

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTÔNIO SEABRA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

WILLIAN APARECIDO DE GODOY DUARTE

**ANÁLISE COMPARATIVA ENTRE OS DESCRITORES DE IMAGENS
SIFT, BRISK E SURF VOLTADOS AO RECONHECIMENTO FACIAL**

**LINS
2º SEMESTRE/2020**

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**FACULDADE DE TECNOLOGIA DE LINS PROF. ANTÔNIO SEABRA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

WILLIAN APARECIDO DE GODOY DUARTE

**ANÁLISE COMPARATIVA ENTRE OS DESCRITORES DE IMAGENS
SIFT, BRISK E SURF VOLTADOS AO RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Lins para obtenção do Título de Tecnólogo(a) em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Felipe Maciel Rodrigues

**LINS
2° SEMESTRE/2020**

D812a Duarte, Willian Aparecido de Godoy
Análise comparativa entre os descritores de imagens Sift, Brisk e Surf voltados ao reconhecimento facial / Willian Aparecido de Godoy Duarte. – Lins, 2020.
35 f.: il.

Monografia (Trabalho de Conclusão de Curso de Tecnologia em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia de Lins Professor Antônio Seabra, 2020.

Orientador: Prof. Me. Felipe Maciel Rodrigues

1.Reconhecimento facial. 2.Descritores de imagens. 3.Brisk. 4.Sift. 5.Surf. I.Rodrigues, Felipe Maciel. II.Faculdade de Tecnologia de Lins Prof. Antônio Seabra. III.Título.

CDD 004.21

WILLIAN APARECIDO DE GODOY DUARTE

**ANÁLISE COMPARATIVA ENTRE OS DESCRITORES DE IMAGENS
SIFT, BRISK E SURF VOLTADOS AO RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de Lins, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas sob orientação da Prof. Me. Felipe Maciel Rodrigues.

Data de aprovação: ____/____/____

Orientador Felipe Maciel Rodrigues

Examinador 1 (Nome do Examinador)

Examinador 2 (Nome do Examinador)

Agradecimentos

Agradeço primeiramente à Deus, que me deu o dom da vida e me abençoa todos os dias. Sou grato aos meus pais, pelo amor incondicional e exemplo de vida. Agradeço ao meu orientador Prof. Me. Felipe Maciel Rodrigues, pelo auxílio no desenvolvimento deste trabalho e por ter confiado este trabalho a mim. Agradeço ao coorientador Prof. Tiago Henrique Trojahn, pelo auxílio prestado durante o desenvolvimento deste trabalho.

Willian Aparecido De Godoy Duarte

RESUMO

É evidente a ascensão do uso de sistemas de processamento de imagens para o auxílio em diversas áreas do conhecimento. Por meio do processamento de imagens, a utilização de algoritmos capazes de descreverem o conteúdo de uma imagem - chamados descritores de imagem - possibilitam que uma aplicação seja capaz de identificar um simples objeto em uma imagem, a face de uma pessoa ou ainda auxiliar médicos no reconhecimento de tumores. Neste contexto, este trabalho visa apresentar uma comparação entre três descritores de imagens (SIFT, SURF e BRISK), a fim de analisar qual algoritmo apresenta melhor resultado quando utilizado no reconhecimento facial, juntamente com o tempo de execução e o custo de processamento. Para isto foi utilizado uma base de imagens YaleFaces que é composta por 165 imagens. Nos testes realizados o algoritmo SURF obteve o melhor acurácia em comparação aos descritores SIFT e BRISK. Já no quesito desempenho o algoritmo BRISK foi o que apresentou melhor resultado comparado com SIFT e SURF. No desenvolvimento do projeto são utilizados a linguagem de programação Python juntamente com a biblioteca de processamento de imagens OpenCV.

Palavras-chave: Reconhecimento Facial. Descritores de imagens. BRISK. SIFT. SURF.

ABSTRACT

The rise in the use of image processing systems to aid in several areas of knowledge is evident. Through image processing, the use of algorithms capable of describing the content of an image - called image descriptors - make it possible for an application to be able to identify a simple object in an image, a person's face or even assist doctors in tumor recognition. In this context, this work aims to present a comparison between three image descriptors (SIFT, SURF and BRISK), in order to analyze which algorithm presents the best result when used in facial recognition, together with the execution time and the processing cost. For this, a YaleFaces image base was used, which is composed of 165 images. In the tests performed, the SURF algorithm obtained the best accuracy compared to the descriptors SIFT and BRISK. In terms performance, the BRISK algorithm showed the best result compared to SIFT and SURF. In the development of the project, the Python programming language is used together with the OpenCV image processing library.

Keywords: Facial Recognition. Image Descriptors. BRISK. SIFT. SURF.

LISTA DE ILUSTRAÇÕES

Figura 1.1 - Etapas do processamento de imagens.	11
Figura 1.2 - Diferença Gaussiana para cada oitava.	13
Figura 1.3 - Histograma dos gradientes.	14
Figura 1.4 - Aplicação da Gaussiana para dar ênfase aos pontos vizinhos.	15
Figura 1.5 - Detecção de ponto de interesse no espaço-escala.....	16
Figura 1.6 - (a) Padrão de amostragem BRISK. (b) Pares de curta distância. (c) Pares de longa distância, cada cor representa um par.....	17
Figura 1.7 - Soma das intensidades da área retangular na imagem integral.	19
Figura 1.8 - Espaço escalar no SURF.....	20
Figura 2.1 - Processo de reconhecimento facial	21
Figura 2.2 - Exemplo de aplicação dos recursos Haar.....	23
Figura 2.3 - Implementação Haar Cascade.....	24

LISTA DE TABELAS

Tabela 1- Quantidade de correspondências e taxa de acerto dos algoritmos.....	28
Tabela 2 - Uso da CPU, Uso de memória e Tempo de execução dos algoritmos	29

LISTA DE ABREVIATURAS E SIGLAS

CPU – *Central Process Unit*

RAM – *Random Access Memory*

SIFT – *Scale Invert Feature Transform*

BRISK – *Binary Robust Invariant Scalable Keypoints*

SURF – *Speeded Up Robust Features*

SUMÁRIO

INTRODUÇÃO	8
1 CONCEITOS BÁSICOS	10
1.1 PROCESSAMENTO DIGITAL DE IMAGENS	10
1.2 DESCRITORES DE CARACTERÍSTICAS	11
1.3 SIFT.....	12
1.4 BRISK.....	15
1.5 SURF.....	18
2 METODOLOGIA.....	21
2.1 BASE DE IMAGENS	22
2.2 DETECÇÃO FACIAL.....	23
2.3 PRÉ-PROCESSAMENTO DA IMAGEM.....	24
2.4 EXTRAÇÃO DE CARACTERÍSTICAS E CLASSIFICAÇÃO	25
3 ANÁLISE E AVALIAÇÃO DOS ALGORITMOS	27
3.1 ANÁLISE DE PRECISÃO DOS ALGORITMOS	28
3.2 ANÁLISE DE CUSTO COMPUTACIONAL.....	29
4 CONCLUSÃO.....	31
REFERÊNCIAS BIBLIOGRÁFICAS	33

INTRODUÇÃO

Toda imagem possui informações que caracterizam a sua aparência e os formatos dos objetos (NIXON; AGUADO, 2012). Estas informações representam contornos, bordas, arestas, pontos e podem ser classificadas como locais (quando se baseiam em estruturas locais para descrever as informações da região de interesse), ou globais (quando descrevem a informação de toda a imagem) (AWAD; HASSABALLAH, 2016).

Atualmente computadores são capazes de identificar pessoas ou objetos em uma imagem. Isso se dá por meio de cálculos matemáticos, que permitem obter informações da imagem e através de uma interpretação (com base em algoritmos de tomada de decisões) é possível que um sistema reconheça e descreva o conteúdo de imagem. A aplicação de descritores de imagem pode ser utilizada para auxiliar na geração de imagens panorâmicas (SILVA et. al, 2012), na construção de mosaicos (TRALLO et. al, 2010) e mapas densos de pontos homólogos (SILVEIRA et. al, 2011). Entretanto, a aplicação dos descritores não se limitam a (re)construção de imagens, podendo identificar objetos em uma imagem (PAVEL; WANG; FENG, 2014), tal como placas de sinalização (SILVA et. al, 2012) e beneficiar também a área de segurança, no reconhecimento facial (GENG; JIANG, 2009), no reconhecimento biométrico por meio da íris (DELGADO, 2013) e a área da saúde, servindo de suporte a médicos no reconhecimento de tumores falsos positivos e falsos negativos (SILVA, 2017).

Tendo em vista a importância e crescente utilização desta tecnologia, este trabalho tem como objetivo desenvolver um estudo sobre os descritores de imagens SIFT, BRISK e SURF aplicados no reconhecimento facial, a fim de definir qual deles tem um melhor desempenho e eficiência para que através destes dados seja possível traçar um equilíbrio entre qualidade e custo computacional com a finalidade de auxiliar no desenvolvimento de novos sistemas que pretendem se favorecer destes algoritmos.

Este estudo comparativo se justifica pela necessidade de diminuir o custo de desenvolvimento de uma aplicação, pois por meio deste, será possível definir qual descritor de imagens atenderá melhor à necessidade desta aplicação, sem que seja preciso gastar tempo e recursos para que seja realizado testes a fim de se tomar conhecimento disso.

No desenvolvimento do projeto são utilizadas a linguagem de programação Python juntamente com o auxílio da biblioteca de processamento de imagens OpenCV, além disso para que seja realizado os testes que irão auferir qual algoritmo apresenta maior acurácia, os três algoritmos são aplicados na base de imagens YaleFaces.

A organização deste trabalho se dá em três capítulos, onde o primeiro apresenta detalhadamente os conceitos de processamento de imagens e dos descritores de imagens utilizados neste projeto. O segundo capítulo aborda os detalhes do desenvolvimento dos testes utilizando os descritores SIFT, BRISK e SURF na base de imagens definida. O terceiro apresenta os resultados obtidos. E finalizando, as conclusões sobre este trabalho.

1 CONCEITOS BÁSICOS

Este capítulo está dividido em cinco sessões, nas quais serão abordados os principais tópicos e tarefas necessárias para realização de reconhecimento de faces. Na seção 1.1 os conceitos sobre o processamento digital de imagens são apresentados. Seção 1.2 é explicado como funciona um descritor de imagem. Na Seção 1.3 são apresentados os conceitos sobre o algoritmo SIFT. Na Seção 1.4 são expostos os conceitos sobre o algoritmo BRISK. Na Seção 1.5 são apontados os conceitos sobre o algoritmo SURF.

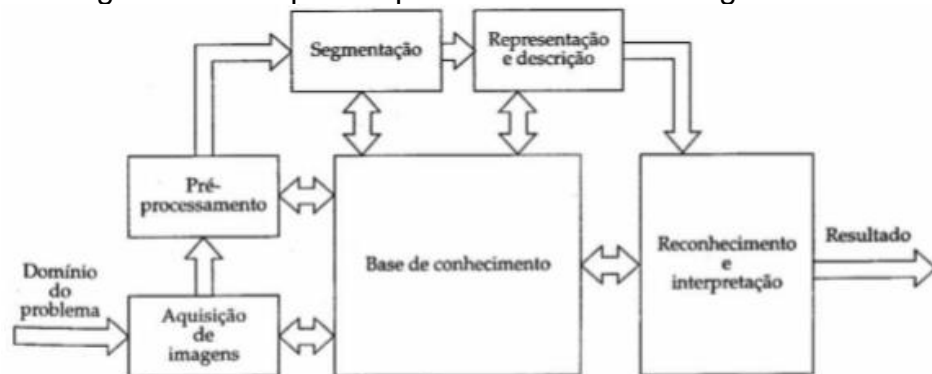
1.1 PROCESSAMENTO DIGITAL DE IMAGENS

Segundo Gonzalez e Woods (2010), o processamento digital de imagens não é uma tarefa simples, seu processo compreende um conjunto de tarefas interconectadas. O processo se inicia com a captura de uma imagem digital, que precisa ser representada de forma apropriada para o tratamento computacional e essa representação pode ser realizada em duas ou mais dimensões. O próximo passo é o pré-processamento, que tem por função melhorar a imagem através da filtragem de ruídos inseridos por meio dos sensores e a correção de distorções geométricas causada pelos sensores.

Em seguida, a etapa de segmentação tem como objetivo além de extrair as características/atributos das imagens, separar fundo e os objetos presentes na cena. A separação entre fundo e objetos se faz importante, pois esta etapa define se a identificação deve ser realizada por meio de um processo de segmentação de características constantes ou descontinuas. O resultado da segmentação é composto por dados em formas de *pixels*, sendo necessária a conversão dos dados a um formato adequado ao processamento computacional. No processo de descrição ocorre a extração das características que possam representar alguma informação relevante para a discriminação entre as classes.

Por fim, ocorre a etapa de reconhecimento e interpretação que, com base nas informações cedidas pelo descritor o objeto recebe uma descrição e posteriormente o processo de interpretação concede um significado ao conjunto de objetos reconhecidos.

Figura 1.1 - Etapas do processamento de imagens.



Fonte: GONZALEZ; WOODS, 2010.

1.2 DESCRITORES DE CARACTERÍSTICAS

Para procurar imagens semelhantes é comum compará-las diretamente. Isso se dá por meio da equivalência dos valores de um *pixel* ou conjunto de *pixels* de uma imagem com os valores de outra imagem (DESELAERS, 2003). Quando analisado esse conjunto de *pixels* pode apresentar as características da imagem, que são propriedades das quais os valores têm de ser parecidos com os objetos em uma classe particular e diferentes para os objetos em outra, inclusive em relação ao fundo da imagem (DOUGHERTY, 2013).

O computador necessita processar os dados de modo a identificar o que está presente na imagem, para isso o primeiro passo é realizar, de maneira efetiva e eficiente, a detecção e extração de características (TIAN, 2013). Para escolher as características corretas da imagem é necessário o domínio do problema ou da imagem, contudo, usualmente são buscadas características invariantes a transformações que possam ocorrer, como translação, mudança no tamanho e rotação (NIXON e AGUADO, 2002).

Essas características podem ser globais ou locais. As globais podem generalizar um objeto inteiro ao utilizar apenas um vetor, além de que os algoritmos descritores globais tendem a ser mais rápidos, mais simples e com boa tolerância a ruídos. Cor, textura e forma podem ser algumas das características globais extraídas, entretanto, por localizar poucos detalhes, essas características não são eficazes quando ocorre alguma alteração na perspectiva da imagem, como por exemplo, alteração em sua rotação ou translação, além de serem sensíveis a oclusão, desordens ou deformação no objeto.

As características locais são descritas como um padrão dentro da imagem que é diferente de sua vizinhança, além de ser associada a mudança das propriedades da mesma e pode ser caracterizada por pontos de interesse, bordas ou trechos da imagem. (TUYTELAARS e MIKOLAJCZYK, 2008).

Os pontos de interesse são normalmente encontrados na imagem na qual ocorre a mudança do sinal em duas dimensões, onde há a intersecção de duas bordas ou onde a borda muda repentinamente. Esses pontos têm uma posição bem definida dentro de uma imagem, onde sua vizinhança é rica em informações, fazendo com que seja considerado um ponto único que não se repete e é diferente dos outros.

1.3 SIFT

Scale Invert Feature Transform ou Descritor de Transformação de Características Invariantes à Escala é um algoritmo proposto por Lowe, e é instruído a extrair características locais de uma imagem que não sofram variações na alteração de escala, rotação, iluminação e no ponto de vista 3D. A identificação dos pontos de interesse invariantes a escala e rotação, ocorrem através das funções gaussianas. Cada ponto de interesse recebe um vetor de características capaz de descrever de forma única regiões de interesse referindo-se ao local de escala e espaço de coordenadas.

O SIFT é formado por quatro etapas, sendo elas: detecção de extremos no espaço de escalas, localização do ponto de interesse, atribuição de orientação e descritor do ponto de interesse.

Durante a etapa de detecção de extremos, ocorre a identificação dos pontos de interesse. Além disso, os possíveis *pixels* de interesse também são localizados (AWAD; HASSABALLAH, 2016). Nesta etapa são selecionados os pontos máximos e mínimos (pirâmide convoluída) com a diferença gaussiana (DoG) aplicada no espaço escalar, para que se obtenha a invariância e o alto nível de eficiência. A função gaussiana 2D é separável, com isso a convolução com a imagem de entrada é computada através da aplicação de dois passos da função gaussiana 1D - uma na horizontal e outra na vertical (LOWE, 1999).

A partir de uma imagem de entrada $I(x,y)$, novas versões da imagem $F(x,y,\sigma)$, são geradas por meio da convolução de um filtro gaussiano $G(x,y,\sigma)$ como mostra a Equação 1:

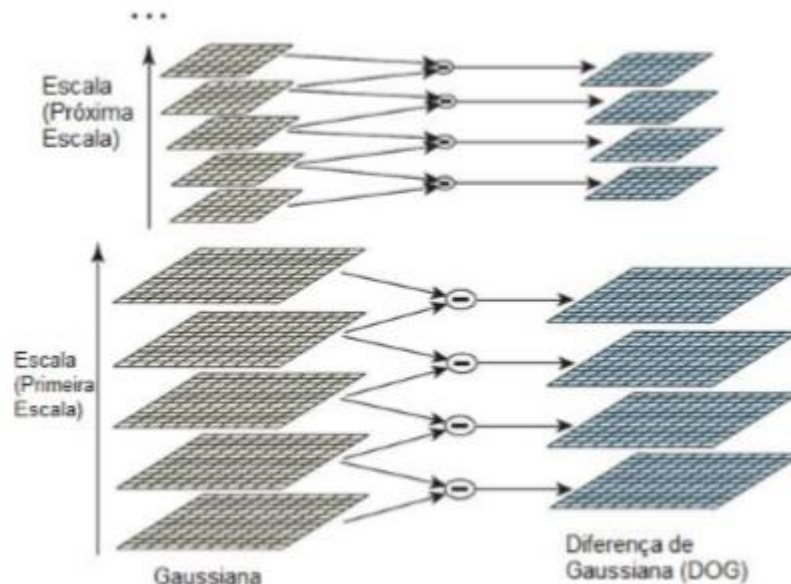
$$F(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

onde

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

Para que as principais regiões sejam localizadas, é realizada a operação de suavização utilizando $\sigma = \sqrt{2}$ e com isso é aplicado a convolução na imagem, resultando uma imagem A. A fim de melhorar a suavização, o processo é repetido utilizando $\sigma = 2$ acarretando uma imagem B.

Figura 1.2 - Diferença Gaussiana para cada oitava.



Fonte: Lowe, 2004.

A pirâmide escalar é gerada por meio da diferença entre as duas funções gaussianas (LOWE, 1999), ou seja, resultado da subtração entre imagem B e imagem A (coeficiente $2/\sqrt{2} = \sqrt{2}$). A imagem B que foi suavizada é redimensionada através de uma interpolação bilinear e espaçamento de *pixel* igual 1.5 em cada direção, este valor indica que cada amostragem é a combinação linear de quatro *pixels* adjacentes (LOWE, 1999). Os níveis da suavização são denominados escala e os níveis do redimensionamento são denominados oitavo (*octave*).

Por intermédio da comparação de cada *pixel* na pirâmide com seu vizinho, são determinados os valores máximos e mínimos da diferença gaussiana no espaço

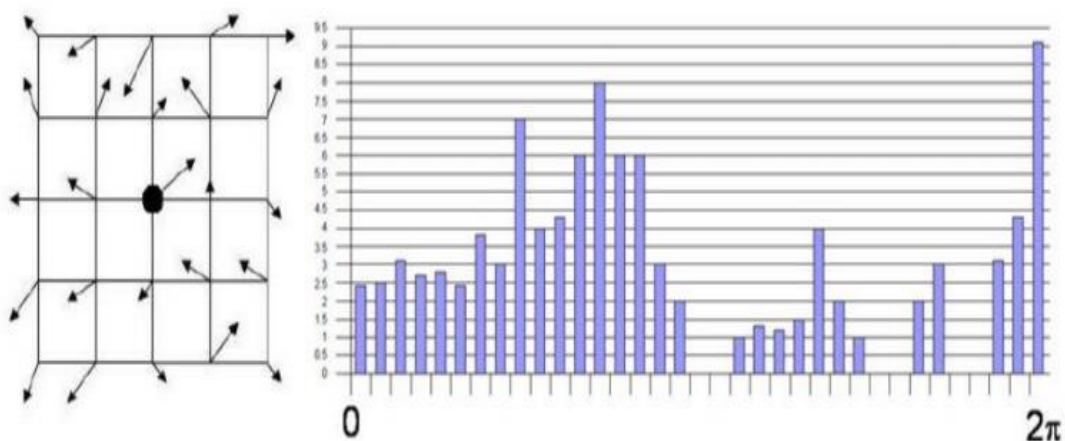
escalar. Neste processo, é realizada a comparação do *pixel* com seus oito vizinhos que se encontram no mesmo nível da pirâmide. Caso o *pixel* permaneça sendo o valor mais alto ou mais baixo que seus vizinhos, o teste é repetido para um nível mais próximo e o *pixel* é comparado com nove vizinhos. Caso este *pixel* ainda possua valor mais alto ou mais baixo que seus vizinhos, este pixel é considerado como um candidato a ser um ponto de interesse (DAS, 2015).

Levando em conta que todos os pontos detectados como extremos são candidatos a pontos de interesse, a próxima etapa é calcular a localização exata destes pontos. Para isso, Lowe (2004) sugere que seja utilizado o cálculo de localização interpolada por meio da expansão quadrática da série de Taylor sobre a função da Diferença Gaussiana. Em seguida, os pontos são filtrados através de um limiar de contraste e é realizado o cálculo de expansão da série de Taylor de segunda ordem da DoG, a fim de descartar os pontos com valores inferiores ao limiar.

A fim de obter a invariância a rotação, é realizada a atribuição de orientação aos pontos chaves. Pensando nisso, Lowe (2004) propõe o cálculo de orientações e magnitudes do gradiente para cada ponto chave, tendo como base as diferenças entre os *pixels* das imagens exibidas nos intervalos do espaço-escala. As orientações entre 0 e 2π são representadas através de um histograma e os pontos vizinhos do ponto chave são adicionados ao histograma.

O pico do histograma é considerado para definição da orientação do ponto chave. Entretanto, são considerados apenas os picos que possuem pelo menos 80% do seu valor, assim o ponto chave pode possuir mais de uma orientação (Lowe, 2004).

Figura 1.3 - Histograma dos gradientes.

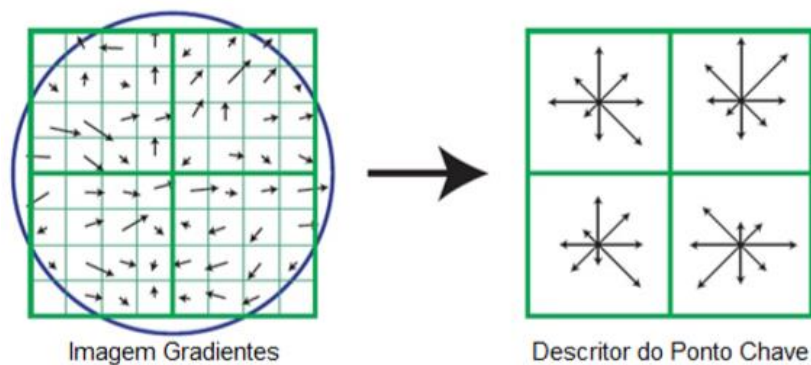


Fonte: Adaptado de Gonzáles & Meggiolaro, 2010.

O descritor do ponto de interesse é construído tendo como base uma região de 4 pixels ao redor do ponto chave dividida em sub-regiões 4x4. Cada sub-região recebe um histograma levando em conta 8 orientações (2 horizontais, 2 verticais e 4 diagonais), resultando na criação de um vetor de tamanho 128 para descrever o ponto chave, contado a começar das magnitudes e orientações dos gradientes (Lowe, 2004).

Por fim, é aplicada uma função Gaussiana com o objetivo de ponderar a magnitude do gradiente em cada ponto na vizinhança do ponto chave. Essa função Gaussiana é utilizada para conter mudanças inesperadas na posição e diminuir a importância dada a gradientes que estão longe do ponto chave, dando ênfase aos pontos vizinhos (González e Meggiolaro, 2010).

Figura 1.4 - Aplicação da Gaussiana para dar ênfase aos pontos vizinhos.



Fonte: Lowe, 2004.

1.4 BRISK

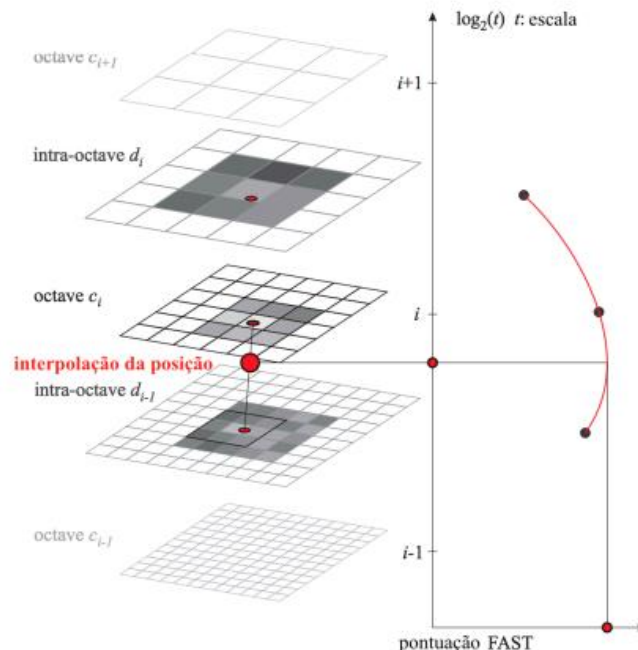
O algoritmo *Binary Robust Invariant Scalable Keypoints* é construído com base em 60 pontos distribuídos em quatro anéis homocêntricos, dado que todos os pares pontos de amostragem são simétricos e circulares. Todo ponto retrata uma área de amostragem circular que aumenta conforme aumenta sua distância em relação ao ponto de interesse (MUHAMMAD, 2015).

Para alcançar a invariância escalar, o máximo local é localizado no plano da imagem e como medição de saliência aplica a pontuação FAST no espaço escalar. Além disso, o BRISK faz uma aproximação do valor dos pontos de interesse no espaço-escalar (LEUTENEGGER; CHLI; SIEGWART, 2011).

A pirâmide escalar no algoritmo BRISK é tipicamente composta de 4 oitavas e 4 intra-oitavas. A formação das oitavas é realizada por meio de sub-amostragens que corresponde à metade da imagem original. A primeira oitava é determinada pela imagem original e as oitavas restantes são as sub-amostras sucessivas da oitava anterior. Já a primeira intra-oitava é obtida através da sub-amostragem por um fator de 1.5 e as sucessivas são sub-amostragens da intra-oitava anterior. Com isso, cada intra-oitava fica localizada entre as camadas de oitavas (LEUTENEGGER; CHLI; SIEGWART, 2011).

O ponto de interesse é aquele que possui saliência máxima, ele é identificado na oitava por meio da análise da saliência de seus oito vizinhos na oitava atual, na oitava anterior e na oitava sucessora. Todas as oitavas de interesse têm a máxima saliência local refinada para três sub-*pixels* e estes são distribuídos em uma parábola 1D, a fim de definir a escala verdadeira do ponto de interesse.

Figura 1.5 - Detecção de ponto de interesse no espaço-escala.



Fonte: LEUTENEGGER; CHLI; SIEGWART, 2011.

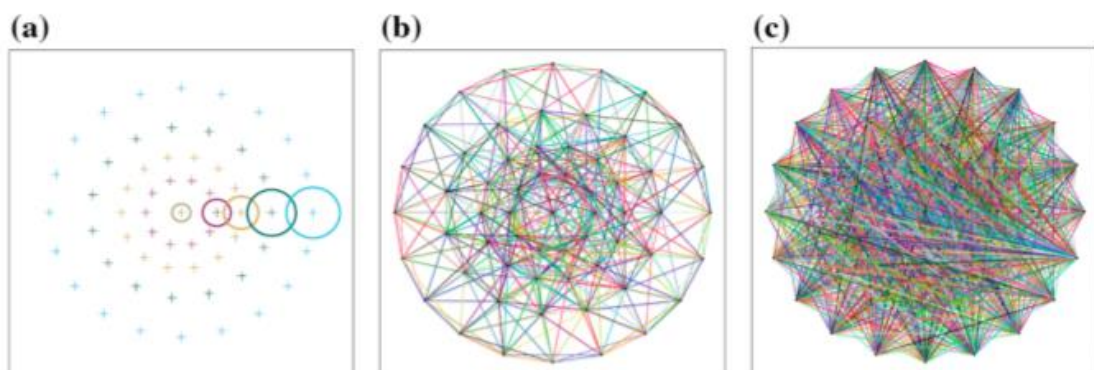
O BRISK usa o detector de característica FAST para detectar os pontos de interesse através das dimensões. Habitualmente é aplicada uma máscara FAST de 9-16, o qual o *pixel* de interesse é identificado quando se testa nove *pixels* consecutivos dentro de uma janela circular de 16 *pixels*. Após isso, os pontos que pertencem a

estas regiões são sujeitos a uma supressão não máxima, onde a pontuação FAST é estabelecida como limite máximo e o ponto em foco tem que satisfazer a condição máxima em relação aos seus oito vizinhos na oitava atual, anterior e sucessora (LEUTENEGGER; CHLI; SIEGWART, 2011).

Através da supressão não máxima são obtidas três pontuações, da oitava atual, da anterior e da posterior. Com isso, para cada oitava é determinado uma função quadrática 2D com o objetivo de ajustar a região 3x3 ao redor dos *pixels* e sub-*pixels* máximos. Então, a localização do ponto de interesse na escala é obtida com a interpolação dos valores máximos utilizando uma função quadrática 1D por meio do espaço escalar (FAN; WANG; WU, 2015).

O BRISK faz uso de um padrão que estipula locais uniformemente espaçados em círculos concêntricos com o ponto de interesse. Este padrão possui 60 pontos de amostragem e aplica suavização Gaussiana a fim de evitar o efeito de *aliasing*¹, o que define o desvio padrão da gaussiana de cada *pixel* de amostragem é a sua distância em relação ao ponto de interesse. Considerando os 60 pontos de amostragem ($N = 60$) temos, então, 1770 combinações de pares possíveis ($\frac{N(N-1)}{2} = 1770$). Por meio do conjunto de todos os pares de amostragem pode ser obtido dois subconjuntos, pares de distância curta C e pares de longa distância L e estes pares são representados matematicamente pelas formulas: $C = [(p_i, p_j) | |p_i - p_j| < \delta_{max}]$ $L = [(p_i, p_j) | |p_i - p_j| < \delta_{min}]$. Sendo $\delta_{max} = 9.78\delta$, $\delta_{min} = 13.67\delta$ e δ a escala do ponto de interesse.

Figura1.6 - (a) Padrão de amostragem BRISK. (b) Pares de curta distância. (c) Pares de longa distância, cada cor representa um par.



Fonte: FAN; WANG; WU, 2015.

¹ *Aliasing* é o aparecimento de um recorte na imagem, que é visualmente semelhante a um serrilhado.

A orientação do ponto de interesse se dá através de L , haja vista que a tendência é que os gradientes locais se anulem uns aos outros não sendo necessários para calcular o gradiente local. Com isso, estima-se a direção do ponto de interesse k como: $(g_x, g_y) = \frac{1}{n} \sum_{(p_i, p_j) \in L} g(p_i, p_j)$. Então, a orientação do ponto de interesse é definida através do padrão de amostragem rotacional $\alpha = \arctan2(g_y, g_x)$ ao redor do ponto de interesse k . Com base na escala e na orientação do ponto de interesse, um descritor binário é determinado por meio das comparações de intensidade mediante a rotação e escala dos pares de curta distância (LEUTENEGGER; CHLI; SIEGWART, 2011).

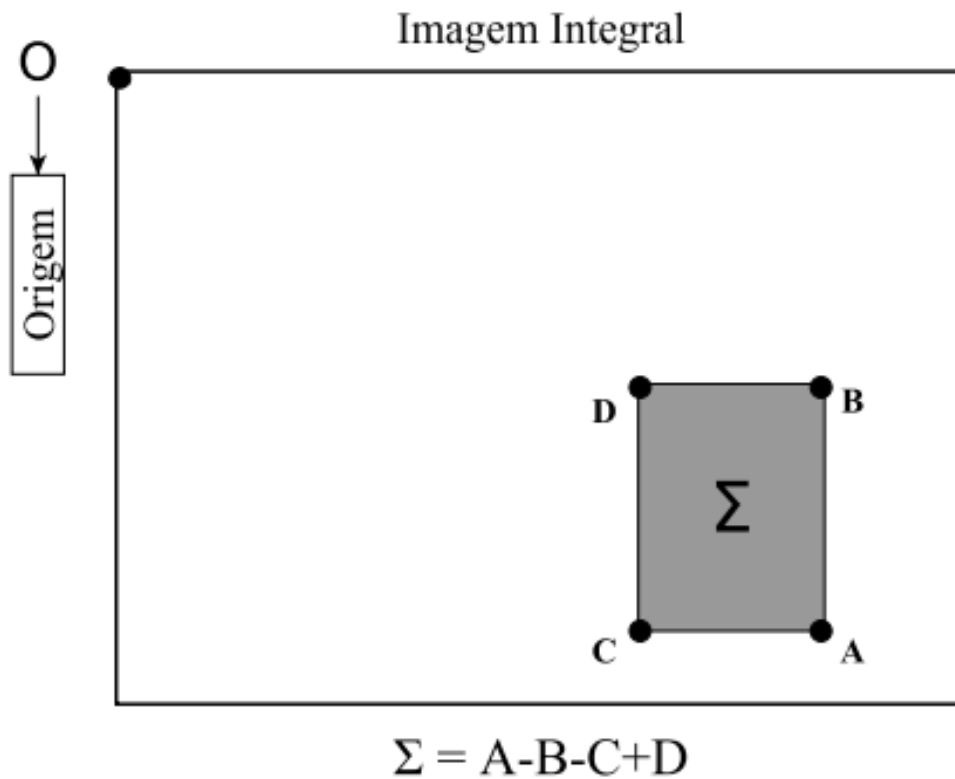
1.5 SURF

Proposto por Bay, Tuytelaars e Gool (2006), *Speeded Up Robust Features* (SURF) é um algoritmo rápido e poderoso para extrair e descrever pontos de interesse em imagens. Se tratando da detecção de objetos específicos na imagem usando o método SURF, não há necessidade de pesquisar por todo o objeto na imagem, uma vez que apenas pontos de interesse relacionados ao objeto são usados para a sua identificação. Dentre as razões para se utilizar este algoritmo, destacam-se a redução do custo computacional através da redução da dimensionalidade a ser examinada e a diminuição da redundância incorporada, dado que os *pixels* não se movem de forma independente e sejam altamente correlacionados (MONTANARI, 2016).

O processo de detecção dos pontos de interesse no algoritmo SURF faz uso de uma aproximação simples da matriz Hessiana, juntamente com o uso de imagens integrais. Quando imagens integrais são utilizadas, um campo qualquer de uma imagem $I_{\mathcal{X}} = (x')$, encontrada em $x' = (x, y)$ representa a soma de todos os *pixels* da imagem de entrada em uma área retangular formada a partir das coordenadas da origem da imagem inteira até x' (BAY et al., 2008).

Devido a utilização de um filtro relativamente extenso, a partir do momento em que a imagem integral é calculada, apenas três adições são necessárias para determinar a soma da intensidade de qualquer área retangular, otimizando assim o processo de detecção (NOVAIS, 2016). A Figura 1.7 ilustra o método de cálculo de uma imagem integral e como são calculadas as sub-regiões.

Figura 1.8 - Soma das intensidades da área retangular na imagem integral.



Fonte: Adaptada de Novais, 2016.

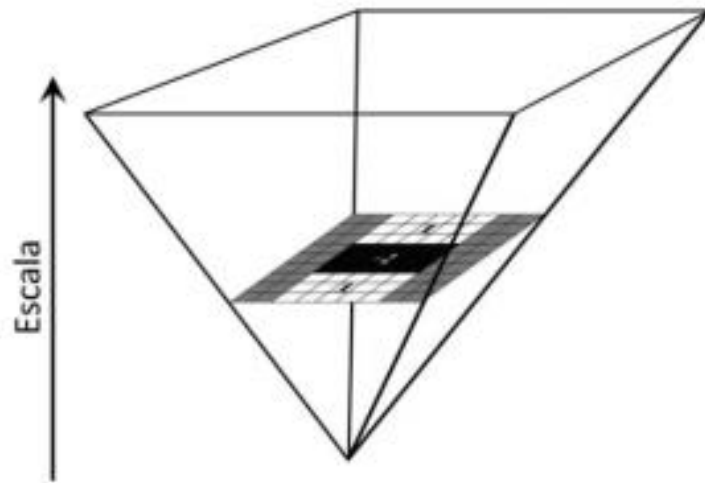
Assim sendo, por conta do uso de imagens integrais, o cálculo é bastante reduzido por ser possível calcular rapidamente o filtro de convolução (BAY et al., 2008). Segundo Novais (2016), determinado um ponto $x' = (x, y)$ em uma imagem I , a matriz Hessiana $H(x', \sigma)$ no ponto x' em uma escala σ é estipulado pela equação:

$$H(x', \sigma) = \begin{vmatrix} Lxx(x', \sigma) & Lxy(x', \sigma) \\ Lxy(x', \sigma) & Lyy(x', \sigma) \end{vmatrix}$$

no qual $Lxx(x', \sigma)$ simboliza a convolução da derivada de segunda ordem da função gaussiana $\frac{\partial^2}{\partial x^2} g(\sigma)$, levando em conta a imagem I no ponto x' . Da mesma forma acontece com $Lxy(x', \sigma)$ e $Lyy(x', \sigma)$ (BAY et al., 2008).

Assim como no descritor SIFT, o espaço escalar do SURF é implementado como uma pirâmide de imagens. A diferença entre o espaço escalar de ambos se encontra na construção da pirâmide escalar, no SIFT são utilizadas imagens com diferentes dimensões, enquanto no SURF são empregadas máscaras gaussianas de diferentes escalas e o tamanho das imagens permanecem inalterados (BAY et al., 2008).

Figura 1.9 - Espaço escalar no SURF



Fonte: Adaptado de Bay et al., 2008.

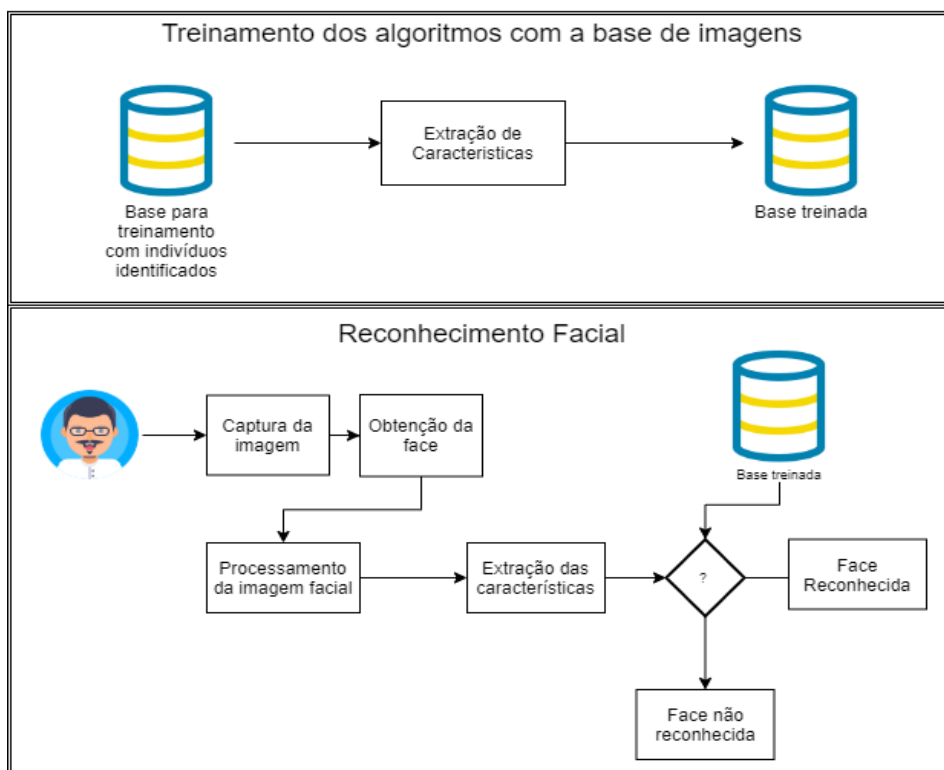
A primeira oitava do espaço escalar é definida com um filtro de tamanho 9×9 e os filtros maiores são gerados com um incremento de 6 resultando filtros 15×15 , 21×21 , 27×27 e assim sucessivamente (KLETTE, 2014).

Com o intuito de detectar os pontos de interesse é executado uma supressão não máxima, ou seja, ocorre a invalidação dos *pixels* onde o valor na direção do gradiente da imagem não é um máximo local (AWAD; HASSABALLAH, 2016). Isso quer dizer que ocorre a separação de duas máximas locais por r pixels em cada nível, em que r é o raio da vizinhança aplicado na supressão não máxima. A fim de estabilizar a localização das amostras é aplicada uma interpolação de maneira similar ao SIFT e com isso é obtido a máxima dos *pixels* por níveis e é utilizada com ponto de interesse (FAN; WANG; WU, 2015).

2 METODOLOGIA

Neste capítulo é apresentada a metodologia aplicada para aferir a acurácia e o desempenho dos algoritmos SIFT, BRISK E SURF utilizados para o reconhecimento facial. Para realizar os experimentos foi utilizado uma máquina com o processador AMD Ryzen 5 e 16 gigabytes de memória RAM, juntamente com o sistema operacional Windows 10 e as versões 3.7.7 do Python e 3.4.2 da biblioteca OpenCV, além de contar com auxílio da base de imagens Yale Faces². A Figura 2.1 ilustra as 2 etapas que compõem o processo, sendo elas, o treinamento dos algoritmos com base nas imagens e o reconhecimento facial.

Figura 2.1 - Processo de reconhecimento facial



Fonte: Elaborada pelo autor, 2020.

Na primeira etapa (treinamento dos algoritmos com a base de imagens) se faz necessário para que o algoritmo passe a conhecer o indivíduo representado em determinada imagem. De modo geral o algoritmo lê a imagem, realiza um pré-

² Yale faces database disponível em: <http://vision.ucsd.edu/content/yale-face-database>

processamento - a imagem é convertida em uma imagem monocromática - extrai as características com base no algoritmo utilizado e armazenadas tais características como pertencendo a um determinado indivíduo. Por fim, é gerado um arquivo com os dados de treinamento que são utilizados posteriormente para encontrar as similaridades entre as imagens.

Já na etapa do reconhecimento facial o algoritmo recebe como entrada uma imagem, efetua a detecção da face, realiza um pré-processamento - a imagem é convertida em monocromática e redimensionada - extrai as características e compara tais características da imagem de entrada com as características das imagens da base de treinamento. Assim é possível classificar se a face da imagem de entrada é similar a alguma imagem da base de treinamento. Os algoritmos utilizados e os detalhes das atividades são apresentados ao longo desse capítulo.

2.1 BASE DE IMAGENS

Uma base de imagens é necessária no processo de reconhecimento facial para que seja realizado o treinamento dos algoritmos que são responsáveis por efetuar a comparação entre as faces. Com isso é fundamental que a base de treinamento contenha imagens faciais dos indivíduos que se deseja identificar. Além disso, é desejável que tais imagens contenham apenas a face, eliminado o fundo e outros detalhes não essenciais para o reconhecimento facial.

O treinamento dos algoritmos exige que a base possua ao menos uma imagem de um indivíduo e que as imagens da base sejam previamente identificadas para que durante o treinamento o algoritmo possa etiquetá-las com uma identificação. A criação de uma base de treinamento, se faz necessária pois ela é utilizada pelo algoritmo para realizar a comparação com as imagens de teste, onde é confrontado as características e ocorre a extração das semelhanças entre as imagens. Outro ponto importante é que as imagens utilizadas nos testes devem ser diferentes das contidas na base de treinamento para que não ocorra a interseção entre as imagens ao realizar análise de performance dos algoritmos.

Neste trabalho foi utilizada uma base de imagens conhecida na literatura, a Yale Database. Esta que contém 165 imagens em escala de cinza de 15 pessoas diferentes, que foram obtidas em ângulos distintos, expressões faciais e graus de

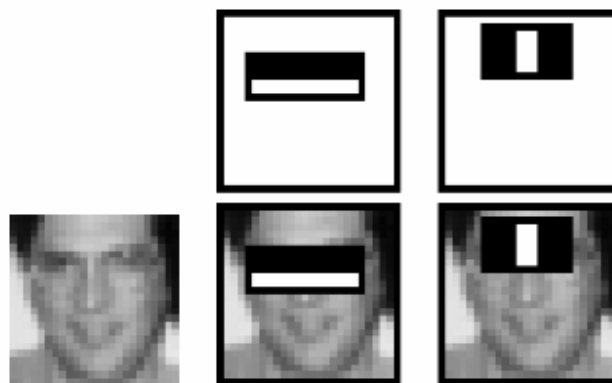
luminosidade diferentes. Todas as imagens estão padronizadas no tamanho de 320 x 243 pixels com fundo em tom de branco.

2.2 DETECÇÃO FACIAL

A detecção da face é fundamental para o reconhecimento facial e consiste em encontrar uma ou mais faces em uma imagem que contenha pessoas e objetos em um mesmo cenário.

O algoritmo utilizado é o *Haar Cascade* (já incluso na biblioteca OpenCV). Proposto por Viola e Jones em 2001, é um algoritmo que utiliza como principal função a *detecMultiScale*, que tem como objetivo detectar objetos e faces nas imagens. O algoritmo analisa a imagem em busca de padrões que caracterizam um objeto e, ao se tratar de uma face o padrão utilizado são as características desse objeto como por exemplo, a região dos olhos. As regiões devem apresentar características que as diferem umas das outras. Um exemplo são os olhos que normalmente se encontram em uma região que é mais escura que a região das bochechas. Essas características são conhecidas como *haar*.

Figura 2.10 - Exemplo de aplicação dos recursos Haar.



Fonte: Adaptado de OpenCV, 2018.

A implementação do Haar Cascade foi realizada na linguagem Python, com o auxílio da biblioteca OpenCV utilizando um modelo já treinado para detectar faces. O código parcial utilizado é apresentado na Figura 2.3 e as linhas em destaque exibem a utilização deste algoritmo. A primeira linha destacada é onde está sendo realizada a execução do modelo treinado para detectar faces e a segunda em destaque mostra

a aplicação da função *detectMultiScale* em cada imagem da base a fim de encontrar faces nestas imagens.

Figura 2.11 - Implementação Haar Cascade

```

1
2 import cv2
3 import os
4 import numpy as np
5 from PIL import Image
6
7 detectorFace = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
8 reconhecedor = cv2.face.LBPHFaceRecognizer_create()
9 reconhecedor.read("classificadorLBPHVale.yml")
10
11 totalAcertos = 0
12 percentualAcerto = 0.0
13 totalConfianca = 0.0
14
15 caminhos = [os.path.join('yalefaces/teste', f) for f in os.listdir('yalefaces/teste')]
16 for caminhoImagem in caminhos:
17     imagemFace = Image.open(caminhoImagem).convert('L')
18     imagemFaceNP = np.array(imagemFace, 'uint8')
19     facesDetectadas = detectorFace.detectMultiScale(imagemFaceNP)
20
21
22     for (x, y, l, a) in facesDetectadas:
23
24
25         idprevisto, confianca = reconhecedor.predict(imagemFaceNP)
26         idatual = int(os.path.split(caminhoImagem)[1].split(".")[0].replace("subject", ""))
27         print(str(idatual) + " foi classificado como " + str(idprevisto) + " - " + str(confianca))
28         if idprevisto == idatual:

```

Fonte: Elaborada pelo autor, 2020.

2.3 PRÉ-PROCESSAMENTO DA IMAGEM

O objetivo do pré-processamento da imagem é a normalização através da redução das variações das imagens obtidas na detecção facial. Nessa etapa é definido o padrão das imagens que serão inseridas na base de treinamento e nas imagens que serão utilizadas para teste. As etapas do pré-processamento envolvem a conversão da imagem em escala de cinza e o redimensionamento para que toda as imagens estejam em um mesmo padrão.

A conversão para a escala de cinza é realizada com o auxílio da função *Image.convert(mode)* que está presente na biblioteca PIL. O parâmetro utilizado é o *mode = "L"*, que é responsável por converter os valores RGB para uma escala de cinza. Essa conversão ocorre de acordo com a equação $L = R * 299/1000 + G * 587/1000 + B * 114/1000$

O redimensionamento da imagem é realizado através da função *resize*, também presente na biblioteca OpenCV. Essa função é responsável por redimensionar a imagem para um tamanho X_i e Y_i especificados (OpenCV, 2000).

2.4 EXTRAÇÃO DE CARACTERÍSTICAS E CLASSIFICAÇÃO

Nesta etapa, o algoritmo analisa a imagem em busca de pontos-chaves que, uma vez encontrados, o algoritmo utiliza-os para o seu treinamento a fim de que seja possível efetuar a classificação da imagem. Através de funções de similaridade o algoritmo é capaz de avaliar as diferenças da face encontrada na imagem de teste com as faces presentes nas imagens da base de treinamento. Assim é possível comparar as diferenças (distâncias) e avaliar se a face da imagem de teste está contida na base de treinamento.

Os algoritmos utilizados possuem parâmetros que influenciam diretamente nas estratégias de extração das características e classificação. Neste projeto, utiliza-se todos os parâmetros dos descritores com seus valores padrões.

No algoritmo SIFT, foram utilizados os parâmetros *nfeatures* (0), *nOctaveLayers* (3), *contrastThreshold* (0.04), *edgeThreshold* (10) e *sigma* (1.6).

O parâmetro *nfeatures* representa o número de melhores características, estas que por sua vez são mensuradas como contraste local. O *nOctaveLayers* é responsável por definir o número de camadas em cada oitava, este número é calculado automaticamente por meio da dimensão da imagem.

O parâmetro *contrastThreshold* é o limiar de contraste que é aplicado a fim de filtrar as características fracas em regiões de baixo contraste e quanto maior este limiar, menor é o número de características produzidas pelo detector.

Utilizado para filtrar os recursos de borda, o parâmetro *edgeThreshold* tem como padrão o valor 10 e quanto maior este limiar menos características são filtradas. Por último, o *sigma* representa Sigma da Gaussiana e é aplicado na imagem de entrada na oitava #0.

Os parâmetros utilizados no algoritmo SURF são *hessianThreshold* (100), *nOctaves* (4), *nOctavesLayers* (3), *extended* (false – 64 elementos), e *upright* (false).

O parâmetro *hessianThreshold* representa o limiar aplicado na detecção dos pontos-chaves e somente as características em que o valor da Hessiana é maior que

este limiar serão retidas, sendo assim, quanto maior for este valor menos pontos-chaves serão obtidas.

Representando a quantidade de oitavas de pirâmide que o detector de ponto chave irá utilizar, o *nOctaves* tem por padrão o valor 4. Já o *nOctavesLayers* define a quantidade de camadas dentro de cada oitava.

O parâmetro *extended* é o indicador de descritor estendido, quando *true* ele utiliza descritores de 128 elementos estendidos e quando seu valor é *false* indica a utilização de descritores de 64 elementos.

Por último o *upright* que é o indicador de características para cima a direita ou para a rotação, quando seu valor é *true* a orientação das características não é calculada e quando o valor é *false* a orientação é calculada.

O algoritmo BRISK possui três parâmetros: *thresh* (30), *octaves* (3), *patternScale* (1.0), o qual representam respectivamente o limiar aplicado na detecção dos pontos chaves, as oitavas de detecção do espaço escalar e o modelo utilizado para amostrar a vizinhança de um ponto chave.

3 ANÁLISE E AVALIAÇÃO DOS ALGORITMOS

Neste capítulo são apresentados os procedimentos utilizados para a condução dos experimentos, utilizando os processos e algoritmos citados anteriormente. A análise dos experimentos tem por objetivo expor o comportamento dos algoritmos quando aplicados na base de imagens, a fim de justificar os resultados obtidos. Este processo é dividido em 3 etapas – Detecção facial, pré-processamento da imagem e extração das características e classificação – conforme abordado no Capítulo 2 Metodologia.

Inicialmente é realizado o treinamento dos algoritmos utilizando a base de imagens de treino disponibilizada. A base de imagens é dividida em dois grupos: imagens de treino e imagens de teste.

O grupo imagens de treino contém 135 imagens compostas por 15 indivíduos em 9 imagens distintas (diferentes expressões faciais iluminações cada um. Já o grupo imagens de teste contém é composta pelos mesmos 15 indivíduos contidos no grupo imagens de treino, porém possui apenas 2 variações de imagens para cada indivíduo, totalizando 30 imagens.

O processo de extração e descrição das imagens foi dividido em três componentes principais. O primeiro e o segundo contemplam a utilização dos algoritmos a fim de obter os pontos chave e o descritor das imagens de treino e das imagens de teste, respectivamente.

Visto que o presente trabalho pretende fornecer informações sobre a acurácia dos algoritmos, faz-se necessário a comparação das imagens a fim de encontrar quais possuem as mesmas características. Para isso, foi escolhido o uso do como método força bruta, *BFMatcher*, disponível na biblioteca OpenCV. Este método faz a comparação dos dados pertencentes ao descritor da imagem de referência com os dados do descritor da imagem que será analisada. Com isso é possível encontrar quais características são iguais/semelhantes nas duas imagens (imagem de treino e imagem de teste), determinando se as imagens são ou não correspondentes.

Com as imagens classificadas e detectadas, é possível realizar o reconhecimento facial e mensurar os resultados de cada algoritmo. Os testes efetuados nesta pesquisa obtiveram dados relacionados a precisão dos algoritmos, tempo de execução, porcentagem de utilização da CPU e uso de memória RAM.

3.1 ANÁLISE DE DESEMPENHO DOS ALGORITMOS

O teste de precisão ocorreu de forma que, ao comparar as imagens de treino com as imagens de teste, fossem obtidos a quantidade de correspondências identificadas, a quantidade de correspondências corretas e a taxa de acerto de cada algoritmo.

Tabela 3.1 - Quantidade de correspondências e taxa de acerto dos algoritmos.

	SIFT	SURF	BRISK
Taxa de Acerto	22/30 (76,67%)	25/30 (83,33%)	19*30 (63,33%)
Total de Correspondências	515944	655961	173452
Total de Correspondências Boas	38246	42349	2378

Fonte: Elaborado pelo autor, 2020.

Ao analisar a Tabela 3.1 é possível perceber a diferença entre a quantidade de correspondências obtidas em cada algoritmo e a quantidade de correspondências corretas, isso reflete na taxa de acerto justificando o melhor desempenho do algoritmo SURF, pois aqueles que obtiveram mais correspondências corretas conseguiram obter maior percentual de acerto ao identificar as quais imagens da base são correspondentes entre si.

Apesar do algoritmo SIFT também apresentar muitas correspondências, a quantidade de correspondências corretas foi menor do que as obtidas pelo SURF, justificando sua acurácia inferior ao SURF. Além disso, ao olhar para o BRISK fica claro que a quantidade de correspondências incorretas e corretas obtidas foi menor em relação aos outros, provando sua baixa taxa de acerto para a base utilizada.

3.2 ANÁLISE DE CUSTO COMPUTACIONAL

As análises relacionadas ao custo computacional referente a execução dos algoritmos estudados nesse projeto se restringem a obtenção de memória consumida, CPU consumida e tempo gasto durante a execução dos algoritmos.

Os dados obtidos nesta análise correspondem a execução dos algoritmos em um computador cujo *hardware* se resume em um processador Ryzen 5 3600 com velocidade máxima de 4.10GHz e 16Gb de memória RAM de 2300MHz. A Tabela 3.2 apresenta o consumo da CPU, consumo de memória e o tempo de execução.

Tabela 3.2 – Custo computacional

	SIFT	SURF	BRISK
Uso da CPU	88,80%	88,10%	75,50%
Uso de memória (MB)	71,40	79,07	87,95
Tempo de Execução (s)	8.81	5.49	3.67

Fonte: Elaborado pelo autor, 2020.

Durante a extração dos pontos chave e descrição das características, cada algoritmo aplica uma técnica diferente e isso interfere na eficiência e quantidade dos pontos detectados influenciando no custo computacional.

Quanto ao uso da CPU, os algoritmos SIFT e SURF apresentam um valor bem próximo, 88,80% e 88,10% respectivamente. Em relação ao uso de memória, a diferença de 7,67 MB para mais do SURF (79,07 MB) em comparação ao SIFT (71,40 MB) se justifica pelo fato deste algoritmo ter identificado 140017 mais correspondências, (655961 SURF e 515944 SIFT). Por fim, a diferença no tempo de execução se dá devido as diferenças em suas técnicas de descrição de características, o SURF utiliza os conceitos de imagens integrais, o que reduz o tempo de execução devido aos filtros de convolução serem aplicados diretamente na imagem original sem a necessidade de criação de novas imagens, enquanto no SIFT são

geradas novas versões da imagem por meio da convolução de um filtro gaussiano o que prolonga seu tempo de execução.

O algoritmo BRISK apresenta uma diferença elevada em seu uso de memória. Durante os testes foi verificado que essa diferença é justificada em função da instanciação do algoritmo. Após uma análise constatou-se que ao instanciar o BRISK cerca de 90% do total de memória é alocado. A diferença no tempo de execução ocorre devido a quantidade de correspondências identificadas por esse algoritmo ser extremamente menor em relação aos outros resultando em um tempo menor de execução – pois há menos correspondências para serem comparadas.

4 CONCLUSÃO

Em se tratando de sistemas de visão computacional, desempenho e precisão são fatores de extrema importância para determinar a qualidade do sistema e quando a obtenção dos resultados depende do fator tempo a problemática se torna ainda maior. Diante disso, analisar e extrair recursos de imagem em um ritmo competitivo para fornecer fluidez ao sistema é de grande valor para o projeto de sistema baseado em visão computacional.

Assim sendo, o presente trabalho propôs um estudo das funções dos algoritmos SIFT, BRISK e SURF no âmbito do reconhecimento facial a fim de obter a qualidade e os custos incorridos na execução, para que futuramente seja possível projetar um sistema de maneira otimizada, considerando as particularidades de cada algoritmo.

Durante a realização deste trabalho os algoritmos citados foram submetidos a base de imagens YaleFaces para que fosse possível a realização dos testes de precisão e custo computacional. Os testes ocorreram de forma que as imagens de treino fossem comparadas com as imagens de teste a fim de obter quais eram correspondentes entre si e com isso foi possível obter a taxa de acerto, tempo de execução, consumo de CPU e o uso de memória RAM de cada algoritmo.

Através dos resultados obtidos foi possível constatar que no quesito acurácia o algoritmo SURF apresentou o melhor resultado, com 83,33% de acertos, enquanto o SIFT e BRISK acertaram 76,67% e 63,33% respectivamente.

No que se refere ao quesito desempenho, o algoritmo BRISK apresentou o melhor resultado, consumindo 87,95MB de memória RAM e com um tempo de execução de apenas 3,67 segundos. Já o algoritmo SURF consumiu 79,07MB de memória RAM e teve um tempo de execução de 5,49 segundos e o SIFT teve um consumo de 71,04MB de memória RAM e um tempo de execução de 8,81 segundos, sendo considerado o pior desempenho entre os algoritmos citados.

Com base nesses dados foi possível concluir que o algoritmo SURF o melhor equilíbrio entre custo computacional e precisão, em relação aos demais, sendo possível através de sua utilização construir um sistema de reconhecimento de imagens capaz de entregar uma performance aceitável.

Fica proposto para trabalhos futuros, um estudo aprofundado sobre o SURF aplicando-o em outras bases de imagens e até mesmo em outras plataformas, como microcontroladores e smartphones, a fim de definir em qual cenário ele apresenta melhor desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

AWAD, Ali Ismail; HASSABALLAH, Mahmoud. **Image Feature Detectors and Descriptors: foundations and applications**. [S.I]: Springer International Publishing, 2016.

BAY, H., ESS, A., TUYTELAARS, T. AND VAN GOOL, L. **Speeded-up robust features (SURF)**. 2008. Disponível em: <<https://homes.esat.kuleuven.be/~konijn/publications/2008/CVIU-HB-08.pdf>> Acesso em: 29 nov. 2020.

DA SILVA, F. A., HIRAGA, A. K., ARTERO, A. O., DE PAIVA, M. S. V., BARBOSA, R. L. Um novo algoritmo para a construção de imagens panorâmicas usando os algoritmos SIFT e RANSAC. In: **ANAIS DO III SIMPÓSIO BRASILEIRO DE GEOMÁTICA**, Presidente Prudente, 2012, p. 151-156.

DAS, Apurba. **Guide to Signals and Patterns in Image Processing: Foundations, Methods and Applications**. [S.I]: Springer International Publishing, 2015.

DELGADO, E. D. H. **Reconhecimento biométrico usando informação da íris e de características perioculares**, 2013. Disponível em: <<http://www.pee.ufrj.br/index.php/pt/producao-academica/teses-de-doutorado/2013/2013102901-2013102901/file>> Acessado em: 29 nov. 2020.

DESELAERS, T. **Features for image retrieval**, 2003. Disponível em: <<http://www-i6.informatik.rwth-aachen.de/publications/download/10/DeselaersThomas--FeaturesforImageRetrieval--2003.pdf>> Acessado em: 29 nov. 2020.

DOUGHERTY, G. **Pattern Recognition and Classification: An Introduction**. Camarillo, California, United States: Springer International Publishing, 2013.

FAN, B.; WANG, Z.; WU, F. **Local Image Descriptor: Modern Approaches**. [S.I]: Springer Berlin Heidelberg, 2015.

GENG, C.; JIANG, X. SIFT features for face recognition. In: **2009 2nd IEEE International Conference on Computer Science and Information Technology**. Beijing, China: IEEE, 2009. p. 598-602.

GONZÁLES, G. L. G.; MEGGIOLARO, M. A. **Aplicação da Técnica SIFT para Determinação de Campos de Deformações de Materiais usando Visão Computacional**. 2010. Dissertação de Mestrado – Pontifícia Universidade Católica, Rio de Janeiro.

GONZALEZ, R. C.; WOODS, R.E. **Digital Image Processing**. New Jersey, United States: Prentice Hall, 2002.

KLETTE, R. **Concise Computer Vision: An Introduction into Theory and Algorithms**. [S.I.]: Springer London, 2014.

LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. Brisk: Binary robust invariant scalable keypoints. In: **Proceedings of the 2011 International Conference on Computer Vision**. Washington, United States: IEEE Computer Society, 2011. p.2548–2555.

LOWE, D. G. **Distinctive Image Features from Scale-Invariant Keypoints**. 1999. Disponível em: < <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>> Acessado em: 29 nov. 2020

LOWE, D. G. Object recognition from local scale-invariant Features. In: **Proceedings of the Seventh IEEE International Conference on Computer Vision**. Kerkyra, Greece: IEEE, 1999. p. 1150-1157

MONTANARI, R. **Detecção e classificação de objetos em imagens para rastreamento de veículos**. 2016. Tese (Doutorado) — Universidade de São Paulo, São Carlos.

MUHAMMAD, A. **OpenCV Android Programming By Example**. [S.l.]: Packt Publishing, 2015.

NIXON, A. S. A. M. **Feature Extraction and Image Processing**. 1. ed. [S.l.]: Newnes, 2002.

NIXON, M. S.; AGUADO, A. S. **Feature extraction & image processing for computer vision**. [S.l.]: Academic Press, 2012.

NOVAIS, J. P. **Aplicação dos Algoritmos SIFT e SURF na Classificação de Sub-Imagens por Discriminação de Textura**. 2016. Monografia – Centro Universitário Eurípides de Marília, Marília.

OPENCV. **OpenCV Documentation**. [S.l.], 2000.

PAVEL, F. A.; WANG, Z.; FENG, D. D. Reliable object recognition using SIFT Features. In: **2009 IEEE International Workshop on Multimedia Signal Processing**. Rio De Janeiro, 2009. p. 1-6.

SILVA, T.; **Análise de Descritores de Imagem e Classificadores Aplicados para Detecção de Lesões em imagens de Mamografia Digital**. 2017. Monografia – Universidade Federal Rural De Pernambuco, Recife.

Silveira, M.T., Feitosa, R.Q., Brito, J.L.N. and Jacobsen, K. **Correspondência eficiente de descritores SIFT para construção de mapas densos de pontos homólogos em imagens de sensoriamento remoto**. 2011. Disponível em: <https://www.scielo.br/scielo.php?pid=S1982-21702011000100008&script=sci_arttext&tlng=en> Acessado em: 29 nov. 2020.

TARALLO, A. de S., RV GONÇALVES, M. S. V. de PAIVA, LA de C. JORGE. **Mosaico automático de imagens agrícolas através da Transformada SIFT**. 2010. Disponível em: <https://www.alice.cnptia.embrapa.br/bitstream/doc/866903/1/p_100100105.pdf> Acessado em: 29 nov. 2020.

TIAN, D. P. **A Review on Image Feature Extraction and Representation Techniques**. 2013. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.450.7400&rep=rep1&type=pdf>> Acessado em: 29 nov. 2020

TUYTELAARS, T.; MIKOLAJCZYK, K. **Local invariant feature detectors: a survey**. [S.l]: Now Publishers Inc, 2008.

VIOLA, P. AND JONES, M., 2001. Rapid object detection using a boosted cascade of simple features. In: **Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition**. Kauai, HI, USA: IEEE, 2001.