

---

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)

T634g 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.  
82 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

---

---

adicionar os links da release do Github (doi) nas metodologias e contribuições

adicionar descrição do algoritmo na legenda

verificar se expliquei todos os métodos de quantização e extração de características utilizados

adicionar figuras de visualização do espaço correspondente a adição de cada tipo diferente de imagem gerada?



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

*(Paul Carus)*



# RESUMO

THUMÉ, G. S.. **Geração de imagens artificiais e quantização aplicadas a problemas de classificação.** 2016. 82 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. A proposta desta dissertação é melhorar a classificação de imagens utilizando métodos de processamento de imagens antes da extração de características. Especificamente analisar a influência do balanceamento de bases de dados e da quantização na classificação. Esse 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), contracenando com o espaço original. A ênfase dessa visualização se dá na facilidade de observação da importância do rebalanceamento das classes quando comparado com valores de métricas estatísticas, como a acurácia da classificação. Os resultados 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 com 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. A principal contribuição desta pesquisa é a investigação de métodos que melhorem a classificação de imagens ao obter melhores espaços de características.

**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. 82 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

An image can be represented as a combination of several features like histograms or texture properties. Those features are composed in a multidimensional vector, which represents the original image. Commonly this vector is given as input to a classification method that can indicate how much separated are the images. The literature suggests that image processing 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 color's number before feature extraction instead of applying quantization methods to vectors already extracted; and generating synthetic images by means of 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 when compared with statistical metrics. The main contribution of this research is in methods to improve image classification by obtaining a better feature space.

**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. . . . .   | 2  |
| Figura 2 – Visualização pixelizada de uma imagem da base COREL-1000. . . . .  | 8  |
| Figura 3 – Sobre a imagem RGB de entrada foram realizadas operações de borramento, realce e de equalização de histograma. A imagem à direita é resultante dessas operações. . . . .   | 9  |
| Figura 4 – Exemplo de filtragem gaussiana como operação de pré-processamento. . . .   | 11 |
| Figura 5 – A imagem original, já em escala de cinza, foi realçada utilizando o método <i>unsharp masking</i> . . . . .  | 12 |
| Figura 6 – Conversão para a escala de cinza com os métodos utilizados nessa pesquisa. O método MSB resulta em uma imagem com 8 bits (64 cores) e os outros em 256 cores. . . . .  | 14 |
| Figura 7 – Método SMOTE: interpolação entre dois exemplos vizinhos no espaço de características. . . . .  | 18 |
| 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. . . . .  | 24 |
| Figura 9 – Plano no espaço RGB, computado pelo método de conversão para escala de cinza <i>Intensidade</i> , quando um dos canais de cor (vermelho, verde ou azul) possui valor 255. Isso porque este método mapeia todas as permutações dos mesmos valores em RGB para a mesma cor. . . . .  | 25 |
| Figura 10 – Resultado da aplicação de métodos de quantização. A imagem original (a) resultou em versões de um canal de cor com 232 cores únicas para o método (e) MSB e 184 cores para os demais métodos. Ao analisar-se as barras de gradiente, assim como as paletas de cores, observa-se que os métodos <i>Luminância</i> e <i>MSB</i> conseguiram uma melhor discriminação entre cores. . . . . | 25 |
| Figura 11 – Duas imagens da base de dados <i>Caltech101</i> com variações no parâmetro de cor utilizando o método <i>MSB</i> . Da esquerda para a direita: imagem original 24-bits e suas versões quantizadas com: 256, 64, 32, 16 e 8 cores. . . . .   | 26 |
| Figura 12 – Geração artificial a classe minoritária para rebalancear as classes. . . . .  | 28 |
| Figura 13 – Geração artificial utilizando borramento com filtro bilateral. . . . .  | 29 |

|  |    |
|--|----|
| Figura 14 – Geração artificial utilizando unsharp masking. . . . .   | 30 |
| Figura 15 – Geração artificial utilizando adição de ruído de Poisson. . . . .  | 31 |
| Figura 16 – Geração artificial utilizando o método SMOTE no espaço visual. . . . .   | 33 |
| Figura 17 – Geração artificial utilizando uma mistura ponderada de duas imagens. . . . .   | 34 |
| Figura 18 – Geração artificial utilizando uma mistura limiarizada de duas imagens. . . . .   | 34 |
| Figura 19 – Geração artificial utilizando uma mistura de duas imagens a partir da saliência da primeira imagem. . . . .  | 35 |
| Figura 20 – Geração artificial utilizando uma composição de imagens. . . . .   | 37 |
| Figura 21 – Essa figura demonstra o fluxo das operações e os métodos utilizados nos experimentos. Após a aquisição da imagem, ela é convertida para escala de cinza por algum método de quantização e seus níveis de cor reduzidos por um parâmetro de quantização. Dependendo do método, a correção <i>gamma</i> é realizada. A imagem quantizada serve então como entrada para um método de extração de características e posteriormente é classificada com <i>SVM</i> . Uma das etapas de experimentos prevê também a concatenação de todos os vetores extraídos e a seleção das características com <i>LPP</i> antes da classificação. . . . . | 40 |
| Figura 22 – Bases de imagens utilizadas para os experimentos de quantização. . . . .   | 41 |
| Figura 23 – Resultados para Corel(a), Produce(b) e Caltech(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. . . . .   | 43 |
| Figura 24 – Resultados de acurácia para o método de quantização <i>MSB</i> considerando 256, 128, 64 e 32 cores com o descriptor BIC. Os boxplots em cinza correspondem às significâncias estatísticas com $p \leq 0.01$ quando comparado a acurácia de 256 cores. . . . .   | 44 |
| Figura 25 – Resultados de acurácia para o método de quantização ‘ <i>Luminância</i> ’ considerando 256, 128, 64 e 32 cores com o descriptor Haralick. Os boxplots em cinza correspondem às significâncias estatísticas com $p \leq 0.01$ quando comparado a acurácia de 256 cores. . . . .   | 44 |
| Figura 26 – Resultados de acurácia para os métodos <i>MSB</i> (quantização), <i>LPP</i> (redução de dimensionalidade) e <i>BIC</i> (extração de características). A comparação do <i>LPP</i> versus <i>MSB</i> foi realizada com a mesma dimensionalidade. Os boxplots em cinza correspondem às significâncias estatísticas com $p \leq 0.01$ quando comparado a acurácia de 256 cores. . . . .  | 45 |
| Figura 27 – Comparação da acurácia alcançada com diferentes métodos de quantização: ‘ <i>Gleam</i> , ‘ <i>Intensidade</i> ’, ‘ <i>Luminância</i> ’ e <i>MSB</i> . Inicialmente com $D = 2310$ e então reduzindo com <i>LPP</i> para $d = 1160, 582, 294$ e $150$ . . . . .   | 46 |
| Figura 28 – Comparação da acurácia com o uso da projeção <i>LPP</i> e o método <i>MSB</i> para quantização das imagens com o objetivo de redução de dimensionalidade. . . . .  | 46 |

|  |    |
|--|----|
| Figura 29 – Resultados para a projeção do LPP sobre o espaço de características produzido pelo método de quantização <i>MSB</i> utilizando 256 ( $d = 2310$ ) e 64 cores ( $d = 582$ ) . . . . .   | 47 |
| Figura 30 – Fluxo de operações para obtenção dos resultados do rebalanceamento de classes . . . . .  | 50 |
| Figura 32 – Classes <i>Cavalo</i> e <i>Elefante</i> utilizadas neste experimento. São duas classes bem discriminadas com 100 imagens cada, originalmente da base de imagens Corel. . . . .   | 53 |
| Figura 33 – Exemplo da geração artificial de imagens com o método de mistura para as classes <i>Elefante</i> e <i>Cavalo</i> da Corel-1000. . . . .  | 54 |
| Figura 34 – À esquerda a projeção dos dois componentes principais obtidos com a aplicação de PCA nas classes <i>Elefante</i> – em azul – e <i>Cavalo</i> – em verde. À direita, as mesmas classes após a remoção de 50% das imagens de treino da classe <i>Cavalo</i> . A diferença dos marcadores consiste na definição de imagens para treino e teste não existente nas classes originais. . . . . | 55 |
| Figura 35 – Resultados de <i>f1-score</i> para as classes <i>Cavalo</i> e <i>Elefante</i> da base Corel. Foi utilizado <i>BIC</i> como método de extração de características e <i>Intensidade'</i> como método de conversão em escala de cinza. Para essa combinação, a geração de imagens utilizando mistura se mostrou favorável. . . . .  | 56 |
| Figura 36 – 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. . . . .  | 57 |
| Figura 37 – Resultado do teste da classificação com 1-NN 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. . . . .   | 58 |
| Figura 38 – 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. . . . .   | 59 |
| Figura 39 – Melhores subespaços encontrados após a geração de novos exemplos para o SMOTE e 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. . . . .   | 60 |
| Figura 40 – 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. . . . .  | 61 |

|   |    |
|---|----|
| Figura 43 – Conversão em escala de cinza com MSB e HOG como método de extração de características. . . . .  | 63 |
| Figura 45 – Boxplot das classes <i>Praia</i> e <i>Montanha</i> para a conversão em escala de cinza com <i>Luminância'</i> e BIC como método de extração de características. . . . .   | 65 |
| Figura 46 – Geração artificial utilizando o método de <i>saliência</i> em duas imagens da classe <i>Praia</i> da base de imagens Corel. . . . .   | 66 |
| Figura 48 – Geração artificial utilizando o método de <i>mistura</i> em duas imagens da classe <i>Montanha</i> da base de imagens Corel. . . . .  | 68 |
| Figura 49 – Base de imagens Corel-1000. <i>Fonte: (PONTI; NAZARÉ; THUMÉ, 2016).</i> . . . . .   | 68 |
| Figura 50 – . . . . .   | 69 |
| Figura 51 – Experimento com as 10 classes da base Corel. Foi utilizado o método Luma para conversão em escala de cinza e LBP como método de extração de características. . . . .  | 69 |
| Figura 52 – Experimento com as 10 classes da base Corel. Foi utilizado o método <i>Intensidade'</i> para conversão em escala de cinza e LBP como método de extração de características. . . . .   | 70 |
| Figura 55 – Artigo publicado na <i>Neurocomputing</i> , referente a quantização de imagens.<br><i>Fonte &lt;<a href="http://www.sciencedirect.com/science/article/pii/S0925231215012771">http://www.sciencedirect.com/science/article/pii/S0925231215012771</a>&gt;</i> . . . . . | 76 |

---

# **LISTA DE ALGORITMOS**

---

|   |    |
|---|----|
| Algoritmo 1 – SMOTE: método para rebalancear classes . . . . .              | 19 |
| Algoritmo 2 – K-NN: método de classificação supervisionada . . . . .        | 20 |
| Algoritmo 3 – Geração artificial: borramento com filtro bilateral . . . . . | 29 |
| Algoritmo 4 – Geração artificial: aguçamento . . . . .                      | 30 |
| Algoritmo 5 – Geração artificial: ruído de Poisson . . . . .                | 31 |
| Algoritmo 6 – Geração artificial: SMOTE visual . . . . .                    | 32 |
| Algoritmo 7 – Geração artificial: mistura ponderada . . . . .               | 33 |
| Algoritmo 8 – Geração artificial: mistura limiarizada . . . . .             | 35 |
| Algoritmo 9 – Geração artificial: mistura saliente . . . . .                | 36 |
| Algoritmo 10 – Geração artificial: composição . . . . .                     | 37 |



# LISTA DE TABELAS

---

---

|   |    |
|---|----|
| Tabela 1 – Resultados de <i>f1-score</i> para as classes <i>Cavalo</i> e <i>Elefante</i> , utilizando <i>Gleam</i> como método para conversão em escala de cinza e <i>ACC</i> para extração de características. . . . .       | 63 |
| Tabela 2 – Resultados de <i>f1-score</i> para as classes <i>Cavalo</i> e <i>Elefante</i> , utilizando <i>MSB</i> como método para conversão em escala de cinza e <i>HOG</i> para extração de características. . . . .         | 64 |
| Tabela 3 – Resultados de <i>f1-score</i> para as classes <i>Praia</i> e <i>Montanha</i> , utilizando <i>Luminância'</i> como método para conversão em escala de cinza e <i>BIC</i> para extração de características. . . . .  | 66 |
| Tabela 4 – Resultados de <i>f1-score</i> para as classes <i>Praia</i> e <i>Montanha</i> , utilizando <i>Intensidade'</i> como método para conversão em escala de cinza e <i>HOG</i> para extração de características. . . . . | 67 |
| Tabela 5 – Resultados de <i>f1-score</i> para as 10 classes da Corel, utilizando <i>Luma</i> como método para conversão em escala de cinza e <i>LBP</i> para extração de características . . . . .                            | 70 |
| Tabela 6 – Resultados de <i>f1-score</i> para as 10 classes da Corel, utilizando <i>Intensidade'</i> como método para conversão em escala de cinza e <i>LBP</i> para extração de características . . . . .                    | 71 |
| Tabela 7 – Resultados de <i>f1-score</i> para as classes XX, utilizando <i>Luminância'</i> como método para conversão em escala de cinza e <i>HOG</i> para extração de características . . . . .                              | 72 |
| Tabela 8 – Resultados de <i>f1-score</i> para as classes <i>Tubarão</i> e <i>Peixe</i> , utilizando <i>Gleam</i> como método para conversão em escala de cinza e <i>HOG</i> para extração de características . . . . .        | 74 |



## **LISTA DE ABREVIATURAS E SIGLAS**

---

---

- ACC ..... *Auto Color Correlogram*  
BIC ..... *Border-interior classification*  
CCV ..... *Color Coherence Vector*  
GCH ..... *Global Color Histogram*  
HOG ..... *Histogram of oriented gradients*  
K-NN ..... *K-Nearest Neighbors*  
LBP ..... *Local Binary Patterns*  
LPP ..... *Locality preserving projections*  
MSB ..... *Bits Mais Significativos*  
PCA ..... *Principal Component Analysis*  
SMOTE .. *Synthetic Minority Over-sampling Technique*



# SUMÁRIO

---

---

|       |  |    |
|-------|--|----|
| 1     | <b>INTRODUÇÃO</b>                        | 1  |
| 1.1   | Contextualização                         | 3  |
| 1.2   | Hipóteses                                | 4  |
| 1.3   | Contribuições                            | 5  |
| 1.4   | Estrutura do documento                   | 5  |
| 2     | <b>FUNDAMENTOS</b>                       | 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   | Classificador de padrões                 | 19 |
| 2.5.1 | <i>K-Nearest Neighbors</i>               | 20 |
| 2.6   | Redução de dimensionalidade              | 20 |
| 2.6.1 | <i>Análise de componentes principais</i> | 21 |
| 2.6.2 | <i>Locality preserving projections</i>   | 22 |
| 2.7   | Considerações finais                     | 22 |
| 3     | <b>QUANTIZAÇÃO DE IMAGENS</b>            | 23 |
| 3.1   | Considerações iniciais                   | 23 |
| 3.2   | Quantização de imagens                   | 23 |
| 3.3   | Considerações finais                     | 26 |
| 4     | <b>GERAÇÃO ARTIFICIAL DE IMAGENS</b>     | 27 |
| 4.1   | Considerações iniciais                   | 27 |
| 4.2   | Borramento                               | 28 |
| 4.3   | Aguçamento                               | 29 |
| 4.4   | Adição de ruído                          | 30 |
| 4.5   | SMOTE visual                             | 32 |

|                    |  |           |
|--------------------|--|-----------|
| <b>4.6</b>         | <b>Mistura ponderada</b>   | <b>33</b> |
| <b>4.7</b>         | <b>Mistura limiarizada</b>                                       | <b>34</b> |
| <b>4.8</b>         | <b>Mistura saliente</b>  | <b>35</b> |
| <b>4.9</b>         | <b>Composição</b>  | <b>36</b> |
| <b>4.10</b>        | <b>Considerações finais</b>                                      | <b>38</b> |
| <b>5</b>           | <b>RESULTADOS: QUANTIZAÇÃO DE IMAGENS</b>                        | <b>39</b> |
| <b>5.1</b>         | <b>Considerações Iniciais</b>                                    | <b>39</b> |
| <b>5.2</b>         | <b>Experimentos</b>  | <b>39</b> |
| <b>5.2.1</b>       | <b><i>Base de Imagens</i></b>                                    | <b>40</b> |
| <b>5.2.2</b>       | <b><i>Protocolo</i></b>  | <b>42</b> |
| <b>5.2.3</b>       | <b><i>Resultados e Discussão</i></b>                             | <b>42</b> |
| <b>5.3</b>         | <b>Considerações Finais</b>                                      | <b>47</b> |
| <b>6</b>           | <b>RESULTADOS: GERAÇÃO ARTIFICIAL DE IMAGENS</b>                 | <b>49</b> |
| <b>6.1</b>         | <b>Considerações Iniciais</b>                                    | <b>49</b> |
| <b>6.2</b>         | <b>Experimentos</b>  | <b>49</b> |
| <b>6.2.1</b>       | <b><i>Experimento 1: duas classes bem discriminadas</i></b>      | <b>53</b> |
| <b>6.2.2</b>       | <b><i>Experimento 2: duas classes bem sobrepostas</i></b>        | <b>64</b> |
| <b>6.2.3</b>       | <b><i>Experimento 3: multiclasses</i></b>                        | <b>68</b> |
| <b>6.2.4</b>       | <b><i>Experimento 4: classes naturalmente desbalanceadas</i></b> | <b>71</b> |
| <b>6.2.5</b>       | <b><i>Experimento 5: classes com muitas imagens</i></b>          | <b>73</b> |
| <b>6.3</b>         | <b>Considerações Finais</b>                                      | <b>74</b> |
| <b>7</b>           | <b>CONCLUSÕES</b>  | <b>75</b> |
| <b>7.1</b>         | <b>Publicações</b>   | <b>76</b> |
| <b>7.2</b>         | <b>Trabalhos Futuros</b>   | <b>76</b> |
| <b>Referências</b> |  | <b>79</b> |



# 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  $256 \times 256$  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, os quais 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 extração dessas características, ao invés de lidar com a má representação das imagens. O uso desses métodos pode revelar características não visíveis nas imagens originais (*latentes*). Tais características podem melhor descrever certas classes, pois melhoram o conjunto de representações de imagens fornecidas à etapa de classificação.

Considerando que é comum realizar a extração de características a partir da imagem original, sem preocupação com a preparação da imagem, o enfoque desse estudo é na etapa de pré-processamento, destacada na Figura 1. Esta ilustra 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 — são normalmente pouco exploradas, quando comparadas com as etapas posteriores.

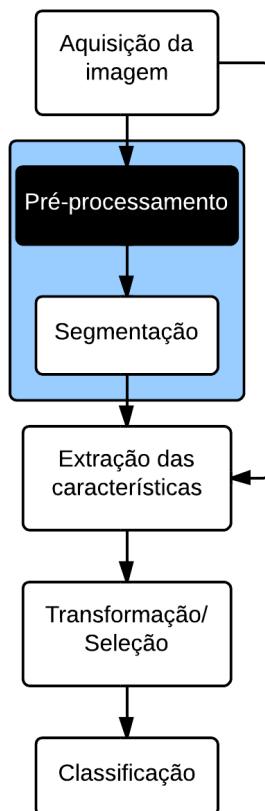


Figura 1 – Etapas canônicas do reconhecimento de padrões desde a aquisição da imagem até sua posterior classificação. As etapas de pré-processamento e segmentação — apresentadas em destaque — são normalmente pouco exploradas, quando comparadas com as etapas posteriores. O enfoque desse estudo é dar maior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

Ao invés de focar em métodos complexos de transformação do espaço de características, 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 na literatura que descrevam o método de quantização e seus parâmetros. Ao negligenciar essa etapa, perde-se a oportunidade de redução da dimensionalidade e/ou do tempo de execução. Dessa forma, essa pesquisa propõe o uso de menos de oito bits para armazenar as informações de cor (imagens quantizadas) e posteriormente extrair as características com dimensionalidade reduzida.

O desbalanceamento de classes também se apresenta como um obstáculo para que a classificação de imagens seja satisfatória. 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. 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. 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. Esse método é comparado com o *Synthetic Minority Over-sampling Technique* (SMOTE), técnica de sobreamostragem dos vetores de características ao interpolar os exemplos mais próximos.

De maneira sumária, **esta pesquisa busca melhorar a classificação de imagens, utilizando métodos de processamento de imagens com foco na quantização e no rebalanceamento de classes, ambos antes da extração de características.** Os resultados obtidos, posteriormente apresentados na Seção ??, demonstram o potencial do processamento de imagens antes da extração de características. Além disso, é fornecido uma visualização do espaço de características após o rebalanceamento das classes, crucial para analisar se as novas características extraídas são relevantes, ou seja, se adicionaram informações que estavam latentes ao aprendizado. Os resultados também demonstram que a quantização das imagens permite ao mesmo tempo obter vetores de características mais compactos e com maior capacidade de discriminação entre classes.

## 1.1 Contextualização

O grupo de pesquisa em Visualização, Imagens e Computação Gráfica (VICG), do Instituto de Ciências Matemáticas e de Computação (ICMC), tem atuado nas áreas de apoio para a classificação de coleções de imagens. Os trabalhos do grupo estão relacionados à visualização de informação com projeções multidimensionais e árvores (JOIA *et al.*, 2011), assim como à extração de características e classificação de coleções de imagens (PAIVA *et al.*, 2011). No que tange o processamento de imagens digitais, PICON, ROSSI e PONTI-JR (2011) e PONTI (2013) focam no pré-processamento para obter melhores resultados de classificação.

Ainda, Paiva *et al.* (2011) mostraram que os espaços de características formados por cor e textura podem ser melhorados, porém há um limite até o qual as características podem ser transformadas, ou selecionadas, de forma a garantir a discriminação entre as classes. Tal projeto atua na investigação de métodos que permitam gerar espaços de características com maior discriminação entre as classes, facilitando a classificação.

Em outros dois trabalhos relacionados é possível ver a diferença da performance para problemas de classificação de imagens. No primeiro, os autores atingem acurácia acima de 98% na classificação de frutas após investigar alterações nos parâmetros de aquisição, realizar pré-processamento e obter a segmentação (ROCHA *et al.*, 2010). No segundo, os autores indicam que o método utilizado para obter a imagem em escala de cinza (comumente utilizada por algoritmos de extração), pode impactar significativamente a classificação final de diversas bases de imagens (KANAN; COTTRELL, 2012).

A geração de imagens artificiais (ou sintéticas) é uma tarefa comum, mas não para o rebalanceamento de classes. Em (??), os autores utilizam um modelo generativo de uma CNN (ou ConvNet) para sintetizar imagens realistas. Essas imagens foram geradas, mas não utilizadas como entrada para um algoritmo de classificação ou no treinamento da própria rede.

## 1.2 Hipóteses

Conforme anteriormente mencionado, muitos aspectos influenciam a performance 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.

O objetivo desta pesquisa é explorar as etapas de processamento de imagens com o intuito de melhorar a discriminação entre classes de uma coleção de imagens. **Utilizar um número reduzido de cores juntamente com um método de quantização apropriado antes da extração de características pode permitir ao mesmo tempo obter vetores de características mais compactos e com maior capacidade de discriminação entre classes.** Analisando para tal, como a extração de características e posterior redução de dimensionalidade são afetados por diferentes métodos de quantização. Melhorando, assim, a classificação e diminuindo a complexidade do sistema.

Além disso, o desbalanceamento de classes é um obstáculo para uma classificação satisfatória, e por isso também é estudado. A hipótese, nesse caso, é que **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. Esse método é comparado com o SMOTE, técnica de sobreamostragem dos vetores de características ao interpolar os exemplos mais próximos.

## 1.3 Contribuições

Dados tais aspectos, pode-se então diferenciar as contribuições desta pesquisa:

### Contribuição Geral

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

### Contribuições Específicas

- Demonstrar que é possível obter vetores de características compactos e efetivos ao extrair características de imagens com reduzido níveis de intensidade a um custo computacional baixo. Reduzindo o tamanho do vetor de características de métodos de descrição de cor após a quantização e possibilidade de 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 combinação 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 comparada à geração de exemplos artificiais no espaço de atributos (SMOTE) e à classificação da base original.

## 1.4 Estrutura do documento

O conteúdo desta dissertação 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 e redução de dimensionalidade.

**Capítulo 3:** a redução do número de intensidades de cor utilizando métodos de quantização antes da etapa de extração de características é descrita.

**Capítulo 4:** descreve-se os métodos de processamento utilizados para a geração artificial de imagens — sobre as imagens originais — com o objetivo de rebalancear classes.

**Capítulo 5:** os resultados obtidos com a quantização de imagens são apresentados e discutidos.

**Capítulo 6:** os resultados obtidos com geração de imagens artificiais são apresentados e discutidos.

**Capítulo 7:** reflete sobre 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 desse 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. Alguns 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 convolução), relevantes para o desenvolvimento desta pesquisa. A extração de características é brevemente definida na Seção 2.3. Por fim, o problema do desbalanceamento de classes é apresentado na Seção 2.4. Nessa seção, além de caracterizar o problema, são apresentadas duas vertentes para solucioná-lo: sobreamostragem e subamostragem. Esses conceitos são necessários para a compreensão do capítulo de proposta desta dissertação.

### 2.2 Pré-processamento de imagens

Uma imagem digital  $I$  pode ser definida como uma função  $f(x,y)$ , onde  $x$  e  $y$  são as coordenadas cartesianas de um determinado ponto e  $f$  a intensidade (brilho) da imagem naquele ponto. 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 vermelha, 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<sup>1</sup>.

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 assim 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, resultou em uma qualidade superior para todos os testes, com valores de erro e desvio padrão menores. No referido estudo, métodos de realce causaram perda de informação e por isso não são indicados para uso em imagens obtidas por microscópio. 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 para recuperar uma imagem degradada que foi borrada por algum processo conhecido. Utiliza

<sup>1</sup> Disponível em <http://wang.ist.psu.edu/docs/related/>



Figura 3 – O pré-processamento de imagens é 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.

*Fonte: Elaborado pela autora.*

uma metodologia probabilística, baseada em EM-ML (*Expectation-Maximization Maximum Likelihood*), 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 uma 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 conectados 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 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  $*$  na Equação (2.1), é o mesmo processo, mas com o filtro previamente rotacionado em 180° ([GONZALEZ; WOODS, 2007](#)).

$$\text{Mapa de características} = \text{imagem de entrada} * \text{filtro} \quad (2.1)$$

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 a 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  $\sigma$  o desvio padrão que determina o raio da distribuição Gaussiana aplicada. Valores altos de variância fazem com que o resultado da função se aproxime da média.



Figura 4 – Exemplo de filtragem gaussiana como operação de pré-processamento. *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 dissennir 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. Soma 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 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.



(a) Original

(b) Unsharp masking

Figura 5 – A imagem original, já em escala de cinza, foi realçada utilizando o método *unsharp masking*. *Fonte: Elaborado pela autora.*

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 outros problemas com a captura. Dessa forma, o processo de “esticar” o contraste expande os níveis de intensidade da imagem (GONZALEZ; WOODS, 2007).

#### Explicar melhor histograma e k

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  $n_k$  é o número de pixels de intensidade  $i_k$ . 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

$$s_k = T(i_k) = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad (2.2)$$

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 apenas para receber imagens de entrada 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 concatenados. 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 (24 bits) em apenas um (8 bits). [Kanan e Cottrell \(2012\)](#) demonstraram que os métodos para a quantização (conversão de uma imagem colorida para escala de cinza) influenciam a performance no 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 nessa dissertação foram escolhidos com base em [Kanan e Cottrell \(2012\)](#): *Gleam* e as versões corrigidas por *gamma* de *Intensidade* e *Luminância*. A operação *gamma* utilizada  $z' = z^{1/2.2}$  é a padrão.

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

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

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:

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

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

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

$$Q_{\text{Luminância}} = 0.299R + 0.587G + 0.114B.$$

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

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

$$Q_{\text{Luma}} = 0.2126R' + 0.7152G' + 0.0722B'.$$

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

**Bits Mais Significativos (MSB):** ao invés de realizar uma combinação linear dos canais de cores, ordena os bits dos canais coloridos em um único canal. Computa quantos bits de cada canal de cor contribuem para a imagem final e extrai os bits do código binário dos canais originais ([PONTI; ESCOBAR, 2013](#)).

A Figura 6 apresenta a conversão na escala de cinza obtida com o uso destes métodos.

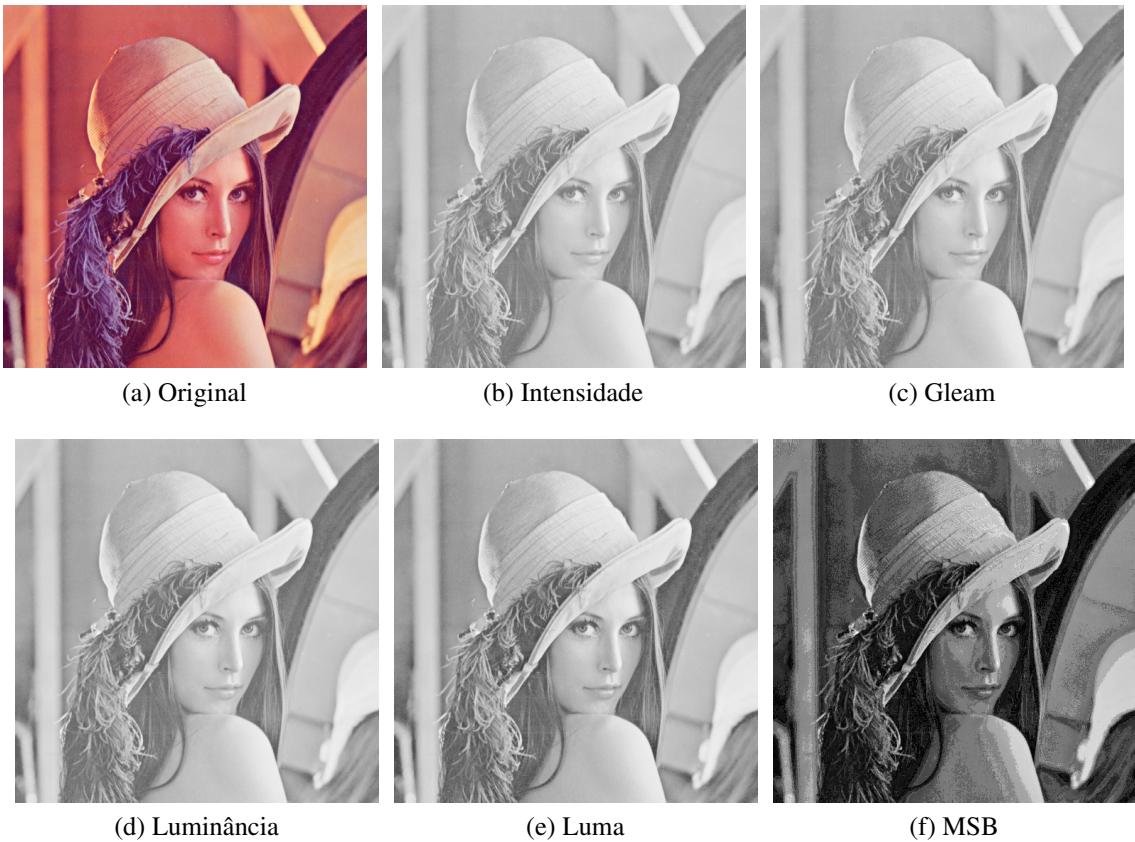


Figura 6 – Conversão para a escala de cinza com os métodos utilizados nessa pesquisa. O método MSB resulta em uma imagem com 8 bits (64 cores) e os outros em 256 cores. *Fonte: Elaborado pela autora.*

## 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. Em aplicações que envolvem a classificação de algas, uma informação muito importante para a discriminação entre classes é a forma ([BORGES et al., 2013](#)). Essas características 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, 2009a), curvatura, 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 da Seção ?? 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*. Classifica os pixels da imagem em coerentes e incoerentes, computa os respectivos histogramas e os concatena (HUANG *et al.*, 1997). 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. Borra levemente a imagem ao substituir os pixels pela média do pixel e seus oito vizinhos;
2. Discretiza o espaço de cor de forma que a imagem contenha apenas  $n$  cores distintas;
3. Classifica os pixels entre coerentes e incoerentes: se o tamanho do seu componente conectado for maior ou igual um dado *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. Os pixels são classificados 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 (PASS; ZABIH; MILLER, 1996). O vetor resultante consiste na concatenação dos auto-correlogramas, um para cada distância. Neste estudo, são consideradas quatro distâncias: 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 vetor resultante possui seis dimensões que representam as seguintes características:

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

**Histogram of oriented gradients (HOG)**: (DALAL; TRIGGS, 2005).

1. Divide a janela da imagem em células;
2. Computa os gradientes;
3. Cada pixel calcula uma ponderação para um canal do histograma de orientação de bordas baseado na orientação do gradiente do elemento em que esta centrado. Esses valores são acumulados em bins sobre as regiões espaciais de células e formam o histograma;
4. Os votos são interpolados bilinearmente entre os centros de vizinhança do bin em orientação e posição;

5. Normaliza o contraste dos blocos da janela que se sobrepõem. Dessa forma cada célula é normalizada em relação a diferentes blocos;
6. Concatena os histogramas de todas as células.

**Local Binary Patterns (LBP) utilizando padrões uniformes:** baseia-se em reconhecer que padrões de textura uniformes são propriedades fundamentais da textura local da imagem e o histograma da sua ocorrência provou-se 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 janela da imagem em células;
2. Para cada pixel em uma célula, compara com seus vizinhos. Esse passo resulta em um código binário de oito dígitos;
3. Computa o histograma da célula;
4. Normaliza o histograma;
5. Concatena os histogramas de todas as células.

## 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 (*oversampling*) 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 desbalanceados. 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; 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 não é um problema, mas em conjunto com a sobreposição de classes pode diminuir significativamente a acurácia da classificação da classe minoritária. Os resultados reportados também mostram que a poda de árvores de decisão raramente levou a alguma melhora na classificação.

(CASTRO; 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 – como 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). Esta pesquisa tem como enfoque o **pré-processamento dos dados**, com um viés no rebalanceamento de classes através da **geração de imagens artificiais** (antes da extração de características).

### 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; 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 SMOTE *Synthetic Minority Over-sampling Technique* é um método desenvolvido por (CHAWLA; HALL; BOWYER, 2002) para rebalancear classes ao gerar artificialmente novos elementos, ao invés de apenas replicá-los. 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 número  $0 \leq x \leq 1$ . Esse valor é adicionado ao vetor original, criando um novo elemento. 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.

---

#### Algoritmo 1: SMOTE: método para rebalancear classes

---

**Entrada:** Imagem colorida  $I$  em formato BGR  
**Saída:** Exemplos sintéticos  $S$

- 1  $N \leftarrow$  vizinhos(minoritária);
- 2 **para cada** exemplo **faça**
- 3      $nn \leftarrow$  vizinho aleatório de  $N$
- 4     **para cada** característica  $(x,y)$  **faça**
- 5         diferença  $\leftarrow nn(x,y) - exemplo(x,y);$
- 6         gap  $\leftarrow$  número aleatório entre 0 e 1;
- 7          $G(x,y) \leftarrow exemplo(x,y) + gap * diferença;$
- 8     **fim**
- 9      $S \leftarrow S \cup G;$
- 10 **fim**

---

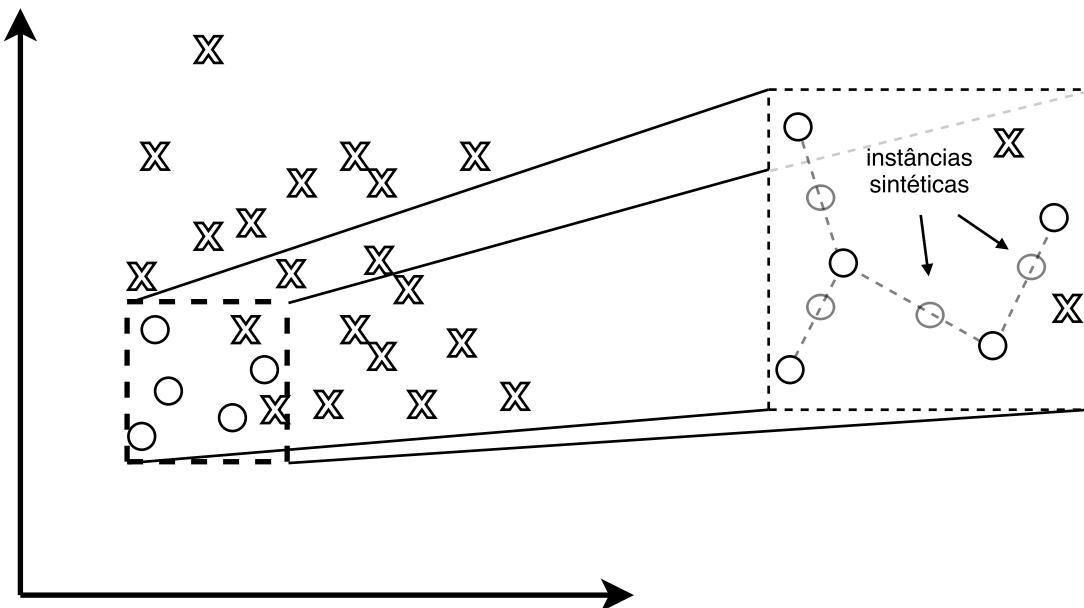


Figura 7 – Método SMOTE: interpolação entre dois exemplos vizinhos no espaço de características. *Fonte: Elaborado pela autora.*

Dessa forma, o SMOTE provê mais elementos para o classificador aprender, ao contrário da replicação de dados. Como trabalhos futuros, os autores [Chawla, Hall e Bowyer \(2002\)](#) apontam que diferentes estratégias para criar esses exemplos sintéticos podem melhorar a performance da classificação. Inclusive salientando exemplos que foram errôneamente classificados.

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 em quais elementos 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 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; num experimento são preditas as classes de um conjunto de imagens de teste.

### 2.5.1 K-Nearest Neighbors

O classificador *K-Nearest Neighbors* (K-NN) considera a proximidade entre os dados para realizar 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).

---

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

---

Esse classificador foi utilizado por ser suscetível à diferenciação entre as classes. Com  $K = 1$ , a previsão da classe corresponde ao exemplo mais próximo.

## 2.6 Redução de dimensionalidade

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 (inclusive antes de imagens serem removidas para provocar o desbalanceamento). 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 (entre SMOTE ou geração de imagens no campo visual) mais se assemelha à distribuição original dos dados.

Considerando que um vetor de características extraído com extratores comuns pode possuir entre 6 (e.g. Haralick) e 512 (e.g. BIC) características, a visualização de um exemplo requer que seja realizado o mapeamento desses valores em apenas duas dimensões. Para isso, uma redução de dimensionalidade mapeia os dados de  $N$  dimensões para um espaço 1D, 2D ou 3D. A partir desses novos dados, pode então ser criada alguma representação visual que tente manter a relação de distância entre os novos e os originais ([PAULOVICH; OLIVEIRA; MINGHIM, 2007](#)).

### **2.6.1 Análise de componentes principais**

O *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. Essa redução permite a projeção dessas imagens no espaço de características. O objetivo é extrair as informações mais importantes dos dados e representá-las 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 sendo:

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

$$C_x = XX^T, \quad (2.3)$$

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

3. Encontra os autovalores e autovetores de  $C_x$ . Um autovetor  $u$  de uma matriz  $A$  pode ser definido por  $Au = \lambda u$ , onde  $\lambda$  é um autovalor escalar associado ao autovetor. Um vetor  $u$  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  $\lambda$  (com outros valores como zero) e o conjunto dos autovetores de  $A$  em uma matriz  $U$ . Assim,

$$A = U\lambda U^T; \quad (2.4)$$

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  $X$ .

## 2.6.2 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 te 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; ZHANG, 2014) o LPP alcançou 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 eles 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 perto, ou seja, conectados por uma aresta e  $W_{ij} = 0$  caso contrário;
3. Computa os autovalores e autovetores

$$XLX^T a = \lambda XDX^T a, \quad (2.5)$$

onde  $D$  é a matriz diagonal na qual a entrada são as somas das colunas de  $W$

A matriz de projeção corresponde aos autovetores, sendo os d menores autovalores.

## 2.7 Considerações finais

Deu-se destaque à discussão das etapas de pré-processamento e realce de características latentes, ambas foco deste estudo, assim como a geração de imagens artificiais para o balançamento de classes. Este capítulo apresentou diversos métodos para exemplificação, além de trabalhos similares.

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. A geração dessas imagens visa rebalancear classes que diferem em número de imagens, e detalhes sobre esse problema também foram fundamentados nesse capítulo.

Esses fundamentos permitem compreender em que contexto esta dissertação de mestrado está inserida. O próximo capítulo abrangerá a proposta deste trabalho.





# QUANTIZAÇÃO DE IMAGENS

## 3.1 Considerações iniciais

Sistemas de reconhecimento de imagens comumente utilizam uma imagem em níveis de cinza (8-bits – 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 \(2012\)](#), com exceção dos métodos HOG e LBP, também descritores frequentemente utilizados na literatura ([WANG; HAN; YAN, 2009b](#)).

Este capítulo descorre sobre a quantização das imagens antes da extração de características, assim como os métodos utilizados, apresentados na Seção [2.2.3](#).

## 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. Obtem-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 apresentados na Seção [2.3](#) para extrair as características. A Figura [8](#) ilustra esses passos, desde a aquisição até a classificação das imagens.

Cada método de quantização se comporta diferentemente para uma dada imagem RGB. Por exemplo, o método *Intensidade* mapeia todas as permutações dos mesmos valores em

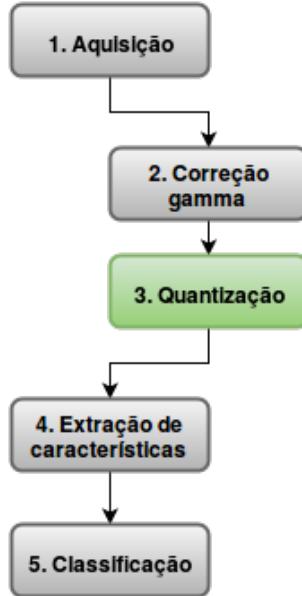


Figura 8 – O pipeline de reconhecimento de imagens pode envolver uma etapa de converter imagens coloridas em imagens em escala de cinza, obtendo uma imagem quantizada que pode ser então processada por métodos de extração de características. O vetor com essas características é então dado como entrada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

RGB para a mesma cor. Dessa forma, produz um plano no espaço RGB conforme mostrado na Figura 9. O efeito do método *Gleam* é similar, mas dada a natureza da função *gamma*, cobre uma superfície curva. Tal resultado também é alcançado utilizando o método *Intensidade*. 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 as cores próximas, podendo aumentar a confusão entre objetos. Os métodos *Luminância* e *Luma* procuram aprimorar a 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 as 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 tais métodos, consulte Seção 2.2.3.

**descrever melhor o pq da superfície curva**

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 discriminá-las. Além disso, o mapa de cores *MSB* obteve um maior número de cores únicas, quando comparado aos demais métodos.

Complementando, a Figura 11 apresenta mais um exemplo de redução de cores utilizando

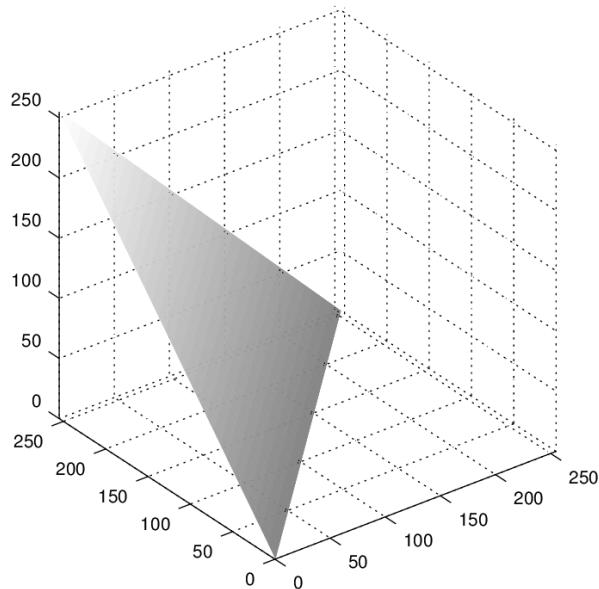


Figura 9 – Plano no espaço RGB, computado pelo método de conversão para escala de cinza *Intensidade*, quando um dos canais de cor (vermelho, verde ou azul) possui valor 255. Isso porque este método mapeia todas as permutações dos mesmos valores em RGB para a mesma cor. *Fonte: (PONTI; NAZARÉ; THUMÉ, 2016).*

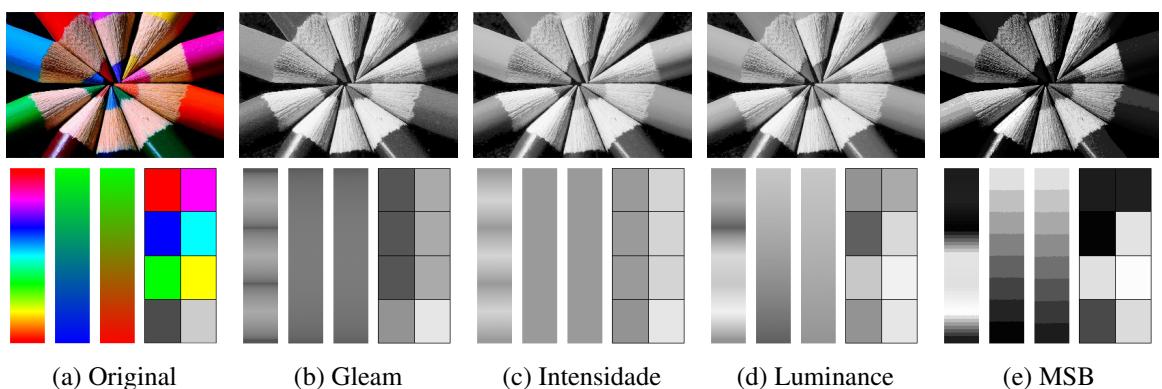


Figura 10 – Resultado da aplicação de métodos de quantização. A imagem original (a) resultou em versões de um canal de cor com 232 cores únicas para o método (e) MSB e 184 cores para os demais métodos. Ao analisar-se as barras de gradiente, assim como as paletas de cores, observa-se que os métodos *Luminância* e *MSB* conseguiram uma melhor discriminação entre cores. *Fonte: (PONTI; NAZARÉ; THUMÉ, 2016).*

o método *MSB* para um par de imagens da base de dados *Caltech-101*<sup>1</sup>. 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.

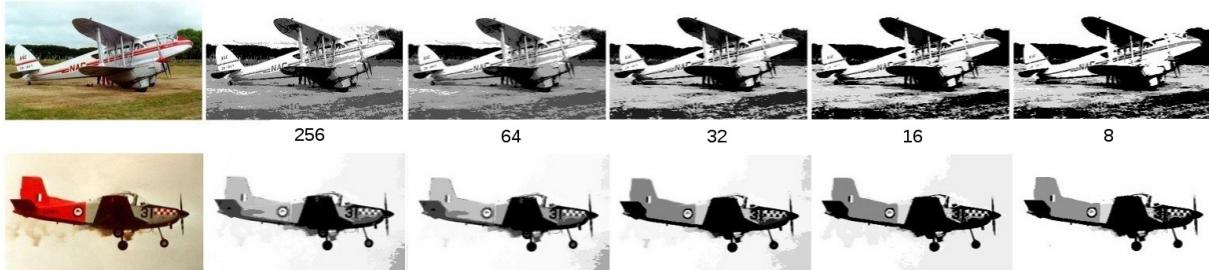


Figura 11 – Duas imagens da base de dados *Caltech101* com variações no parâmetro de cor utilizando o método *MSB*. Da esquerda para a direita: imagem original 24-bits e suas versões quantizadas com: 256, 64, 32, 16 e 8 cores. *Fonte:* ([PONTI; NAZARÉ; THUMÉ, 2016](#)).

### 3.3 Considerações finais

Ao aplicar a quantização na etapa de pré-processamento é esperado a redução da dimensionalidade do vetor de características no início do sistema, beneficiando todas as etapas posteriores. A hipótese é que utilizar um número reduzido de cores vai reduzir significativamente a dimensionalidade, enquanto melhora ou mantém a classificação do sistema.

Espera-se, também, que o método MSB para a quantização tenha uma performance melhor, dada a preservação das cores observada na Figura 11. Os resultados da utilização dos métodos descritos neste capítulo estão no Capítulo 5.

<sup>1</sup> Disponível em <<http://www.vision.caltech.edu/ImageDatasets/Caltech101/>>



# GERAÇÃO ARTIFICIAL DE IMAGENS

---

## 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 tipo de 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 borramento – pode ocasionar perda da textura. Assim, justifica-se a exploração de um vasto número de métodos de processamento de imagens e bases.

Nesta pesquisa oito métodos de processamento de imagens são aplicados nas imagens minoritárias originais, gerando imagens artificiais a partir destas. Isso é realizado a fim de permitir a extração de informações latentes com o objetivo de melhorar a classificação com alguma técnica de Aprendizado de Máquina, o que reflete a melhora da diferenciação entre as classes. Dada a quantidade de imagens necessárias para rebalancear a base original, são geradas imagens utilizando cada um dos métodos, além de uma versão combinando todos eles (ou seja, compondo um conjunto com algumas imagens processadas por cada método) e outra apenas replicando as imagens como *baseline*. Como demonstrado na Figura 12, 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 e posteriormente essas imagens resultantes são utilizadas como treinamento.

Neste capítulo os métodos de geração artificial para o rebalanceamento de classes de imagens são descritos. Os experimentos posteriormente destacados no Capítulo ?? foram realizados utilizando: borramento; mistura ponderada; *unsharp masking*; composição; combinação de thresholds; combinação com saliência; visual SMOTE; e adição de ruído.

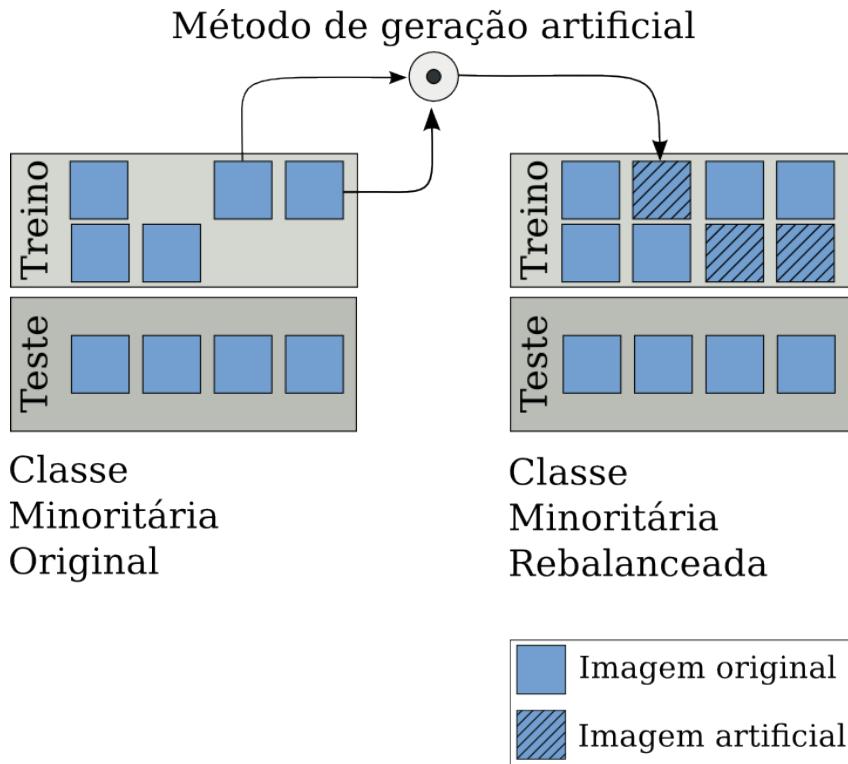


Figura 12 – Geração artificial a classe minoritária para rebalancear as classes. *Fonte: Elaborado pela autora.*

## 4.2 Borramento

Também conhecido como filtro de suavização, o borramento é uma operação de processamento comumente utilizada com o objetivo de filtrar as frequências de uma imagem, removendo ruídos e detalhes não relevantes. Normalmente esse tipo de filtro provoca também um certo borramento das bordas, como pode ser observado na Figura 4.

Esse comportamento não é esperado quando devemos gerar novas imagens, pois informações relevantes podem ser removidas. Dessa forma, a operação de borramento utilizada é a de filtro bilateral. Ela substitui o valor do pixel  $(x, y)$  por uma média dos pixels de intensidade similar na imagem e dos pixels vizinhos (TOMASI; MANDUCHI, 1998). Ou seja, é uma média ponderada das intensidades. O Algoritmo 3 descreve os passos desse filtro na sua versão mais simples, de força bruta. A Figura 13 exemplifica o seu funcionamento: à esquerda está demonstrada a imagem original e à direita a imagem borrada.

**Parâmetros e suas variações** Conforme descrito no Algoritmo 3, os parâmetros para essa geração são: o diâmetro  $d$  de pixinhança de pixels, o  $\sigma$  do espaço de cor e o  $\sigma$  do espaço de coordenadas. Esses parâmetros dependem das propriedades das imagens e dos resultados pretendidos. Dessa forma, o tamanho do filtro é um valor escolhido arbitrariamente para cada aplicação em específico (TOMASI; MANDUCHI, 1998). Como o nosso objetivo com

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

---

**Entrada:** Imagem colorida  $I$  em formato BGR  
**Entrada:** Diâmetro  $d$  de pixinhança de pixels  
**Entrada:** Sigma do espaço de cor  
**Entrada:** Sigma do espaço de coordenadas  
**Saída:** Imagem gerada  $BF$

```

1 para cada pixel  $(x,y)$  faça
2    $BF(x,y) \leftarrow 0;$ 
3    $W(x,y) \leftarrow 0;$ 
4   para cada pixel  $(i,j)$  faça
5      $w \leftarrow G_{\sigma_s}(\|(x,y) - (i,j)\|)G_{\sigma_x}(|I(x,y) - I(i,j)|);$ 
6      $BF(x,y) \leftarrow BF(x,y) + wI(i,j);$ 
7      $W(x,y) \leftarrow W(x,y) + w;$ 
8   fim
9    $BF(x,y) \leftarrow I(x,y)/W(x,y);$ 
10 fim
11 
```

---

[Conferir algoritmo](#)



(a) Original



(b) Imagem artificial

Figura 13 – Geração artificial utilizando borramento com filtro bilateral. *Fonte: Elaborado pela autora.*

a geração das imagens não foi especializar no comportamento de uma classe de imagens específica, um valor foi escolhido aleatoriamente e a partir dele os parâmetros de entrada foram definidos.

**Limitações** Esse filtro tende a remover texturas e a 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, a de mediana e a de médias.

## 4.3 Aguçamento

Diferentemente da suavização, o processamento de aguçamento procura enfatizar as transições de intensidade. Um processo 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 (Ver Algoritmo 4). A imagem resultante, ilustrada na Figura 14, é uma versão realçada da imagem original, dado que soma à imagem justamente o que é removido com um filtro de suavização.

---

### Algoritmo 4: Geração artificial: aguçamento

---

**Entrada:** Imagem colorida  $I$  em formato BGR

**Saída:** Imagem gerada  $G$

```

1 borrada  $\leftarrow$  filtro de suavização( $I$ ) para cada pixel  $(x,y)$  faça
2   |   diferença  $\leftarrow I(x,y) - \text{borrada}(x,y);$ 
3   |    $G(x,y) \leftarrow I(x,y) + k * \text{diferença};$ 
4 fim

```

---



(a) Original



(b) Imagem artificial

Figura 14 – Geração artificial utilizando unsharp masking. *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 de aguçamento conhecidos são: utilizar primeira derivada (grandiente) ou a segunda derivada da imagem (Laplaciano).

## 4.4 Adição de ruído

O ruído de Poisson ocorre na contagem de fótons de dispositivos ópticos. Ele segue a distribuição de Poisson, que expressa a probabilidade de um certo número de eventos ocorrer em um intervalo fixo de tempo e/ou espaço se esses eventos ocorrem com uma taxa média conhecida. O efeito da adição de ruído pode ser visto na Figura 15.



(a) Original



(b) Imagem artificial

Figura 15 – Geração artificial utilizando adição de ruído de Poisson. *Fonte: Elaborado pela autora.*

### descrever o algoritmo

A distribuição de Poisson segue a equação:

Uma possível implementação para encontrar os valores de Poisson foi desenvolvida por Knuth e pode ser vista no Algoritmo 5.

---

#### Algoritmo 5: Geração artificial: ruído de Poisson

---

**Entrada:** Imagem colorida  $I$  em formato BGR

**Saída:** Imagem gerada  $G$

```

1 para cada pixel  $(x,y)$  faça
2    $L \leftarrow \exp(-I(x,y));$ 
3    $p \leftarrow 1;$ 
4    $k \leftarrow 0;$ 
5   faça
6      $| k \leftarrow k + 1;$ 
7      $| p \leftarrow p * \text{número aleatório entre } 0 \text{ e } 1;$ 
8   enquanto  $p > L;$ 
9    $| G(x,y) \leftarrow k - 1;$ 
10 fim
```

---

### Parâmetros e suas variações

**Limitações** A adição de ruído é normalmente indesejável e a utilizamos para englobar mais um processamento de imagens. (??)

## Métodos relacionados

### 4.5 SMOTE visual

Conforme visto na Seção 2.4.1, o SMOTE é um método de rebalanceamento aplicado após a extração de características. A ideia dessa geração, chamada de SMOTE visual, é imitar esse funcionamento 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 adicionado na imagem original (Ver Algoritmo 6). O efeito que esse processamento causa na imagem pode ser visualizado na Figura 16.

---

#### Algoritmo 6: Geração artificial: SMOTE visual

---

**Entrada:** Imagem colorida  $I$  em formato BGR  
**Entrada:** Imagem colorida  $I_2$  em formato BGR  
**Saída:** Imagem gerada  $G$

```

1 para cada pixel  $(x, y)$  faça
2   |   diferença  $\leftarrow I(x, y) - I_2(x, y);$ 
3   |   gap  $\leftarrow$  número aleatório entre 0 e 1;
4   |    $G(x, y) \leftarrow I(x, y) + gap * \text{diferença};$ 
5 fim
6 mínimo  $\leftarrow$  menor valor de  $G;$ 
7 máximo  $\leftarrow$  maior valor de  $G;$ 
8 para cada pixel  $(x, y)$  faça
9   |    $G(x, y) \leftarrow G(x, y) - \text{mínimo};$ 
10 fim
11 para cada pixel  $(x, y)$  faça
12   |    $G(x, y) \leftarrow G(x, y) * (255 / (\text{máximo} - \text{mínimo}));$ 
13 fim

```

---

**Nova imagem**

**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.



Figura 16 – Geração artificial utilizando o método SMOTE no espaço visual. *Fonte: Elaborado pela autora.*

## 4.6 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 17, onde dadas duas imagens como entrada, a imagem da direita corresponde a soma delas.

---

### **Algoritmo 7:** Geração artificial: mistura ponderada

---

**Entrada:** Primeira imagem colorida  $I$  em formato BGR

**Entrada:** Segunda imagem colorida  $I_2$  em formato BGR

**Saída:** Imagem gerada  $G$

- 1  $\alpha \leftarrow$  número aleatório entre 10 e 80;
  - 2  $\beta \leftarrow 100 - \alpha;$
  - 3 **para cada** pixel  $(x,y)$  **faça**
  - 4   |  $G(x,y) \leftarrow \beta.I(x,y) + \alpha.I_2(x,y);$
  - 5 **fim**
- 

**Parâmetros e suas variações** Os parâmetros  $\alpha$  e  $\beta$  são escolhidos de forma aleatória. Um valor



Figura 17 – Geração artificial utilizando uma mistura ponderada de duas imagens. *Fonte: Elaborado pela autora.*

entre 10% e 80% é escolhido para  $\alpha$ ; e o  $\beta$  é o 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 threshold e saliência descritos a seguir.

## 4.7 Mistura limiarizada

A combinação de *thresholds* é uma composição do fundo (*background*) de uma imagem e do objeto da cena (*foreground*) de outra imagem. A Figura 18 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.



Figura 18 – Geração artificial utilizando uma mistura limiarizada de duas imagens. *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.

**Algoritmo 8:** Geração artificial: mistura limiarizada

**Entrada:** Imagem colorida  $I$  em formato BGR

**Entrada:** Imagem colorida  $I_2$  em formato BGR

**Saída:** Imagem gerada  $G$

- 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  $G \leftarrow I_{background} + I_{foreground}$ ;

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

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

## 4.8 Mistura saliente

A combinação de regiões salientes é muito similar com o método anterior de combinação de *thresholds*, porém, utiliza um algoritmo mais robusto que detecta a saliência da imagem a partir do método SLIC. A Figura 19 mostra a combinação da região saliente da imagem original à esquerda com a imagem central, resultando na imagem combinada à direita.

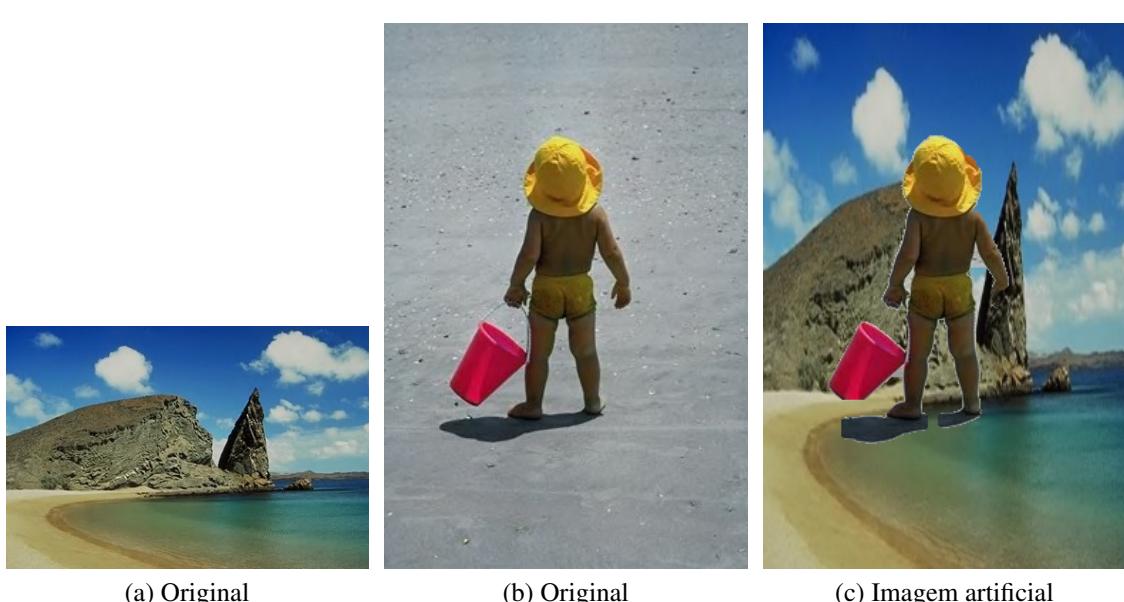


Figura 19 – Geração artificial utilizando uma mistura de duas imagens a partir da saliência da primeira imagem.  
Fonte: Elaborado pela autora.

As operações aplicadas na imagem para extrair a região mais saliente são: 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:** Imagem colorida  $I$  em formato BGR

**Entrada:** Imagem colorida  $I_2$  em formato BGR

**Saída:** Imagem gerada  $G$

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

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

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

**Métodos relacionados** Similar à mistura por *thresholds*.

## 4.9 Composição

Essa geração pretende compor informações de diversas imagens em uma única imagem. Assim é feito um mosaico com várias imagens, conforme pode ser visto na Figura 20. Se as imagens possuem um elemento centralizado, essa geração pode resultar em uma imagem de um objeto que parece uma mutação dos objetos centrais, conforme pode ser visualizado na Figura ??.

**figura da mistura central**

Para cada quadrado a ser preenchido sorteia uma imagem do conjunto de treinamento, realiza uma operação de borramento, aguçamento, mistura ponderada ou visual SMOTE e adiciona essa imagem no quadrado respectivo. Os passos para tal composição estão descritos no Algoritmo 10.



Figura 20 – Geração artificial utilizando uma composição de imagens. *Fonte: Elaborado pela autora.*

---

#### **Algoritmo 10:** Geração artificial: composição

---

**Saída:** Imagem gerada  $G$

- 1 **enquanto**  $total < \text{número de quadrados } q$  **faça**
- 2      $I \leftarrow$  imagem aleatória do conjunto de treinamento;
- 3     operação  $\leftarrow 1 + (\text{rand}() \% 3)$ ;
- 4     **selecione** 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              $| I \leftarrow \text{visual SMOTE}(I);$
- 13     **fim**
- 14      $x \leftarrow$  posição aleatória em x de I;
- 15      $y \leftarrow$  posição aleatória em y de I;
- 16      $qx \leftarrow$  posição atual para o quadrado em x de G;
- 17      $qy \leftarrow$  posição atual para o quadrado em y de G;
- 18      $G(qx, qy) \leftarrow I(x, y);$
- 19      $total ++;$
- 20 **fim**

---

**Parâmetros e suas variações** O parâmetro  $q$  controla quantos quadrados serão criados na nova imagem. Nesta pesquisa foram realizados testes com 4 e 16.

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

### Métodos relacionados

## 4.10 Considerações finais

Esse capítulo descreveu como o rebalanceamento é realizado: gerar imagens para a classe (ou classes) minoritárias e utilizá-las para o treinamento. Os métodos para gerar tais imagens foram apresentados e exemplificados. Foram também descritos os algoritmos, parâmetros utilizados, limitações e métodos relacionados a cada um dos métodos.



# RESULTADOS: QUANTIZAÇÃO DE IMAGENS

---



---

## 5.1 Considerações Iniciais

Este capítulo apresenta os resultados encontrados quando os métodos de quantização de imagens foram aplicados no pipeline de reconhecimento de imagens. Para cada experimento realizado, são descritos: a base de imagens, o protocolo utilizado, os resultados encontrados e a discussão da relevância de tais resultados.

Os experimentos são relacionados à quantização de imagens, quando realizada antes da extração de características, e portanto no campo visual. Os resultados devem refletir melhorias nas etapas subsequentes, como uma melhor acurácia na etapa de classificação ou a redução do tempo de processamento.

## 5.2 Experimentos

O objetivo dessa seção é mostrar os efeitos da etapa de quantização e como ela pode ser utilizada para reduzir a dimensionalidade do espaço de características. A Figura 21 demonstra o fluxo das operações e os métodos utilizados nos experimentos. Inicialmente, as imagens foram quantizadas em 256, 128, 64, 32 e 16 cores. Após, suas características foram extraídas e duas etapas 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.



Figura 21 – Essa figura demonstra o fluxo das operações e os métodos utilizados nos experimentos. Após a aquisição da imagem, ela é convertida para escala de cinza por algum método de quantização e seus níveis de cor reduzidos por um parâmetro de quantização. 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.*

### 5.2.1 Base de Imagens

Três bases de imagens, ilustradas na Figura 22, foram utilizadas nestes experimentos:

**Corel-1000<sup>1</sup>:** consiste em dez classes balanceadas de imagens naturais, com algumas classes bem definidas e algumas não;

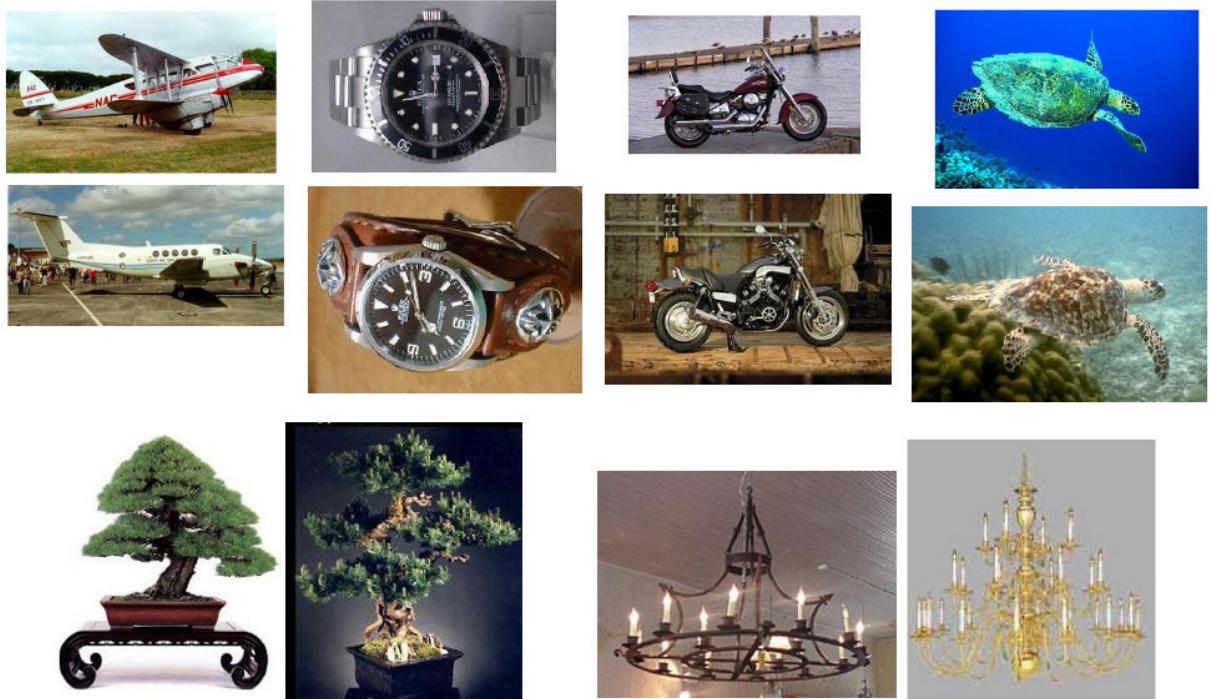
**Caltech101-600<sup>2</sup>:** contém fotos e desenhos. Desta base, foi utilizado um conjunto de seis classes balanceadas: aviões, bonsais, candelabros, tartarugas, motocicletas e relógios;

**Produce<sup>3</sup>** (também conhecido como base de vegetais e frutas tropicais): composta por imagens com um fundo similar mas muitas mudanças na iluminação, no número de objetos e na escala. Apesar da oclusão parcial de objetos ser observada, essa classe possui dados bem comportados.

<sup>1</sup> Disponível em <http://www.wang.ist.psu.edu/docs/related/>

<sup>2</sup> Disponível em <http://www.vision.caltech.edu/ImageDatasets/Caltech101/>

<sup>3</sup> Disponível em ??



(a) Base de imagens Caltech101



(b) Base de imagens Corel-1000



(c) Base de imagens Produce

Figura 22 – Bases de imagens utilizadas para os experimentos de quantização. *Fonte: (PONTI; NAZARÉ; THUMÉ, 2016).*

Estes experimentos possuem foco na redução na dimensionalidade. Para evitar o problema do desbalanceamento, as bases *Produce* e *Caltech101* foram modificadas. Dessa forma, as classes disponíveis foram balanceadas ao remover imagens das classes majoritárias.

### 5.2.2 Protocolo

Os experimentos foram realizados com uma validação cruzada de *10-fold*. Considerando que as bases estão平衡adas e que a seleção de exemplos para a validação cruzada é estratificada, a medida estatística de *acurácia* foi utilizada para avaliar a performance 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:
  - *ACC*: utilizando um conjunto de quatro distâncias  $D = 1, 3, 5, 7$  e a distância tabuleiro de xadrez  $D_8(p, q) = \text{Max}(|x - s|, |y - t|)$  entre os pixels  $p(x, y)$  e  $q(s, t)$ ;
  - *BIC*: com uma vizinhança de quatro pixels;
  - *CCV*: adotando um valor de *threshold* = 25 para a classificação dos pixels entre coerentes e incoerentes;
  - *Haralick-6*: o pixel vizinho para o qual computar a matriz de correlação foi definido como sendo o pixel à direita.
3. **Redução da dimensionalidade:** a projeção 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:** com o classificador SVM (*Support Vector Machines*). Os parâmetros para essa etapa foram encontrados utilizando uma *grid search* no conjunto de treino.

### 5.2.3 Resultados e Discussão

A Figura 23 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 uma impacto significativo na acurácia da classificação. Além disso, a redução de 256 para um menor número de cores normalmente manteve as acurácia e em alguns casos resultou em uma ligeira melhora, especialmente para os níveis de 128 e 64.

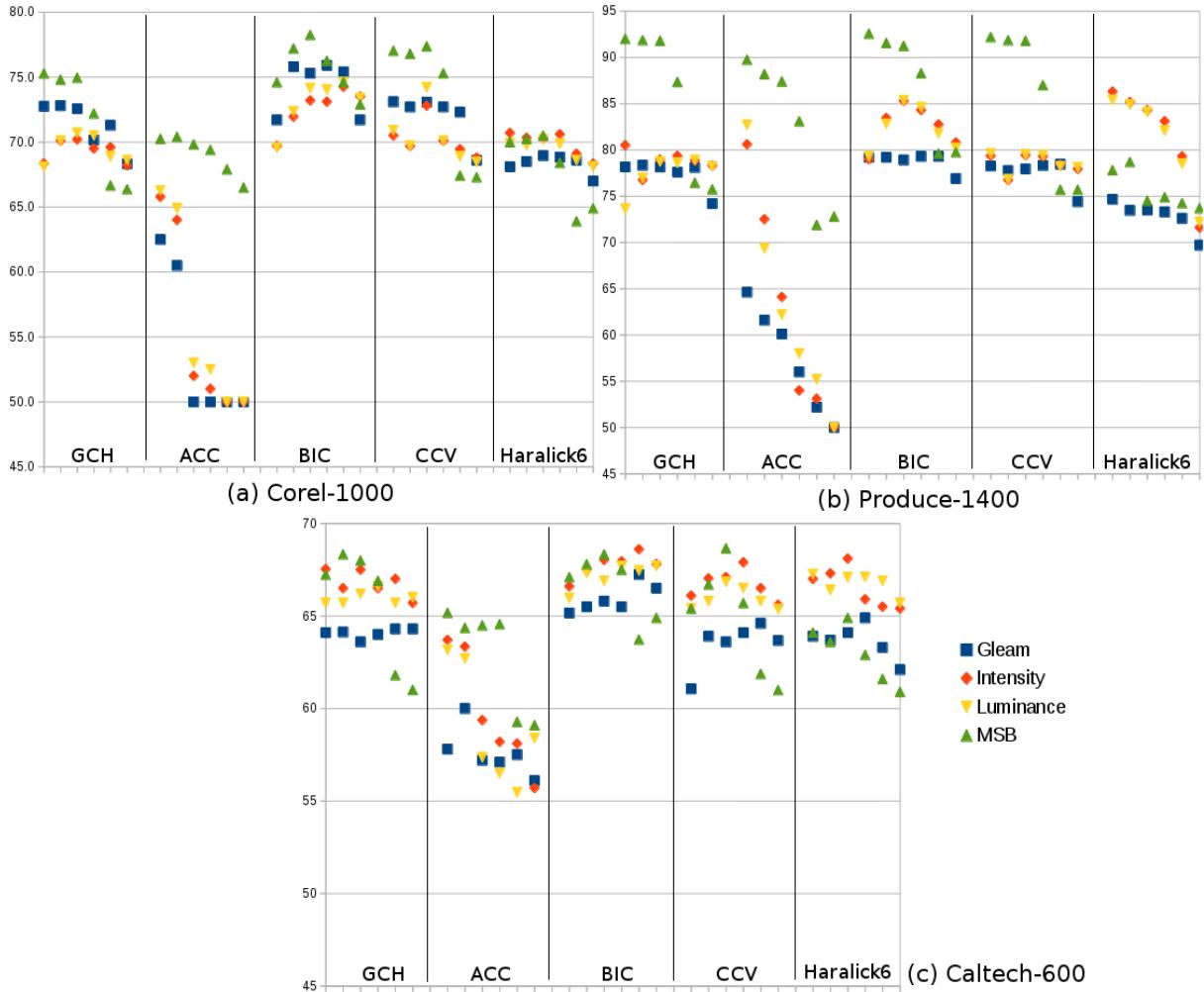


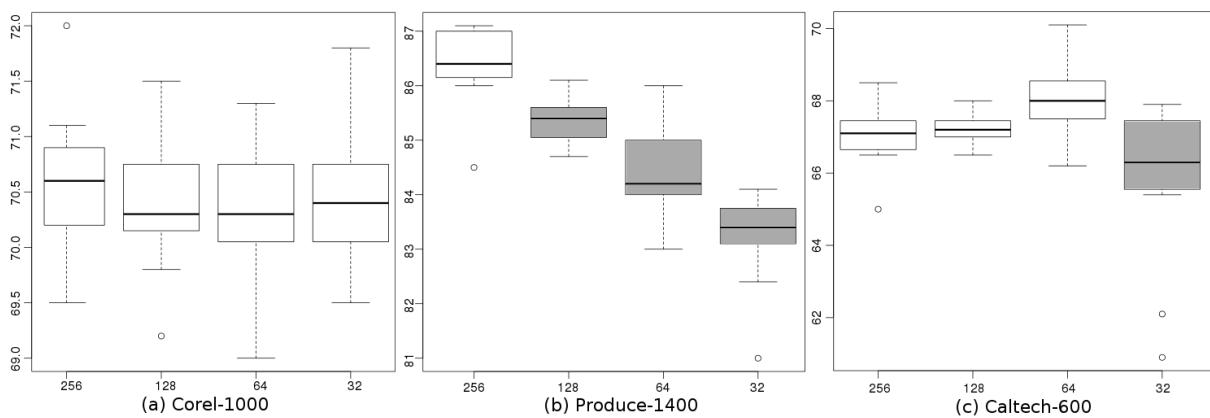
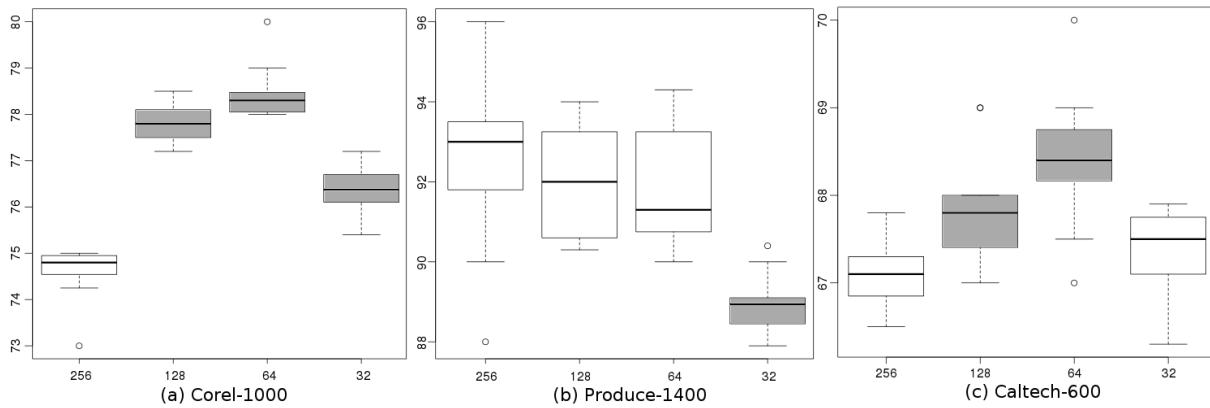
Figura 23 – Resultados para Corel(a), Produce(b) e Caltech(c), com todos os métodos de quantização. Para cada método de extração de características a acurácia é resultante da sua aplicação utilizando 256, 128, 64, 32, 16 e 8 cores, da esquerda para a direita. Fonte: ([PONTI; NAZARÉ; THUMÉ, 2016](#)).

A partir dessa análise geral, uma análise mais específica foi realizada com a combinação dos métodos BIC e *MSB*; e Haralick e *Luminância*. 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.

#### descrever o teste estatístico e o resultado

O teste estatístico ANOVA foi realizado para comparar as acuráciais dos experimentos da Figura 24 e 25. O boxplot para 256, 128, 64 e 32 cores com os métodos BIC e *MSB* está demonstrado na Figura 24. De acordo com o teste estatístico representado, utilizar características de cor e níveis de quantização providos pelo método *MSB* demonstrou resultados melhores do que com 256 cores para as bases Corel (128, 64 e 32 cores) e Caltech (64 cores). O único resultado que piorou significativamente foi para 32 cores da base de imagens *Produce*. Portanto, reduzir converter as imagens de 3 canais de cores para um e reduzir os 256 possíveis valores para apenas 64 provou uma boa escolha de processamento anterior a extração de características.

Menores valores podem degradar os resultados em características de textura, como mostrado na Figura 25.



A redução de dimensionalidade obtida utilizando os métodos de quantização e redução da dimensionalidade com o LPP está ilustrada na Figura 26. 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 comparadas com a classificação dos vetores reduzidos apenas pela quantização. O método de quantização obteve valores de acurácia menores à utilização do LPP em três experimentos: de 256 dimensões com a base *Corel* e com 256 e 64 na base *Produce*. Para a base *Caltech* 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 acurárias após a redução da dimensionalidade. Isso pode ser observado na Figura 26 referente à base de dados *Caltech*.

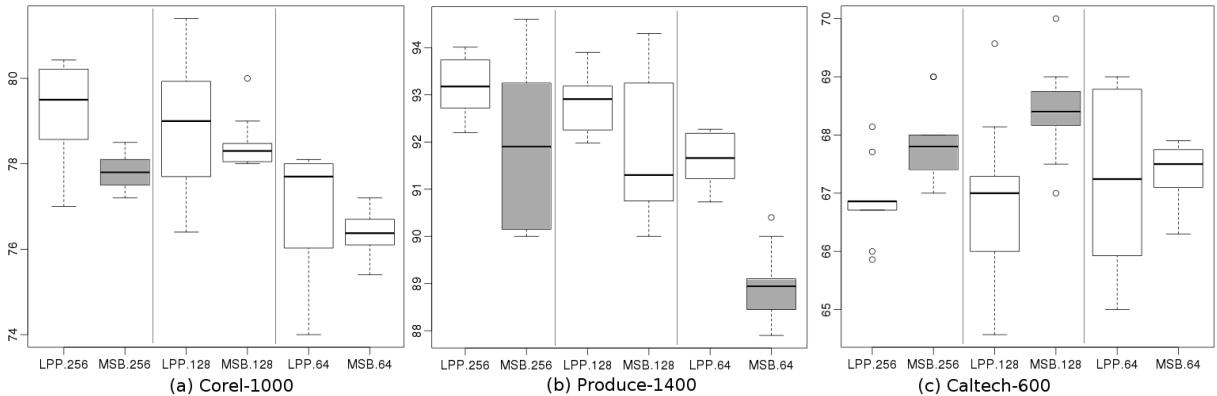


Figura 26 – Resultados de acurácia para os métodos *MSB* (quantização), *LPP* (redução de dimensionalidade) e *BIC* (extração de características). A comparação do *LPP* versus *MSB* foi realizada com a mesma dimensionalidade. Os boxplots em cinza correspondem às significâncias estatísticas com  $p \leq 0.01$  quando comparado a acurácia de 256 cores. Fonte: ([PONTI; NAZARÉ; THUMÉ, 2016](#)).

O número de dimensões de um vetor resultante de apenas um método de extração de características pode ser considerado baixo. Ainda mais que é comum extrair diversos descritores para uma situação, considerando que normalmente não está claro qual método deveria ser utilizado em cada caso. Por conta disso, os próximos experimentos utilizaram a 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 a redução da dimensionalidade com métodos convencionais (*LPP*, neste caso). A melhor configuração encontrada, até então, entre tamanho do vetor e acurácia foi utilizando 128 e 64 cores.

Inicialmente, foi testada a configuração de um vetor  $D = 2310$  com *LPP* para redução de dimensionalidade com  $d = 1160, 582, 294$  e  $150$ . Ou seja, produzindo vetores com o mesmo tamanho dos obtidos apenas com a quantização como redução da dimensão. 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. A Figura 27 mostra os resultados utilizando *LPP*. Note que o método de quantização *MSB* resultou em acurárias melhores que os outros métodos.

A utilização da concatenação de todos os vetores melhorou a acurácia em relação ao melhor descritor individual. A Figura 28 apresenta a comparação do espaço original com *LPP* e *MSB* para redução da dimensionalidade. O teste estatístico ANOVA foi realizado utilizando  $\alpha = 0.01$  como nível de significância. Os resultados que não mudaram significativamente as acurárias foram: *MSB* com 582 características para a base de dados *Corel*; e *MSB* com 1160 nas três bases. O único resultado de piora significativa foi para 32 cores com a base *Produce*. Por

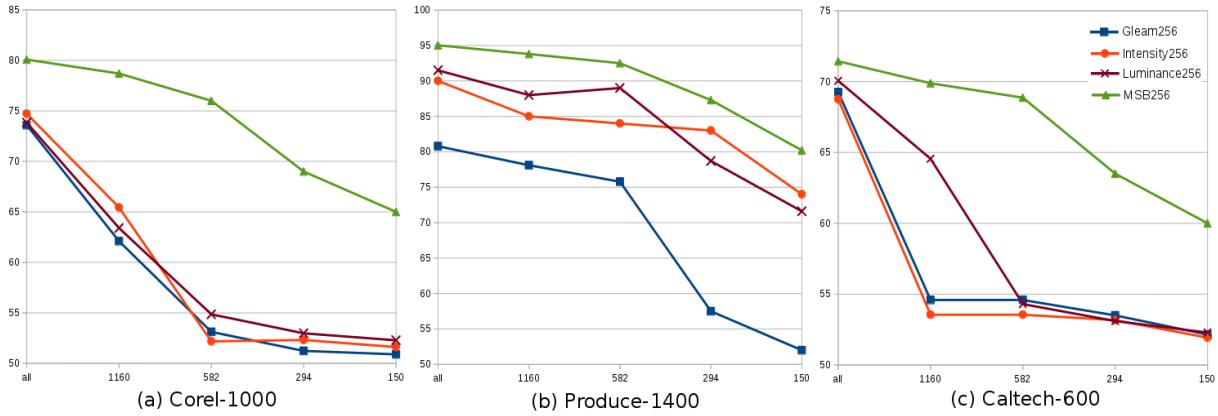


Figura 27 – Comparação da acurácia alcançada com diferentes métodos de quantização: *Gleam*, *Intensidade*, *Luminância* e *MSB*. Inicialmente com  $D = 2310$  e então reduzindo com LPP para  $d = 1160, 582, 294$  e  $150$ . Fonte: ([PONTI; NAZARÉ; THUMÉ, 2016](#)).

conta disso, 64 cores parece ser uma boa escolha de parâmetro de quantização.

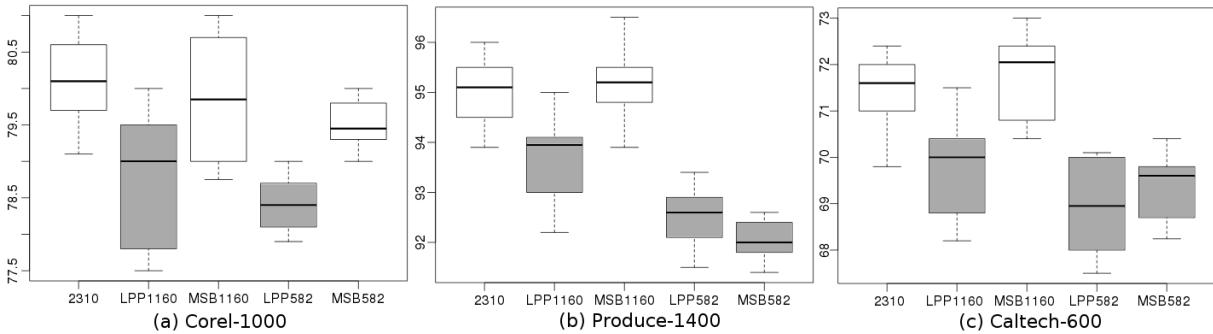


Figura 28 – Comparação da acurácia com o uso da projeção LPP e o método *MSB* para quantização das imagens com o objetivo de redução de dimensionalidade. Fonte: ([PONTI; NAZARÉ; THUMÉ, 2016](#)).

Os resultados indicam que a quantização pode ser utilizada como redução da dimensão de dados visuais, especialmente utilizando 128 e 64 cores. Como experimento, a Figura 29 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.

O vetor concatenado com todos os descritores possui  $9C + 6$  dimensões, onde  $C$  é o número de cores da imagem de entrada. E 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.

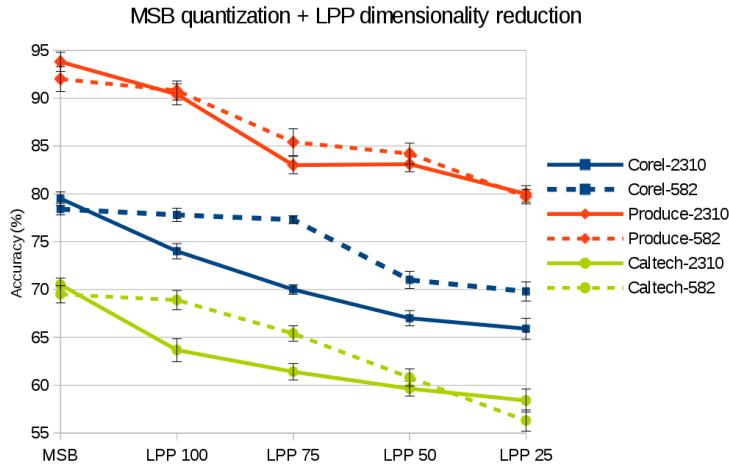


Figura 29 – Resultados para a projeção do LPP sobre o espaço de características produzido pelo método de quantização *MSB* utilizando 256 ( $d = 2310$ ) e 64 cores ( $d = 582$ ). Fonte: (PONTI; NAZARÉ; THUMÉ, 2016).

Ao comparar o uso da quantização com a utilização de métodos mais complexos para redução da dimensionalidade, esse procedimento 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. 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%.

### 5.3 Considerações Finais





# RESULTADOS: GERAÇÃO ARTIFICIAL DE IMAGENS

---

## 6.1 Considerações Iniciais

Os resultados encontrados ao rebalancear classes a partir da geração de imagens artificiais estão descritos neste capítulo. Para cada experimento realizado, são descritos: o protocolo utilizado (incluindo a base de imagens e 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 com métodos de processamento, para melhorar a acurácia da classificação de bases de imagens. Como entrada são utilizadas imagens originais provenientes de diversas coleções disponíveis na literatura. Como resultado, são calculadas medidas estatísticas da classificação dessas coleções após a alteração dessas imagens com os métodos de realce de características relevantes.

Os experimentos a serem relatados são relacionados à geração de imagens para rebalancear classes. Tal processamento é realizado antes da extração de características, e portanto no campo visual. Por conta disso, os resultados devem refletir melhorias nas etapa subsequente de classificação.

## 6.2 Experimentos

Esta seção descreve os resultados encontrados ao rebalancear as classes de imagens aplicando os processamentos — descritos no Capítulo 4 — nas imagens originais. 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.

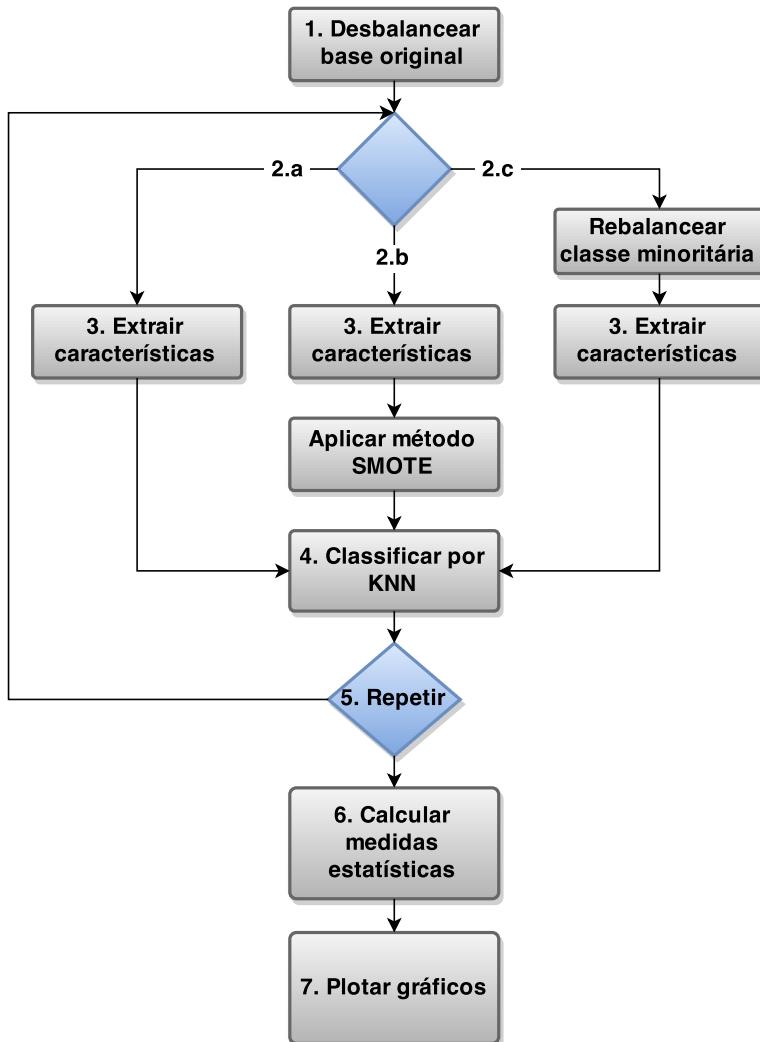


Figura 30 – Fluxo de operações para obtenção dos resultados do rebalanceamento de classes. *Fonte: Elaborado pela autora.*

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 foi realizada, utilizando como exemplo uma base com duas classes de imagens. Primeiramente as imagens foram separadas de forma aleatória em  $k = 5$  folds em cada classe. Depois, as duas classes compõem 40 configurações, consistindo em: um fold para teste e os outros para treino na classe que permanecerá balanceada; e um de teste e um de treino para a classe que os métodos de processamento irão rebalancear. Tal validação é repetido para todas as classes, ou seja, cada classe contribui para o desbalanceamento. Porém, se originalmente a base é naturalmente desbalanceada, um fold é utilizado para teste e os restantes para treino para todas as classes.

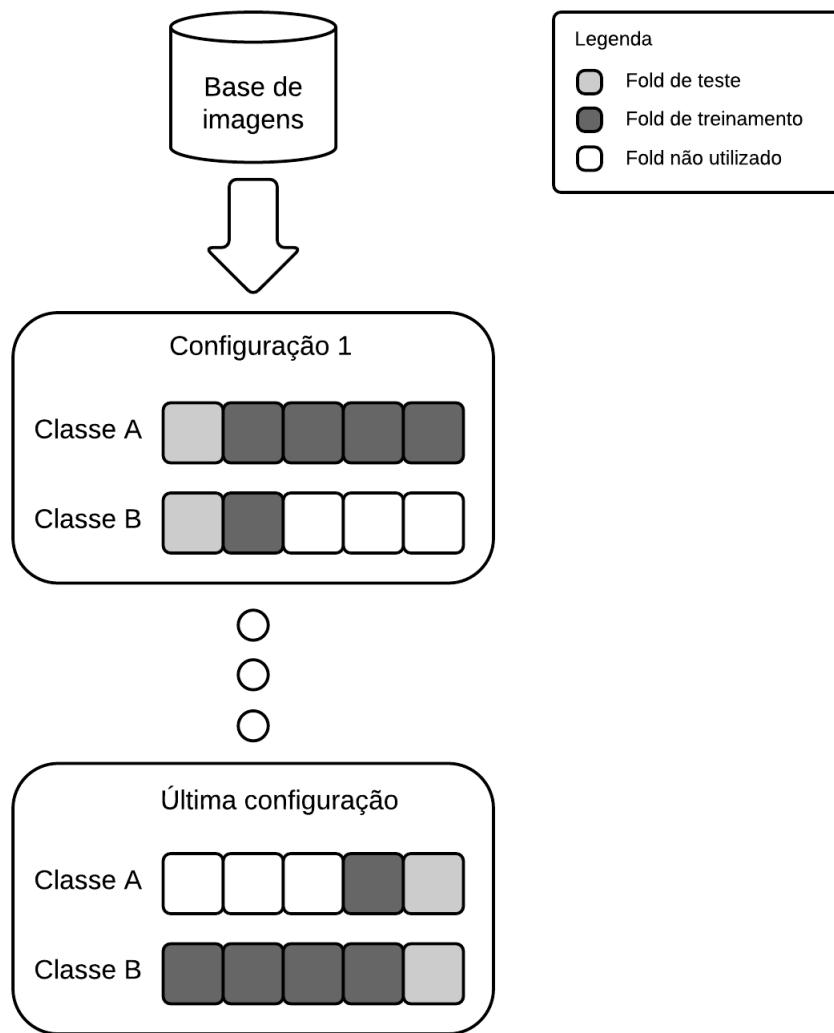


Figura 31 – Fonte: Elaborado pela autora.

A medida estatística mais comum para avaliação é a razão do número de acertos pela quantidade de imagens testadas. Essa medida, conhecida por 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, a acurácia normal poderá ser muito alta, mesmo considerando que nenhuma imagem da classe minoritária foi corretamente classificada. Dessa forma, considera que os erros são igualmente importantes. Mas em se tratando de bases desbalanceadas, deve-se diferenciar o erro em, por exemplo, diagnosticar um paciente doente – classe minoritária – como sendo saudável e um paciente saudável – classe majoritária – como estando doente ([BATISTA; PRATI; MONARD, 2004](#)). No primeiro caso, o paciente corre risco de diagnóstico tardil, enquanto o paciente saudável realiza outros testes para refutação.

Pode-se estender essa medida obtendo-se a acurácia k-fold: 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 pela Equação 6.1, que considera problemas de desbalanceamento de classes.

$$Acc = 1 - \frac{\sum_{i=1}^c E(i)}{2c}, \quad (6.1)$$

onde  $c$  é o número de classes e  $E(i) = e_{i,1} + e_{i,2}$  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  $FN(i)$  (falsos negativos) são os exemplos pertencentes a  $i$  e incorretamente classificados, e  $FP(i)$  (falsos positivos) são os exemplos erroneamente rotulados como  $i$ .

Uma outra medida para bases desbalanceadas é a medida-F1 (conhecida como *F1-Measure* ou *F1-Score* e apresentada na Equação (6.4)), que combina precisão e revocação como medida de efetividade da classificação (GARCIA, 2009). A precisão (Equação 6.2) é a medida da exatidão: dos exemplos classificados como positivos, quantos realmente são. E a revocação (Equação 6.3) é a medida de completude: quantos exemplos positivos foram corretamente classificados como tal.

$$P = \frac{VP}{VP + FP}, \quad (6.2)$$

onde  $VP$  são os exemplos positivos corretamente classificados.

$$R = \frac{VP}{VP + FN} \quad (6.3)$$

$$F1 = 2 \frac{PR}{P + R} \quad (6.4)$$

A partir dessas medidas, o teste estatístico de Friedman pode ser usado para determinar se há diferença significante em uma amostra de resultados gerados (GARCÍA *et al.*, 2010). As performances dos algoritmos são analisados e um *rank* é atribuído para cada conjunto de dados. Ele considera que a hipótese nula a ser testada é que não há diferença estatística relevante entre as observações. Para analisar se o teste da hipótese é significativo, pode ser utilizado o p-valor, que indica o quão estatisticamente significante o resultado é: quanto menor o seu valor, maior a evidência contra a hipótese nula (geralmente o limiar utilizado é 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.

### 6.2.1 Experimento 1: duas classes bem discriminadas

Neste experimento foram utilizadas duas classes com cores bem distintas entre elas, ou seja, de fácil diferenciação. Por tal razão, um sub-experimento de visualização foi realizado para análise do espaço de características. Como o foco é na visualização do espaço de características, é relevante ter o modelo do espaço ideal das classes balanceadas, por isso esse experimento em específico não trata de uma base naturalmente desbalanceada.

#### Protocolo

1. **Imagens originais:** Duas classes da base Corel: *Cavalo* e *Elefante*. As classes estão exemplificadas na Figura 32. A principal característica dessas imagens é a diferença das cores das imagens. Apesar de haverem casos de confusão, são classes que podem ser consideradas bem discriminadas.



Figura 32 – Classes *Cavalo* e *Elefante* utilizadas neste experimento. São duas classes bem discriminadas com 100 imagens cada, originalmente da base de imagens Corel. *Fonte: Elaborado pela autora.*

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 *Cavalo* 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;



Figura 33 – Exemplo da geração artificial de imagens com o método de mistura para as classes *Elefante* e *Cavalo* da Corel-1000. *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 para a análise estatística;
6. **Classificação:** Inicialmente o classificador *Naive Bayes* foi explorado, apresentando melhora na acurácia ao apenas replicar as imagens. Esse comportamento não é desejado em um classificador para a avaliação de rebalanceamento de classes. Por essa razão e por permitir uma análise da melhora no comportamento da classificação, o classificador supervisionado KNN com  $K = 1$  (para mais detalhes ver Seção 2.5.1) foi utilizado;
7. **Projeção multidimensional:** dois componentes principais encontrados ao aplicar PCA (Seção 2.6.1) nos vetores de características para redução de dimensionalidade foram projetados.

## Visualização

As classes *Elefante* e *Cavalo* 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 *Cavalo*, originalmente balanceada. Essa e as próximas projeções desta seção foram obtidas com a técnica para redução de dimensionalidade PCA, descrita na Seção 2.6.1, e são referentes aos dois componentes principais com maiores autovalores.

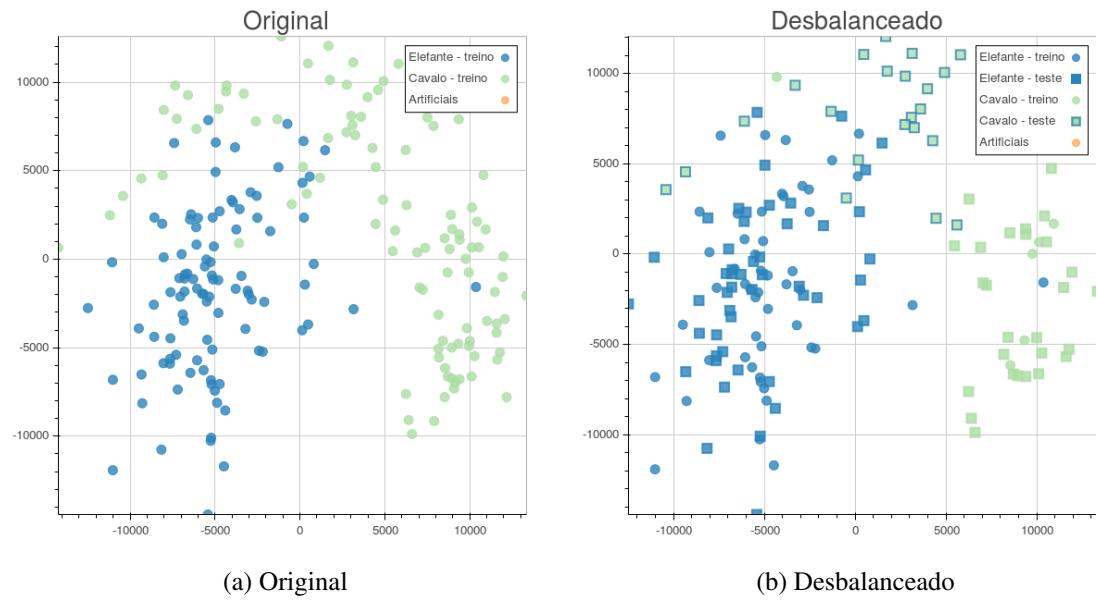


Figura 34 – À esquerda a projeção dos dois componentes principais obtidos com a aplicação de PCA nas classes *Elefante* – em azul – e *Cavalo* – em verde. À direita, as mesmas classes após a remoção de 50% das imagens de treino da classe *Cavalo*. A diferença dos marcadores consiste na definição de imagens 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 KNN com  $K = 1$  reportou que o *f1-score* da geração de imagens utilizando o método de mistura teve um ganho satisfatório em relação ao rebalanceamento no espaço de características com o SMOTE (apresentado na Figura 35 e Tabela ??). 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.

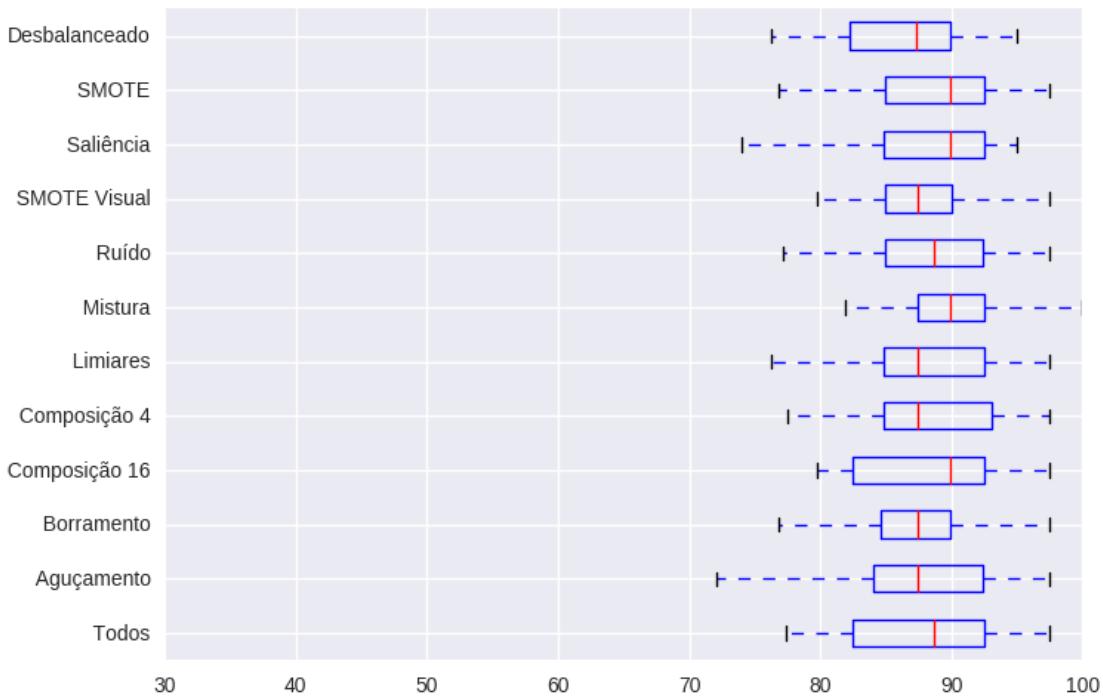


Figura 35 – Resultados de *f1-score* para as classes *Cavalo* e *Elefante* da base Corel. Foi utilizado *BIC* como método de extração de características e *Intensidade'* como método de conversão em escala de cinza. Para essa combinação, a geração de imagens utilizando *mistura* se mostrou favorável.

Fonte: Elaborado pela autora.

necessário a tabela reference aos f-scores da figura acima? está comentado

Para verificar se a geração de imagens inseriu mais informação na classe minoritária do que apenas povoar os espaços entre os exemplos (i.e. SMOTE), a classe rebalanceada utilizando ambos métodos está demonstrada na Figura 36. Em laranja estão representados os novos exemplos de treinamento, projetados no plano da base original balanceada.

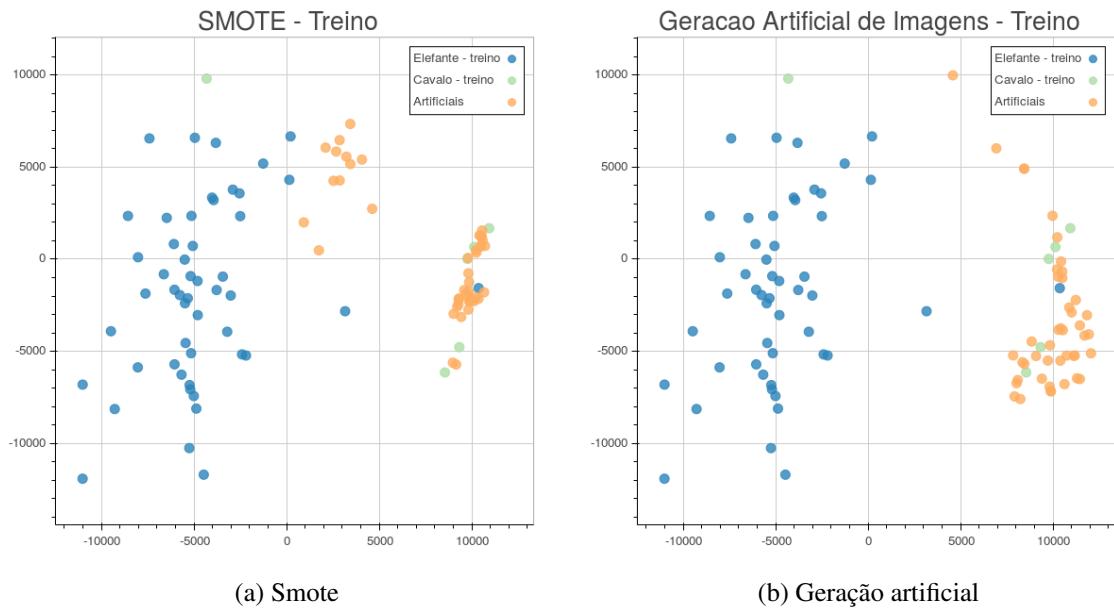


Figura 36 – Comparação dos exemplos de treinamento da geração com SMOTE e no campo visual. Em laranja estão representados os novos exemplos, projetados no plano da base original balanceada.

*Fonte: Elaborado pela autora.*

Após o treinamento realizado com as novas imagens geradas e as originais, o conjunto de teste foi fornecido ao classificador 1-NN e o resultado das predições está ilustrado na Figura 37. 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*.

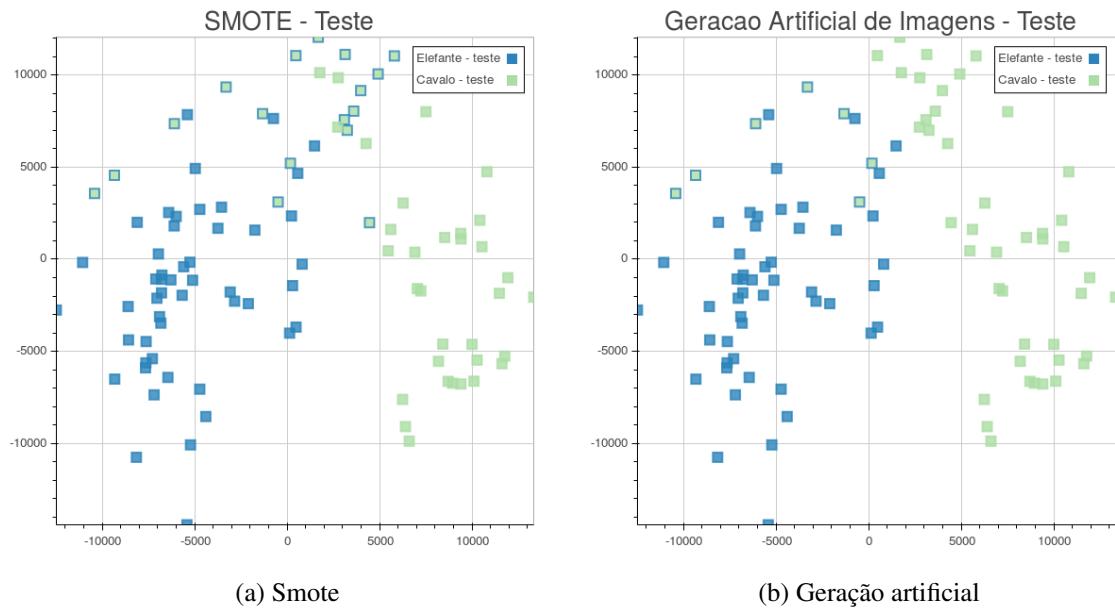


Figura 37 – Resultado do teste da classificação com 1-NN após o treinamento realizado com as bases rebalanceadas. A cor no interior dos marcadores quadrados representa a classe real dos exemplos e a borda representa a classe predita pelo classificador. *Fonte: Elaborado pela autora.*

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 possui capacidade de extrapolar a sua região, como pode ser observado no grupo de exemplos gerados 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 38 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.



Figura 38 – 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.*

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 39), 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.

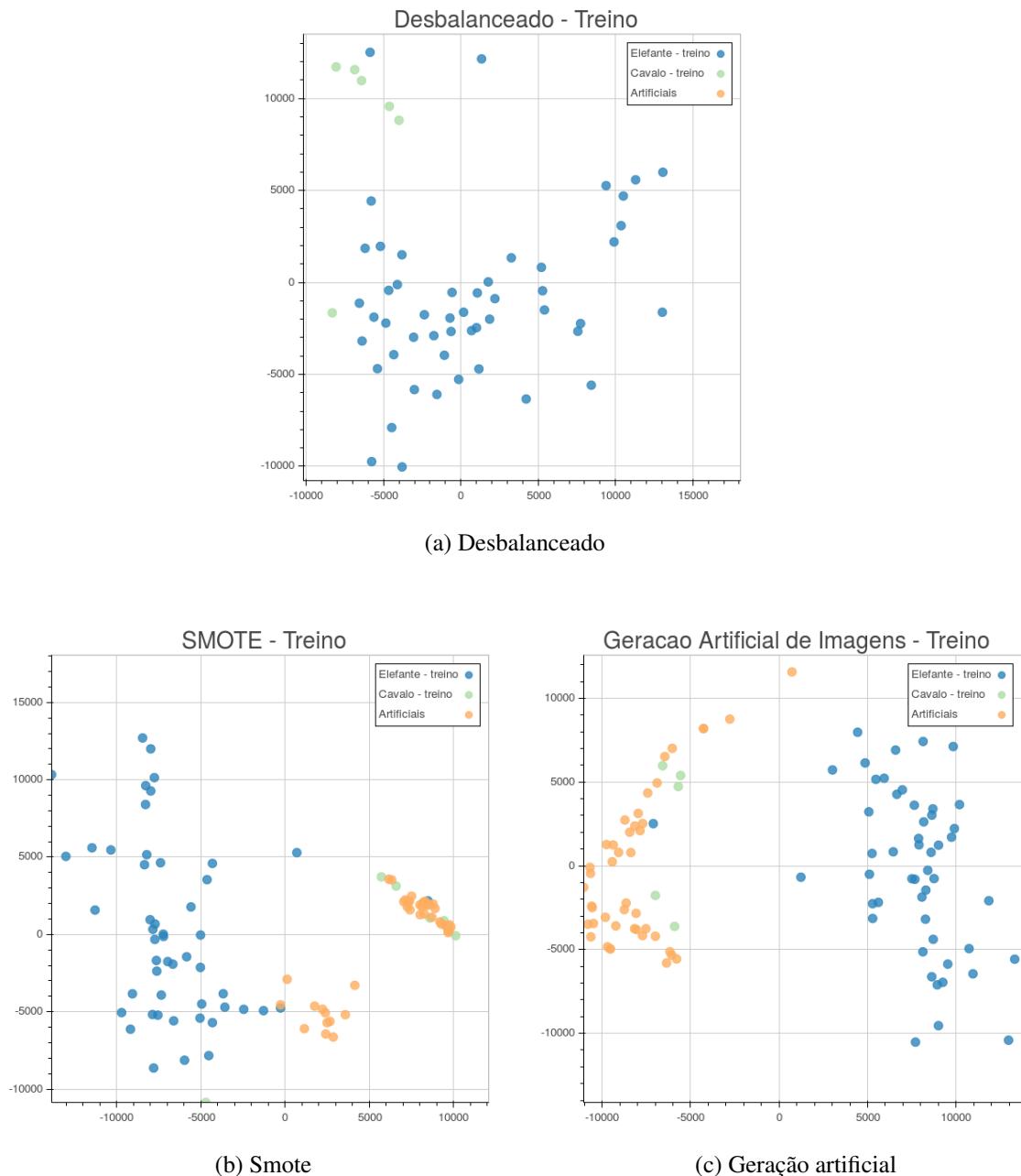


Figura 39 – Melhores subespaços encontrados após a geração de novos exemplos para o SMOTE e para a geração artificial de imagens, e após a remoção de imagens para a projeção dos dados desbalanceados. Pode-se notar que a geração de imagens artificiais proporciona a criação de um subespaço que melhor discretiza as classes, quando comparado com SMOTE ou com a base desbalanceada. *Fonte: Elaborado pela autora.*

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 40 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.

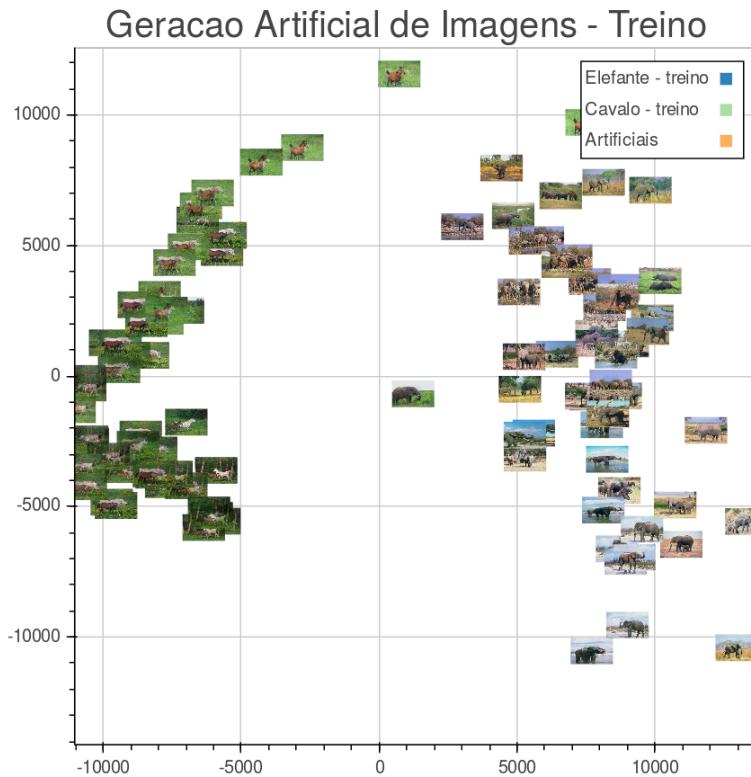


Figura 40 – Visualização do impacto do método de extração de características na separação entre classes.

Possível verificar que o BIC utiliza as intensidades como principal representação de uma imagem.

*Fonte: Elaborado pela autora.*

### **Resultados: melhor combinação dos métodos de extração de características e conversão para escala de cinza**

Para análise estatística, todas as combinações de conversão para escala de cinza e métodos de extração de características foram utilizados. A combinação que obteve o melhor *f-score* para as classes *Elefante* e *Cavalo* foi utilizando *Gleam* e *ACC*. O *boxplot* apresentado na Figura 41 retrata a média dos *f-scores* das 40 configurações deste experimento. O método *mistura*, exemplificado na Figura 42, obteve o melhor *f-score*. A Tabela 1 mostra os valores de tal métrica com valores decimais para o cálculo dos testes estatísticos.

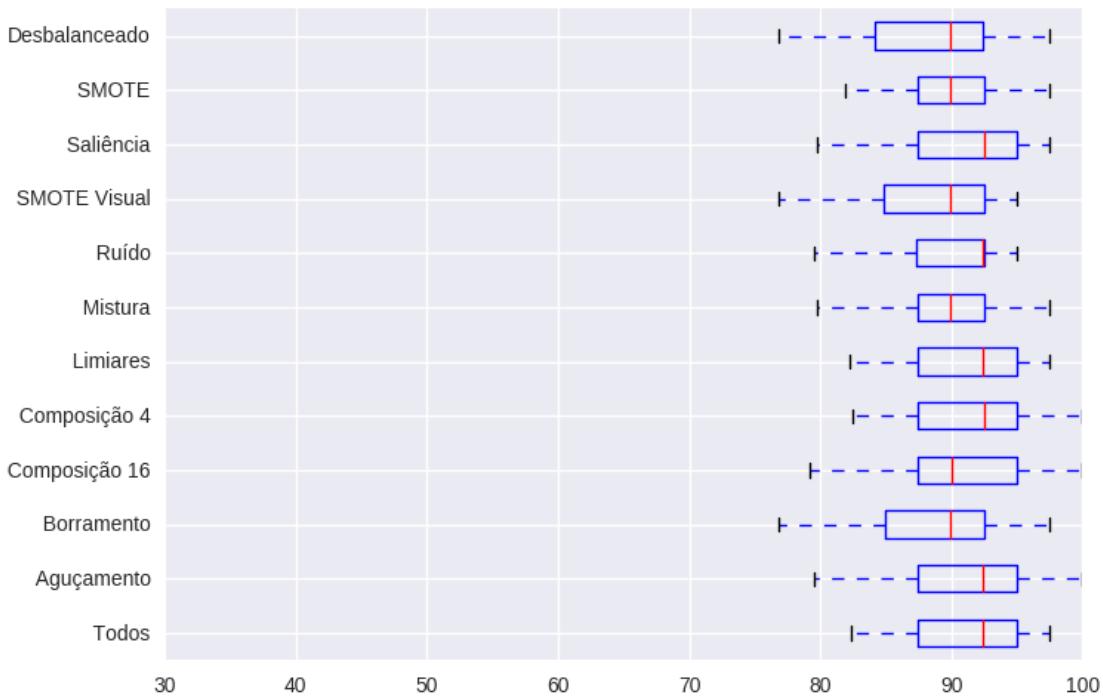


Figura 41 – Conversão em escala de cinza com Gleam e ACC como método de extração de características.

Fonte: Elaborado pela autora.

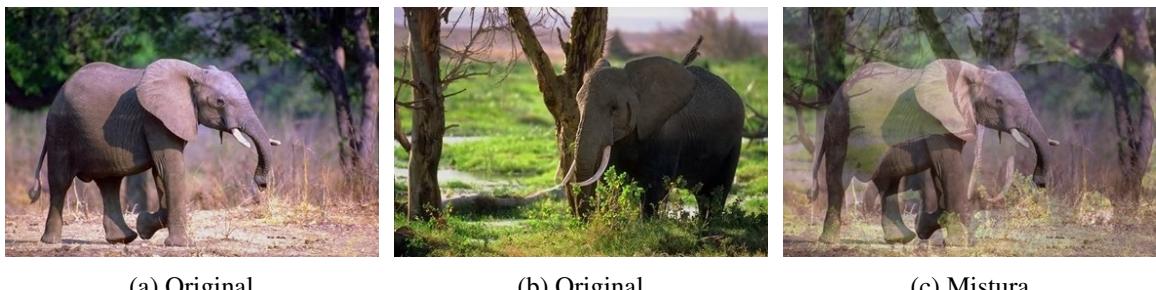


Figura 42 – Fonte: Elaborado pela autora.

### Resultados: maior variância obtida com a combinação dos métodos de extração de características e conversão para escala de cinza

Considerando a análise da melhor combinação dos métodos de representação da imagem, foi verificado também a performance dos rebalanceamentos em um cenário mais complicado: o de maior variância dos *f-scores* para as 40 configurações da validação. A Figura 43 mostra o *boxplot* referente aos resultados da Tabela 2. O melhor método de rebalanceamento para tal cenário foi de geração artificial de imagens aplicando *ruído* nas imagens originais e as utilizando como treinamento (exemplificado na Figura 44).

Tabela 1 – Resultados de *f1-score* para as classes *Cavalo* e *Elefante*, utilizando *Gleam* como método para conversão em escala de cinza e *ACC* para extração de características.

| <b>Gleam &amp; ACC</b> | <b>Média</b>     | <b>Desvio Padrão</b> |
|------------------------|------------------|----------------------|
| Todos                  | 91.090913        | 4.559066             |
| Aguçamento             | 91.002678        | 4.907016             |
| Borramento             | 89.394500        | 5.103498             |
| Composição 16          | 90.934305        | 4.399334             |
| Composição 4           | <b>91.773528</b> | 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                  | 90.169505        | 4.498590             |
| Desbalanceado          | 88.258567        | 5.538461             |

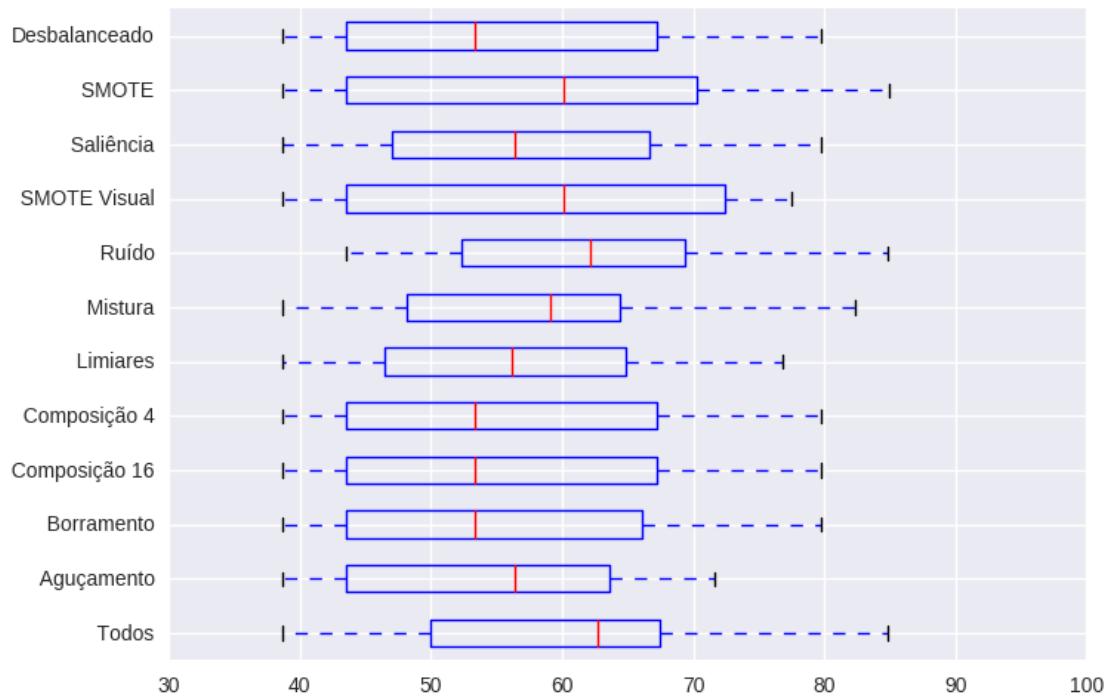


Figura 43 – Conversão em escala de cinza com MSB e HOG como método de extração de características.

Fonte: Elaborado pela autora.

Tabela 2 – Resultados de *f1-score* para as classes *Cavalo* e *Elefante*, utilizando *MSB* como método para conversão em escala de cinza e *HOG* para extração de características.

| <b>MSB &amp; HOG</b> | <b>Média</b>     | <b>Desvio Padrão</b> |
|----------------------|------------------|----------------------|
| 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                | <b>62.174910</b> | 10.746760            |
| SMOTE Visual         | 58.920085        | 14.765860            |
| Saliência            | 56.322367        | 12.169296            |
| SMOTE                | 58.342450        | 13.768688            |
| Desbalanceado        | 55.667145        | 13.341421            |



Figura 44 – Fonte: Elaborado pela autora.

## Discussão

Adicionar um item de discussão, para recapitular tudo aqui? talvez fique repetitivo

### 6.2.2 Experimento 2: duas classes bem sobrepostas

O experimento anterior considerou classes distintas, por isso classes de difícil diferenciação também foram testadas.

#### Protocolo

- Classes de imagens originais:** as classes *Praia* e *Montanha* foram escolhidas por serem as classes que possuem maior dificuldade de diferenciação da base Corel, havendo alta taxa de sobreposição de intensidades de cores e texturas, conforme testes realizados.

2. **Desbalanceamento:** as duas classes contém 100 imagens cada, portanto são balanceadas. Para o experimento, foram utilizadas as 40 configurações de  $k = 5$  folds.
3. **Método para geração artificial:** todas as gerações foram testadas, as que melhor se sobressaíram são relatadas nas seções de resultados a seguir.
4. **Quantização:** todos os métodos de conversão em escala de cinza foram testados.
5. **Extração de características:** todos os métodos para extração foram testados.
6. **Classificação:** o classificador utilizado foi o KNN com  $K = 1$ .

### **Resultados: melhor combinação dos métodos de extração e conversão para escala de cinza**

A combinação de métodos que resultou em um melhor  $f\text{-score}$  para esse contexto foi a utilização de *Luminância*'ância e BIC. Na Figura 53 é possível verificar o *boxplot* dos  $f\text{-scores}$  para as classes desbalanceadas, rebalanceadas com SMOTE após a extração de características e rebalanceadas com os métodos de geração artificial como primeiro passo do *pipeline*. Como pode ser visto na Tabela 3, diversos métodos foram melhores que o SMOTE, e todos melhores do que a base desbalanceada. O método de saliência, mistura e composição obtiveram os melhores resultados.

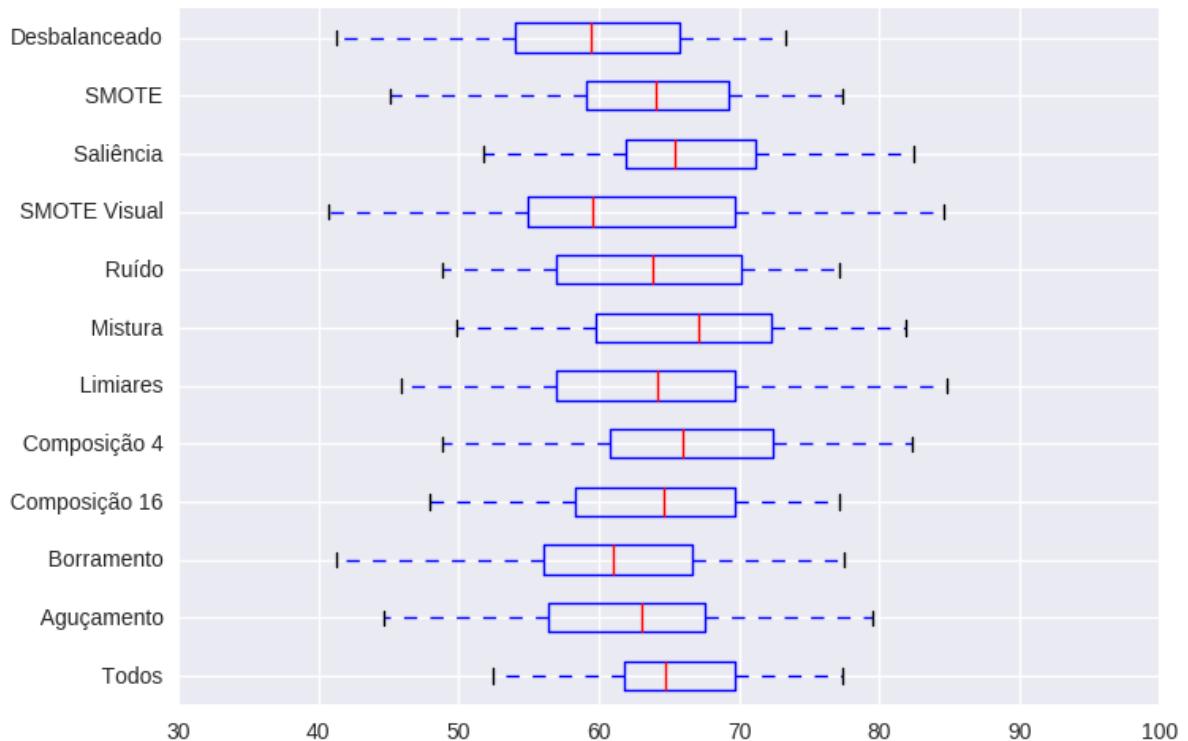


Figura 45 – Boxplot das classes *Praia* e *Montanha* para a conversão em escala de cinza com *Luminância'* e BIC como método de extração de características. Fonte: Elaborado pela autora.

A Figura 46 exemplifica uma geração artificial deste experimento, utilizando o método de *saliência*. Interessante notar que essa imagem faz sentido visualmente.

Tabela 3 – Resultados de *fI-score* para as classes *Praia* e *Montanha*, utilizando *Luminância'* como método para conversão em escala de cinza e *BIC* para extração de características.

| <b>Luminância' e BIC</b> | <b>Média</b> | <b>Desvio padrão</b> |
|--------------------------|--------------|----------------------|
| Todos                    | 65.21        | 7.54                 |
| Aguçamento               | 62.30        | 7.85                 |
| Borramento               | 61.24        | 7.55                 |
| Composição 16            | 64.23        | 7.31                 |
| Composição 4             | 65.65        | 8.27                 |
| Limiares                 | 63.69        | 8.77                 |
| Mistura                  | 65.41        | 8.20                 |
| Ruído                    | 63.50        | 8.05                 |
| SMOTE Visual             | 61.55        | 9.44                 |
| Saliência                | <b>65.69</b> | 6.63                 |
| SMOTE                    | 63.58        | 7.74                 |
| Desbalanceado            | 59.49        | 8.11                 |



(a) Original

(b) Original

(c) Saliência

Figura 46 – Geração artificial utilizando o método de *saliência* em duas imagens da classe *Praia* da base de imagens Corel. *Fonte: Elaborado pela autora.*

### Resultados: maior variância obtida com a combinação dos métodos de extração e conversão para escala de cinza

Os métodos que combinados obtiveram a maior variância foram *Intensidade'* e *HOG* (ver Figura 47 e Tabela 4). O rebalanceamento com a geração de imagens obteve a melhor performance. Porém, o único método que melhorou a variância foi o *aguçamento* (um exemplo de tal geração pode ser visto na Figura ??). A Figura 48 exemplifica a geração artificial com o método *mistura*, utilizando duas imagens de treinamento da classe *Montanha*.

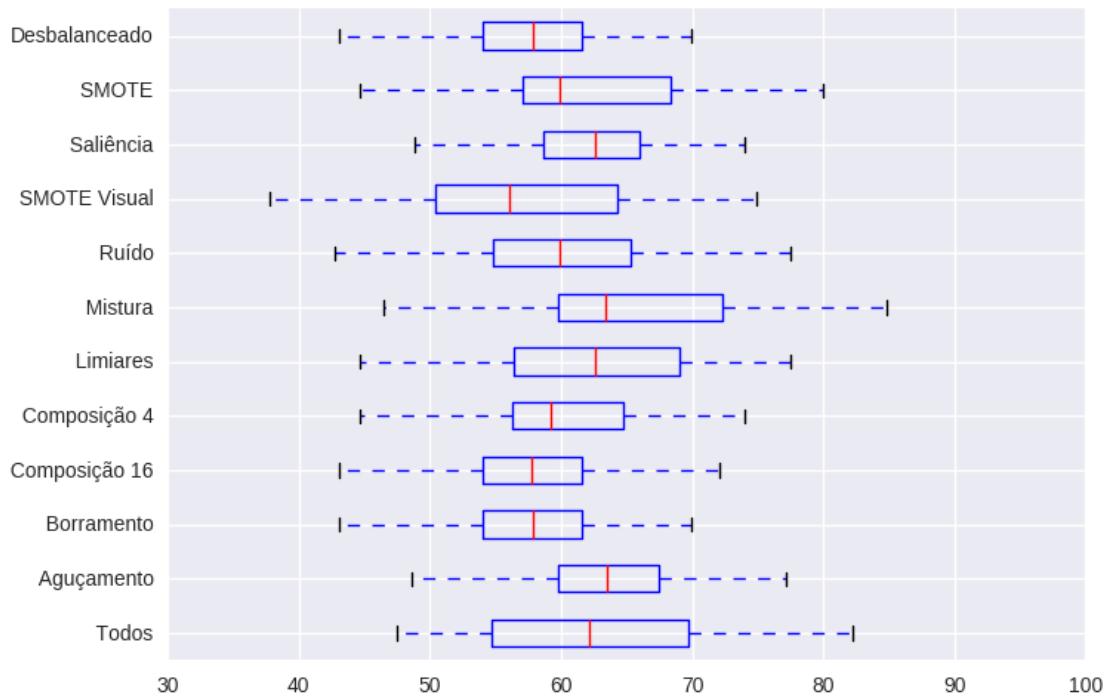


Figura 47 – Conversão em escala de cinza com *Intensidade'* e HOG como método de extração de características. Fonte: Elaborado pela autora.

Tabela 4 – Resultados de *f1-score* para as classes *Praia* e *Montanha*, utilizando *Intensidade'* como método para conversão em escala de cinza e *HOG* para extração de características.

| Intensidade' e HOG | Média            | Desvio padrão |
|--------------------|------------------|---------------|
| Todos              | 62.184222        | 9.310391      |
| Aguçamento         | 63.455343        | 6.719545      |
| Borramento         | 57.599052        | 8.332506      |
| Composição 16      | 57.669325        | 8.424716      |
| Composição 4       | 59.239965        | 8.254027      |
| Limiares           | 63.138800        | 9.132368      |
| Mistura            | <b>65.255990</b> | 9.073246      |
| Ruído              | 60.577283        | 8.690926      |
| SMOTE Visual       | 56.998855        | 9.050991      |
| Saliência          | 61.728330        | 8.154174      |
| SMOTE              | 62.210017        | 8.318301      |
| Desbalanceado      | 57.651543        | 8.323832      |



Figura 48 – Geração artificial utilizando o método de *mistura* em duas imagens da classe *Montanha* da base de imagens Corel. *Fonte:* Elaborado pela autora.

## Discussão

### 6.2.3 Experimento 3: multiclasses

Os dois experimentos anteriores analisaram o rebalanceamento de apenas duas classes. Este experimento apresenta a geração artificial de imagens aplicada a uma base com 10 classes balanceadas, contendo 100 imagens cada.

#### Protocolo

O seguinte protocolo foi seguido para a obtenção dos resultados:

- Classes de imagens originais:** esse experimento foi realizado com a base de imagens Corel-1000<sup>1</sup>, composta por fotografias que representam classes variadas: tribos africanas, praia, construções, ônibus, dinossauros, elefantes, flores, cavalos, montanhas e tipos de comidas.. Para fins de exemplificação, são apresentadas amostras das imagens que representam essas classes na Figura 49.



Figura 49 – Base de imagens Corel-1000. *Fonte:* (PONTI; NAZARÉ; THUMÉ, 2016).

- Desbalanceamento:** 200 configurações de folds para que cada classe fosse desbalanceada.
- Método para geração artificial:** o melhor foi a mistura.

<sup>1</sup> Disponível em <http://wang.ist.psu.edu/docs/related/>

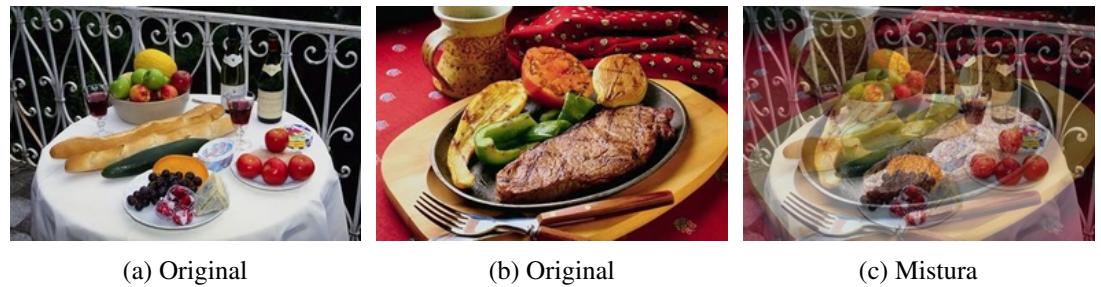


Figura 50

**4. Quantização:**

**5. Extração de características:**

**6. Classificação:** KNN com  $K = 1$ .

### Resultado da melhor combinação dos métodos de extração e conversão para escala de cinza

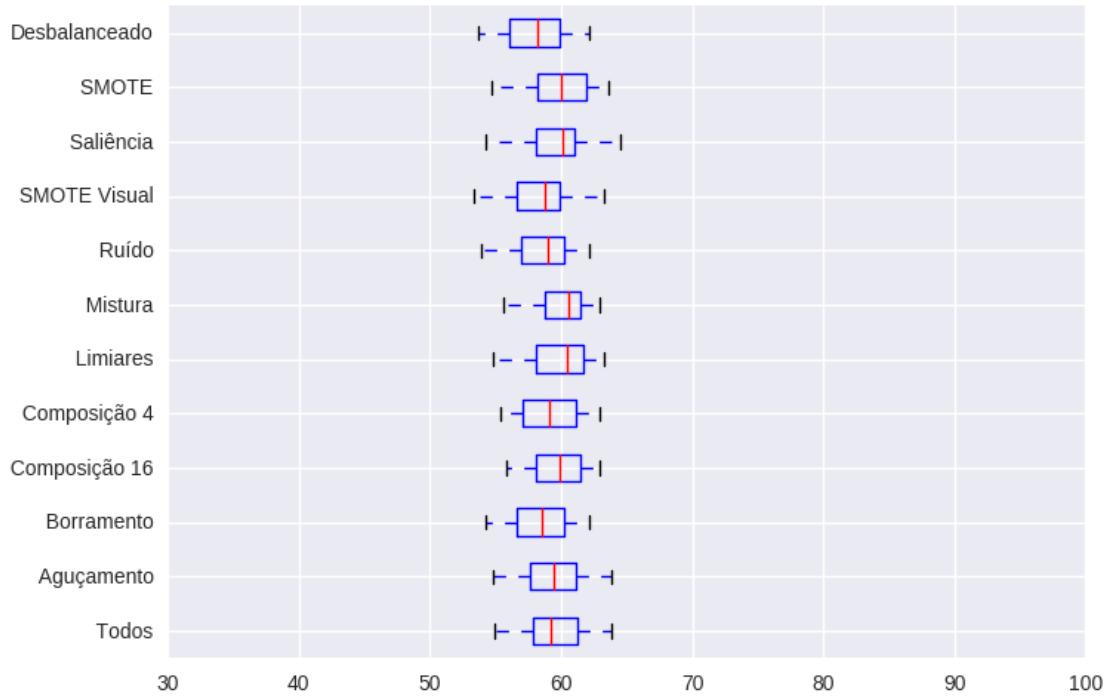


Figura 51 – Experimento com as 10 classes da base Corel. Foi utilizado o método Luma para conversão em escala de cinza e LBP como método de extração de características. *Fonte: Elaborado pela autora.*

### Resultado da maior variância obtida com a combinação dos métodos de extração e conversão para escala de cinza

Tabela 5 – Resultados de *f1-score* para as 10 classes da Corel, utilizando *Luma* como método para conversão em escala de cinza e *LBP* para extração de características

| <b>Luma e LBP</b> | <b>Média</b> | <b>Desvio padrão</b> |
|-------------------|--------------|----------------------|
| Todos             | 59.46        | 2.23                 |
| Aguçamento        | 59.32        | 2.33                 |
| Borramento        | 58.41        | 2.36                 |
| Composição 16     | 59.71        | 1.92                 |
| Composição 4      | 58.97        | 2.20                 |
| Limiares          | 59.89        | 2.33                 |
| Mistura           | <b>59.99</b> | 2.02                 |
| Ruído             | 58.48        | 2.21                 |
| SMOTE Visual      | 58.43        | 2.35                 |
| Saliência         | 59.63        | 2.07                 |
| SMOTE             | 59.78        | 2.36                 |
| Desbalanceado     | 58.09        | 2.31                 |

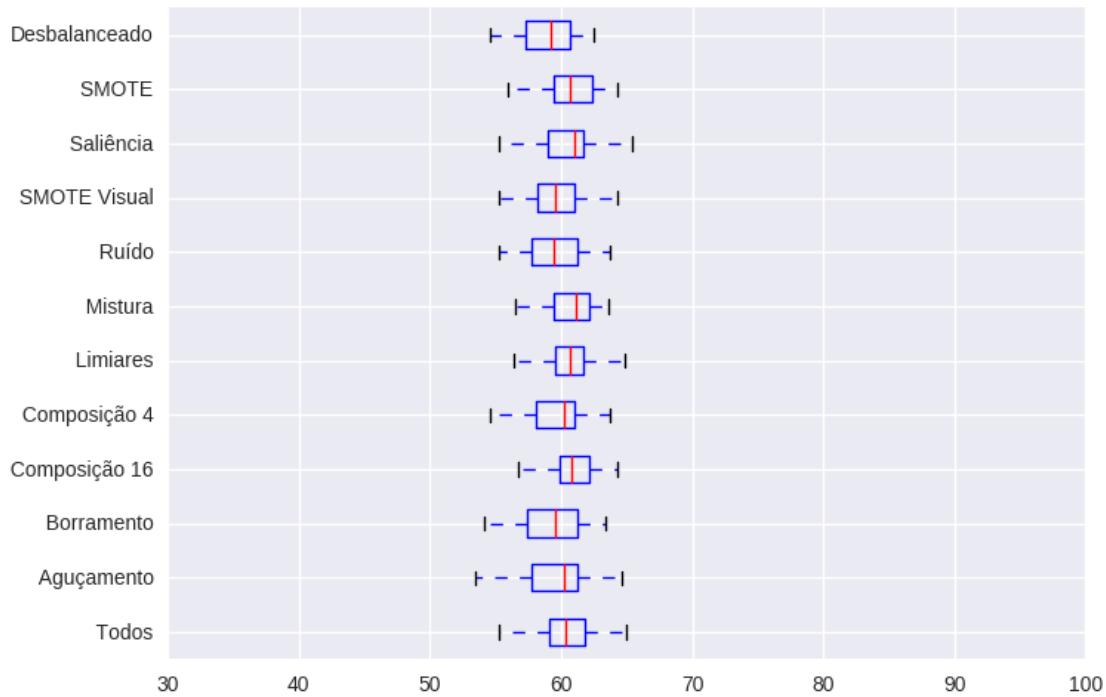


Figura 52 – Experimento com as 10 classes da base Corel. Foi utilizado o método *Intensidade'* para conversão em escala de cinza e LBP como método de extração de características. *Fonte: Elaborado pela autora.*

## Discussão

Tabela 6 – Resultados de *f1-score* para as 10 classes da Corel, utilizando ‘*Intensidade*’ como método para conversão em escala de cinza e *LBP* para extração de características

| <b>LBP Intensity</b> | <b>Média</b> | <b>Desvio Padrão</b> |
|----------------------|--------------|----------------------|
| Todos                | 60.39        | 2.35                 |
| Aguçamento           | 59.78        | 2.51                 |
| Borramento           | 59.30        | 2.34                 |
| Composição 16        | <b>60.73</b> | 2.23                 |
| Composição 4         | 59.77        | 2.30                 |
| Limiares             | 60.59        | 2.24                 |
| Mistura              | 60.71        | 2.16                 |
| Ruído                | 59.39        | 2.44                 |
| SMOTE Visual         | 59.44        | 2.26                 |
| Saliência            | 60.36        | 2.18                 |
| SMOTE                | 60.49        | 2.22                 |
| Desbalanceado        | 58.91        | 2.20                 |

#### **6.2.4 Experimento 4: classes naturalmente desbalanceadas**

##### **Base de Imagens**

##### **Protocolo**

O seguinte protocolo foi seguido para a obtenção dos resultados:

1. **Classes de imagens originais:**
2. **Desbalanceamento:**
3. **Método para geração artificial:**
4. **Quantização:**
5. **Extração de características:**
6. **Classificação:**

##### **Resultados e Discussão**

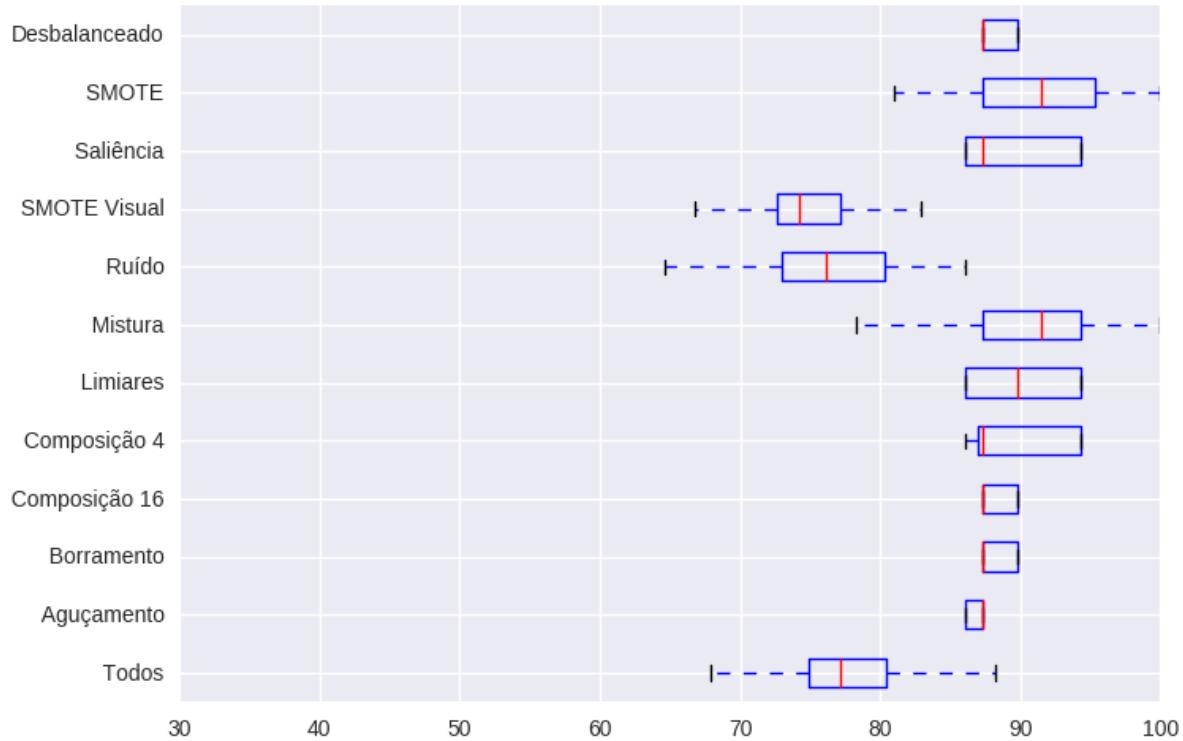


Figura 53 – Fonte: Elaborado pela autora.

Tabela 7 – Resultados de *f1-score* para as classes XX, utilizando *Luminância'* como método para conversão em escala de cinza e *HOG* para extração de características

| Luminância & HOG | Média            | Desvio Padrão |
|------------------|------------------|---------------|
| Todos            | 77.373797        | 5.112180      |
| Aguçamento       | 87.831980        | 7.039036      |
| Borramento       | 85.051780        | 9.816291      |
| Composição 16    | 85.051780        | 9.816291      |
| Composição 4     | 85.392703        | 10.069356     |
| Limiares         | 86.213200        | 10.528769     |
| Mistura          | 90.312440        | 7.357446      |
| Ruído            | 75.523828        | 6.209384      |
| SMOTE Visual     | 74.109795        | 4.093989      |
| Saliência        | 85.713360        | 10.399830     |
| SMOTE            | <b>90.711873</b> | 6.423170      |
| Desbalanceado    | 85.051780        | 9.816291      |

### 6.2.5 Experimento 5: classes com muitas imagens

#### Base de Imagens

#### Protocolo

O seguinte protocolo foi seguido para a obtenção dos resultados:

1. Classes de imagens originais:
2. Desbalanceamento:
3. Método para geração artificial:
4. Quantização:
5. Extração de características:
6. Classificação:

**Resultado da melhor combinação dos métodos de extração e conversão para escala de cinza**

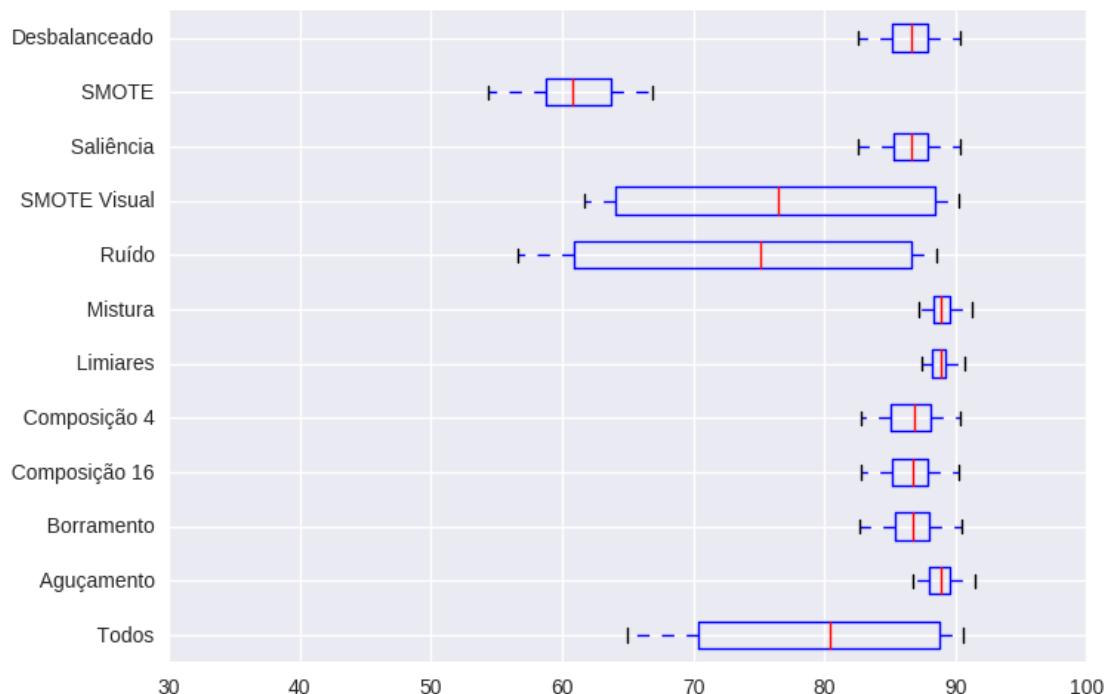


Figura 54 – Fonte: Elaborado pela autora.

Tabela 8 – Resultados de *f1-score* para as classes *Tubarão* e *Peixe*, utilizando *Gleam* como método para conversão em escala de cinza e *HOG* para extração de características

| <b>Gleam e HOG</b> | <b>Média</b> | <b>Desvio padrão</b> |
|--------------------|--------------|----------------------|
| Todos              | 79.38        | 9.88                 |
| Aguçamento         | 88.88        | 1.13                 |
| Borramento         | 86.69        | 1.81                 |
| Composição 16      | 86.61        | 1.84                 |
| Composição 4       | 86.69        | 1.97                 |
| Limiares           | 88.93        | 1.04                 |
| Mistura            | <b>88.99</b> | 0.98                 |
| Ruído              | 73.72        | 13.18                |
| SMOTE Visual       | 76.23        | 12.51                |
| Saliência          | 86.61        | 1.86                 |
| SMOTE              | 60.90        | 3.46                 |
| Desbalanceado      | 86.60        | 1.86                 |

### 6.3 Considerações Finais

Este estudo apresentou evidências experimentais de que, em problemas de duas classes, pode haver ganho estatístico do *f-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). 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

---

Os resultados encontrados apontam para uma alternativa – ou adição – a seleção de características, ao usar os métodos de quantização de imagens. 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 computa utilizando uma memória proporcional o 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. Outra possibilidade é utilizar esse métodos como um primeiro passo de redução e então utilizar o LPP para obter apenas 100 características que melhor representam os dados, atingindo acurárias maiores ou similares.

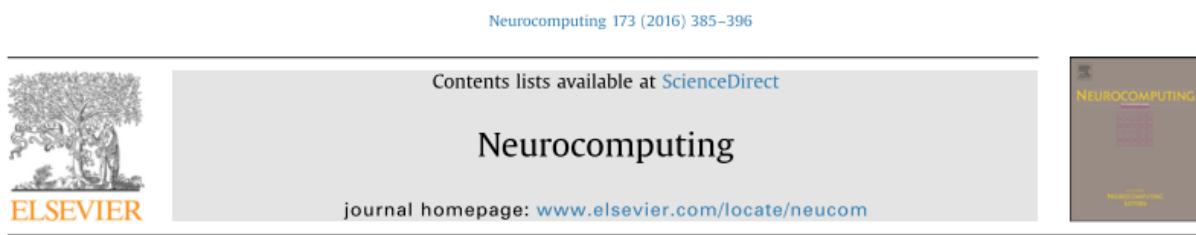
É importante ressaltar que realizar a quantização de imagens não requere treinamento e já faz parte de uma tarefa do pipeline de reconhecimento. Por esta razão, seu uso não aumenta o custo computacional do sistema, e ainda simplifica os passos subsequentes. Isso reduz a dimensão do vetor de características 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.

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 (em se tratando de problemas de classes desbalanceadas), melhorando a acurácia de algoritmos de classificação, quando comparada à geração de exemplos artificiais no espaço de atributos (i.e. SMOTE). Para validar a ideia da geração artificial de imagens, as características das novas imagens – extraídas com o método BIC – 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.

## 7.1 Publicações

Foi publicado um artigo na revista *Neurocomputing* (Figura 55), referente aos resultados de quantização desta pesquisa. O artigo referente à geração de imagens artificiais para o rebalanceamento de classes está em preparo.



### Image quantization as a dimensionality reduction procedure in color and texture feature extraction



Moacir Ponti\*, Tiago S. Nazaré, Gabriela S. Thumé

Instituto de Ciências Matemáticas e de Computação Universidade de São Paulo São Carlos, 13566-590 SP Brazil

#### ARTICLE INFO

##### Article history:

Received 2 July 2014

Received in revised form

9 April 2015

Accepted 16 April 2015

Available online 1 September 2015

##### Keywords:

Image quantization

LPP

Compact features extraction

#### ABSTRACT

The image-based visual recognition pipeline includes a step that converts color images into images with a single channel, obtaining a color-quantized image that can be processed by feature extraction methods. In this paper we explore this step in order to produce compact features that can be used in retrieval and classification systems. We show that different quantization methods produce very different results in terms of accuracy. While compared with more complex methods, this procedure allows the feature extraction in order to achieve a significant dimensionality reduction, while preserving or improving system accuracy. The results indicate that quantization simplify images before feature extraction and dimensionality reduction, producing more compact vectors and reducing system complexity.

© 2015 Elsevier B.V. All rights reserved.

Figura 55 – Artigo publicado na *Neurocomputing*, referente a quantização de imagens. Fonte <<http://www.sciencedirect.com/science/article/pii/S0925231215012771>>.

## 7.2 Trabalhos Futuros

Ao usar 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 esparços.

### trabalhos futuros da quantização

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 extrator.

Atualmente, o estado da arte de extração e classificação de imagens corresponde ao uso de redes neurais de convolução, conhecidas por 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 uma CNN. Essa rede permite encontrar as características mais relevantes da base de imagens, que os extratores de características canônicos não capturam. Isso porque ela possui uma hierarquia de camadas, desde a imagem original até uma etapa de classificação, com o objetivo de aprender qual o melhor processamento para as imagens de modo a melhor discriminá-las. Podem também indicar possíveis operações para auxiliar na geração de imagens artificiais.

Além de analisar o processamento realizado por uma rede de convolução para a classificação das imagens, uma RBM (máquinas de Boltzmann restritas) 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 21.
- 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 4 vezes nas páginas 17, 18, 19 e 51.
- 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 20.
- 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 17.
- 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://link.springer.com/chapter/10.1007/3-540-45428-4{\\\_}5](http://link.springer.com/chapter/10.1007/3-540-45428-4{\_}5)<http://arxiv.org/abs/1106.1>>. Citado na página 18.
- DALAL, N.; TRIGGS, B. Histograms of Oriented Gradients for Human Detection. In: **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**. 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.
- 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 77.
- 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 52.

GARCÍA, S.; FERNÁNDEZ, A.; LUENGO, J.; HERRERA, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. **Information Sciences**, v. 180, n. 10, p. 2044–2064, may 2010. ISSN 00200255. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025509005404>>. Citado na página 52.

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, 27 e 30.

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 19.

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 22.

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 15.

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 19.

JOIA, P.; PAULOVICH, F. V.; COIMBRA, D.; CUMINATO, J. A.; NONATO, L. G. Local Affine Multidimensional Projection. **IEEE Transactions on Visualization and Computer Graphics**, v. 17, n. 12, p. 2563–2571, 2011. Citado na página 3.

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/>&tool=pmcentrez&rendertype=ab>. Citado 2 vezes nas páginas 4 e 13.

LOWE, 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.

- 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.
- PAIVA, J. G. S.; FLORIAN, L.; PEDRINI, H.; TELLES, G. P.; MINGHIM, R. Improved similarity trees and their application to visual data classification. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 17, n. 12, p. 2459–2468, 2011. Citado 2 vezes nas páginas 3 e 4.
- PASS, G.; ZABIH, R.; MILLER, J. Comparing images using color coherence vectors. In: **Proceedings of the fourth ACM international conference on Multimedia**. New York, USA: ACM Press, 1996. p. 65–73. ISBN 0897918711. Disponível em: <<http://dl.acm.org/citation.cfm?id=244130.244148>>. Citado na página 16.
- 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 21.
- 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 23.
- PICON, C.; ROSSI, I.; PONTI-JR, M. Análise da classificação de imagens por descritores de cor utilizando várias resoluções. In: **Workshop of Undergraduate Works**. SIBGRAPI, 2011. Disponível em: <[http://www.icmc.usp.br/{~}moacir/papers/Picon{\\\_}WUW201](http://www.icmc.usp.br/{~}moacir/papers/Picon{\_}WUW201)>. Citado na página 3.
- PONTI, M. Segmentation of Low-Cost Remote Sensing Images Combining Vegetation Indices and Mean Shift. **IEEE Geoscience and Remote Sensing Letters**, IEEE, v. 10, n. 1, p. 67–70, jan 2013. ISSN 1545-598X. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6202674>>. Citado na página 3.
- 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 10 vezes nas páginas xviii, 25, 26, 41, 43, 44, 45, 46, 47 e 68.

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 na página 4.

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 77.

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 28.

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 na página 15.

\_\_\_\_\_. \_\_\_\_\_. In: **2009 IEEE 12th International Conference on Computer Vision**. IEEE, 2009. p. 32–39. ISBN 978-1-4244-4420-5. ISSN 1550-5499. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5459207>>. Citado na página 23.

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.

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 22.