Universidade Federal de Pernambuco

Centro de Informática

Graduação em Engenharia da Computação

Detecção de Imagens Manipuladas utilizando Descritores Locais

José Antônio da Silva

Trabalho de Graduação

Orientador: Tsang Ing Ren

Recife

Junho de 2017

Universidade Federal de Pernambuco
Centro de Informática

José Antônio da Silva

Detecção de Imagens Manipuladas utilizando Descritores Locais

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: *Tsang Ing Ren*

Recife
Junho de 2017

Ao meu amigo, Bruno Manoel dos Santos (in memorian).

Agradecimentos

Gostaria de agradecer primeiramente a Deus pela minha vida, e pela oportunidade que Ele me deu de estar vivendo exatamente nesta época, compartilhando minha existência com pessoas incríveis que pude conhecer e conviver, além de aprender cada vez um pouco mais com cada uma delas, garantidamente não sou mais o mesmo no meu ano anterior: a evolução é progressiva, e o aprendizado, consolidado.

Aos meus pais (Josefa e Mário) e irmãos (Mário, Marcos e Flávio), que através de seus amores incondicionais, puderam, através de inúmeros exemplos e palavras, ensinar-me a importância dos valores humanos e do respeito ao próximo, tanto em palavras quanto em ações, acompanhadas de um irresistível incentivo a nunca desistir do que projeta-se em meus sonhos. Ao professor Tsang Ing Ren por todo apoio, paciência e mentoria durante a Iniciação Científica e o Trabalho de Graduação.

Agradeço em especial a todos os meus professores, tanto a nível universitário, quanto nível infantil/básico, que puderam me transformar para sempre em inúmeros aspectos, que sequer posso enumerar aqui. A todos vocês, meu grande obrigado, é um privilégio aprender tanta coisa diferente com pessoas dispostas a transformar o mundo começando transformando pessoas.

Agradeço, por fim, a todos os meus amigos que, ao longo do curso (e muito antes dele) puderam fazer com que eu pudesse compartilhar tantos momentos diferentes, e mostrar que não existe chances de vivermos plenamente sem nossos amigos. Obrigado a todos vocês, e mesmo perto ou longe, acredito que a amizade legítima perdura, apesar das barreiras do espaço e do tempo.

Resumo

Imagens digitais são utilizadas em campos como medicina, mercado financeiro (digitalização de cheques e promissórias), jornalismo, investigações criminais, entre outros campos de atuação. No caráter jornalístico em especial, são comumente empregadas em veículos de comunicação de forma a provar visualmente algum elemento documentado na notícia. No entanto, com a facilidade do uso de ferramentas de edição em imagens, a adulteração desse formato de conteúdo tornou-se cada vez mais acessível e, como consequência, a manipulação de imagens para corroborar matérias de *fake news* tem se tornado mais comum. Neste trabalho, é proposto um método de detecção de imagens com manipulações do tipo *Copy-Move Forgery Detection* (CMFD) e *Splicing* através da combinação de diferentes extratores de características, mostrando que tal combinação pode melhorar as taxas de acerto na classificação entre autêntica ou manipulada. Primeiro, utilizaram-se os extratores *Histogram of Oriented Gradients* (HOG), *Weber Local Descriptors* (WLD) e *Local Binary Patterns* (LBP); a partir deles, gera-se um vetor de características da imagem, podendo aplicá-lo como entrada para um classificador *Support Vector Machine* (SVM), classificando a imagem entre autêntica ou manipulada a partir das informações disponíveis no vetor de características. Para a eliminação dos atributos menos relevantes, utilizou-se o método de seleção de características *Local Linear-Based* (LLB). O treinamento e o teste do SVM foram executados sobre o banco de imagens manipuladas CASIA. Os canais RGB e YCbCr foram analisados de forma comparativa, de forma a decidir qual dos dois concede melhores resultados. Os resultados obtidos atingiram uma acurácia de 97,19% utilizando os três extratores combinados com LLB na base CASIA v2.0, chegando próximo ao estado-da-arte, que é de 97,83% usando a abordagem *Deep Learning*. Tais resultados mostram que o uso combinado de extratores aumenta a variabilidade do modelo, fazendo com que seja superior em precisão comparado ao uso individual destes; verifica-se também que o uso de seleção de atributos pode aumentar ainda mais os valores, desde que se escolha adequadamente o número de características.

Palavras-chave: WLD, Detecção de manipulação, LBP, detecção de *copy-move*, HOG, LLB, detecção de *Splicing*, extração de características.

Abstract

Digital images are used in fields such as medicine, financial market (digitalization of promissory notes and checks), journalism, criminal investigations, among many fields of application. In journalistic field in particular, are commonly used in communication vehicles in order to visually prove element documented in the news. However, with the ease of use of image editing tools, the tampering with this content has become increasingly accessible. As a consequence, the manipulation of images to corroborate contents of fake news has become more common. In this work, a method of image detection is proposed with Copy-Move Forgery Detection (CMFD) manipulations and splicing through the combination of different feature extractors, showing that such a combination can improve hit rates in the classification between authentic or manipulated. First, it was used the Histogram Of Oriented Gradients (HOG), Weber Local Descriptors (WLD), and Local Binary Patterns (LBP) extractors. From them, it is generated an image characteristics vector, being able to apply it as input to Support Vector Machine (SVM) classifier, classifying either as authentic or manipulated from the information available in the characteristic vector. For The elimination of less relevant attributes, we used the method of selection of features Linear-Based Local (LLB). The training and test steps were performed in CASIA image dataset. The RGB and YCbCr channels were studied in a comparative way, in order to decide which of the two gives better results. The results obtained reached an accuracy of 97.19% using the three extractors combined with LLB in the CASIA base V2.0, coming close to state-of-the-art, which uses the Deep Learning approach with accuracy of 97,83%. Such results Show that the combination of extractors increases the variability of the model, Making them superior in accuracy compared to the individual use of these; it is also verified that the use of attribute selection can increase even more the values, since the number of features is properly chosen.

Keywords: WLD, Forgery Detection, LBP, Copy-Move Forgery HOG, LLB, Splicing Detection, feature extraction

Sumário

1 Introdução **1**

1.1 Objetivo 4

1.2 Estrutura do trabalho 5

2 Revisão bibliográfica **6**

2.1 Imagem Digital 6

2.1.1 Espaço de Cores 6

2.1.1.1 Espaço de cores YCbCr 7

2.1.1.2 Espaço de cores RGB 8

2.2 Seleção de Atributos com *Local Linear Based* (LLB) 9

2.3 *Local Binary Patterns* 9

2.3.1 LBP Uniforme e Invariante a Rotação 12

2.4 *Histogram of Oriented Gradient (HOG)* 13

2.4.1 Passos para extração de features com HOG 13

2.5 *Weber Local Descriptor* 15

2.6 *Support Vector Machine* (SVM) 17

3 Método proposto e Metodologia **20**

4 Experimentos e Resultados **23**

4.1 Resultados para *CASIA v1.0* 25

4.2 Resultados para *CASIA v2.0* 27

5 Conclusão e trabalhos futuros **30**

Lista de Figuras

1.1 Um exemplo de *splicing forgery*, onde uma região de uma imagem origem é copiada e colada para uma destino; (A) é a imagem fonte, e (B) é a destino, cuja composição pode ser vista em (C). 2

1.2 Um exemplo de *copy-move forgery*: (A) apresenta a imagem original, e (B) mostra o desaparecimento de um dos garotos, devido à cópia e colagem de regiões de folhas para outra área. 3

1.3 Falsificações recentes em imagens documentadas: (a) Composição de Brad Pitt e Cher (Johnson and Farid, 2005); (b) Montagem de John Kerry e Jane Fonda (Johnson and Farid, 2005); (c) Jeffery Wong Su En recebendo o prêmio da Rainha Elizabeth II (Redi et al., 2011); (d) Primeiro-ministro paquistanês Yousaf Gilani (www.fourandsix.com, 2012); (e) Montagem iraquiana de Mísseis (Irene et al., 2011); (f) Capa da revista *Times* documentando o caso de O. J. Simpson (Redi et al., 2011). 3

2.1 Mandrill. No canto superior à esquerda, a imagem original; seguindo da esquerda para a direita, da segunda até a quarta imagem, os canais RGB visualizados separadamente, cada um tendo suas cores mostradas de forma caracterizada (originalmente, cada matriz apresenta apenas valores de intensidade de acordo com a sensibilidade de seus respectivos canais). 7

2.2 Imagem original em (a), e seus canais com cores: luminância (b), diferença do vermelho (c) e diferença do azul (d). Na segunda linha de imagens, cada canal em escala de cinza (*single channel image*, respectivamente. 8

2.3 Representação visual da vizinhança g_c em função do raio e número de pixels vizinhos. 10

2.4 Vizinhanças circulares para diferentes valores de (P,R). 11

2.5 Representação do pixel central com seus vizinhos, considerando a limiarização. 12

2.6 Esquerda: células 8x8 do HOG. Direita: Visualização dos descritores HOG da imagem exibindo os histogramas normalizados nas células 8x8. Pode-se detectar que a direção dominante do histograma captura as formas na imagem, sendo mais evidente nos membros inferiores. 14
2.7 Filtros utilizados para o cálculo do WLD. 16
2.8 Vizinhança simétrica quadrada para diferentes valores de P e R. 17
2.9 Exemplo de um problema linearmente separável com duas classes, e com dois possíveis classificadores. 18

3.1 Método proposto. 21
3.2 Imagem manipulada no canal RGB (A), e seus componentes de luminância Y (B), crominância Cb (C) e Cr (D). 22

4.1 Resultado da comparação de acurácia entre os extratores utilizando os canais RGB e YCbCr, especificamente no canal Cr. 26
4.2 Resultado da comparação de acurácia entre os extratores utilizando os canais RGB e YCbCr, especificamente no canal Cr. 28

Lista de Tabelas

4.1 Tabela com os resultados comparativos entre os canais RGB e YCbCr por descritor sobre a base CASIA 1.0. 25

4.2 Tabela com os resultados para *CASIA v1.0* sem *feature selection.* 26

4.3 Tabela com os resultados para CASIA v1.0 utilizando *feature selection*. 27

4.4 Tabela com os resultados comparativos entre os canais RGB e YCbCr por descritor para a base CASIA 2.0. 28

4.5 Tabela com os resultados para *CASIA v2.0* sem *feature selection.* 29

4.6 Tabela com os resultados para CASIA v2.0 utilizando *feature selection*. 29

4.7 Comparação dos melhores resultados obtidos com deteção utilizando *Deep Learning*[1]. 29

CAPÍTULO 1

Introdução

Imagens digitais possuem várias aplicações, como: mercado financeiro (digitalização de cheques e notas promissórias, por exemplo), diagnósticos médicos, formas de arte, fotografia, etc. Nos últimos anos, a disponibilidade de ferramentas de edição de imagens, tanto livres quanto pagas (Adobe Photoshop,GIMP, entre outros) tem aumentado, tornando-se mais fácil de duplicar e manipular o conteúdo das imagens sem causar degradações de forma significativa, nem mesmo deixar vestígios de edição a olhos destreinados. Falsificação é um termo empregado na área de imagens digitais para definir quaisquer modificações (mesmo a nível de poucos pixels) efetuadas em uma imagem de forma a adulterar seu conteúdo. Também é uma palavra cujo teor pode ser considerado subjetivo. Uma imagem pode ser considerada uma falsificação dependendo do contexto que foi apresentada. Ainda mais, a larga utilização de redes sociais podem facilmente propagar imagens alteradas, isoladas ou com conteúdos de *fake news* que corroboram ainda mais a sua falsa alegação, propagando uma falsa autenticidade ao leitor *a priori*.

Análise forense de Imagens Digitais ou *Digital Image Forensis* [2] é a área de estudo que lida com a análise de manipulações maliciosas e autenticação de imagens digitais. Detecção de manipulação em imagens pode ser primariamente categorizado em duas abordagens, a ativa e a passiva. [3]. A abordagem ativa é baseada em informações adicionais embutidas na imagem digital para detecção de adulterações, como assinaturas digitais e marcas-d'água. Tal informação pode ser usada para certificar a originalidade de uma imagem; no entanto, a abordagem ativa requer que informações adicionais estejam embutidas nela durante o processo de captura, ou num estágio posterior por uma pessoa autorizada. Caso as informações da imagem original sejam desconhecidas (imagens presentes na web), então a abordagem ativa é impossível ou ineficiente.

Por outro lado, a abordagem passiva (ou *blind-based*) é capaz de detectar manipulações sem informações adicionais. Ela detecta a manipulação extraindo características intrísecas da

imagem, tendo como objetivo a detecção de falsificação, e identificação do dispositivo que as obteve. Pode ser sub-categorizada em manipulação dependente e independente. A que será objeto de estudo deste trabalho será a dependente, que trata das ações de *copy-move* e *splicing*. *Copy-Move Forgery Detection* (CMFD) aborda a cópia e colagem de um elemento original da imagem sobre ela mesma (como mostra a Figura 1.3), e *splicing*, que baseia-se no princípio de inserir em uma imagem um elemento oriundo de outra imagem, alterando seu contexto original, como mostra a Figura 1.1. Das técnicas na literatura, CMFD é amplamente utilizada [3]. Já a manipulação independente trata de outras manipulações digitais, como compressão, reamostragem e outras inconsistências inerentes.

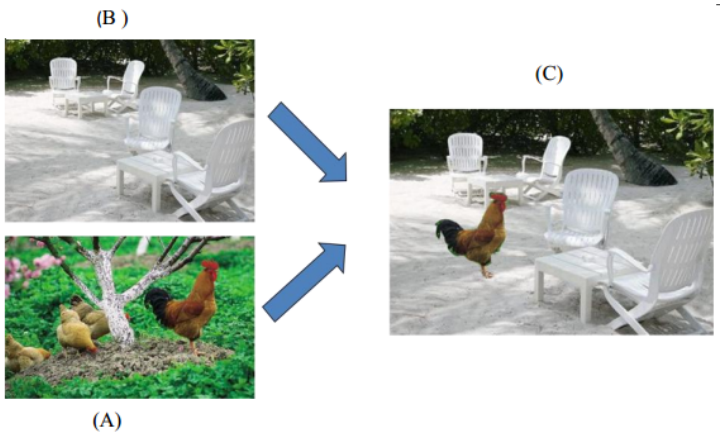

Figura 1.1 Um exemplo de *splicing forgery*, onde uma região de uma imagem origem é copiada e colada para uma destino; (A) é a imagem fonte, e (B) é a destino, cuja composição pode ser vista em (C).

Fonte: Nor Bakiah, Ainuddin Wahid, *et al.* [3]

Uma das primeiras tentativas de identificar regiões adulteradas foi proposta por Fridich *et al* (2003), onde os autores propuseram um método de deteção de *copy-move* usando a Transformada Discreta do Cosseno (DCT - *Discrete Cosine Transform*). Neste método, a imagem é dividida em blocos de tamanho 16x16 para extração de características. Então, os coeficientes DCT dos blocos são ordenados e sobrepostos de forma a ordenar suas representações lexicográficas, já que, havendo um agrupamento dos elementos por similaridades, faz com que as comparações entre os blocos deixem de ser realizadas entre um e todos, evitando comparações repetitivas, e em consequência, reduzindo o custo computacional. Popescu and Farid

Figura 1.2 Um exemplo de *copy-move forgery*: (A) apresenta a imagem original, e (B) mostra o desaparecimento de um dos garotos, devido à cópia e colagem de regiões de folhas para outra área.

Fonte: Nor Bakiah, Ainuddin Wahid, *et al.* [3]

Figura 1.3 Falsificações recentes em imagens documentadas: (a) Composição de Brad Pitt e Cher (Johnson and Farid, 2005); (b) Montagem de John Kerry e Jane Fonda (Johnson and Farid, 2005); (c) Jeffery Wong Su En recebendo o prêmio da Rainha Elizabeth II (Redi et al., 2011); (d) Primeiro-ministro paquistanês Yousaf Gilani (www.fourandsix.com, 2012); (e) Montagem iraquiana de Mísseis (Irene et al., 2011); (f) Capa da revista *Times* documentando o caso de O. J. Simpson (Redi et al., 2011).

Fonte: Vidhi P. Raval [2]

(2004) apresentaram um método utilizando Análise de Componentes Principais (PCA - *Principal Component Analysis*) para a representação dos segmentos, criando a sobreposição de blocos quadrados; métodos baseados em PCA resultam em uma redução de custo computacional, pois é gerada uma representação das características em uma dimensão reduzida; tal representação é robusta a pequenas variações na imagem devido à compressão com perdas e ruído aditivo, justi-

ficando seu uso. Kang e Wei (2008) propuseram o uso de Decomposição em Valores Singulares (SVD - *Singular Value Decomposition*), que é um processo de fatoração de uma matriz real ou complexa; é assumido que tal procedimento é útil para identificar as regiões manipuladas em uma imagem digital, devido ao fato de as características obtidas por tal método são invariantes a transformações algébricas, geométricas, entre outras pertubações . O SVD foi usado de forma a obter vetores de características, bem como redução de dimensionalidade. Ordenação lexicográfica também é aplicada nas linhas e vetores de colunas, e blocos similares são identificados para detectar regiões adulteradas. Huang *et al.* (2009) usaram o algoritmo de Transformada de Características Invariante à Escala (SIFT - *Scale-Invariant Feature Transform*), - que nada mais é que um algoritmo de detecção e descrição de características locais - para detectar manipulações do tipo *copy-move*; neste artigo, os autores introduziram o algoritmo SIFT usando casamento de características (*feature matching*). O algoritmo apresenta bons resultados mesmo quando a imagem está ruidosa ou comprimida. Ghorbani *et al.*(2011) propuseram o método DWT-DCT, que é uma combinação da Transformada Discreta de Wavelet (DWT- *Discrete Wavelet Transform*), uma transformada que representa o sinal em uma decomposição dinâmica de sub-bandas, e a Transformada Discreta do Cosseno. Autores utilizaram DWT e dividiram a imagem em sub-bandas e então utilizaram o método de Transformada Discreta do Cosseno com decomposição e quantização de coeficientes (DCT-QCD - *Discrete Cosine Transform Quantization Coefficients Decomposition*) em vetores-linha para reduzir o comprimento do vetor. Após uma ordenação Lexicográfica dos vetores-linha, é executado deslocamento de vetores. Finalmente, o vetor deslocado é comparado com os limiares, e a região manipulada na imagem é realçada. Cao *et al.* (2012) Propôs um algoritmo robusto de detecção de *copy-move* em 2012. Autores também usaram DCT para detectar coeficientes DC, onde cada bloco é representado por um círculo de blocos, e a extração de características é feita em cada círculo de blocos; após isto, é feita uma busca por pares de blocos similares, detectando regiões manipuladas.

1.1 Objetivo

Este trabalho tem como objetivo propor um modelo de detecção de imagens manipuladas através do uso conjunto de um classificador e diferentes extratores de características, comparando

as combinações de extratores entre si de forma a encontrar a combinação com a maior taxa de acerto.

A partir das contribuições em [4], que utiliza apenas o WLD, é efetuado o uso de múltiplos extratores de características na coleta de informações de uma imagem, de modo a aumentar a precisão na classificação entre autêntica ou manipulada, levando-se em consideração manipulações do tipo *copy-move* e *splicing*. Mostra também como etapas de pré-processamento acerca de conversão de canais de cores podem tornar a operação de detecção de *forgeries* mais eficiente com determinados *feature extractors*.

1.2 Estrutura do trabalho

A divisão deste trabalho é efetuada da seguinte forma: No Capítulo 2, é apresentado um *background* acerca dos conceitos que serão explorados ao longo do trabalho, como os espaços de cores que serão utilizados, bem como extração de características, e as técnicas de extração de características e classificação; o Capítulo 3 apresenta uma descrição do método proposto, apresentado ao leitor onde o problema será atacado; o Capítulo 4 demonstra os passos que se seguem para a execução da solução para o problema apresentado, mostrando como o uso de variadas técnicas de extração de características podem aumentar as taxas de acerto, se comparadas com extrações, bem como as ferramentas que serão utilizadas para a execução da proposta. O Capítulo 5 apresenta como foi realizada a execução dos experimentos, as bases de dados utilizadas e seus respectivos resultados, demonstrando a eficácia da metodologia proposta; Capítulo 6 apresenta as conclusões que foram obtidas através dos experimentos, e o quão eficiente foi para solucionar o problema inicialmente apresentado; também são apresentadas algumas ideias futuras que podem ser empregadas como novas soluções e possíveis aprimoramentos.

CAPÍTULO 2

Revisão bibliográfica

Este capitulo tem como objetivo apresentar, de uma forma geral, termos e conceitos da área de Processamento de Imagens e Aprendizagem de Máquina, de modo a facilitar a compreensão do trabalho elaborado.

2.1 Imagem Digital

Uma imagem digital é uma representação bidimensional e finita de pontos, na qual a menor unidade constituinte é denominada pixel. Sua construção é realizada através da aquisição de sinais eletromagnéticos em sensores digitais, na qual é efetuada uma conversão para o domínio discreto de uma representação contínua e tridimensional do mundo real.

Tal procedimento envolve princípios de quantização, na qual determina a cor e resolução de uma determinada imagem, ou seja, qual o número máximo de cores distintas podem ser representadas, além do tamanho que a matriz bidimensional que representa a imagem pode possuir, definindo o conceito de resolução espacial.

2.1.1 Espaço de Cores

O intuito de criação de um espaço de cores é organizar a representação dos coeficientes que transportam as informações relevantes de cor e intensidade presentes em uma imagem seguindo critérios devidamente embasados dentro de um sistema de coordenadas. Geralmente, o uso de tais modelos tanto são voltados para facilitação de uso em *hardware* (sistema de cores para impressoras, ou monitores de Tubos de Raios Catódicos (*Cathodic Tube Ray Television*- CRT) ou Telas de Cristal Líquido (*Liquid Crystal Display* - LCD), quanto para *software* (aplicações que trabalham com manipulação de cores). A figura 2.1 ilustra a separação de uma imagem em seus três canais no espaço RGB (*Red/Green/Blue*).

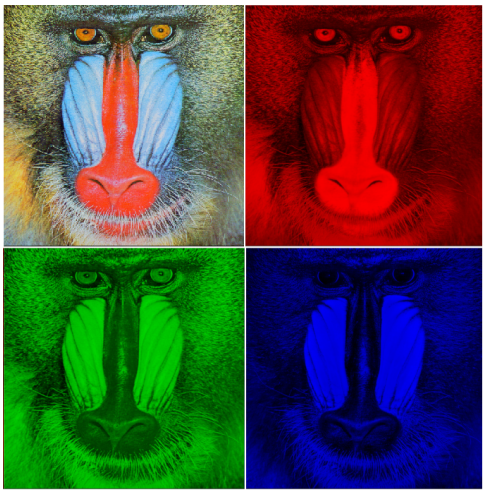

Figura 2.1 Mandrill. No canto superior à esquerda, a imagem original; seguindo da esquerda para a direita, da segunda até a quarta imagem, os canais RGB visualizados separadamente, cada um tendo suas cores mostradas de forma caracterizada (originalmente, cada matriz apresenta apenas valores de intensidade de acordo com a sensibilidade de seus respectivos canais).

Fonte: https://www.r-bloggers.com/color-quantization-in-r/

2.1.1.1 Espaço de cores YCbCr

Trata-se de um espaço de cores utilizado em sistemas de vídeo digital. A informação de luminosidade (luminância) é carregada pela componente Y, e a informação de cor é transportada pelos outros dois canais resultantes. Gera vantagens para transmissão e compressão de vídeo, pois ao separar o canal de luminância Y, este pode ser armazenado com resolução superior ou transmitido em alta largura de banda, enquanto os canais CbCr podem ser transmitidos em uma largura de banda reduzida, subamostrados ou comprimidos, melhorando a eficiência de envio, baseado no fato de que os seres humanos são mais sensíveis às informações em preto-e-branco do que acerca das cores; este processo é chamado de *chroma subsampling*, e também é empregada em muitos métodos de codificação, tanto analógicos quanto digitais, e na codificação JPEG. Um exemplo da separação dos canais pode ser visualizada na Figura 4.1.

O espaço YCbCr [5] pode ser convertido para o espaço RGB através dos seguintes mapeamentos:

$$Y = 0.257R + 0.504G + 0.098B + 16 \tag{2.1}$$

$$C_r = 0.439R - 0.368G - 0.071B + 128 \tag{2.2}$$

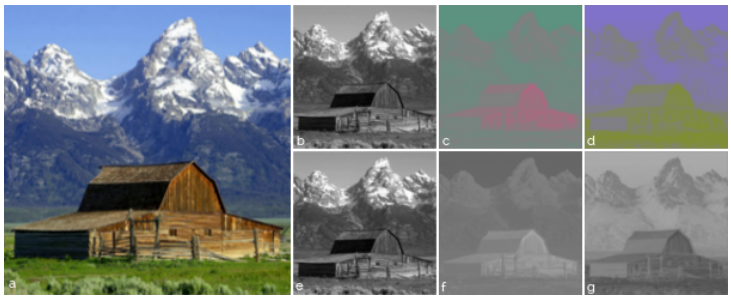

Figura 2.2 Imagem original em (a), e seus canais com cores: luminância (b), diferença do vermelho (c) e diferença do azul (d). Na segunda linha de imagens, cada canal em escala de cinza (*single channel image*, respectivamente.

Fonte: https://www.r-bloggers.com/color-quantization-in-r/

$$C_b = -0.148R - 0.291G + 0.439B + 128 \tag{2.3}$$

2.1.1.2 Espaço de cores RGB

O espaço de cores RGB (*red,green* e *blue)* possui caráter aditivo, isto é, realiza a combinação de cores, chamadas primárias, para produzir um novo conjunto de cores, cujos resultados dependem da cromaticidade dos canais primários. Sua especificação completa também requer uma curva de correção gama e cromaticidade de ponto branco. É um modelo de cores conveniente para computação gráfica devido ao fato de o sistema visual humano funcionar de forma similar, mas não idêntica, ao sistema de cores RGB. É o principal modelo utilizado em equipamentos como monitores de TV e computadores, além de câmeras digitais usufruírem da mesma técnica. A Figura 2.1 ilustra a separação dos canais RGB.

O espaço RGB [5] pode ser convertido para o espaço RGB através dos seguintes mapeamentos:

$$R = 1.164(Y - 16) + 1.596(C_r - 128) \tag{2.4}$$

$$G = 1.164(Y - 16) + 0.813(C_r - 128) - 0.392(C - b - 128) \tag{2.5}$$

$$B = 1.164(Y - 16) + 2.017(C_b - 128) \tag{2.6}$$

2.2 Seleção de Atributos com *Local Linear Based* (LLB)

Seleção de atributos (*features*) é o processo de escolha de um subconjunto de atributos considerados relevantes (variáveis, preditores), para que sejam utilizados em um modelo de classificação. Entende-se que os dados contêm muitos atributos que são redundantes/irrelevantes, portanto utilizar tal procedimento possibilita vantagens, como: prevenção do *curse of dimensionality* [6], melhoria na generalização evitando *overfitting*, tempos de treinamento menores, além de simplificar os modelos, tornando-os mais fáceis de serem interpretados por usuários. É importante ressaltar que a técnica de seleção de atributos deve ser diferenciada da extração de atributos: enquanto a primeira exclui atributos presentes nos dados sem alterá-los, a segunda cria novas combinações de atributos. Para este trabalho, será utilizada a seleção de atributos utilizando LLB (*Local Linear Based*).

LLB é um método de *feature selection* proposto em [7], em que a ideia-chave é decompor um problema complexo não-linear em um conjunto de porções menores lineares utilizando aprendizado local [8]; utilizar tal processo é preferível em vez de projetar o conjunto de informações em um espaço dimensional superior[7]. O algoritmo proposto baseia-se em técnicas de análise numérica e aprendizagem de máquina, sem fazer suposições a respeito da distribuição de dados inerente. Dada a entrada $D = (x_n, y_n)_{n=1}^{N}$ (x_n representa os exemplos, e y_n determina o n-ésimo rótulo associado ao n-ésimo exemplo), são executadas sucessivas iterações de modo a ajustar os pesos w_k de cada atributo, até que um critério de parada seja satisfeito, que é definido por um parâmetro θ.

2.3 *Local Binary Patterns*

É um tipo de descritor de textura em tons de cinza bastante eficiente, na qual rotula os pixels na imagem limiarizando a vizinhança de cada pixel, e considera o resultado como um número binário. Uma de suas propriedades mais importantes no mundo real é sua robustez à mudanças monotômicas na escala de cinza causadas, por exemplo, por variações na iluminação. Outra propriedade importante reside no fato de que possui uma simplicidade de uso computacional, na qual torna possível analisar imagens em condições de tempo real. A ideia básica é resumir a

estrutura de textura local em um dado ponto da imagem comparando-o com cada pixel em sua vizinhança.

A partir da definição formal em [9], a textura T em uma vizinhança local de uma imagem em tons de cinza pode ser definida como uma distribuição conjunta de níveis de cinza dos pixels P(P>1) de acordo com a seguinte equação:

$$T = t(g_c, g_0, ..., g_{P-1}) \tag{2.7}$$

onde g_c é o valor de cinza do pixel central correspondente à sua vizinhança $g_p(p = 0, 1, ..., P-1)$, cujas dimensões da vizinhança são regidas pelo raio da circunferência R ao redor do pixel. Supondo que a coordenada deste seja (0,0), a coordenada de um g_p qualquer seja dada por $(-R \cdot sen(2\pi p/P), R \cdot cos(2\pi p/P))$. As figuras 2.4 e 2.3 representam graficamente de maneira mais intuitiva a relação entre a quantidade de pixels e o raio da circunferência projetada.

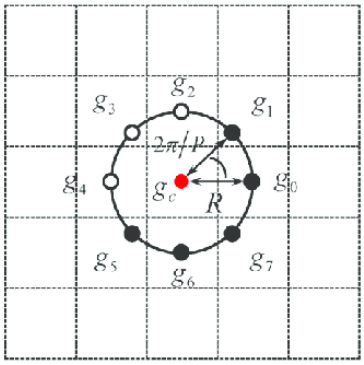

Figura 2.3 Representação visual da vizinhança g_c em função do raio e número de pixels vizinhos.

Fonte:

https://www.researchgate.net/profile/Jin_Tae_Kwak/publication/272682413/figure/fig2/

É possível subtrair g_c dos valores em níveis de cinza da vizinhança circular g_p:

$$T = t(g_c, g_0 - g_c, ..., g_{P-1} - g_c) \tag{2.8}$$

Quando se assume que os valores de g_c são independentes de de g_i, onde $(0 \leq i \leq P-1)$, é

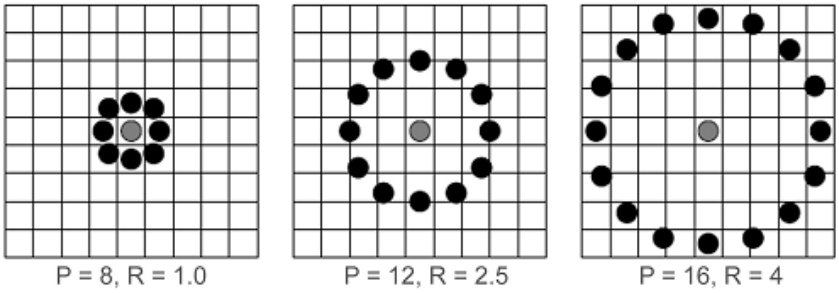

Figura 2.4 Vizinhanças circulares para diferentes valores de (P,R).

Fonte: http://circabook.com/local-binary-patterns-and-its-application-to-facial-image-/

possível aproximar T como:

$$T \approx t(g_0 - g_c, ..., g_{P-1} - g_c) \tag{2.9}$$

Com os valores de textura obtidos, um vetor binário de tamanho P pode ser obtido a partir da estimativa de *threshold* sobre cada um dos P elementos da vizinhança de g_c em relação a este elemento; sendo assim, a cada g_i, pode ser aplicada uma função de estimativa de limiarização s que é dada por:

$$s(p_n - p_c) = \begin{cases} 1, p_n - p_c \geq 0 \\ 0, p_n - p_c < 0 \end{cases} \tag{2.10}$$

Portanto, a textura pode ser agora composta da seguinte maneira:

$$T \approx t(s(g_0 - g_c), ..., s(g_{P-1} - g_c)) \tag{2.11}$$

Os valores altamente discriminativos fazem registros de valores dos padrões na vizinhança de cada pixel em um histograma com P-dimensões. O LBP de um pixel p_c com vizinhança circular (P,R) pode ser representado por $LBP_{P,R}$, e calculado através da equação [10]:

$$LBP_{P,R} = \sum_{n=0}^{P-1} 2^n s(p_n - p_c)^2 \tag{2.12}$$

Quando cada sinal s(x) é multiplicado por um fator binomial 2^p, a equação 2.10 devolve um valor que caracteriza a estrutura espacial da textura local da imagem naquele pixel. Quando o cálculo do LBP em todos os pixels é efetuado, o histograma é calculado com 2^p pixels; ele é tratado como um descritor LBP. O processo pode ser visualizado na figura 2.5.

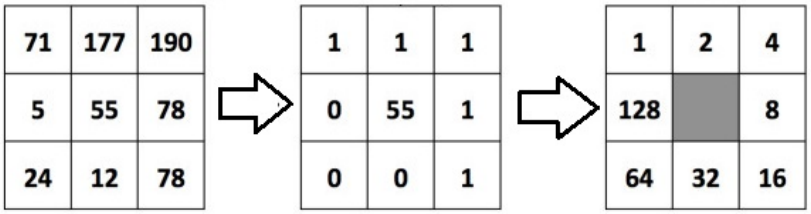

Figura 2.5 Representação do pixel central com seus vizinhos, considerando a limiarização.

Existem diversos variantes de operadores de LBP [11]: Invariante a rotação , uniforme, e uniforme e invariante a rotação [10], sendo este o selecionado para a *feature extraction* no trabalho atual, por se tratar de um operador invariante a medidas, isto é, sua saída não é afetada por qualquer transformação monotônica na escala de cinza, sendo um excelente medidor de padrão espacial mas, por definição, não é capaz de captar contrastes.

2.3.1 LBP Uniforme e Invariante a Rotação

De acordo com [8], o operador LBP possui 2^p valores de saída que correspondem a 2^p padrões binários que são formados por P pixels no grupo de vizinhos. À medida que a imagem gira, os valores em tons de cinza p_i movem-se ao longo do perímetro circular ao redor de p_c, na qual gira em um padrão espacial binário resultando em valores distintos de LBP. Isso não se aplica a padrões que incluem 0's e 1's que permanecem constantes a cada ângulo de rotação. Para identificar os padrões uniformes, a medida de uniformidade U denota o número de transições espaciais (mudanças de 0 para 1 ou 1 para 0) no padrão em questão:

$$LBP^{uir} = \begin{cases} \sum_{n=0}^{P-1} s(g_i - g_c), U(LBP) \leq 2 \\ P+1, C.C. \end{cases}$$

(2.13)

Onde

$$U(LBP) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{n=0}^{P-1} |s(g_n - g_c) - s(g_{n-1} - g_c)| \tag{2.14}$$

2.4 *Histogram of Oriented Gradient (HOG)*

É um descritor de características utilizado na área de Visão Computacional e Processamento de Imagens para detectar objetos. De um modo geral, a técnica HOG contabiliza as ocorrências de orientação no gradiente em porções localizadas da imagem, chamado também de janela de detecção ou região de interesse (ROI).Ele é similar ao *Edge Orientation Histograms* [12] e transformadas com *features* invariantes a escala, mas difere pelo fato de computar uma grade de células uniformemente espaçadas com alta densidade, além de usar sobreposição na normalização do contraste local, a fim de melhorar a acurácia. O algoritmo para a extração de features é apresentado na seção a seguir.

2.4.1 Passos para extração de features com HOG

O processo de extração de *features* é composta pelas principais etapas:

1. A primeira etapa é realizar um pré-processamento na imagem de interesse, para certificar uma normalização de cores e valores gama, embora [13] considere uma etapa dispensável, por possuir baixos ganhos de desempenho.

2. De forma a calcular os valores do gradiente, é aplicada uma máscara de derivadas, tanto na direção horizontal quanto na vertical. A intensidade é filtrada com os seguintes *kernels* de filtro:

$$D_x = [-1, 0, 1] \tag{2.15}$$

$$D_y = [-1, 0, 1]^T \tag{2.16}$$

Dada uma imagem I, é possível obter as derivadas x e y usando a operação de convolução $I_x = I_x * D_x$, e $I_y = I_y * Dy$, com a magnitude calculada por $|G| = \sqrt{I_X^2 + I_Y^2}$ e a orientação do gradiente obtida por $\theta = arctan\frac{I_y}{I_X}$. No entanto, [13] mostra que também é possível aplicar uma máscara de Sobel, embora essas máscaras sejam piores para detectar certos objetos em imagens.

3. Em seguida, divide-se a imagem em pequenas regiões conectadas chamadas células, e cada pixel dentro da célula contribui em votação de pesos para um canal de histograma baseado em orientações, que entrega uma orientação de acordo com o *bin* mais próximo no intervalo entre entre 0 e 180 graus. A imagem 2.6 (a) ilustra a divisão de uma imagem em células de tamanho 8x8.

Figura 2.6 Esquerda: células 8x8 do HOG. Direita: Visualização dos descritores HOG da imagem exibindo os histogramas normalizados nas células 8x8. Pode-se detectar que a direção dominante do histograma captura as formas na imagem, sendo mais evidente nos membros inferiores.

Fonte: http://www.learnopencv.com/histogram-of-oriented-gradients/

4. O próximo passo é fazer o cálculo do histograma de gradientes para cada célula. Gradientes de uma imagem são sensíveis à uma iluminação geral. Para levar em conta mudanças de iluminação e contraste, a intensidade do gradiente deve ser normalizada localmente, a fim de o histograma não seja afetado por variações de luminosidade, e isto requer que um grupo

de células se agrupem em blocos maiores, mas conectados espacialmente. O grupo de células em um bloco serve de base para o agrupamento e normalização dos histogramas. O grupo normalizado de histogramas representa o histograma do bloco, cuja normalização é dada por:

$$b' = \frac{b}{\sqrt{||b||^2 + e^2}} \tag{2.17}$$

Onde b é o vetor normalizado contendo todos os histogramas em um dado bloco, e e é uma constante de pequeno valor (e.g.: $e \approx 1$). O conjunto destes componentes compõem o descritor de características da imagem.

2.5 *Weber Local Descriptor*

É um descritor robusto, baseado no conceito de que a sensibilidade visual humana é baseada na mudança da intensidade de estímulo [14]. Possui seus fundamentos embasados na Lei de Weber, que é determinada através de dois componentes importantes: excitação diferencial, definido como D e orientação do gradiente $\Phi(p_c)$, definida por [4]:

$$\Phi(p_c) = arctan\left(\frac{k_s^{11}}{k_s^{10}}\right) \tag{2.18}$$

Na qual k_s^{11} e k_s^{10} são as saídas dos filtros f_{11} e f_{10}.

Ernst Weber conceituou no século 19 que a razão do aumento com a intensidade de fundo é uma constante [14]. Esta relação pode ser definida através da seguinte equação que define a Lei de Weber:

$$\frac{\Delta I}{I} = k \tag{2.19}$$

Onde ΔI representa o limiar de aumento (diferença perceptível para discriminação), e I indica que a proporção do lado esquerdo da equação é sempre um valor constante, apesar das variações no denominador I.

O elemento de excitação diferencial D é utilizado para estimar diferenças de intensidade entre um pixel central e seus vizinhos, fazendo com que seja possível encontrar variações de saliências dentro da imagem para simular o padrão de percepção de um ser humano. A excitação diferencial $D(p_c)$ para um pixel p_c é calculado da seguinte forma:

1. Calcula-se a diferença entre o pixel p_c e seus vizinhos através do filtro f_{00}, vide figura 2.8, que é dada por:

$$k_s^{00} = \sum_{i=0}^{N-1} (\Delta p_i) = \sum_{i=0}^{N-1} (p_i - p_c) \tag{2.20}$$

Na qual p_i é o i-ésimo vizinho do pixel p_c e N é a quantidade total de vizinhos.

2. Calcula-se a proporção das diferenças em relação à intensidade do pixel atual pelas saídas dos filtros f_{00} e f_{01}, que também podem ser vistos na figura 2.8:

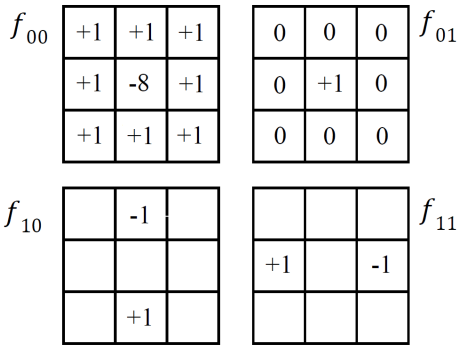

Figura 2.7 Filtros utilizados para o cálculo do WLD.

$$I = \frac{k_s^{00}}{k_s^{01}} = \sum_{i=0}^{N-1} \left(\frac{p_i - p_c}{p_c} \right) \tag{2.21}$$

A excitação diferencial $D(p_c)$ do pixel atual p_c é:

$$D(p_c) = arctan \left[\sum_{i=0}^{N-1} \left(\frac{p_i - p_c}{p_c} \right) \right] \tag{2.22}$$

A componente de orientação Φ é o gradiente de orientação $\Phi(p_c)$, e é calculada a partir da Equação 2.18.

Posteriormente, há um mapeamento de Φ para Φ', onde o intervalo original [-π/2, π/2] é transformado em [0, 2π]. Então, através de um processo de quantização, ele é mapeado em T direções dominantes. Após a computação de D e Φ, o histograma WLD é calculado dividindo em *bins* de acordo com as orientações dominantes. O histograma é tratado como um descritor WLD envolvendo três parâmetros (T,M,S), na qual T representa a quantidade de direções

dominantes, M é a quantidade de segmentos de excitação, e S é a quantidade de *bins* nos segmentos dos sub-histogramas; os detalhes podem ser encontrados em [14]. Um modelo simples de descritor utiliza uma vizinhança 3x3 circundando o pixel central; este modelo não consegue capturar adequadamente microestruturas existentes em diferentes escalas. Para contornar tal entrave, é empregado o uso de um descritor *multiscale* WLD, utilizando vizinhanças simétricas quadradas com diferentes tamanhos (P,R) (vide Figura 2.8, onde P representa a quantidade de pixels vizinhos em relação ao central, e R representa o raio (escala) de distância dos pixels. Para fins de *forgery detection*, utiliza-se três níveis de vizinhanças de pixels com P=8,16 e 24, bem como a escala variando com R=1,2 e 3. A versão final do histograma WLD *multiscale* é obtida através de concatenação dos três histogramas obtidos em cada um dos valores de vizinhança e escala [11].

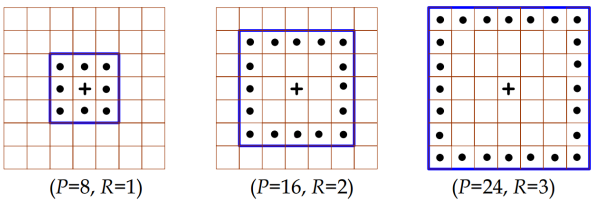

Figura 2.8 Vizinhança simétrica quadrada para diferentes valores de P e R.

Fonte: [14]

2.6 *Support Vector Machine* (SVM)

Trata-se de uma abordagem de Aprendizagem de Máquina bastante utilizada para problemas de classificação, bem como também os de regressão. Ela é uma técnica robusta que maximiza a precisão de predição de um modelo sem que haja um *overfitting*.

A sua aplicação para detectar manipulações é através da abordagem de divisão do problema em dois tipos de classes (autênticas e manipuladas), havendo uma alimentação do modelo com as *features* obtidas pelas imagens de entrada [8]. Com esta abordagem, os padrões de treinamento (x_i, y_i) são dados de modo que i = 1, 2, ... , M, $a_i \in R^d$, $b_i \in -1, +1$; a_i é um vetor de características do conjunto de treinamento, e b_i é o rótulo das classes, onde +1 e -1 definem as classes C_1 (imagens autênticas) e C_2 (imagens manipuladas). O objetivo é construir um

modelo de classificador a partir dos padrões existentes, na qual seja minimizada a probabilidade de classificação incorreta de um conjunto de exemplos. Sendo assim, o SVM elabora um hiperplano ótimo:

$$g(x) = C^T x + C_0 \tag{2.23}$$

Onde é encontrada a margem máxima de classificação dentro do melhor desempenho de generalização. A margem é definida como a distância de separação entre os pontos de dados mais próximos de cada classe e o hiperplano esboçado. De um modo geral, SVM é um classificador linear que possui a capacidade de categorizar dados separáveis, no entanto os vetores de características geralmente não conseguem ser linearmente separáveis. Para suprimir tal problema, são utilizadas funções de *kernel*, na qual mapeiam o espaço de características de entrada para um espaço de dimensão superior, de tal modo que os dados possam se tornar linearmente separáveis; o desempenho do classificador será afetado pela escolha da função de *kernel* de acordo com o problema a ser tratado. Para este trabalho, foi utilizado o *kernel* polinomial, que é dado por:

$$K(x_i, x_j) = (\lambda x_i^T x_j + 1)^d, \lambda > 0 \tag{2.24}$$

Onde λ e d são parâmetros de ajuste na função de *kernel* [15].

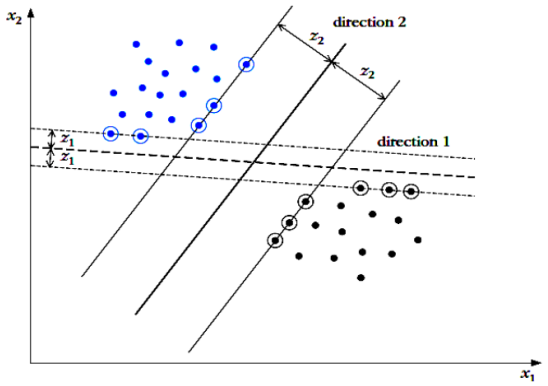

Figura 2.9 Exemplo de um problema linearmente separável com duas classes, e com dois possíveis classificadores.

Fonte: [16]

A Figura 2.9 ilustra um exemplo de um problema linearmente separável de duas classes. O hiperplano com uma linha escura ao longo da direção 2 é a selecionada, pois a margem $2Z_2$ é bem maior que a margem de separação $2Z_1$, que se estende ao longo da direção 1, pois o objetivo geral é encontrar um hiperplano que possibilite oferecer uma margem máxima de separação entre as classes; O classificador SVM tenta explorar e encontrar os coeficientes que maximizam tal margem.

CAPÍTULO 3

Método proposto e Metodologia

Apesar de atualmente já existirem vários métodos para a detecção de *copy-move forgery detection*, este ainda é um campo aquecido e, por conta disto, vários estudos têm realizado melhorias, e buscado um aumento nas taxas de acurácia em comparação com o estado da arte. A maioria dos métodos divulgados e abordados apostam em métodos adaptativos [17], envolvendo técnicas de *matching* [18], transformadas como a DCT [19], DWT (*Discrete Wavelet Transform*) [20], além de PCA [21], e via extratores de características [4]. Indo além do último mencionado, alguns resultados de pesquisas formentam o uso de diferentes extratores de características aplicados a uma mesma imagem, de forma a aumentar a representação de descrição desta imagem, implicando numa maior variabilidade no modelo, possibilitando taxas ainda maiores de acerto. Em [4], faz-se uso apenas do WLD multi-escala para coletar características de uma imagem, onde tais *features* servem como entrada de um classificador de forma a classificar uma imagem entre autêntica ou forjada.

A proposta deste trabalho de graduação é, a partir de [4], que faz uso apenas do WLD, acrescentar os extratores HOG e LBP para a coleta de características de uma imagem, e em seguida utilizar um classificador SVM para confirmar ou não a sua autenticidade; para fins de comparação, faz-se a construção de diversos modelos com a combinação dos diferentes extratores mencionados, verificando em qual configuração é possível obter a mais alta taxa de acurácia. Apresentamos em detalhes o método proposto para o Trabalho de Graduação, o conjunto de dados utilizados, juntamente com o planejamento dos experimentos. O objetivo é avaliar e apresentar qual dos métodos obtém a melhor taxa de acerto na detecção de uma imagem adulterada.

A Figura 3.1 apresenta, na forma de diagrama de blocos, o funcionamento do método proposto, ilustrando todas as etapas desenvolvidas. Inicialmente, a imagem de entrada é convertida para o canal de cores YCbCr ou RGB. Tal uso justifica-se pelo fato dos falsificadores realizarem as edições na imagem no canal de cores RGB; pelo fato da visão humana, no sistema

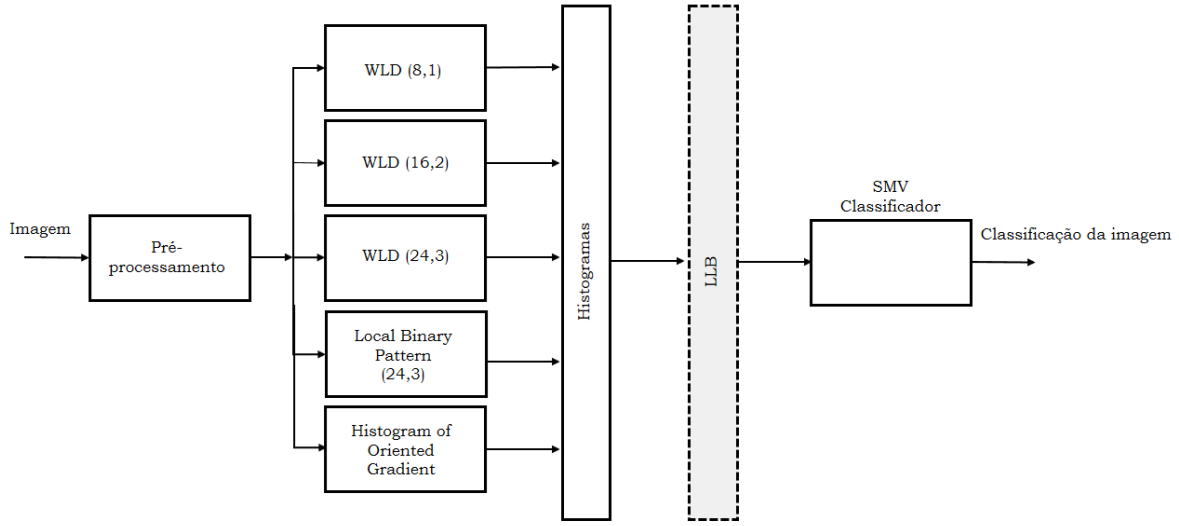

Figura 3.1 Método proposto.

YCbCr, ser menos sensível aos canais de crominância (a tendência é de perceber melhor as diferenças de luminância), pode-se encontrar traços de manipulações mais discretos inerente em tais canais, justificando o uso dest canal além do RGB; no entanto, mais à frente, com edições mais caprichadas, a detecção de anomalias na crominância tornam-se ineficientes. Nos casos em que não utiliza-se o canal de crominância, o canal RGB é simplesmente convertido para tons de cinza (canal de luminância); a conversão com pesos é empregada, e sua fórmula é expressa pela equação em (2.1); a imagem 3.2 mostra a representação de uma imagem que sofreu manipulação decomposta nos canais de luminânica e crominância.

Em seguida, o componente escolhido (luminância ou crominância) é utilizado para extrair as suas características na forma de cada um dos *feature descriptors* elucidados no capítulo 2, que são: WLD-*multiscale*, LBP e HOG; Cada uma de suas combinações geram modelos que serão executados e terão suas métricas comparadas entre si. Para o WLD-*multiscale*, considera-se a concatenação de histogramas de diferentes operadores de variação (P,R), de forma a representar as *features* da imagem: P é a contagem de pixels da vizinhança, e R é a escala espacial do operador. No caso do LBP e do HOG, são escolhidos um conjunto fixo de parâmetros (P,R) e tamanho NxN do bloco, respectivamente. Em seguida, um processo de *feature extraction* também é realizado, e seguindo a proposta em [4], será escolhido o LLB; modelos com e sem o

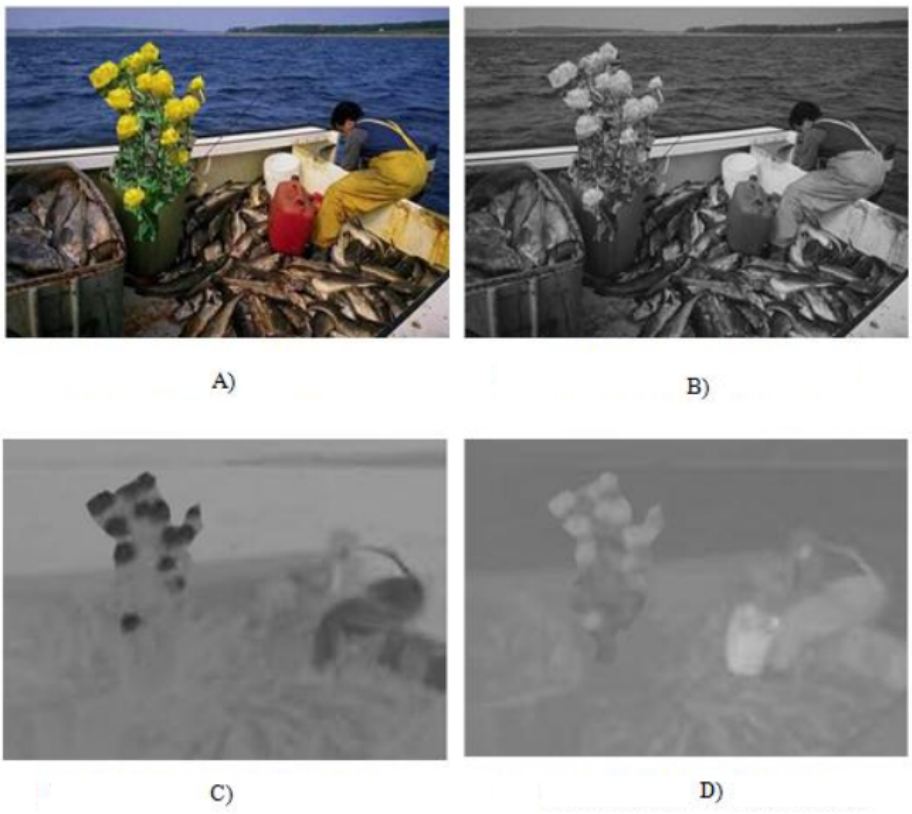

Figura 3.2 Imagem manipulada no canal RGB (A), e seus componentes de luminância Y (B), crominância Cb (C) e Cr (D).

processo de seleção de características serão empregados para fins de comparação. Finalmente, um classificador do tipo SVM, que foi previamente treinado e testado com imagens presentes em CASIA 1.0 e CASIA 2.0, é utilizado para decidir se a imagem foi manipulada. A métrica de comparação entre os modelos elaborados neste trabalho será a acurácia, que é a quantidade total de acertos dividida pela quantidade total de imagens testadas.

CAPÍTULO 4

Experimentos e Resultados

Neste capítulo são apresentados os resultados obtidos por cada um dos descritores de características sob cada base de imagem que foram utilizadas nesse trabalho. Os modelos foram implementados utilizando o *Matlab* 2014 em conjunto com a ferramenta gratuita *Octave*. O computador utilizado possui um processador Intel®Core i5, com 2,3Ghz de clock e 8Gb de memória RAM.

Conforme visto em [4], foi constatado experimentalmente que o canal de crominância Cr gera um melhor resultado de acurácia em relação ao canal Cb, mostrando que este componente é mais sensível à detecção de *features* relevantes para detecção de falsificações. Foi também evidenciado a superioridade do canal Cr em relação ao RGB no tocante à detecção de características relevantes para detecção de eventuais manipulações do tipo *splicing*, reforçando o seu uso para este tipo de procedimento; tais resultados estão descritos no início das seções 5.1 e 5.2.

Foram utilizadas as bases de dados CASIA v1.0 e v2.0. CASIA 1.0 [22] possui um total de 1725 imagens, nas quais 800 imagens são autênticas, e 925 imagens foram manipuladas na forma *spliced* com tamanhos de 384x256 pixels, no formato JPEG. As imagens autênticas foram na maioria coletadas do banco do conjunto de imagens da *Corel*, e outras foram obtidas através das câmeras digitais dos criadores do projeto. Além disso, as imagens autênticas foram divididas em várias categorias (cenas, animais, arquitetura, pessoas, plantas, natureza e textura) de acordo com o conteúdo da imagem, e também foram considerados alguns critérios baseados na informação das categorias quando foi criado o banco de imagens. Já o conjunto v2.0 [23] possui um total de 12.323 imagens coloridas com resoluções que variam entre 240x160 a 900x600 pixels, nas quais 7491 imagens são autênticas e 5123 são falsificadas; neste conjunto, além do JPEG, foram adicionados formatos sem compressão, além de serem consideradas imagens JPEG com diferentes fatores de qualidade Q. As imagens foram coletadas da mesma forma que na primeira versão, com o adicional de versões em páginas web que foram devidamente au-

torizadas para uso. Nesta versão, imagens foram divididas na mesma quantidade de categorias que a versão anterior, com o adicional de uma versão de ambientes internos (*indoor*), para que seja considerado o impacto das iluminações na imagem. Também levou-se em consideração um pós-processamento das fronteiras das regiões recortadas e coladas nas imagens.

Para ambos os conjuntos, foi simulado o processo de *splicing* de várias maneiras distintas:

- Aleatoriamente utilizou *crop-and-paste* em regiões com diferentes formas (circulares, triangulares, retangulares, e com formas arbitrárias).
- Regiões cortadas da imagem podem ser processadas com *resizing*, *rotation* ou outra distorção, então é colada para gerar uma imagem forjada.
- Diferentes tamanhos (pequeno, médio e grande) das regiões embutidas são levadas em consideração.
- A maioria das imagens geradas que foram manipuladas foram levadas em consideração que fossem realísticas aos olhos humanos.

Exclusivamente para o CASIA v2.0, houve um processo aleatório de *crop-and-paste* em certas regiões de imagens. Diferentes combinações de extratores possibilita a criação de diferentes histogramas, possibilitando agrupar informações distintas sobre a mesma imagem, possibilitando assim observar distintos resultados de precisão na classificação. Para o WLD, várias combinações de T, M e S foram testadas, e ajustadas para valores que dão resultados ótimos, que que foram T=4, M=4 e S=20 [4]. O HOG possui foi aplicado com células de tamanho 64x64; um tamanho grande de célula faz com que se perca detalhes em pequena escala, mas possibilita operar com uma quantidade menor de *features*, diminuindo o tempo de treinamento do modelo do classificador, e prevenindo o efeito de *curse of dimensionality*[6]. No caso do LBP, [9] mostra que além de suas vantagens originais de robustez a operações tradicionais de processamento de imagens, bem como transformações espaciais como rotação ou *fliping*, afirma também que features obtidas a baixas frequências espaciais são mais estáveis do que comparado com as obtidas com frequências altas; seus experimentos mostraram que adicionar um filtro passa-baixas melhora o desempenho na detecção de *forgeries*; por isto, antes de a imagem ser enviada para o LBP, é aplicado um filtro gaussiano com dimensões 5x5 e $\sigma = 2$, de forma a remover componentes espaciais de frequência elevada. Utiliza-se apenas uma escala

de LBP, com vizinhança de 24 pixels e raio 3; a justificativa de usar apenas uma escala de LBP é a mesma que a utilizada no HOG, que é preferível preservar uma quantidade menor de características, sem que prejudique a qualidade das informações extraídas.
Para uma boa classificação do método proposto, foi empregada uma classificação SVM envolvendo um kernel RBF (*Radial Basis Function*), em conjunto com um kernel do tipo polinomial, sendo avaliada com um método de validação cruzada com 10 *folds*. Foi constatado em [4] que o kernel polinomial possui um melhor desempenho de resultados comparado com o kernel RBF, sendo neste caso empregado apenas o polinomial. É utilizado um *grid search method* para descobrir os parâmetros ótimos da SVM. Foi utilizada a implementação LIBSVM para MATLAB, usando os parâmetros C=1.0 e $\varepsilon = 10^{-3}$. O desempenho dos métodos é dado em termos de acurácia (valor médio em cima de dez iterações). [24].

4.1 Resultados para *CASIA v1.0*

Na primeira etapa dos experimentos, foram consideradas combinações de extratores sem realizar a redução de dimensionalidade. Primeiramente, realizou-se uma comparação de acurácia entre os 3 definidos utilizando o canal RGB e YCbCr, a fim de encontrar o melhor canal de cores para seu respectivo extrator; os resultados podem ser vistos na figura 4.2

Os resultados da tabela mostraram que, para a base de imagens atual, o uso do canal YCbCr foi superior em todos os testes com os extratores isolados, visíveis no gráfico da figura 4.1, mostrando que, para operações puras de *copy-move* e *splicing*, o canal mencionando armazenou melhor as anomalias resultantes das operações de edição nos canais de crominância, gerando *features* importantes para detecção no classificador. A tabela 4.1 exibe os valores obtidos em detalhes.

Feature Descriptor	RGB (%)	Crominância Cr (%)
HOG	$51,8023 \pm 2,1659$	$58,7791 \pm 2,2675$
LBP	$53,4884 \pm 0,0074$	$77,6131 \pm 3,6796$
WLD	$55,5233 \pm 1,3497$	$84,1552 \pm 0,6403$

Tabela 4.1 Tabela com os resultados comparativos entre os canais RGB e YCbCr por descritor sobre a base CASIA 1.0.

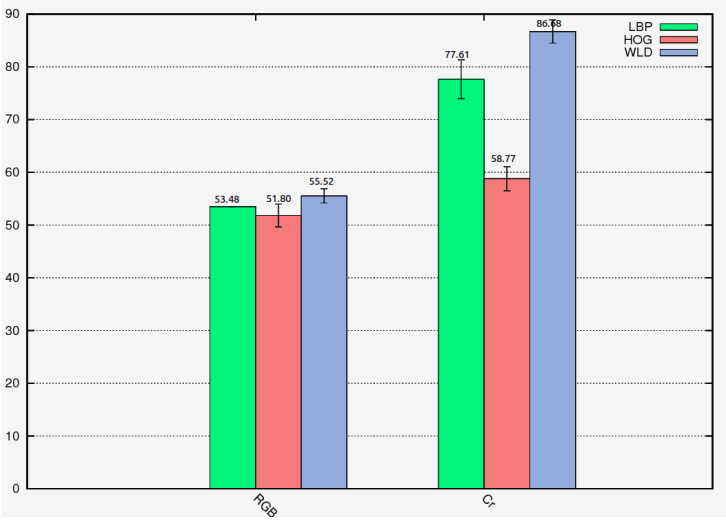

Figura 4.1 Resultado da comparação de acurácia entre os extratores utilizando os canais RGB e YCbCr, especificamente no canal Cr.

Os resultados da Tabela 4.2 foram gerados usando todas as combinações. É possível constatar que a associação de todos os extratores gerou a maior taxa de acurácia, mostrando a eficácia que o aumento de variabilidade gerado pelo uso de diferentes extratores numa mesma imagem proporciona.

Feature Descriptor	Acurácia (%)	N° de *features*
HOG	$58,7791 \pm 2,2675$	540
LBP	$77,6163 \pm 3,6796$	555
WLD	$86,6860 \pm 2,2003$	960
HOG + LBP	$77,3256 \pm 2,5268$	1095
HOG + WLD	$86,9186 \pm 2,3775$	1500
WLD + LBP	$88,0814 \pm 2,5157$	1515
HOG + WLD + LBP	$88,6628 \pm 2,0372$	2055

Tabela 4.2 Tabela com os resultados para *CASIA v1.0* sem *feature selection.*

Utilizando o LLB como *feature selection*, obteve-se novamente a acurácia mais elevada no caso em que todos os extratores são utilizados; encontrando o número ótimo de atributos escolhidos que maximize a métrica de comparação, é possível reduzir o tempo de treinamento e teste do classificador, além de aumentar as taxas de acerto; no entanto, o aumento não chega

a ser substancial em alguns casos, revelando uma baixa variância entre os atributos.

Feature Descriptor	Acurácia (%)	N° de *features* (k)
HOG	$60,0581 \pm 1,7334$	497
LBP	$78,7698 \pm 2,4534$	302
WLD	$88,1820 \pm 1,3077$	612
HOG + LBP	$79,4409 \pm 2,0844$	708
HOG + WLD	$88,0012 \pm 1.5778$	1325
WLD + LBP	$88,9535 \pm 3,6262$	925
HOG + WLD + LBP	$89,5491 \pm 0,1445$	1237

Tabela 4.3 Tabela com os resultados para CASIA v1.0 utilizando *feature selection*.

4.2 Resultados para *CASIA v2.0*

Com a segunda versão do database CASIA, foi possível submeter os modelos propostos a imagens com manipulação *copy-move* e *splicing* mais bem elaborada, envolvendo pós-processamento nas regiões alteradas com borramento, bem como operações de *scaling* e *rotation*. A quantidade maior de exemplos possibilitou a construção de um modelo de classificação que calculou uma melhor acurácia, embora o tempo de treinamento e teste do modelo de classificador tenha aumentado enormemente. A tabela abaixo mostra os valores de acurácia para os extratores, considerando os diferentes canais RGB e YCbCr; desta vez, constata-se a vantagem do canal RGB sobre o YCbCr nos extratores LBP ($94,5484\% \pm 2,4884$) e HOG ($91,8784\% \pm 0,9535$) (visível na figura 4.2), e há vantagem no YCbCr apenas com o WLD multi-escala ($84,1552 \pm 0,6403$) (detalhes visíveis na tabela 4.4); isto mostra que as técnicas de pós-processamento empregadas nessa nova base de imagens (como o borramento com máscara gaussiana) possibilitaram uma melhor coleta de atributos a imagem. Na tabela 4.5, é possível observar que a combinação de extratores no canal RGB com YCbCr apenas no WLD multi-escala, combinada com uma vasta quantidade de imagens para treinamento e teste, é possível ver um aumento na acurácia das combinações, reforçando mais uma vez que a utilização de diversos extratores combinados melhora a detecção de alterações causadas por manipulações. A tabela 4.2 apresenta os

resultados das combinações após um processo de seleção de atributos; do mesmo modo que o CASIA 1.0, resultado que são estatisticamente equivalentes mostram a baixa variância das features coletadas.

Feature Descriptor	RGB (%)	Crominância Cr (%)
HOG	$91,8784 \pm 0,9535$	$81,6153 \pm 1,3226$
LBP	$94,5484 \pm 2,4884$	$79,8300 \pm 0,2970$
WLD	$78,3316 \pm 0,2032$	$84,1552 \pm 0,6403$

Tabela 4.4 Tabela com os resultados comparativos entre os canais RGB e YCbCr por descritor para a base CASIA 2.0.

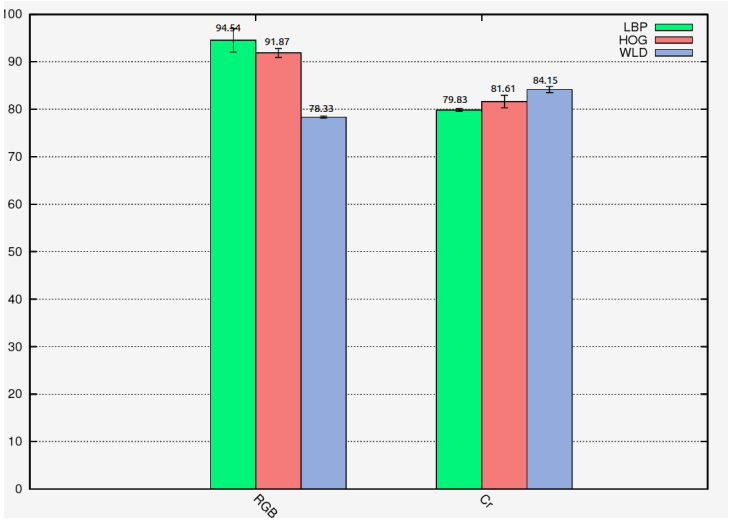

Figura 4.2 Resultado da comparação de acurácia entre os extratores utilizando os canais RGB e YCbCr, especificamente no canal Cr.

Quando compara-se os resultados obtidos neste trabalho com os produzidos através de *deep learning* em [1], pode-se verificar que as diferenças são evidentes quando as duas abordagens são comparadas na base CASIA v1.0; devido à pouca variabilidade e ausência de etapas de pré-processamento, houve uma maior dificuldade de se obter características concisas de imagens que sofreram alterações; tal diferença é mitigada quando foram executados sobre a base CASIA v2.0, onde o método proposto chegou próximo ao estado da arte. A tabela 5.7 apresenta os valores de forma comparativa.

Feature Descriptor	Acurácia (%)	*features* (k)
HOG	$81,6153 \pm 1,3226$	540
LBP	$79,8300 \pm 0,2970$	555
WLD	$84,1552 \pm 0,6403$	960
HOG + LBP	$95,5154 \pm 1,8876$	1095
HOG + WLD	$86,3231 \pm 1,1078$	1500
WLD + LBP	$96,9713 \pm 0,3856$	1515
HOG + WLD + LBP	$97,1413 \pm 0,3663$	2055

Tabela 4.5 Tabela com os resultados para *CASIA v2.0* sem *feature selection.*

Feature Descriptor	Acurácia (%)	*features* (*k*)
HOG	$92,6567 \pm 0,8449$	367
LBP	$80,9876 \pm 0,9784$	245
WLD	$84,9775 \pm 0,5889$	887
HOG + LBP	$95,6854 \pm 2,0418$	871
HOG + WLD	$86,9791 \pm 0,8756$	1278
WLD + LBP	$97,5452 \pm 0.3628$	1169
HOG + WLD + LBP	$97,6945 \pm 0,4514$	1476

Tabela 4.6 Tabela com os resultados para CASIA v2.0 utilizando *feature selection*.

Feature Descriptor	CASIA v1.0 (%)	CASIA v2.0 (%)
Método Proposto	89,5491	97,6945
Deep Learning [1]	98,04	97,83

Tabela 4.7 Comparação dos melhores resultados obtidos com deteção utilizando *Deep Learning*[1].

CAPÍTULO 5

Conclusão e trabalhos futuros

Com base nos resultados, observa-se que os extratores combinados possibilitaram taxas mais elevadas do que seria possível se fossem trabalhados isoladamente; no entanto, o acréscimo destes deve ser ponderado com cautela, pois é preciso preservar o equilíbrio entre a quantidade de *features* obtidas, e a quantidade de exemplos que se irá utilizar para treinar e testar o modelo de classificador; pois um número excessivo de atributos eleva grandemente o tempo de execução do classificador, além de incorrer no problema da *curse of dimensionality*, na qual afirma que a quantidade ideal de exemplos disponíveis para o classificador deve ser muito maior que o número de atributos; valores muito abaixo deste prejudicam grandemente as taxas de acurácia dos classificadores, entre eles o SVM.

Algumas alternativas que podem ser aplicadas em trabalhos futuros:

- Utilizar novos canais de cores e checar quais deles apresentam melhores resultados comparados com os obtidos neste trabalho;
- Executar as combinações em novas bases públicas de imagens;
- Utilizar SIFT e SURF como *feature extractors* substitutos aos extratores com menores taxas de acurácia, e checar o desempenho das combinações;
- Utilizar o *Ada Boost* e *Random Forest* como classificadores, e checar suas acurácias em comparação com o SVM.
- Experimentar uma abordagem utilizando *Deep Learning* de forma combinada com as técnicas existentes.

Referências Bibliográficas

[1] J. N. Yuan Rao, "A deep learning approach to detection of splicing and copy-move forgeries in images", *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.

[2] V. P. Raval, "Analysis and detection of image forgery technologies," *International Journal for Scientific Research & Development,* vol. 1, pp. 1796–1798, 2013.

[3] Nor Bakiah Abd Warif, Ainuddin Wahid Abdul Wahab, "Copy-move forgery detection: Survey, challenges and future directions", *Journal of Network and Computer Applications*, 2016.

[4] G. M. G. B. Sahar Q. Saleh, Muhammad Hussain, "Evaluation of image forgery detection using multi-scale weber local descriptors", *International Symposium on Visual Computing*, 2013.

[5] "Espaços de cores." `http://www.ic.unicamp.br/~cpg/material-didatico/mo815/9802/curso/node23.html`. Acessado em 11/06/2017.

[6] "The curse of dimensionality." `https://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/tutorials/dimensionality.pdf`. Acessado em 10/06/2017.

[7] S. G. Yijun Sun, Sinisa Todorovic, "Local learning based feature selection for high dimensional data analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2009.

[8] S. Q. Saleh, "IMAGE SPLICING AND COPY-MOVE FORGERY DETECTION," Master's thesis, King Saud University - College of Computer Information Sciences, 2012.

[9] S. . Z. H. C. S.-C. R. J. F. P. J.-S. Li, Leida ;Li, "An efficient scheme for detecting copy-move forged images by local binary patterns," *Jornal of Information Hiding and Multimedia Signal Processing*, 2013.

[10] T. M. T. Ojala, M. Pietikainen, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1610–1626, 2002.

[11] M. H. G. M.-G. B. Hatim Aboalsamh, Sahar Q. Saleh, “Comparsion between wld and lbp descriptors using non-intrusive image forgery detection,” *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, 2014.

[12] L. T. . W.-X. X. . J.-J. Huang, “Finding main road seeds based on symmetrical edge orientation histogram”, *Electronics Letters,* vol. 40, pp. 235 – 237, 2014.

[13] M. W. Meera Mary Isaac, “A key point based copy-move forgery detection using hog features”, *International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2016.

[14] S. S. H. C. Z. G.-P. M. C. X. G. W. Cheng, J., “Wld: A robust local image descriptor”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 1705–1720, 2010.

[15] A. E. M. Hussain, S. K. Wajid and M. Berbar, “A comparison of svm kernel functions for breast cancer detection”, *Eighth International Conference Computer Graphics, Imaging and Visualization,* pp. 145–150, 2011.

[16] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. 2009.

[17] X. Y. P. Chi-Man and X. Bi, “Image forgery detection using adaptive over-segmentation and feature points matching,in *IEEE Transactions on Information Forensics and Security*, 2015.

[18] A. B. A. Edoardo and G. Mazzola, “Copy-move forgery detection by matching triangles of keypoints", *in IEEE Transactions on Information Forensics and Security,* 2015.

[19] Q. G. H. Jie, H. Zhang and H. Huang, “An improved lexicographical sort algorithm of copy-move forgery detection,” *in Second International Conference on Networking and Distributed Computing, 2011.*

[20] S. Khan and A. Kulkarni, “Robust method for detection of copy-move forgery in digital images,” *in International Conference on Signal and Image Processing*, 2010.

[21] H. F. A.C. Popescu, “Exposing digital forgeries by detecting duplicated image regions”, *TR2004-515, Dartmouth College, Computer Science*, 2004.

.

[22] “Casia v1.0 forgery image dataset. http://forensics.idealtest.org/. Acessado em 11/06/2017.

[23] “Casia v2.0 forgery image dataset.” http://forensics.idealtest.org/casiav2/. Acessado em 11/06/2017.

[24] C.-J. Chang, Chih-Chung; Lin, “Libsvm: A library for support vector machines,” in *ACM Transactions on Intelligent Systems and Technology*, 2011.

[25] V. H. M. Gajanan K. Birajdar, “Digital image forgery detection using passive techniques: A survey", *Journal of Digital Investigation,* vol. 100, pp. 226–245, 2013.