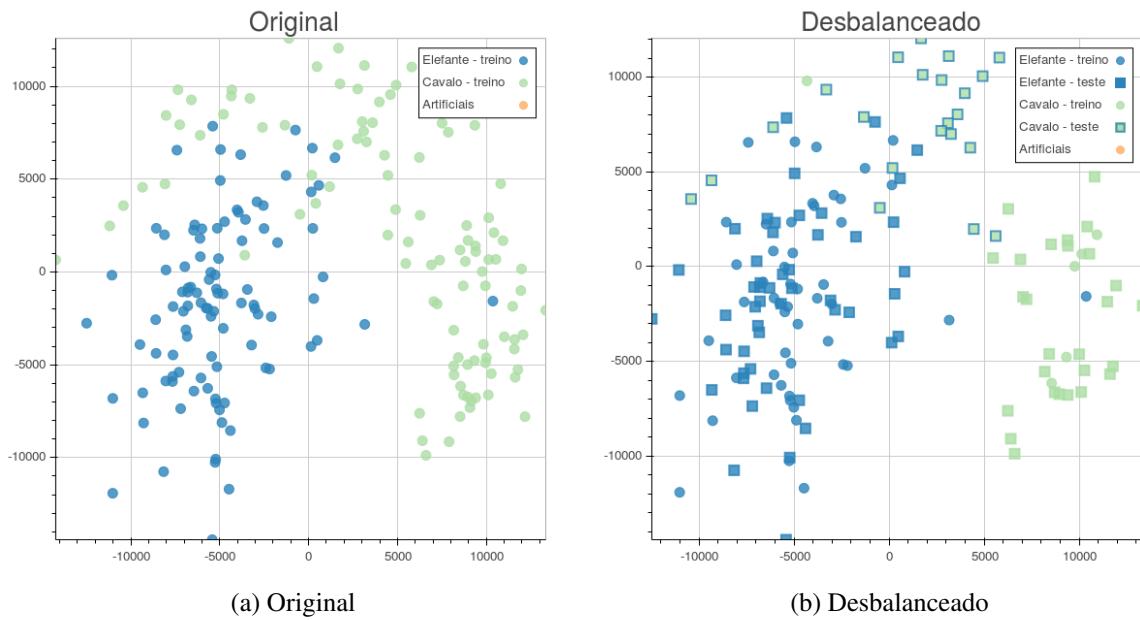


Figura 34 – À esquerda a projeção dos dois componentes principais obtidos com a aplicação de PCA nas classes *Elephant* — em azul — e *Horse* — em verde. À direita, as mesmas classes após a remoção de 50% das imagens de treino da classe *Horse*. A diferença dos marcadores consiste na definição de imagens para treino e teste não existente nas classes originais.

Fonte: Elaborado pela autora.

Os resultados da classificação dos três experimentos (desbalanceado, SMOTE e geração artificial) utilizando K-NN com $K = 1$ reportou que o *F1-Score* da geração de imagens utilizando o método de *mistura* obteve um ganho de mais de 10% em relação ao rebalanceamento no espaço de características com o SMOTE. Foi utilizado BIC como método de extração de características e *Intensidade'* como método de conversão em escala de cinza. Para essa combinação, a geração de imagens utilizando *mistura* se mostrou favorável e portanto a visualização do espaço de características apresenta esse método como geração.

Para verificar se a geração de imagens inseriu mais informação na classe minoritária do que apenas povoar os espaços entre os exemplos, a classe rebalanceada utilizando ambos métodos está demonstrada na Figura 35. Em laranja estão representados os novos exemplos de treinamento, projetados no plano da base original balanceada.

Após o treinamento realizado com as novas imagens geradas e as originais, o conjunto de teste foi fornecido ao classificador K-NN com $K = 1$ e o resultado das predições está ilustrado

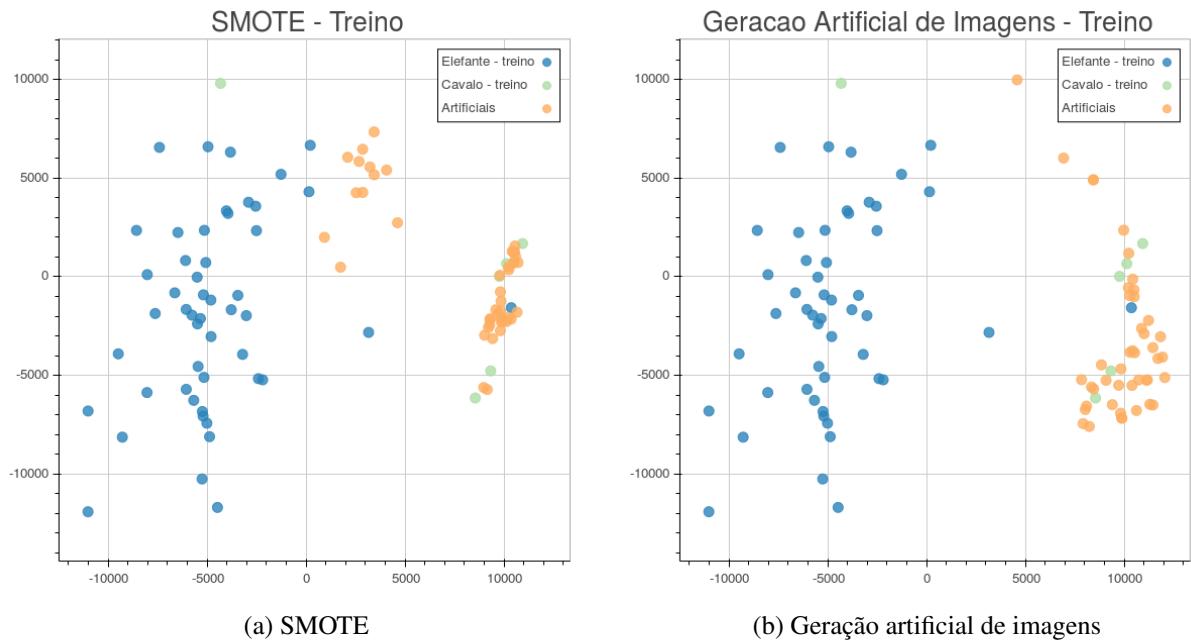


Figura 35 – Comparação dos exemplos de treinamento da geração com SMOTE e no campo visual. Em laranja estão representados os novos exemplos, projetados no plano da base original balanceada.

Fonte: Elaborado pela autora.

na Figura 36. A cor no interior dos marcadores quadrados representa a classe real dos exemplos e a borda representa a classe predita pelo classificador. Nota-se que a melhoria na classificação com a geração de imagens fica visível e corresponde ao aumento do *F1-Score*.

De uma forma geral, pode-se dizer que a geração de imagens melhorou a definição da classe minoritária e foi o método que mais se assemelhou à distribuição dos dados originais. Além disso, um dos problemas do SMOTE pode ser verificado nessas projeções: **ao realizar a interpolação dos vetores de características originais, exemplos podem ser criados em regiões do espaço que fazem parte da outra classe**. Ficou claro também que o método SMOTE não apresentou capacidade de extrapolar a sua região, como pode ser observado no grupo de exemplos à direita do espaço de características. O SMOTE gerou novos elementos próximos a uma linha reta, enquanto a geração de imagens proporcionou uma abrangência maior em volta desse espaço, com maior dispersão.

Na Figura 37 é possível visualizar a região de decisão, observando suas modificações frente aos métodos. Pode ser observado que, em ambas técnicas, a região da classe minoritária apresenta-se melhor representada. Além disso, é possível verificar que o SMOTE ocasionou uma certa invasão do espaço de características da classe majoritária.

Em todas as figuras anteriores relacionadas a essa visualização, os exemplos foram projetados no plano criado pelas suas componentes principais com maior autovalores da base original balanceada. Se após a geração de novos exemplos essas componentes forem recalculadas (Figura 38), pode-se notar que a geração de imagens artificiais proporciona a criação de um

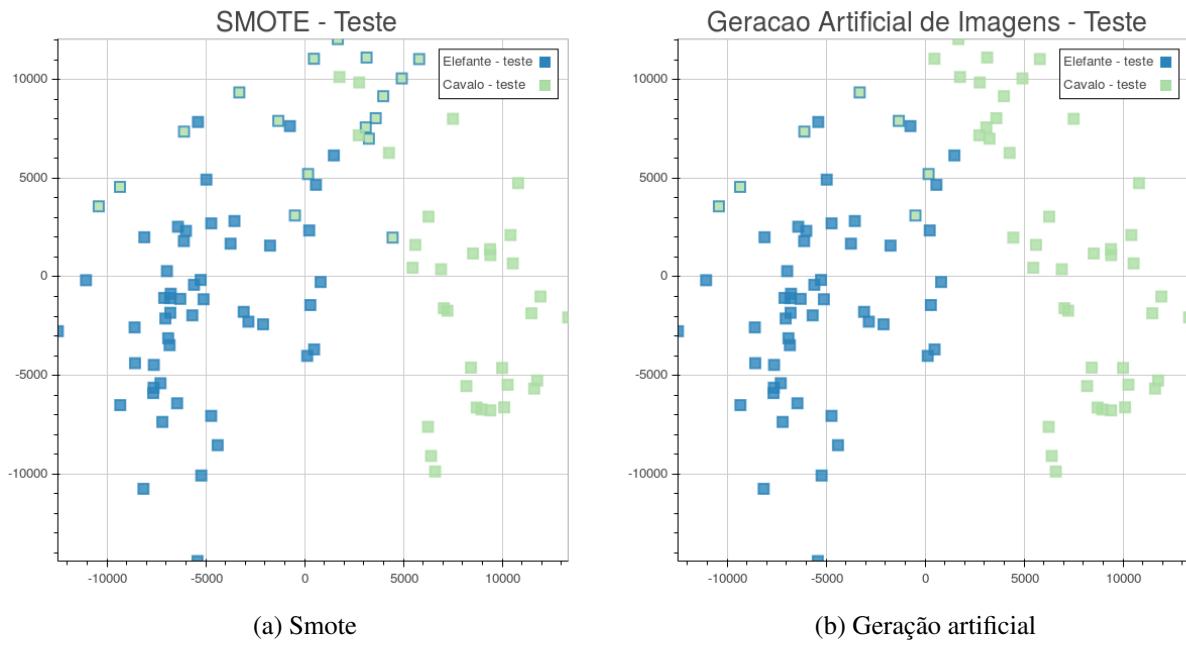


Figura 36 – Resultado do teste da classificação com K-NN com $K = 1$ após o treinamento realizado com as bases rebalanceadas. A cor no interior dos marcadores quadrados representa a classe real dos exemplos e a borda representa a classe predita pelo classificador.

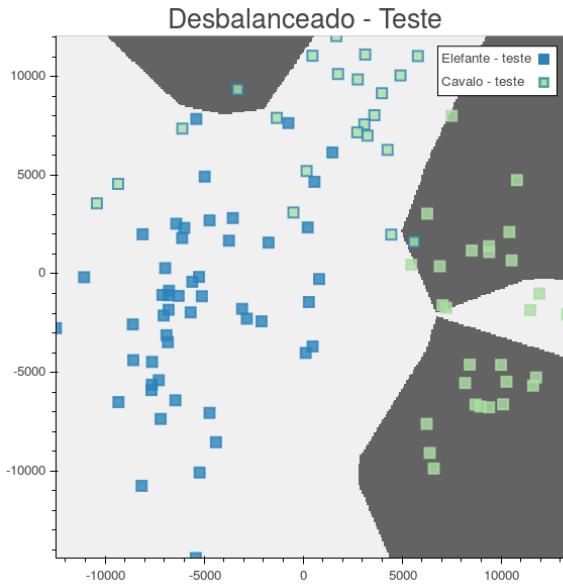
Fonte: Elaborado pela autora.

subespaço que melhor discretiza as classes, quando comparado com SMOTE ou com a base desbalanceada.

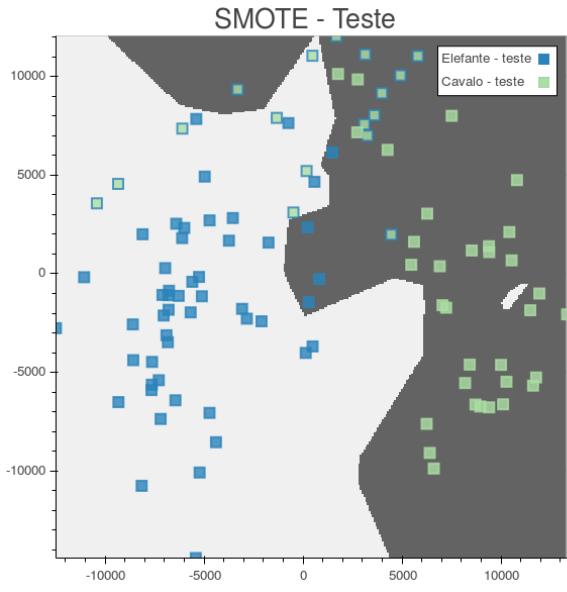
Como relatado no início desse experimento, o extrator de características utilizado foi o BIC. Fundamentalmente ele captura informações de intensidade de cor das imagens. Na Figura 39 as próprias imagens foram utilizadas como marcadores na projeção do melhor subespaço após a geração artificial com o método de *mistura*. É nítido o impacto da etapa de extração de características na separação das classes e também no método de geração de imagens antes de tal extração.

Resultados

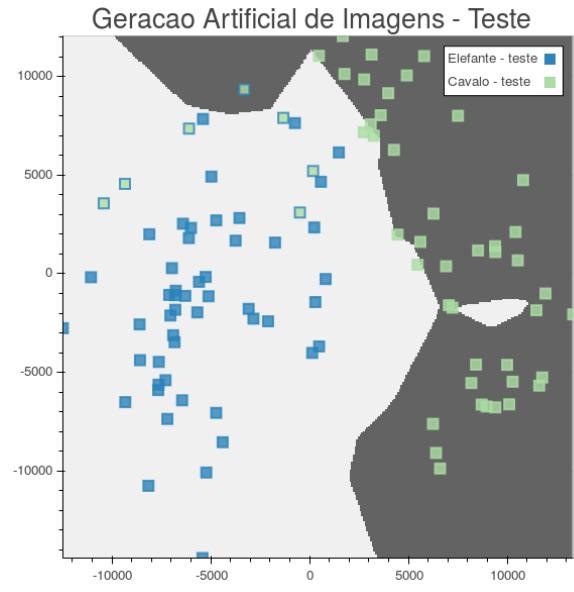
Para análise estatística, todas as combinações de conversão para escala de cinza e métodos de extração de características foram testados. A combinação *Gleam* e ACC obteve o melhor *F1-Score* para as classes *Elephant* e *Horse*. O boxplot apresentado na Figura 40 retrata a média dos *F1-Scores* das 40 combinações deste experimento. A Tabela 1 mostra os valores de tal métrica para o cálculo dos testes estatísticos. Como pode ser observado, o método de *composição*, exemplificado na Figura 41, obteve o melhor *F1-Score*. Como reportado anteriormente para o método BIC, a melhor geração para essas classes foi o método de *mistura*. Interessante notar que, ao menos para este caso, a preferência pelo método de geração parece estar relacionada ao método de extração. Isso se deve, possivelmente, ao fato de o melhor método de extração indicar



(a) Desbalanceado



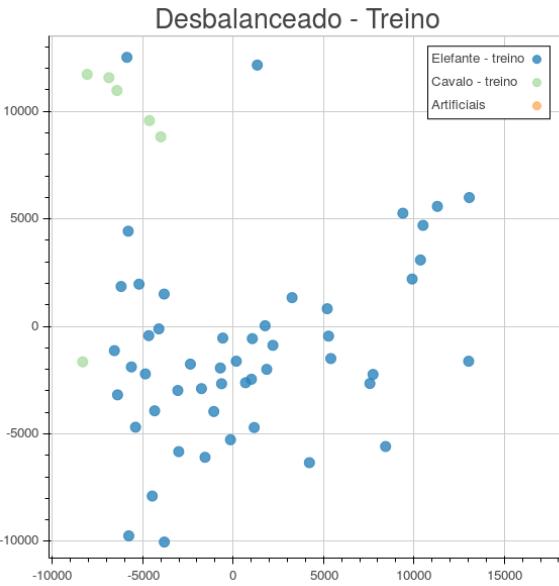
(b) Smote



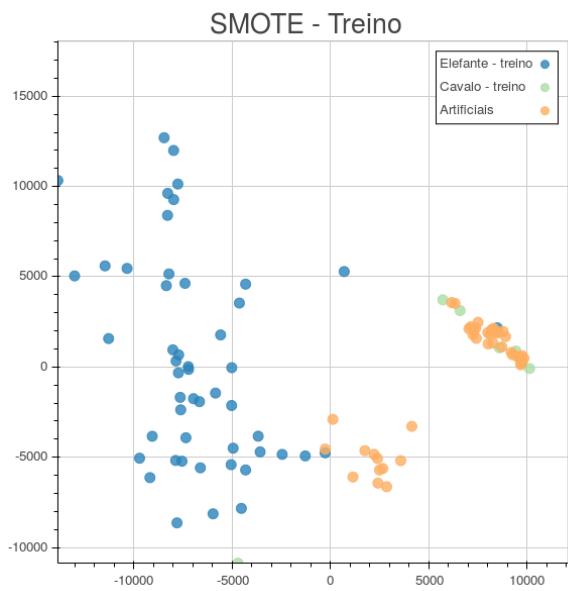
(c) Geração artificial

Figura 37 – Região de decisão com K-NN ($K = 1$). Pode ser observado que em ambas técnicas a região da classe minoritária apresenta-se melhor representada. Além disso, é possível verificar que o SMOTE ocasionou uma certa invasão do espaço de características da classe majoritária.

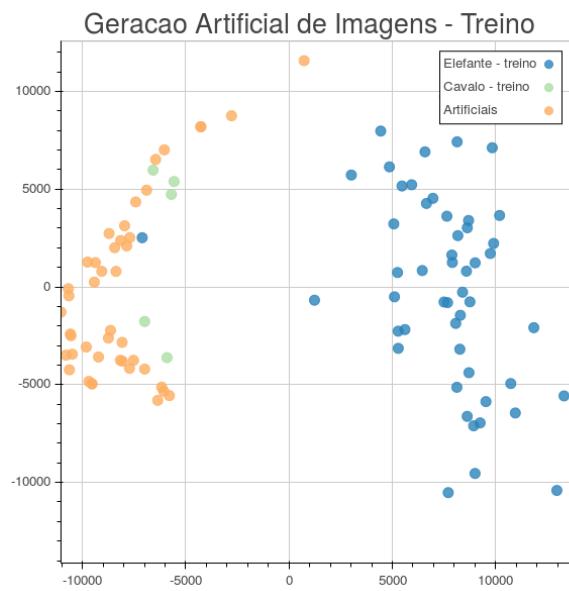
Fonte: Elaborado pela autora.



(a) Desbalanceado



(b) Smote



(c) Geração artificial

Figura 38 – Melhores subespaços encontrados após a geração de novos exemplos para o SMOTE, para a geração artificial de imagens, e após a remoção de imagens para a projeção dos dados desbalanceados. Pode-se notar que a geração de imagens artificiais proporciona a criação de um subespaço que melhor discretiza as classes, quando comparado com SMOTE ou com a base desbalanceada.

Fonte: Elaborado pela autora.

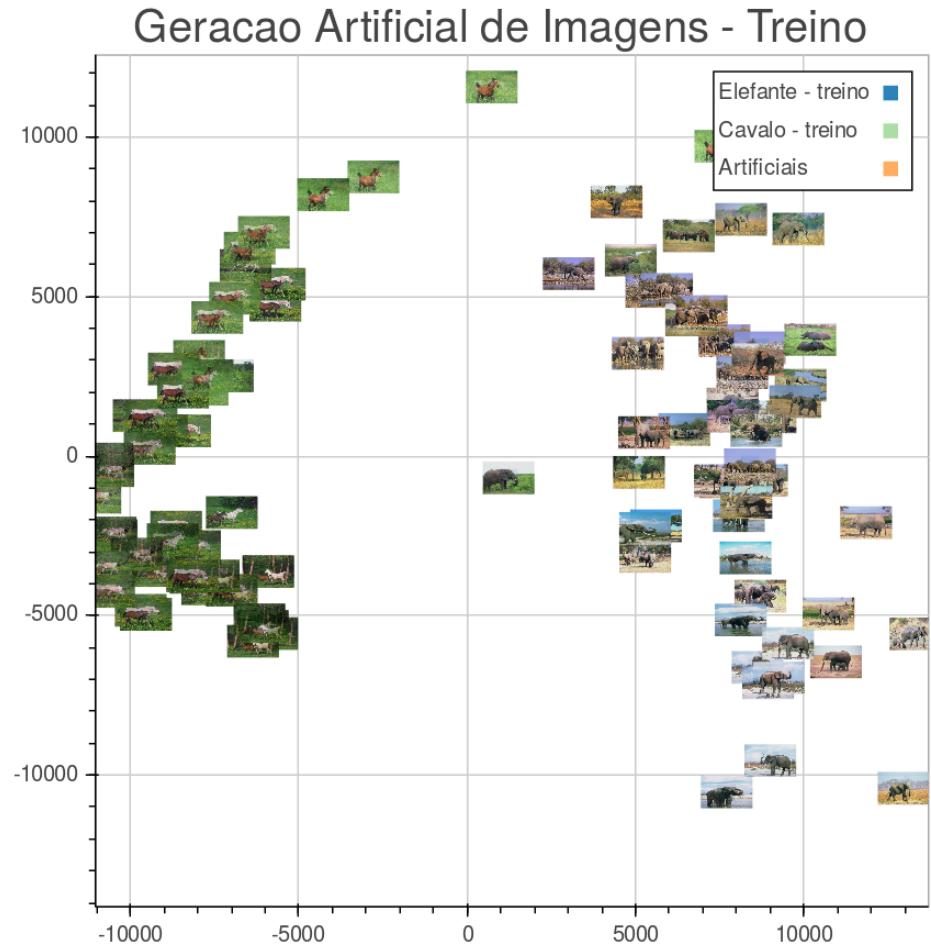


Figura 39 – Visualização do impacto do método de extração de características na separação entre classes. Possível verificar que o BIC utiliza as intensidades como principal representação de uma imagem.

Fonte: Elaborado pela autora.

as características mais relevantes das imagens.

Considerando a análise da melhor combinação dos métodos de representação da imagem, foi verificado também o desempenho dos rebalanceamentos em um cenário mais complexo: o de maior variância dos *F1-Scores*, dadas as 40 combinações da validação. A Figura 42 mostra o gráfico de *boxplot* referente aos resultados da Tabela 2. O melhor método de rebalanceamento para tal cenário foi o de geração artificial de imagens de *adição de ruído*, exemplificado na Figura 43. O par de métodos MSB e HOG obteve maior valor de variância — para conversão de escala de cinza e extração de características, respectivamente.

Discussão

Para o resultado da combinação dos melhores métodos de conversão em escala de cinza e extração de características, o teste *post-hoc* HSD de Tukey revelou que não há diferença estatística entre a base desbalanceada e o SMOTE (*p-value* = 0.2073). Mas indicou que existe uma significância entre o desbalanceamento e a geração artificial (*p-value* = 0.0062). Isso

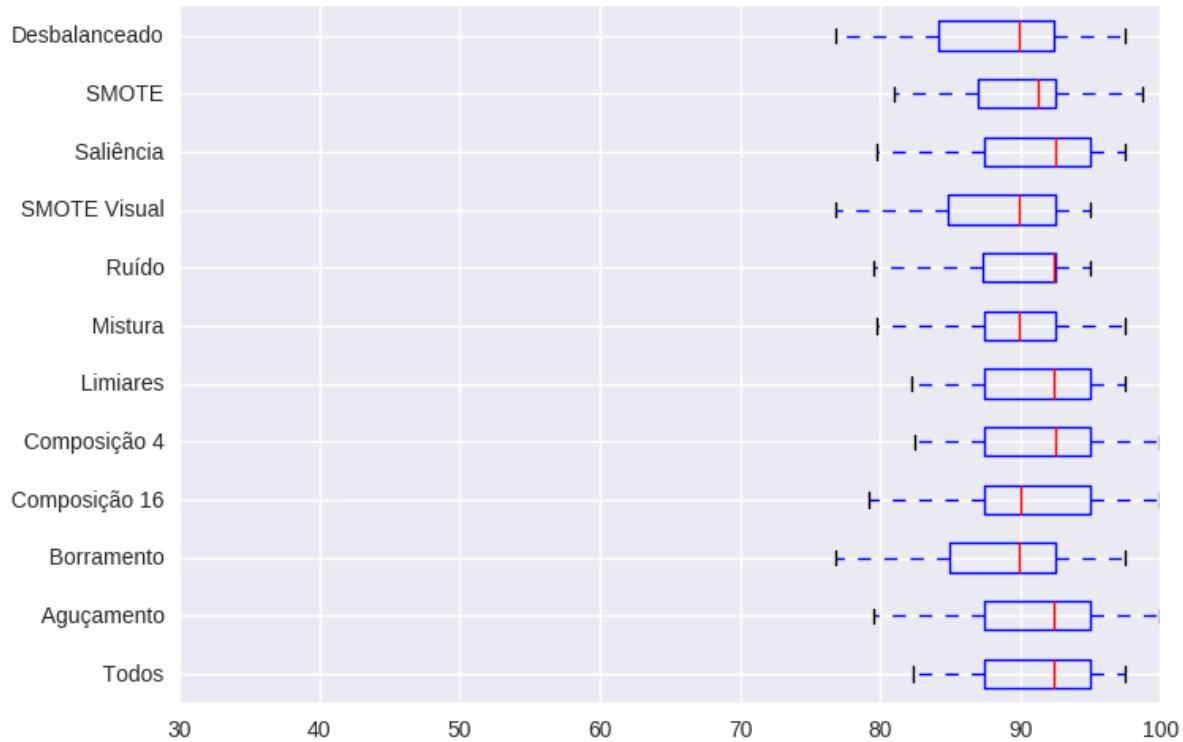


Figura 40 – Conversão em escala de cinza com *Gleam* e ACC como método de extração de características. Nota-se que o método de geração baseado em Composição 4 obteve maior valor de *F1-Score*.

Fonte: Elaborado pela autora.

Tabela 1 – Resultados de *F1-Score* para as classes *Horse* e *Elephant*, utilizando *Gleam* como método para conversão em escala de cinza e ACC para extração de características. Nota-se que o método de geração baseado em Composição 4 obteve maior valor de *F1-Score*.

<i>Gleam & ACC</i>	Média	Desvio Padrão
Todos	91.090913	4.559066
Aguçamento	91.002678	4.907016
Borramento	89.394500	5.103498
Composição 16	90.934305	4.399334
Composição 4	91.773528	4.909852
Limiares	90.893133	5.285833
Mistura	90.177055	4.409787
Ruído	89.337770	5.169757
SMOTE Visual	88.616535	5.567976
Saliência	91.282655	4.230281
SMOTE	90.173808	4.566863
Desbalanceado	88.258567	5.538461



(a) Imagem gerada

Figura 41 – A imagem gerada apresenta uma *composição* de quatro imagens da classe *Elephant*.
Fonte: Elaborado pela autora.

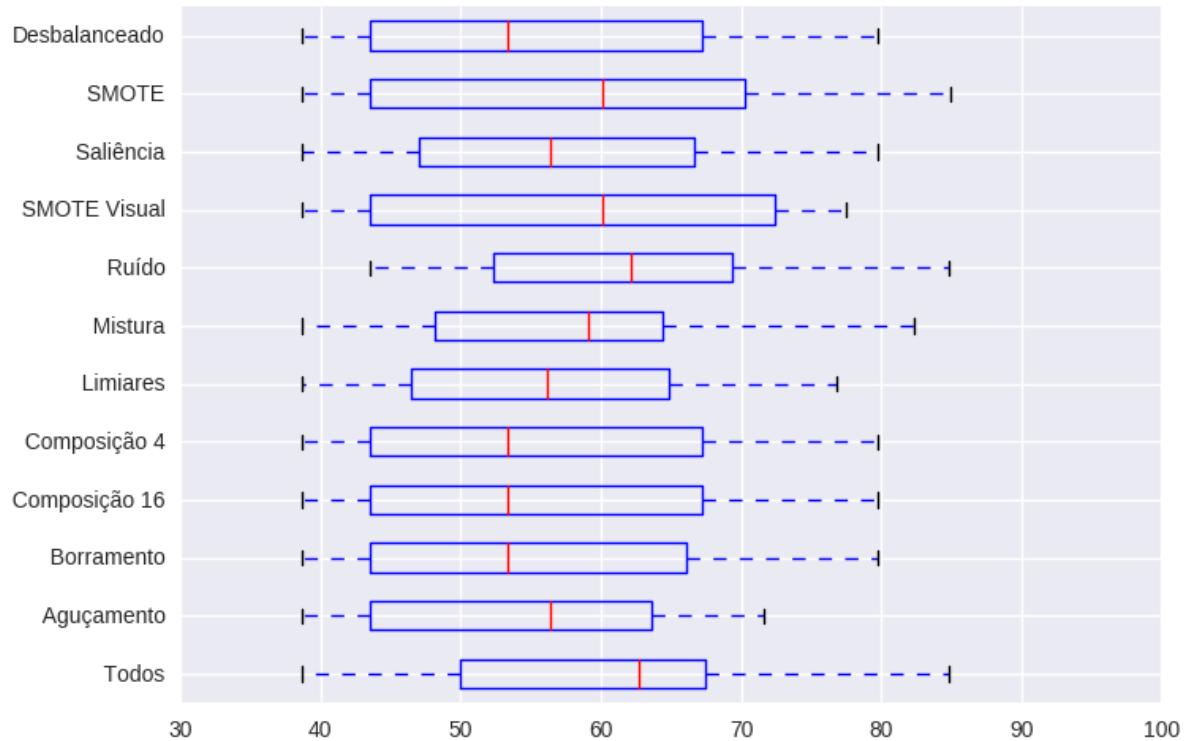


Figura 42 – Conversão em escala de cinza com MSB e HOG como método de extração de características. Essa combinação de métodos obteve a maior variância de *F1-Score*. Nota-se que a *adição de ruído* apresentou-se como o melhor método.

Fonte: Elaborado pela autora.

Tabela 2 – Resultados de *F1-Score* para as classes *Horse* e *Elephant*, utilizando MSB como método para conversão em escala de cinza e HOG para extração de características. O método de *adição de ruído* foi aquele que obteve melhor valor de *F1-Score*.

MSB & HOG	Média	Desvio Padrão
Todos	60.000127	12.063967
Aguçamento	54.809555	10.610213
Borramento	55.588173	13.275734
Composição 16	55.667145	13.341421
Composição 4	55.652205	13.323408
Limiares	55.652268	11.547820
Mistura	57.826535	10.882912
Ruído	62.174910	10.746760
SMOTE Visual	58.920085	14.765860
Saliência	56.322367	12.169296
SMOTE	58.342450	13.768688
Desbalanceado	55.667145	13.341421



Figura 43 – Imagem gerada utilizando *adição de ruído* em imagens da classe *Horse*. Regiões mais claras apresentam maior ruído na imagem resultante (b).

Fonte: Elaborado pela autora.

significa que o melhor método para rebalancear essas classes é a geração artificial utilizando o método de *misturas* de duas imagens. Ainda de acordo com o teste, não há evidência estatística da relevância do resultado da combinação de maior variância. Portanto, todos os próximos experimentos relatam apenas os resultados da melhor combinação.

4.3.2 Experimento 2: duas classes sobrepostas

O experimento anterior considerou classes relativamente distintas. Por isso, classes de difícil diferenciação também foram testadas e são reportadas a seguir, com o objetivo de verificar o desempenho do método.

Protocolo

- 1. Imagens originais:** as classes *Beach* e *Mountain* foram escolhidas por serem as classes que possuem maior dificuldade de diferenciação da base Corel-1000 ([WANG; WIEDERHOLD, 2001](#)). Apresentam alta taxa de sobreposição de intensidades de cores e texturas, conforme testes realizados. Uma imagem de exemplo de cada classe é apresentada na Figura 44.



(a) Mountain



(b) Beach

Figura 44 – Imagens representativas das classes *Beach* e *Mountain* da base de imagens Corel-1000. Tais classes apresentam sobreposição de características por ambas apresentarem paisagens naturais. O objetivo desse experimento é verificar o desempenho da geração artificial em um problema de difícil classificação.

Fonte: Wang e Wiederhold (2001).

- 2. Desbalanceamento:** as duas classes contém 100 imagens cada, portanto são balanceadas. Para esse experimento, foram utilizadas as 40 combinações de desbalanceamento dos $k = 5$ folds.
- 3. Método para geração artificial:** todos os métodos de geração foram testados. Os que obtiveram melhores resultados foram os métodos de: *mistura saliente*, *aguçamento* e a utilização de todos os tipos de gerações. A Figura 45 exemplifica uma geração artificial deste experimento, utilizando o método de saliência.
- 4. Conversão em escala de cinza:** todos os métodos de conversão em escala de cinza foram testados. O método *Luma* resultou em melhor *F1-Score*.
- 5. Extração de características:** todos os métodos para extração foram testados. O método CCV destacou-se como o melhor método para extrair as características que melhor diferenciam essas classes.
- 6. Classificação:** o classificador KNN com $K = 1$ foi utilizado.

Resultados

A combinação dos métodos *Luma* e CCV resultou em melhores valores de *F1-Score*. Como pode ser visto na Tabela 3, diversos métodos de geração artificial de imagens obtiveram



Figura 45 – Geração artificial utilizando o método de *saliência* em duas imagens da classe *Beach* da base de imagens Corel-1000. Detalhes salientes da imagem (b) aparecem sobrepostos à imagem (a) na imagem resultante (c).

Fonte: Elaborado pela autora.

resultados melhores que o SMOTE, e na sua maioria maiores do que a base desbalanceada. Destacam-se os métodos de *mistura saliente*, *aguçamento* e a combinação de todos os métodos.

Tabela 3 – Resultados de *F1-Score* para as classes *Beach* e *Mountain*, utilizando *Luma* como método para conversão em escala de cinza e CCV para extração de características. As gerações com aguçamento e saliência obtiveram os melhores resultados.

Luma & CCV	Média	Desvio Padrão
Todos	66.015325	5.621643
Aguçamento	66.684300	5.619944
Borramento	62.186430	6.509084
Composição 16	63.225965	7.920787
Composição 4	63.824235	5.621168
Limiares	64.453515	6.769440
Mistura	59.506260	6.472903
Ruído	64.202075	7.231610
SMOTE Visual	59.512530	7.737273
Saliência	66.260870	5.732209
SMOTE	65.531135	4.714502
Desbalanceado	59.640090	8.675836

Discussão

De acordo com o teste HSD de Tukey realizado, foi encontrado $p\text{-value} = 0.0003$ na comparação da base desbalanceada com a base rebalanceada utilizando o método SMOTE e $p\text{-value} = 0.0000$ com a geração artificial. Isso indica que ambos os métodos obtiveram relevância estatística quando comparados ao original. Porém, ao comparar o SMOTE com a geração artificial, o teste resultou em $p\text{-value} = 0.7122$. Tal valor indica que não há diferença significativa entre os métodos SMOTE e os métodos de geração artificial de imagens.

4.3.3 Experimento 3: multiclasses

Os dois experimentos anteriores analisaram o rebalanceamento de apenas duas classes. Este experimento apresenta a geração artificial de imagens aplicada a três bases multiclasses: Corel-1000, Caltech101-600 e Produce. Seus resultados são discutidos individualmente.

As bases são originalmente balanceadas, portanto a cada combinação dos *folds*, uma classe é considerada minoritária e três dos seus *folds* são descartados. Restando, dessa forma, um para treino e um para teste. Isso permitiu testar as bases como desbalanceadas.

4.3.3.1 Base de imagens Corel

Protocolo

- Imagens originais:** esse experimento foi realizado com a base de imagens Corel-1000, composta por fotografias que representam 10 classes: tribos africanas, praia, construções, ônibus, dinossauros, flores, elefantes, cavalos, montanhas e tipos de comidas ([WANG; WIEDERHOLD, 2001](#)). Para fins de exemplificação, são apresentadas imagens que representam essas classes na Figura 46.



Figura 46 – Base de imagens Corel-1000 ([WANG; WIEDERHOLD, 2001](#)). Essa base é composta por fotografias que representam 10 classes: tribos africanas, praia, construções, ônibus, dinossauros, flores, elefantes, cavalos, montanhas e tipos de comidas.

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

- Desbalanceamento:** realizaram-se 200 combinações de desbalanceamento dos 5 *folds* para as 10 classes.
- Método para geração artificial:** o método que atingiu o maior *F1-Score* de geração artificial foi a *mistura*. Por esse motivo, ela está exemplificada na Figura 47.
- Conversão em escala de cinza:** todos os métodos foram testados e o método *Gleam* obteve os melhores resultados.
- Extração de características:** dada a combinação de todos os métodos, o LBP se sobressaiu entre eles.



Figura 47 – Geração artificial de *mistura* para imagens da base Corel-1000. A imagem resultante (c) apresenta uma mistura das imagens (a) e (b).

Fonte: Elaborado pela autora.

6. Classificação: KNN com $K = 1$ foi o classificador utilizado.

Resultados

Ao realizar o experimento com as 10 classes da base Corel-1000, a combinação de *Gleam* para conversão em escala de cinza; e LBP como método de extração de características, resultou em melhores *F1-Scores*. A Tabela 4 apresenta a média e o desvio padrão dos *F1-Scores* encontrados nesse experimento. Note que a geração artificial com o método *mistura* obteve *F1-Score* similar ao SMOTE.

Tabela 4 – Resultados de *F1-Score* para as 10 classes da Corel-1000, utilizando *Gleam* como método para conversão em escala de cinza e LBP para extração de características. A geração artificial com o método *mistura* obteve *F1-Score* similar ao SMOTE.

<i>Gleam & LBP</i>	Média	Desvio Padrão
Todos	61.216388	2.205391
Aguçamento	61.098384	2.275732
Borramento	60.369376	2.254895
Composição 16	60.630235	2.212156
Composição 4	60.568624	2.254904
Limiares	61.296003	2.101686
Mistura	61.366671	2.225635
Ruído	60.825884	2.358098
SMOTE Visual	60.886122	2.321783
Saliência	61.050988	2.271443
SMOTE	61.368896	2.148675
Desbalanceado	60.362256	2.290263

Discussão

O rebalanceamento com o método de *mistura* e com o SMOTE tiveram um desempenho quase idêntico (i.e. $p\text{-value} = 0.9948$). Mas apesar dos *F1-Scores* também parecerem similares à

base desbalanceada, o teste HSD de Tukey confirmou que há diferença estatística significante entre eles ($p\text{-value} = 0.0$). Dessa forma, tanto o SMOTE quanto a geração artificial rebalancearam a base satisfatoriamente.

4.3.3.2 Base de imagens Caltech101-600

Protocolo

- Imagens originais:** foi utilizada a base Caltech101-600 ([FEI-FEI; FERGUS; PERONA, 2007](#)). Desta base, foi utilizado um conjunto de seis classes balanceadas: aviões, bonsais, candelabros, tartarugas, motocicletas e relógios. Imagens que representam essas classes são apresentadas na Figura 48.

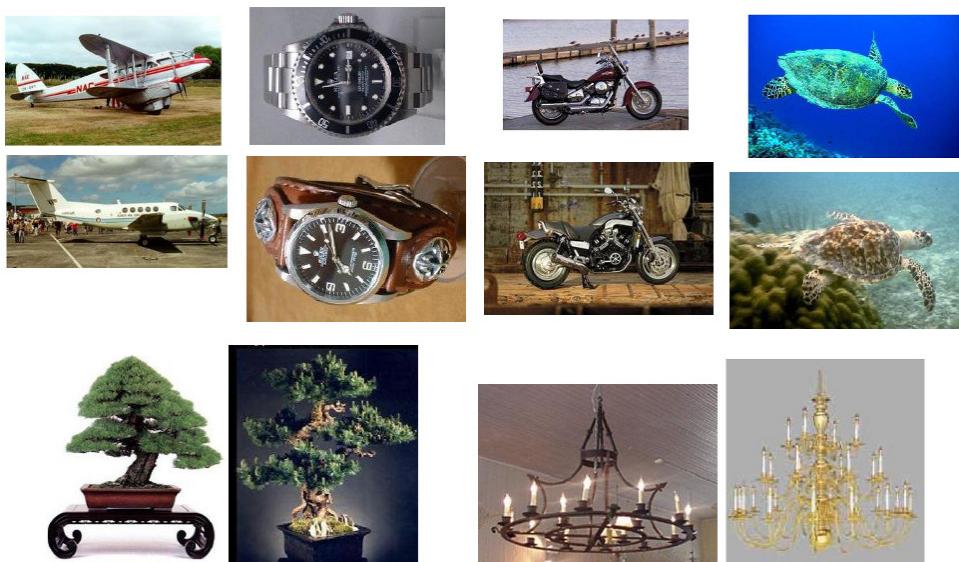


Figura 48 – Conjunto de seis classes balanceadas: aviões, bonsais, candelabros, tartarugas, motocicletas e relógios da base de imagens Caltech ([FEI-FEI; FERGUS; PERONA, 2007](#)).

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

- Desbalanceamento:** 120 combinações de *5 folds* para as 6 classes foram testadas. Cobrindo, assim, todas as possibilidades para garantir que cada imagem pudesse pertencer a majoritária, minoritária, treino, teste ou não ser utilizada na classificação.
- Método para geração artificial:** o melhor resultado em termos de geração artificial foi utilizando todas as gerações. A Figura 49 apresenta um exemplo da geração com *mistura de limiares*, uma das possíveis gerações artificiais.
- Conversão em escala de cinza:** todos os métodos foram testados. O que obteve melhores resultados foi o *Intensidade'*.
- Extração de características:** de todos os métodos foram testados, o HOG foi o melhor método de extração de características.

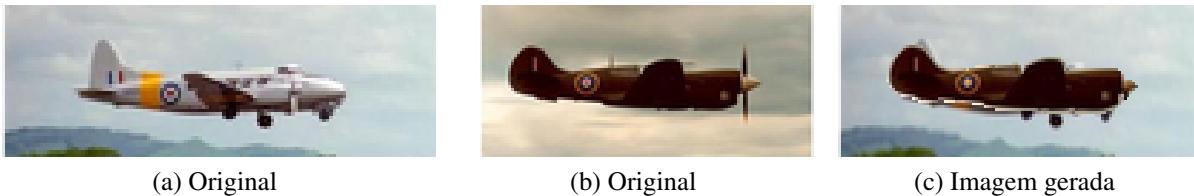


Figura 49 – Exemplo de uma geração artificial utilizando o método de *mistura limiarizada* para a base de imagens Caltech101-600.

Fonte: Elaborado pela autora.

6. Classificação:

classificador KNN com $K = 1$.

Resultados

Considerando que essa base possui 6 classes, 120 sub-experimentos foram necessários para cobrir todas as possibilidades de desbalanceamento. Os resultados de tais experimentos são apresentados na Tabela 5. É possível notar que tanto o SMOTE quanto a geração artificial utilizando todos os métodos obtiveram um *F1-Score* melhor que a versão desbalanceada. A combinação aleatória uniforme de todos os métodos obteve um *F1-Score* melhor do que cada método individualmente.

Tabela 5 – Resultados do experimento com a base Caltech101-600. É possível notar que tanto o SMOTE quanto a geração artificial utilizando todos os métodos obtiveram um *F1-Score* melhor que a versão desbalanceada.

Intensidade' & HOG	Média	Desvio Padrão
Todos	77.591493	3.387543
Aguçamento	76.982412	3.482750
Borramento	75.736587	3.812333
Composição 16	75.742438	3.823391
Composição 4	75.787022	3.794589
Limiares	76.785628	3.652596
Mistura	77.186702	3.382386
Ruído	77.344959	3.664212
SMOTE Visual	75.486170	4.405685
Saliência	76.587268	3.600158
SMOTE	77.755417	3.529355
Desbalanceado	75.732382	3.833682

Discussão

Com o teste *post-hoc* HSD de Tukey ficou comprovada a suposição de que os resultados do SMOTE e da geração artificial utilizando todos os métodos não possuem diferença estatística. Isso porque resultaram em $p\text{-value} = 0.9333$. Porém, entre o desbalanceado e o SMOTE ou a

geração obteve-se $p\text{-value} < 0.0$. O que indica que todos os métodos de rebalanceamento foram satisfatórios para a base Caltech101-600.

4.3.3.3 Base de imagens Produce

Protocolo

- 1. Imagens originais:** base de imagens Produce, composta por imagens de vegetais e frutas tropicais ([ROCHA et al., 2010](#)). Posuem fundo similar mas muitas mudanças na iluminação, no número de objetos e na escala. Amostras das imagens dessa base são apresentadas na Figura 50. Essa base contem 14 classes e um total de 1400 imagens.



Figura 50 – Base de imagens Produce, composta por imagens de vegetais e frutas tropicais ([ROCHA et al., 2010](#)). A base possui classes com fundo similar, porém com variações na iluminação, número de objetos, cor e escala.

Fonte: [Ponti, Nazaré e Thumé \(2016\)](#).

- 2. Desbalanceamento:** considerando que a base Produce é balanceada, 280 combinações dos 5 *folds* das 14 classes foram testadas. Dessa forma todas as classes tiveram chance de serem a classe minoritária e, assim, rebalanceadas.
- 3. Método para geração artificial:** o melhor método de geração para a base Produce foi a *mistura*, exemplificada na Figura 51.
- 4. Conversão em escala de cinza:** todas as combinações de métodos foram testadas. *Luma* destacou-se como o melhor método para converter a imagem em escala de cinza.
- 5. Extração de características:** todos os métodos foram testados. O LBP obteve os melhores resultados como método para extrair as características.
- 6. Classificação:** classificador KNN com $K = 1$.



Figura 51 – Exemplo de geração artificial utilizando a *mistura* de duas imagens para a base Produce.

Fonte: Elaborado pela autora.

Resultados

A Tabela 6 apresenta a média e o desvio padrão dos *F1-Scores* encontrados nos 280 sub-experimentos para a base Produce. Esses cobrem todas as possibilidades de desbalanceamento. O método de *mistura* resultou no melhor *F1-Score*.

Tabela 6 – Resultados de *F1-Scores* para a base de imagens Produce. O método *mistura* apresentou-se como o melhor.

Luma & LBP	Média	Desvio Padrão
Todos	91.682606	1.985052
Aguçamento	91.463371	2.039631
Borramento	91.493838	2.022884
Composição 16	91.476496	2.037559
Composição 4	91.478369	2.021889
Limiares	91.843197	1.934367
Mistura	91.985465	1.949951
Ruído	91.489904	2.028988
SMOTE Visual	91.473853	2.031678
Saliência	91.772214	2.006734
SMOTE	91.869133	1.952464
Desbalanceado	91.489904	2.028988

Discussão

O teste *post-hoc* HSD de Tukey indicou que há relevância estatística entre os *F1-Scores* da base desbalanceada quando comparada com o rebalanceamento utilizando o método de *mistura* (*p-value* = 0.0087). Mas comparado com SMOTE, o *p-value* = 0.0607 indica que não há diferença significante entre ele e a versão original. Ainda assim, o teste resultou em *p-value* = 0.7658 para a comparação entre o SMOTE e a geração artificial, indicando que não há significância entre os dois. De qualquer forma, tais resultados indicam que a geração artificial de *mistura* obteve resultado significativo.

4.3.4 Experimento 4: classes naturalmente desbalanceadas

Os experimentos anteriores foram realizados em bases já平衡adas, com o desbalanceamento sendo provocado. Este experimento utiliza duas combinações de classes de imagens naturalmente desbalanceadas.

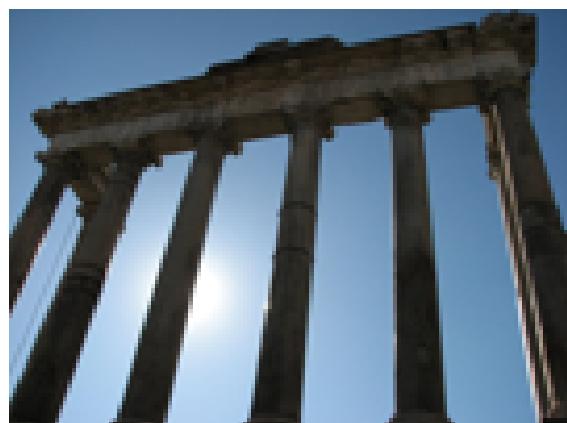
4.3.4.1 Duas classes desbalanceadas: Eiffel Tower e Rome antica

Protocolo

- Imagens originais:** as classes utilizadas para esse experimento foram *Eiffel Tower* com 1607 imagens, e *Rome antica* com apenas 125 imagens ([HU; COLLOMOSSE, 2013](#)). Essas classes possuem fundo similar, mas formas distintas.



(a) *Eiffel Tower*



(b) *Rome antica*

Figura 52 – Imagens das classes naturalmente desbalanceadas *Eiffel Tower*, com 1607 imagens, e *Rome antica*, com apenas 125 imagens.

Fonte: [Hu e Collomosse \(2013\)](#).

- Desbalanceamento:** natural da base. A classe minoritária é a *Rome antica* com 125 imagens. Dessa forma, a cada combinação dos *folds*, $1607 - 125 = 1482$ imagens foram geradas para rebalancear a base.
- Método para geração artificial:** todas as gerações foram testadas, a que obteve melhor resultado foi a *mistura*. Essa geração está exemplificada na Figura 53.
- Conversão em escala de cinza:** todos os métodos foram testados. O que obteve melhor *F1-Score* foi o MSB.
- Extração de características:** de todos os métodos para extração testados, o HOG obteve os melhores resultados. Isso confirma que as classes são diferenciadas pela sua forma.
- Classificação:** KNN com $K = 1$.

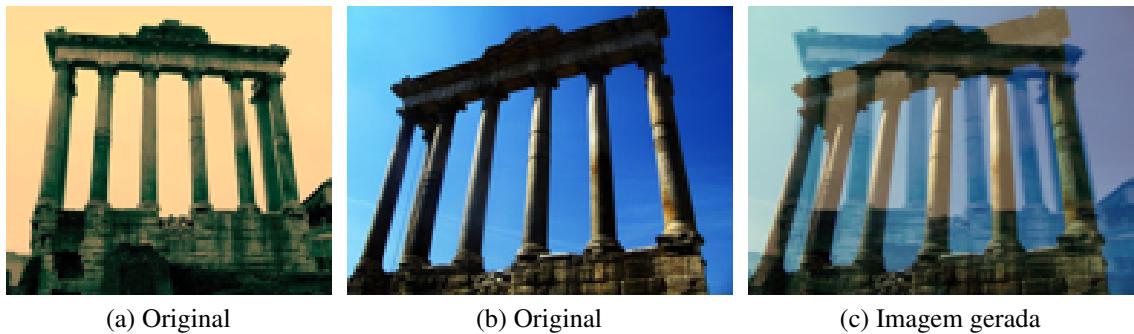


Figura 53 – Exemplo de geração artificial utilizando a *mistura* de duas imagens para a base Produce.

Fonte: Elaborado pela autora.

Resultados

O rebalanceamento foi realizado de forma a balancear a classe minoritária *Rome antica*. A Tabela 7 apresenta os resultados de tal rebalanceamento. Nela estão os resultados de média e desvio padrão do rebalanceamento realizado com a combinação dos *folds* das duas classes. O método que obteve maior valor de *F1-Score* foi o de geração artificial com o método de *mistura* de duas imagens.

Tabela 7 – Resultados de *F1-Score* para as classes *Eiffel Tower* e *Rome antica*, naturalmente desbalanceadas. O método *mistura* destacou-se como o melhor método para rebalanceamento dessas classes.

MSB & HOG	Média	Desvio Padrão
Todos	88.616130	1.952745
Aguçamento	98.345625	1.755180
Borramento	97.173480	2.760269
Composição 16	97.206953	2.712589
Composição 4	97.301088	2.381761
Limiares	98.628050	1.142830
Mistura	99.112148	0.837440
Ruído	91.499965	1.762218
SMOTE Visual	82.963755	2.678785
Saliência	98.246915	1.526565
SMOTE	98.471765	0.788747
Desbalanceado	97.173480	2.760269

Discussão

Apesar de valores de *F1-Score* próximos, de acordo com o teste *post-hoc* de Tukey, foi encontrado $p\text{-value} = 0.0030$ para a base desbalanceada quando comparado com o rebalanceamento com SMOTE. Isso indica que há significância estatística relevante entre eles. O teste também indicou que há significância entre a base desbalanceada e ela rebalanceada com

a geração artificial ($p\text{-value} = 0.0000$). Porém, se comparados o SMOTE e a geração artificial temos $p\text{-value} = 0.2255$, indicando que não há diferença significante.

4.3.4.2 Multiclasses desbalanceadas: Trafalgar Square, Madeleine Church e Pantheon

Protocolo

- Imagens originais:** foram utilizadas *Trafalgar Square*, *Madeleine Church* e *Pantheon* ([HU; COLLOMOSSE, 2013](#)). Essas classes são apresentadas na Figura 54. Elas contêm tanto imagens parecidas (difícil classificação), quanto imagens bem divergentes (fácil diferenciação).

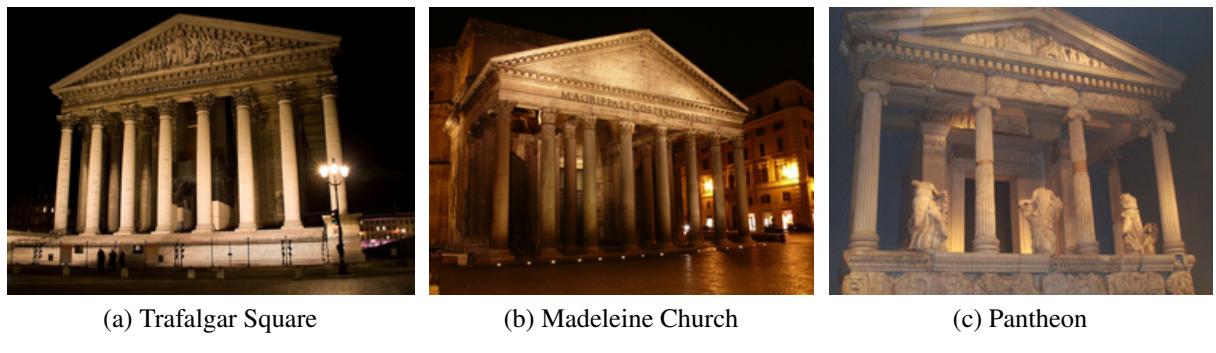


Figura 54 – Imagens representativas das classes *Trafalgar Square*, *Madeleine Church* e *Pantheon*. Elas contêm tanto imagens parecidas (difícil classificação), quanto imagens bem divergentes (fácil diferenciação).

Fonte: [Hu e Collomosse \(2013\)](#).

- Desbalanceamento:** natural da base. A classe *Trafalgar Square* contém 132 imagens, a *Madeleine Church* 316 e a *Pantheon* 23. Para rebalancear foram necessárias $316 - 132 = 147$ imagens para a *Trafalgar Square* e $316 - 23 = 234$ para a classe *Pantheon*.
- Método para geração artificial:** todas as gerações foram testadas, mas a melhor foi a de *composição* de quatro imagens, retratada na Figura 55.
- Conversão em escala de cinza:** de todos os métodos foram testados, o melhor para essas imagens foi o *Gleam*.
- Extração de características:** todos os métodos para extração foram testados, mas o HOG obteve melhores resultados.
- Classificação:** KNN com $K = 1$ foi o classificador utilizado.

Resultados

Os melhores métodos para extração e conversão em escala de cinza foram HOG e *Gleam*, respectivamente. O método de geração artificial de imagens que obteve o melhor *F1-Score* foi a



Figura 55 – Exemplo de geração artificial realizada com a *composição* de quatro imagens para a classe *Pantheon*.
Fonte: Elaborado pela autora.

composição de quatro imagens e a Tabela 8 apresenta tais resultados. O método SMOTE piorou significativamente a classificação, mas a geração artificial ficou muito similar a classificação sem gerar imagem nenhuma.

Tabela 8 – Resultados utilizando os métodos HOG e *Gleam* para as classes *Trafalgar Square*, *Madeleine Church* e *Pantheon*. O método SMOTE piorou significativamente a classificação, mas a geração artificial ficou muito similar a classificação sem gerar imagem nenhuma.

Gleam & HOG	Média	Desvio Padrão
Todos	57.285167	6.383771
Aguçamento	69.726448	8.069939
Borramento	70.621480	8.314352
Composição 16	70.478478	8.366100
Composição 4	70.837125	7.811876
Limiares	67.965067	5.699239
Mistura	65.644708	6.112751
Ruído	60.777810	8.257640
SMOTE Visual	53.912270	7.907406
Saliência	67.584110	6.236710
SMOTE	62.322870	5.726037
Desbalanceado	70.621480	8.314352

Discussão

Utilizando o teste *post-hoc* de Tukey, foi encontrado $p\text{-value} = 0.0$ para a base desbalanceada versus o rebalanceamento com SMOTE. Isso indica que há significância estatística relevante entre eles. O teste também indicou que não há significância entre a base desbalanceada e ela rebalanceada com a geração artificial ($p\text{-value} = 0.9858$). Além disso, se comparados o SMOTE e a geração artificial temos $p\text{-value} = 0.0$, indicando que há diferença significante. Ou

seja, o método SMOTE foi significativamente pior que a base original e o método de geração artificial teve uma melhora não estatisticamente significativa.

4.3.5 Experimento 5: classes com muitas imagens

Por fim, este experimento foi realizado com duas classes contendo uma grande quantidade de imagens.

4.3.5.1 Classes balanceadas, distintas e com grande número de imagens

Protocolo

1. **Imagens originais:** classes *Deer* e *Ship*, cada uma com 5000 imagens cada ([Krizhevsky; Hinton, 2009](#)). Essas imagens possuem dimensão de apenas 32x32 pixels, o que dificulta a representatividade da extração de características (ver Figura 56).

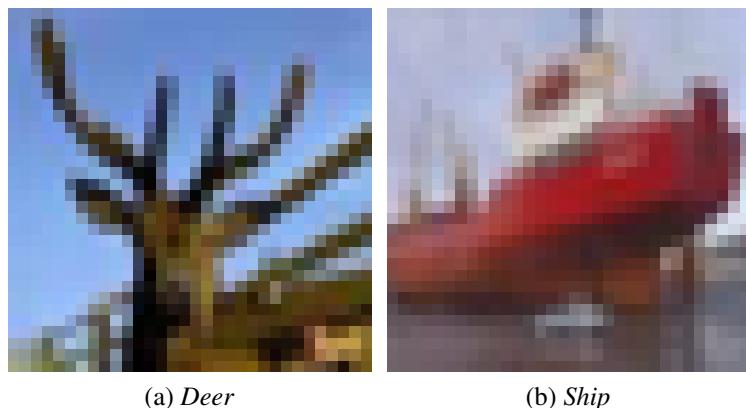


Figura 56 – Imagens representativas das classes *Deer* e *Ship*. São imagens em baixa resolução (32x32 pixels), o que dificulta o seu processamento e posterior classificação.

Fonte: Krizhevsky e Hinton (2009).

2. **Desbalanceamento:** são classes balanceadas. Por conta disso, todas as combinações dos 5 *folds* foram testadas.
3. **Método para geração artificial:** a operação de *aguçamento* obteve maiores valores de *F1-Score* dentre todos os métodos de rebalanceamento.
4. **Conversão em escala de cinza:** todos os métodos de conversão em escala de cinza foram testados. O que resultou em melhor *F1-Score* foi o *Gleam*.
5. **Extração de características:** todos os métodos para extração foram testados, mas o método HOG destacou-se como o melhor.
6. **Classificação:** o classificador utilizado foi o KNN com $K = 1$.

Resultados

Os resultados encontrados ao rebalancear as classes *Deer* e *Ship* estão apresentados na Tabela 9. Os rebalanceamentos com imagens artificiais geradas com os métodos *aguçamento*, *mistura limiarizada* e *mistura ponderada* apresentaram os melhores resultados.

Tabela 9 – Resultados de *F1-Score* para as classes *Deer* e *Ship*, utilizando *Gleam* como método para conversão em escala de cinza e HOG para extração de características. As gerações de imagens com *aguçamento*, *mistura limiarizada* e *mistura ponderada* obtiveram os melhores resultados.

Gleam & HOG	Média	Desvio Padrão
Todos	88.939785	1.035462
Aguçamento	89.473075	0.961293
Borramento	88.150530	1.006920
Composição 16	88.086065	0.987147
Composição 4	88.360675	0.949842
Limiares	89.356875	0.942907
Mistura	89.445505	0.939809
Ruído	86.762190	1.165064
SMOTE Visual	88.395965	1.081594
Saliência	88.136675	0.988034
SMOTE	63.183905	2.355925
Desbalanceado	88.121310	0.985599

Discussão

Ao realizar o teste *post-hoc* de Tukey, foi encontrado $p\text{-value} = 0.0$ para a base desbalanceada quando comparada com o rebalanceamento com SMOTE. Isso indica que há significância estatística relevante entre eles e também entre a base desbalanceada e ela rebalanceada com a geração artificial ($p\text{-value} = 0.0006$). Além disso, se comparados o SMOTE e a geração artificial tem-se $p\text{-value} = 0.0$, indicando que há diferença significante. Assim, os resultados apontam que SMOTE piorou significativamente a classificação, enquanto a geração artificial forneceu uma melhora considerável.

4.3.5.2 Classes balanceadas, similares e com grande número de imagens

Protocolo

1. **Imagens originais:** classes *Shark* e *Fish* com 1300 imagens cada (RUSSAKOVSKY *et al.*, 2015). Essas classes são exemplificadas na Figura 57.
2. **Desbalanceamento:** as duas classes estão balanceadas. Assim, esse experimento realizou a combinação dos 5 *folds* de cada classe para testar o rebalanceamento.

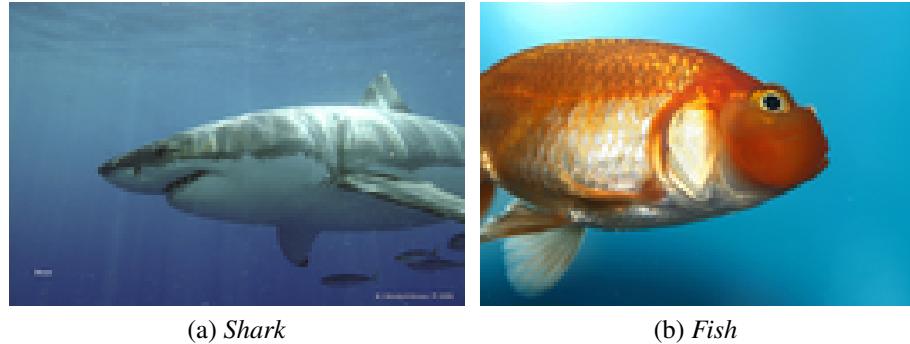


Figura 57 – Imagens das classes *Shark* e *Fish*. Essas classes apresentam fundo similar.

Fonte: Russakovsky et al. (2015).

3. **Método para geração artificial:** a *mistura limiarizada* foi o método de geração artificial que obteve maior *F1-Score*, exemplificada na Figura 58.

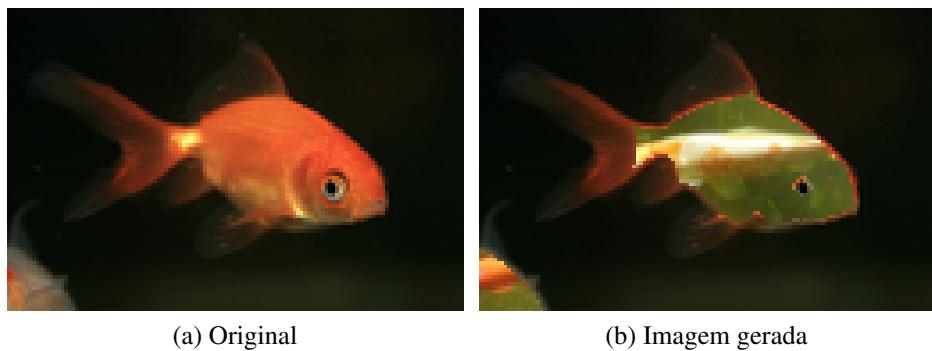


Figura 58 – Exemplo de geração artificial utilizando a *mistura limiarizada* de duas imagens para a classe *Fish*. Por se tratarem de imagens com *foreground* facilmente destacado, a combinação de limiares oferece uma mistura interessante dos objetos.

Fonte: Elaborado pela autora.

4. **Conversão em escala de cinza:** todos os métodos de conversão em escala de cinza foram testados. O que resultou em melhor *F1-Score* foi o *Luma*.
5. **Extração de características:** todos os métodos para extração foram testados, mas o que melhor se destacou foi o LBP.
6. **Classificação:** KNN com $K = 1$ foi o classificador utilizado.

Resultados

A Tabela 10 apresenta a média e o desvio padrão encontrados ao rebalancear as classes *Shark* e *Fish*. Pode ser verificado que a geração artificial com a *mistura limiarizada* aparenta ser o melhor método para rebalanceá-las.

Tabela 10 – Resultados de *F1-Score* para as classes *Shark* e *Fish*. A geração artificial com a *mistura limiarizada* apresenta-se como o melhor método para rebalanceá-las.

Luma & LBP	Média	Desvio Padrão
Todos	77.128293	2.064232
Aguçamento	76.862198	1.786898
Borramento	73.919025	1.942236
Composição 16	75.435960	2.254193
Composição 4	74.359165	1.974563
Limiares	78.330465	1.768789
Mistura	77.381940	2.588103
Ruído	74.094233	1.743535
SMOTE Visual	74.569545	1.648100
Saliência	76.934077	2.114528
SMOTE	78.066890	2.107176
Desbalanceado	73.378227	2.110355

Discussão

Como esperado, de acordo com o teste *post-hoc* de Tukey foi encontrado $p\text{-value} = 0.0$ para a base desbalanceada quando comparada com o SMOTE. Isso indica que há significância estatística relevante entre eles. Tal teste também indicou que há significância entre a base desbalanceada e ela rebalanceada com a geração artificial ($p\text{-value} = 0.0$). Porém, não há diferença significante na comparação entre o SMOTE e a geração artificial ($p\text{-value} = 0.8264$). Assim, ambos métodos rebalancearam as classes satisfatoriamente.

4.4 Considerações finais

Este capítulo descreveu como o rebalanceamento é realizado, através da geração de imagens artificiais para a classe minoritária e utilizando-a para o treinamento. Os métodos para gerar tais imagens foram apresentados e exemplificados. Foram também descritos os algoritmos e seus parâmetros, limitações e métodos relacionados.

A Tabela 11 apresenta o *ranking* dos métodos de rebalanceamento ao acumular os resultados de todos os experimentos. Esse valor é dado pela soma da posição de cada método em relação ao *F1-Score*, em ordem ascendente. Por exemplo, na Tabela 10 o método de *mistura limiarizada* ocupa a primeira posição e o método SMOTE a segunda. No geral, contabilizando todos os experimentos, a *mistura limiarizada* apresenta-se como melhor método de rebalanceamento.

Essa tabela considera que diferenças não significativas do valor do *F1-Score* possuem o mesmo peso que diferenças significantes. A Tabela 12 apresenta a média dos *F1-Scores* para cada método. É possível verificar que, no cenário multiclass, apesar de ter sido o melhor método em relação à sua posição, o SMOTE apresentou piores resultados mesmo comparando com as bases

Tabela 11 – *Ranking* dos métodos de rebalanceamento ao acumular os resultados de todos os experimentos. Esse valor é dado pela soma da posição de cada método em relação ao *F1-Score*, em ordem ascendente.

Cenários de duas classes	Cenário Multiclasses	Todos
Aguçamento (16)	SMOTE (13)	Limiares (35)
Limiares (17)	Mistura (14)	Mistura (39)
Saliência (22)	Limiares (18)	SMOTE (42)
Todos (25)	Todos (22)	Aguçamento (43)
Mistura (25)	Saliência (24)	Saliência (46)
Composição 4 (29)	Aguçamento (27)	Todos (47)
SMOTE (29)	Composição 4 (28)	Composição 4 (57)
Composição 16 (37)	Ruído (28)	Composição 16 (69)
Borramento (44)	Borramento (29)	Borramento (73)
Ruído (47)	Composição 16 (32)	Ruído (75)
SMOTE Visual (47)	Desbalanceado (34)	Desbalanceado (86)
Desbalanceado (52)	SMOTE Visual (42)	SMOTE Visual (89)

desbalanceadas. Nesse caso, o melhor método de rebalanceamento é a operação de geração de imagens artificiais com *aguçamento*.

Tabela 12 – Apresenta a média dos *F1-Scores* para cada método, ordenada pela coluna de todos os experimentos. É possível verificar que, no cenário multiclasses, apesar de ter sido o melhor método em relação à sua posição, o SMOTE apresentou piores resultados mesmo comparando com as bases desbalanceadas.

Métodos	Cenários de duas classes	Cenário Multiclasses	Todos
Aguçamento	84,473575	74,817654	80,182055
Limiares	84,332408	74,472474	79,950215
Saliência	84,172238	74,248645	79,761752
Composição 4	83,123738	74,667785	79,365537
Composição 16	82,977850	74,581912	79,246322
Mistura	83,124582	74,045887	79,089606
Borramento	82,164793	74,555320	78,782805
Desbalanceado	81,314335	74,551506	78,308633
Todos	82,358089	71,943914	77,729567
Ruído	81,179247	72,609639	77,370532
SMOTE	79,085501	73,329079	76,527091
SMOTE Visual	78,811666	70,439604	75,090749

Este estudo apresentou evidências experimentais de que pode haver ganho estatístico do *F1-Score* ao gerar imagens, quando comparado à geração de exemplos artificiais no espaço de atributos (ou seja, depois que as características já foram extraídas das imagens). Além disso, na maioria dos experimentos, a geração artificial obteve significância estatística relevante quando comparada a base desbalanceada. Com os experimentos realizados foi possível notar que a geração de imagens artificiais pode gerar novas informações para a classificação das imagens. O que indica que um estudo aprofundado de cada contexto pode relatar quais operações podem ser aplicadas nas imagens originais de forma a auxiliar o cenário de bases desbalanceadas.



CONCLUSÕES

São muitos os aspectos que influenciam a classificação de imagens. É comum lidar com as particularidades das características extraídas ao invés de se preocupar com o tratamento das imagens no início do *pipeline*. Portanto, o enfoque desta pesquisa foi na investigação de métodos de processamento de imagens antes da extração de características. Para tal, esta dissertação propôs duas abordagens para melhorar a discriminação entre classes: realizar a quantização com o objetivo de obter vetores de características mais compactos; e gerar imagens artificiais a partir das imagens originais de forma a rebalancear a base de imagens.

Os resultados encontrados ao usar os métodos de quantização de imagens apontam para uma alternativa — ou complemento — à seleção de características. Dado o número de cores limitado na imagem original, a quantidade de possíveis características a serem extraídas é reduzido, especialmente as de cor. A extração de características de textura também é facilitada, considerando que normalmente é computada utilizando uma memória proporcional ao número de intensidades.

Ficou constatado que um vetor original de D dimensões pode ser reduzido a $d \approx D/4$ modificando apenas o parâmetro de quantização e produzindo bons resultados. Ou seja, de 256 intensidades para apenas 64. Outra possibilidade é utilizar esses métodos como um primeiro passo de redução e então utilizar o LPP para obter apenas 100 características que melhor representam os dados, atingindo acurácia maiores ou similares.

É importante ressaltar que realizar a quantização de imagens não requer treinamento e já faz parte de uma tarefa do próprio *pipeline* de reconhecimento. Por esta razão, seu uso não aumenta o custo computacional do sistema, e ainda simplifica os passos subsequentes. Isso reduz a dimensão do vetor de características para os vetores de cor e o tempo de computação para os descritores de textura. Outra observação importante é que a quantização é usada especialmente para dados visuais, então não é um método geral de redução de dimensionalidade.

Além disso, com os experimentos realizados foi possível notar que a geração de imagens

artificiais pode gerar novas informações para a classificação das imagens. Assim a geração de elementos no espaço visual provou contribuir com o balanceamento entre classes. Melhorando assim, a acurácia de algoritmos de classificação, inclusive quando comparada à geração de exemplos artificiais no espaço de atributos (i.e. SMOTE). Para validar a proposta da geração artificial de imagens, uma visualização do espaço de características foi realizada. As características das novas imagens e os exemplos resultantes da interpolação de vetores originais foram projetados no plano das imagens originais antes do desbalanceamento. Dessa forma foi possível visualizar que a geração de imagens artificiais proposta foi capaz de ocupar uma região do espaço mais abrangente do que o SMOTE. Este último, comprovadamente, possui o ponto negativo de não extrapolar os limites da classe minoritária. Ainda, está suscetível à criação de novos exemplos em regiões da classe majoritária, o que também prejudica a classificação.

5.1 Publicações

Foi publicado um artigo na revista *Neurocomputing*, referente aos resultados de quantização desta pesquisa ([PONTI; NAZARÉ; THUMÉ, 2016](#)). O artigo referente à geração de imagens artificiais para o rebalanceamento de classes está em preparo.

5.2 Trabalhos Futuros

Ao utilizar imagens com reduzido número de cores (quantizadas), os métodos de extração de características baseados em orientação (HoG, SIFT), *bag of visual words* e *Fisher vectors*, seriam provavelmente mais esparsos. Portanto, estudos futuros da influência da quantização em outros métodos de extração de características podem ser realizados.

Visto que o método MSB se sobressaiu nos resultados do uso da quantização, variações desse método podem produzir melhores mapas de cor para a etapa de reconhecimento. A investigação dos espaços gerados por tais métodos também pode fornecer uma melhor análise da discriminação entre classes.

Estudos futuros da influência da quantização em outros métodos de extração de características podem ser realizados. Por exemplo, ao utilizar imagens com reduzido número de cores (quantizadas), métodos de extração de características globais mais complexos como VLAD produziriam provavelmente vetores mais esparsos, o que poderia ser explorado em diversas aplicações.

Como extensão dos experimentos reportados, pode ser feita a análise dos espaços encontrados para os diferentes métodos de geração artificial de imagens. Além disso, o impacto de tais métodos em diferentes extratores de características pode sugerir quais são as características latentes percebidas com cada método de extração de características. E, dessa forma, outros métodos para a geração podem ser sugeridos. Por exemplo, o SMOTE visual poderia ser

realizado tomando-se imagens próximas (ao invés de aleatórias), dada uma medida de distância (e.g. diferença entre histogramas). Tornando esse método mais similar à sua implementação original, realizada no espaço de características.

Atualmente, o estado da arte de extração e classificação de imagens corresponde ao uso de *Convolutional Neural Network* (CNN) ([SCHMIDHUBER, 2014](#)). Essas redes são compostas por camadas de neurônios que têm por objetivo aprender quais são as melhores características que diferenciam as classes de imagens. O aprendizado, nesse caso, corresponde ao ajuste dos parâmetros para reduzir a diferença entre a saída esperada — classe verdadeira — e a produzida. Dessa forma, tais redes aprendem quais são as características latentes nas imagens de entrada. Uma possível continuação a esta pesquisa é analisar as características aprendidas por essas redes. Elas permitem encontrar as características mais relevantes da base de imagens, que os extratores de características canônicos não capturam. Isso porque possuem uma hierarquia de camadas, desde a imagem original até a etapa de classificação, com o objetivo de aprender qual o melhor processamento para as imagens de modo a melhor discriminá-las, em níveis gradativos de abstração. Podem também indicar possíveis operações para a geração de imagens artificiais.

Além de analisar o processamento realizado por uma rede de convolução para a classificação das imagens, uma *Restricted Boltzmann Machine* (RBM) também pode ser treinada para análise da sua memória associativa (matriz de características aprendida). Uma rede neural RBM treina um modelo a partir dos vetores de entrada ([FISCHER; IGEL, 2014](#)). As imagens artificialmente geradas foram adicionadas no conjunto de treino sem verificação da sua relevância, o que pode ter prejudicado a classificação. A representação das imagens de entrada, aprendida pela etapa de treinamento da RBM, pode ser utilizada para definir quais imagens são relevantes para o aprendizado ou não. Além disso, pode servir como escolha para qual imagem original utilizar, ao invés do método aleatório utilizado nos resultados preliminares.

REFERÊNCIAS

- ABDI, H.; WILLIAMS, L. J. Principal Component Analysis. **Wiley Interdisciplinary Reviews: Computational Statistics**, v. 2, n. 4, p. 433–459, 2010. ISSN 19395108. Disponível em: <<http://dx.doi.org/10.1002/wics.101>>. Citado na página 20.
- AHONEN, T.; HADID, A.; PIETIKÄINEN, M. Face description with local binary patterns: application to face recognition. **IEEE transactions on pattern analysis and machine intelligence**, v. 28, n. 12, p. 2037–2041, dec 2006. ISSN 0162-8828. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/17108377>>. Citado na página 9.
- BATISTA, G. E.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. **ACM Sigkdd Explorations Newsletter**, v. 6, n. 1, p. 20–29, 2004. Disponível em: <<http://dl.acm.org/citation.cfm?id=1007735>>. Citado 2 vezes nas páginas 18 e 20.
- BOIMAN, O.; SHECHTMAN, E.; IRANI, M. In defense of Nearest-Neighbor based image classification. **IEEE Conference on Computer Vision and Pattern Recognition**, n. i, p. 1–8, 2008. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587598>>. Citado na página 22.
- BORGES, V. R. P.; OLIVEIRA, M. de; FERREIRA, T. G.; VIEIRA, A. A. H.; OLIVEIRA, M. C. F. de. Feature Extraction and Interactive Visualization to Assist Green Algae Taxonomic Classification. In: **XXVI SIBGRAPI - Conference on graphics, patterns and images**. Arequipa, Peru: [s.n.], 2013. p. 4. Disponível em: <<http://www.ucsp.edu.pe/sibgrapi2013/eproceedings/wip/115151.pdf>>. Citado 2 vezes nas páginas 9 e 14.
- CASTRO, C. L.; BRAGA, A. P. Aprendizado supervisionado com conjuntos de dados desbalanceados. **Sba Controle & Automação**, v. 22, n. 5, p. 441 – 446, 2011. Citado na página 18.
- CHAPELLE, O.; HAFFNER, P.; VAPNIK, V. N. Support vector machines for histogram-based image classification. **IEEE transactions on neural networks**, v. 10, n. 5, p. 1055–64, jan 1999. ISSN 1045-9227. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/18252608>>. Citado na página 23.
- CHAWLA, N. V.; HALL, L. O.; BOWYER, K. W. SMOTE: Synthetic Minority Over-sampling Technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321–357, 2002. Disponível em: <<http://www.jair.org/papers/paper953.html>>. Citado 2 vezes nas páginas 2 e 18.
- CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, sep 1995. ISSN 0885-6125. Disponível em: <<http://link.springer.com/10.1007/BF00994018>>. Citado na página 22.
- DALAL, N.; TRIGGS, B. Histograms of Oriented Gradients for Human Detection. In: **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. IEEE, 2005. v. 1, p. 886–893. ISBN 0-7695-2372-2. ISSN 1063-6919. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1467360>>. Citado na página 16.

FEI-FEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. **Computer Vision and Image Understanding**, v. 106, n. 1, p. 59–70, apr 2007. ISSN 10773142. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1077314206001688>>. Citado 3 vezes nas páginas 28, 29 e 67.

FISCHER, A.; IGEL, C. Training restricted Boltzmann machines: An introduction. **Pattern Recognition**, v. 47, n. 1, p. 25–39, jan 2014. ISSN 00313203. Citado na página 83.

GARCIA, E. A. Learning from Imbalanced Data. **IEEE Transactions on Knowledge and Data Engineering**, v. 21, n. 9, p. 1263–1284, sep 2009. ISSN 1041-4347. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5128907>>. Citado na página 51.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. [S.l.]: Prentice-Hall, 2007. ISBN 013168728X. Citado 8 vezes nas páginas 8, 10, 11, 12, 14, 15, 37 e 41.

GROSS, R.; BRAJOVIC, V. An image preprocessing algorithm for illumination invariant face recognition. **Audio and Video-Based Biometric Person Authentication**, p. 10–18, 2003. Citado na página 11.

HAN, H.; WANG, W.-Y.; MAO, B.-H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. **Advances in intelligent computing**, Alvey Vision Club, v. 17, n. 12, p. 878–887, 2005. Citado na página 18.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural Features for Image Classification. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 3, n. 6, p. 610–621, nov 1973. ISSN 0018-9472. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4309314>>. Citado na página 16.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: **Proceedings of the Alvey Vision Conference**. Alvey Vision Club, 1998. p. 147—152. Disponível em: <<http://www.bmva.org/bmvc/1988/avc-88-023.html>>. Citado na página 15.

HE, X.; NIYOGI, P. Locality preserving projections. In: **Neural information processing systems**. [S.l.: s.n.], 2004. v. 16, p. 153–160. ISBN 0262201526. Citado na página 21.

HU, R.; COLLOMOSSE, J. A performance evaluation of gradient field HOG descriptor for sketch based image retrieval. **Computer Vision and Image Understanding**, v. 117, n. 7, p. 790–806, jul 2013. ISSN 10773142. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1077314213000349>>. Citado 3 vezes nas páginas 52, 71 e 73.

HUANG, J.; KUMAR, S. R.; MITRA, M.; ZHU, W.-J.; ZABIH, R. Image indexing using color correlograms. In: **Computer Society Conference on Computer Vision and Pattern Recognition**. IEEE, 1997. p. 762–768. ISBN 0-8186-7822-4. ISSN 1063-6919. Disponível em: <<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=609412>>. Citado na página 16.

JAPKOWICZ, N.; STEPHEN, S. The class imbalance problem : A systematic study. **Intelligent data analysis**, IOS Press, v. 6, n. 5, p. 429–449, 2002. Citado na página 20.

KANAN, C.; COTTRELL, G. W. Color-to-grayscale: does the method matter in image recognition? **PloS one**, v. 7, n. 1, p. e29740, jan 2012. ISSN 1932-6203. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3254613/>>. Citado 2 vezes nas páginas 2 e 13.

- KNUTH, D. E. No Title. In: **Art of Computer Programming, Volume 2: Seminumerical Algorithms**. 3. ed. [S.l.: s.n.], 1997. cap. 3.4.7. Citado na página 41.
- KRIZHEVSKY, A.; HINTON, G. **Learning Multiple Layers of Features from Tiny Images**. [S.l.], 2009. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&ty>>. Citado na página 75.
- KUNCHEVA, L. **Combining pattern classifiers: methods and algorithms**. [S.l.]: John Wiley & Sons, 2004. Citado na página 2.
- LOWE, D. G. D. G. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, v. 60, n. 2, p. 91–110, 2004. Disponível em: <<http://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94>>. Citado na página 15.
- MARSLAND, S. Machine Learning: An Algorithmic Perspective. In: **Machine Learning: An Algorithmic Perspective**. [S.l.]: CRC Press, 2015. cap. Support Ve, p. 457. ISBN 1498759785. Citado na página 23.
- OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 7, p. 971–987, jul 2002. ISSN 0162-8828. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1017623>>. Citado na página 17.
- PASS, G.; ZABIH, R.; MILLER, J. Comparing images using color coherence vectors. In: **Proceedings of the fourth ACM international conference on Multimedia**. New York, New York, USA: ACM Press, 1996. p. 65–73. ISBN 0897918711. Disponível em: <<http://portal.acm.org/citation.cfm?doid=244130.244148>>. Citado na página 15.
- PAULOVICH, F. V.; OLIVEIRA, M. C. F.; MINGHIM, R. The Projection Explorer: A Flexible Tool for Projection-based Multidimensional Visualization. In: **XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)**. IEEE, 2007. p. 27–36. ISBN 0-7695-2996-8. ISSN 1530-1834. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4368165>>. Citado na página 20.
- PENATTI, O. A.; VALLE, E.; TORRES, R. d. S. Comparative study of global color and texture descriptors for web image retrieval. **Journal of Visual Communication and Image Representation**, v. 23, n. 2, p. 359–380, feb 2012. ISSN 10473203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1047320311001465>>. Citado na página 25.
- PONTI, M.; ESCOBAR, L. Compact color features with bitwise quantization and reduced resolution for mobile processing. In: **Global Conference on Signal and Information Processing**. Austin, TX, Estados Unidos: [s.n.], 2013. p. 751–754. Disponível em: <<http://link.springer.com/article/10.1007/s11760-011-0216-x>>. Citado na página 13.
- PONTI, M.; MASCARENHAS, N.; FERREIRA, P.; SUAZO, C. Three-dimensional noisy image restoration using filtered extrapolation and deconvolution. **Signal, Image and Video Processing**, v. 7, n. 1, p. 1–10, feb 2011. ISSN 1863-1703. Disponível em: <<http://link.springer.com/10.1007/s11760-011-0216-x>>. Citado na página 8.
- PONTI, M.; NAZARÉ, T. S.; THUMÉ, G. S. Image quantization as a dimensionality reduction procedure in color and texture feature extraction. **Neurocomputing**, v. 173, p. 385–396, jan 2016. ISSN 09252312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/>

S0925231215012771>. Citado 12 vezes nas páginas 27, 28, 30, 32, 33, 34, 35, 36, 65, 67, 69 e 82.

PONTI, M. P. Microscope Volume Segmentation Improved through Non-Linear Restoration. **International Journal of Natural Computing Research**, IGI Global, v. 1, n. 4, p. 37–46, jan 2010. ISSN 1947-928X. Disponível em: <<http://www.igi-global.com/article/microscope-segmentation-improved-through-non/52614>>. Citado na página 8.

ROCHA, A.; HAUAGGE, D. C.; WAINER, J.; GOLDENSTEIN, S. Automatic fruit and vegetable classification from images. **Computers and Electronics in Agriculture**, v. 70, n. 1, p. 96–104, jan 2010. ISSN 01681699. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S016816990900180X>>. Citado 3 vezes nas páginas 2, 29 e 69.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision**, v. 115, n. 3, p. 211–252, apr 2015. ISSN 0920-5691. Disponível em: <<http://link.springer.com/10.1007/s11263-015-0816-y>>. Citado 2 vezes nas páginas 76 e 77.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural Networks**, p. 88, oct 2014. ISSN 08936080. Disponível em: <<http://arxiv.org/abs/1404.7828>>. Citado na página 83.

SMITH, S. M.; BRADY, J. M. SUSAN—a new approach to low level image processing. **International journal of computer vision**, v. 23, n. 1, p. 45–78, 1997. Disponível em: <<http://link.springer.com/article/10.1023/A:1007963824710>>. Citado na página 15.

STEHLING, R. O.; NASCIMENTO, M. A.; FALCÃO, A. X. A compact and efficient image retrieval approach based on border/interior pixel classification. In: **Proceedings of the eleventh international conference on Information and knowledge management**. New York, USA: ACM Press, 2002. p. 102–109. ISBN 1581134924. Disponível em: <<http://dl.acm.org/citation.cfm?id=584792.584812>>. Citado na página 15.

TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: **Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)**. Narosa Publishing House, 1998. p. 839–846. ISBN 81-7319-221-9. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=710815>>. Citado na página 39.

WANG, J.; WIEDERHOLD, G. SIMPLIcity: semantics-sensitive integrated matching for picture libraries. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 23, n. 9, p. 947–963, 2001. ISSN 01628828. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=955109>>. Citado 5 vezes nas páginas 8, 28, 52, 63 e 65.

WANG, X.; HAN, T. X.; YAN, S. An HOG-LBP human detector with partial occlusion handling. **IEEE 12th International Conference on Computer Vision**, IEEE, 2009. Citado 2 vezes nas páginas 15 e 25.

XIE, J.; LU, Y.; ZHU, S.-C.; WU, Y. N. A Theory of Generative ConvNet. feb 2016. Disponível em: <<http://arxiv.org/abs/1602.03264>>. Citado na página 3.

XU, Y.; ZHANG, Z.; LU, G.; YANG, J. Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. **Pattern Recognition**, jan 2016. ISSN 00313203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320316000121>>. Citado na página 10.

YANG, C.; ZHANG, L.; LU, H.; RUAN, X.; YANG, M.-H. Saliency Detection via Graph-Based Manifold Ranking. In: **IEEE Conference on Computer Vision and Pattern Recognition**. IEEE, 2013. p. 3166–3173. ISBN 978-0-7695-4989-7. ISSN 1063-6919. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6619251>>. Citado na página 46.

ZHUO, L.; CHENG, B.; ZHANG, J. A comparative study of dimensionality reduction methods for large-scale image retrieval. **Neurocomputing**, Elsevier, v. 141, p. 202–210, oct 2014. ISSN 09252312. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0925231214004238>>. Citado na página 21.