



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

Ricardo Luna da Silva

Classificação de banhistas na faixa segura de praia

Recife

2018

Ricardo Luna da Silva

Classificação de banhistas na faixa segura de praia

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Ciência da Computação

Orientador: Valmir Macário Filho

Recife

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

S586c Silva, Ricardo Luna da
Classificação de banhistas na faixa segura de praia / Ricardo
Luna da Silva. – 2018.
66 f. : il.

Orientador: Valmir Macário Filho.
Trabalho de Conclusão de Curso (Graduação) – Universidade
Federal Rural de Pernambuco, Departamento de Estatística e
Informática, Recife, BR-PE, 2018.
Inclui referências.

1. Algoritmos computacionais 2. Inteligência artificial
3. Classificação I. Macário Filho, Valmir, orient. II. Título

CDD 004

"Eis que vos dou poder para pisar em serpentes e escorpiões, e sobre toda força do inimigo, e nada vos fará dano algum."

Lucas 10:19

Agradecimentos

Agradeço aos meus pais, Aluizio Severino e Mariza Bento, pela ajuda, paciência e todo suporte que me foi prestado durante toda meu período de graduação.

Aos professores do curso, especialmente aos fundamentos da computação, professor Valmir Macário, meu orientador, professor João Paulo e ao professor Luciano Pacifico por toda a ajuda durante o período na graduação e por acreditar em mim.

Agradeço também aos funcionários de toda Universidade Federal Rural de Pernambuco, especialmente para Sandra Xavier que foi praticamente uma segunda mãe durante todo meu tempo na graduação.

Aos meus amigos de curso, independente de período ou turma, que sempre estiveram ao meu lado e compartilhando conhecimento. Gostaria ainda mais de agradecer aos alunos da minha turma, de 2013.2, que sempre foram pacientes e me ajudaram a crescer como pessoa e academicamente. E a todas as outras pessoas que conheci durante esse período de curso na UFRPE que tornaram essa fase da minha vida algo maravilhoso.

Finalmente gostaria de agradecer a Deus, por todas as oportunidades proporcionadas durante essa fase da minha vida, desde a aprovação até o término do curso, e também pelas futuras oportunidades, pois sem ele não conseguiria chegar até onde cheguei. Agradeço a todos vocês de coração!

"O Senhor é o meu pastor, nada me faltará."

Salmos 23:1

Resumo

Visando evitar riscos em ambientes aquáticos, afogamentos e ataque de tubarão, áreas de praia devem ser monitoradas constantemente. Quando necessário, as equipes de resgate devem responder com velocidade ao caso. Este trabalho visa propor um algoritmo de classificação de pessoas como parte de um sistema para monitoramento automático em áreas de praia. Certos fatores do ambiente são bastante desafiadores, como variação de brilho em dias nublados, a posição do sol em diferentes momentos do dia, dificuldade em segmentação de imagens, pessoas submersas e posição afastada da câmera. Para esse tipo de problema na literatura é comumente encontrado, para detecção de pessoas, o uso de descritores de imagem em conjunto com um classificador. Este trabalho realiza um estudo em imagens de praia usando os seguintes descritores de imagem e suas combinações em pares: Momentos de Hu, Momentos de Zernike, Filtro de Gabor, Histograma de Gradientes Orientados (HOG), Padrões Binários Locais (LBP) e Haar. Além disso, uma técnica de redução de dimensionalidade (PCA) é aplicada para seleção de características. A taxa de detecção é avaliada com os seguintes classificadores: *Random Forest*, classificador em cascata e *Support Vector Machine* (SVM) com *kernel* linear e radial. Os experimentos demonstraram que o classificador SVM com *kernel* radial usando os descritores HOG e LBP aplicando a técnica PCA mostrou resultados promissores, obtendo 90,31% de precisão. **Palavras-chave:** Visão computacional, descritores de características, classificação.

Abstract

In order to avoid risks in aquatic environments, drownings and shark attack, beach areas should be constantly monitored. When necessary, rescue workers must respond quickly to the case. This work aims to propose a classification algorithm for people as part of a system for automatic monitoring in beach areas. Certain environmental factors are quite challenging, such as varying brightness on cloudy days, the position of the sun at different times of the day, difficulty in segmenting images, submerged people, and position away from the camera. For this type of problem in the literature is commonly found, for people detection, the use of image descriptors in conjunction with a classifier. This work performs a beach image study using the following image descriptors and their combinations in pairs: Hu Moments, Zernike Moments, Gabor Filter, Guided Gradient Histogram (HOG), Local Binary Patterns (LBP) and Haar. In addition, a dimensionality reduction technique (PCA) is applied for feature selection. The detection rate is evaluated with the following classifiers: textit Random Forest, cascade classifier and textit Support Vector Machine (SVM) with linear and radial textit kernel. The experiments demonstrated that the SVM classifier with radial kernel using the HOG and LBP descriptors applying the PCA technique showed promising results, obtaining 90.31% accuracy. **Keywords:** Computer Vision, feature descriptors, classification.

Lista de ilustrações

Figura 1 – Placas de alerta contra ataques de tubarões.	14
Figura 2 – Imagem da praia de Boa Viagem.	17
Figura 3 – (a) Imagem projetada em uma matriz. (b) Imagem resultante após o processo de quantização e amostragem.	22
Figura 4 – Cubo RGB.	23
Figura 5 – Imagem em RGB.	23
Figura 6 – Imagem após o processo de conversão para tons de cinza, (a) imagem em RGB e em (b) imagem convertida para tons de cinza.	24
Figura 7 – Operações lógicas.	25
Figura 8 – Máscara de gradientes utilizadas no descritor HOG.	28
Figura 9 – Uma amostra positiva (a) e negativa (d) da base de dados com os respectivos gradientes, magnitude (b) (e) e orientação (d) (f).	28
Figura 10 – Conjunto de células do descritor HOG.	28
Figura 11 – Geração dos histogramas para o conjunto de blocos.	28
Figura 12 – Um exemplo de características LBP.	29
Figura 13 – Resultado da aplicação do filtro de Gabor em uma imagem.	30
Figura 14 – Polinômios de Zernike.	32
Figura 15 – Retângulos utilizados na versão antiga das características de Haar.	33
Figura 16 – Retângulos utilizados na versão atual das características de Haar.	33
Figura 17 – Árvore de decisão.	34
Figura 18 – Ilustração de hiperplanos canônicos e separador.	36
Figura 19 – Classificador em cascata.	37
Figura 20 – Processo de extração de características.	40
Figura 21 – Imagem da praia de Boa Viagem.	40
Figura 22 – Fluxograma do trabalho.	42
Figura 23 – Imagem rotulada.	45
Figura 24 – Imagem binária.	46
Figura 25 – Um exemplo de imagens positivas (a) (b) (c) e negativas (d) (e) (f).	46
Figura 26 – Filtro de Gabor com janelas 5×5 aplicada (a) com três diferentes orientações: 0° (b), 45° (c) e 90° (d).	48

Lista de tabelas

Tabela 1 – Experimento P-1. Taxa de reconhecimento (%) durante a seleção do parâmetro do filtro de Gabor (orientação e tamanho da janela). . . .	53
Tabela 2 – Experimento P-2. Taxa de reconhecimento para diferentes ordens do momento de Zernike.	53
Tabela 3 – Experimento P-3. Taxa de reconhecimento (%) para diferentes tamanhos de células (linhas) e blocos (colunas) para o descritor HOG. .	53
Tabela 4 – Experimento P-4. Taxa de reconhecimento (%) para diferentes parâmetros do LBP.	54
Tabela 5 – Tamanho dos vetores de característica e suas combinações em pares	55
Tabela 6 – Experimento C-1. Taxa de reconhecimento (%) na base de testes usando classificador Floresta aleatória.	55
Tabela 7 – Experimento C-2. Taxa de reconhecimento (%) na base de testes usando classificador <i>Random Forest</i> e PCA	56
Tabela 8 – Experimento C-2. Taxa de reconhecimento (%) na base de testes usando classificador SVM com <i>kernel</i> linear	56
Tabela 9 – Experimento C-2. Taxa de reconhecimento (%) na base de testes usando classificador SVM com <i>kernel</i> linear e PCA	56
Tabela 10 – Experimento C-3. Taxa de reconhecimento (%) na base de testes usando classificador SVM com <i>kernel</i> radial	57
Tabela 11 – Experimento C-3. Taxa de reconhecimento (%) na base de testes usando classificador SVM com <i>kernel</i> radial e PCA	57
Tabela 12 – Experimento C-4. Taxa de reconhecimento (%) na base de testes usando classificador em cascata	58
Tabela 13 – Tempo de extração médio de características em milissegundos de uma imagem em tons de cinza com dimensões de 29×25 <i>pixels</i> . .	58

Lista de abreviaturas e siglas

CMY	Cyan, Magenta e Yellow
CUDA	Compute Unified Device Architecture
EMD	Earth mover's distance
GPU	Graphics Processing Unit
HOG	Histograma de Gradientes Orientados
HOOF	Histograma Orientado a Fluxo Óptico
KNN	K – Nearest Neighbors
LBP	Padrões Binários Locais
MBH	Histograma de fronteira de movimento
OpenCv	Open Source Computer Vision Library
PCA	Análise de componentes principais
RCM	Region Covariance Matrix
RGB	Red, Green e Blue
SIFT	Scaleinvariant Feature Transform
SURF	Speeded up Robust Features
SVM	Máquina de Vetores de Suporte
UFRPE	Universidade Federal Rural de Pernambuco
UKF	Unscented Kalman Filter

Sumário

	Lista de ilustrações	8
1	INTRODUÇÃO	13
1.1	Problema de Pesquisa	16
1.1.1	Justificativa	17
1.2	Objetivos	20
1.3	Estrutura do trabalho	20
2	PROCESSAMENTO DE IMAGEM	21
2.1	Introdução	21
2.2	Espaço de cores	22
2.2.1	Modelo RGB de cores	22
2.2.2	Conversão de RGB para tons de cinza	24
2.3	Vizinhança de um pixel	24
2.4	Operações lógicas	25
2.5	Descritores de características	26
2.5.1	Momentos de Hu	26
2.5.2	HOG (<i>Histogram of Oriented Gradients</i>)	27
2.5.3	LBP (<i>Local binary patterns</i>)	29
2.5.4	Características de Gabor	29
2.5.5	Momentos de Zernike	31
2.5.6	Características de Haar	32
2.6	Classificação de Banhista	33
2.6.1	Floresta Randômica (<i>Random forest</i>)	33
2.6.2	Máquina de vetores de suporte (<i>support vector machine – SVM</i>)	35
2.6.3	Cascata de classificadores (<i>Train cascade</i>)	37
2.6.4	Análise de componentes principais - PCA	38
3	CLASSIFICAÇÃO DE BANHISTA	39
3.1	Extração de características	39
3.2	Classificação de Banhista	41
4	AVALIAÇÃO EXPERIMENTAL	44
4.1	Base de dados	44
4.2	Seleção de parâmetros	47
4.2.1	Experimento P	47

4.2.2	Experimento P-1	47
4.2.3	Experimento P-2	48
4.2.4	Experimento P-3	48
4.2.5	Experimento P-4	48
4.2.6	Combinação de descritores e classificadores	49
4.2.7	Experimento C-1	49
4.2.8	Experimento C-2	49
4.2.9	Experimento C-3	50
4.2.10	Experimento C-4	50
4.3	Ambiente experimental	50
4.4	Métricas de validação	51
5	RESULTADOS	52
5.1	Resultados do experimentos do tipo P	52
5.1.1	Características de Gabor	52
5.1.2	Momentos de Zernike	53
5.1.3	HOG	53
5.1.4	LBP	54
5.1.5	Tamanho do vetores de características	54
5.2	Resultados do experimentos do tipo C	55
5.2.1	Floresta aleatória	55
5.2.2	SVM com <i>kernel</i> linear	56
5.2.3	SVM com <i>kernel</i> radial	57
5.2.4	Classificadores em cascata	57
5.3	Tempo de extração médio	57
5.4	Discussão	58
6	CONCLUSÃO E TRABALHOS FUTUROS	61
6.1	Trabalhos Futuros	61
6.2	Contribuições	62
	REFERÊNCIAS	63

1 Introdução

Ataques de tubarões aos seres humanos não são um problema recente nas regiões litorâneas do Brasil, mas só depois de muito tempo foi que se teve início ao monitoramento desse problema. Desde o surgimento, já foram estudadas e aplicadas inúmeras providências, mas até o momento nenhuma delas se mostrou bastante efetiva. Especificamente no estado de Pernambuco, esses incidentes começaram a ser monitorados aproximadamente 25 anos atrás. No decorrer de pouco mais de duas décadas ocorreu o registro de 61 vítimas que foram atacadas e desse total um número de 24 pessoas morreram segundo dados do G1(PE, 2017). Com esses números nossa cidade entrou para as estatísticas de incidentes com esses animais.

Mas esse problema não é único do estado de Pernambuco, segundo dados da Universidade Federal Rural de Pernambuco (UFRPE) (PERNAMBUCO, 2017), temos 11 unidades do total de 17 da federação que são cortadas pelo mar, somando um total de 100 ocorrências na costa brasileira. Em Pernambuco, nosso primeiro caso registrado ocorreu na praia de Piedade em 1992 e tivemos o nosso maior índice em 1994 quando um total de 10 pessoas foram atacadas.

Com o objetivo de diminuir esses ataques na região metropolitana, as áreas de risco foram sinalizadas com placas que informam sobre o risco de ataques de tubarão naquele local, e em alguns pontos são encontrados postos de salva-vidas que monitoram o movimento das pessoas com o intuito de prevenir esses e outros acidentes com os banhistas. O monitoramento pode ser uma estratégia efetiva, mas com o tamanho da área litorânea do Recife, que possui um total de 32 quilômetros de áreas restritas ao banho, tal abordagem se torna bastante difícil de ser aplicada. Na maioria dessas praias, um perímetro de proteção é dado por barreiras naturais (recifes), pois a maioria dos acidentes são causados pelo avanço de pessoas além do perímetro seguro ou em áreas que não possuem essas barreiras naturais.

Figura 1 – Placas de alerta contra ataques de tubarões.



Fonte – Marlon Costa/Pernambuco Press.

Para trabalhar na abordagem desse problema, uma estratégia é o monitoramento por câmeras e tal atividade sendo feita de forma autônoma. A implantação de tal estratégia deve mitigar aspectos relacionados ao reconhecimento de pessoas em si e outros problemas relacionados ao ambiente.

O monitoramento por câmeras já vem desde muito tempo sendo utilizado para observação de várias situações que acontecem diariamente em muitas cidades do mundo. Normalmente esse monitoramento é feito por uma equipe responsável por acompanhar vídeos em um monitor próximo e tomar alguma decisão quando necessário. E não muito recentemente essas atividades começaram a ser auxiliadas por sistemas computacionais aumentando assim sua eficiência e permitindo que uma quantidade menor de pessoas conseguissem monitorar áreas cada vez maiores.

As etapas normalmente utilizadas nesses sistemas envolvem segmentação, classificação e rastreamento. A segmentação é a etapa responsável por fragmentar a imagem em segmentos que são similares entre si, dessa forma as partes que não são de interesse na imagem podem ser descartadas, diminuindo assim o espaço de busca do objeto de interesse. Durante a classificação cada um dos componentes, segmentos ou partes da imagem será avaliado por um classificador que indica a que classe pertence, que para nosso problema pode ser indicado como pessoa ou não-pessoa, com uma

certa precisão. E o rastreamento é a etapa responsável por acompanhar um determinado objeto no decorrer de uma sequência de imagens. Este trabalho centra-se na segunda etapa (classificação) que utilizando valores computados a partir de um determinado segmento da imagem pode indicar com um certa precisão o que seria aquele determinado segmento.

Com o intuito de obter um resultado robusto, algumas abordagens precisam ser desenvolvidas para este sistema de monitoramento, que pode avaliar e posteriormente utilizar o resultado dessa pesquisa, dado que as pessoas estarão submersas e podem desaparecer temporariamente da visão da câmera. E de forma adicional escolher uma representação poderosa para representar aspectos da imagem que está sendo avaliada. Além desse fato, o ambiente envolvido é não controlado e dinâmico de forma a propiciar um maior desafio para o sistema. Até onde os autores sabem, nenhum outro trabalho detecta pessoas, especificamente em imagens de praia.

Para detecção de pessoas, uma prática comum encontrada na literatura é a utilização de descritores de características em conjunto com um classificador (ZENG; MA, 2010). O descritor de características cria uma representação matemática da imagem através de valores que podem representar particularidades da imagem. Podendo representar medidas estatísticas que relacionam características intrínsecas de uma imagem específica; formas mais especificamente como retas e outras formas bem definidas ou texturas que podem ser vistas como variação entre uma vizinhança de um pixel ou região. A extração de característica é uma parte fundamental para a pesquisa e os algoritmos utilizados devem possuir um baixo custo computacional levando em conta que o sistema deve retornar uma resposta com pouco atraso.

O classificador é responsável por verificar se os candidatos que foram fornecidos como entrada são banhistas ou não, levando em consideração uma determinada taxa de precisão. Neste projeto, o classificador deve obter um bom resultado quando se comparar à relação entre precisão do classificador versus custo computacional, tal fato deve ser levando em conta visando o funcionamento em tempo real do sistema como um todo. A escolha de um algoritmo de classificação para um dado conjunto de características reflete substancialmente no custo computacional, ou seja, tempo de processamento e na qualidade da detecção de objetos. Classificadores como Funções com base radial (RBF) (CHANG; FU, 2006) e máquina de vetor de suporte (SVM) com *kernel* não linear (MAJI; BERG; MALIK, 2008), geralmente produzem bons resultados, porém tem uma velocidade baixa de processamento. Classificadores como SVM com *kernel* linear (CHANG; LIN, 2008) ou *Random Forests* (MARIN et al., 2013) são geralmente utilizados por causa disso.

1.1 Problema de Pesquisa

O problema da pesquisa é a detecção de pessoas em regiões de praia, porém o foco é um pouco diferente se comparado à maioria das abordagens encontradas no estado da arte. Um dos trabalhos mais conhecidos na literatura, para detecção de pessoas é o de Viola *et al.* (VIOLA; JONES; SNOW, 2005), que propôs uma estrutura de detecção de pedestres que usa características semelhantes a Haar e, em seguida, aplica algoritmos de aprendizagem de máquina usando o AdaBoost para obter resultados de classificação satisfatórios.

E mais recentemente, as pessoas são detectadas através do rastreamento de cabeças (HSU; GUBBI; PALANISWAMI, 2015). O trabalho de Xie *et al.* (XIE; DANG; TONG, 2012) extrai Histograma de Gradientes Orientados (HOG) alimentado por uma GPU (*Graphics Processing Unit*). Para se obter uma melhor detecção é utilizada a detecção de cabeça e ombros, visando obter uma maior robustez para essa tarefa, que é quase sempre visível na cena. Outros trabalhos levam em consideração que a forma, cabeça e ombros, é mais importante (ARDIYANTO; MIURA, 2014).

Dado todo esse conjunto de técnicas apresentadas é possível observar a complexidade do problema em questão. Utilizando todo o conhecimento obtido acerca do problema foi observado que o uso de descritores seria uma solução simples e que poderia ter um baixo custo computacional.

O contexto também apresenta problemas que são específicos do local, a praia, um ambiente que tem alta taxa de incidência de luz praticamente durante todo o ano, o movimento constante das ondas e a variação da maré são fatores que limitam o uso de algumas técnicas na literatura. Visando bons resultados o algoritmo de extração de características deve atender a certas particularidades do problema, lidar bem com a variação de iluminação, invariância a escala e rotação e ter um custo computacional relativamente baixo. A Figura 2 apresenta uma imagem da praia da Boa Viagem.

Figura 2 – Imagem da praia de Boa Viagem.



Fonte – O autor.

1.1.1 Justificativa

Na literatura pode ser encontrada uma vasta gama de técnicas para o reconhecimento de pessoas, mas a maioria dessas técnicas muitas vezes são formuladas levando em consideração um conjunto de premissas, onde uma delas é o aparecimento total da região dos ombros e a cabeça (ZENG; MA, 2010). Outro trabalho utiliza uma técnica para remoção de fundo dividida em duas etapas, na primeira uma etapa é feita a remoção de objetos estáticos e a segunda é para encontrar pessoas que podem ter sido removidas da imagem no modelo estático (TU; ZHANG; HAO, 2013). Podendo utilizar também um conjunto de câmeras para conseguir detectar as pessoas (ESHEL; MOSES, 2008). Nesse contexto cada técnica fica sujeita ao cumprimento de certos requisitos e a não ocorrência ou variação dessas premissas pode fazer o resultado final não cumprir o objetivo esperado.

Para a solução do problema abordado, no caso a detecção de pessoas, foi possível encontrar uma vasta gama de técnicas na literatura e cada uma conta com sua resolução, como na técnica usada por Birchfield (BIRCHFIELD, 1998), que aproximou um descritor em forma de elipse para rastrear cabeças e utilizou a informação da diferença de gradiente da fronteira da cabeça e do histograma de cores da cabeça. Uma

elipse também foi utilizada no trabalho de Charif e McKenna (ROUGIER et al., 2006) para rastrear a cabeça de várias pessoas em uma sala. Nosso problema tem características semelhantes, mas alguns dos fatores, principalmente o de hardware leva a buscar por forma com o menor custo computacional possível.

Técnicas mais recentes contam com mais de um estágio para classificação e usa descritores combinados (HU et al., 2014). No primeiro estágio são utilizados dois descritores, HOG e LBP, em conjunto com um classificador Viola-Jones para eliminar a maioria dos candidatos, no segundo estágio é utilizada RCM (*Region Covariance Matrix*) para refinar a classificação.

Para detecção e rastreamento pessoas, no trabalho de Xie et al. (XIE; DANG; TONG, 2012) são utilizadas imagens adquiridas a partir de uma câmera de segurança. Como o artigo anterior, o mesmo extrai características utilizando o descritor HOG, que é alimentado por uma GPU. Todos os verdadeiros positivos, regiões na imagem que de fato se encontram as pessoas, são acompanhados por uma trilha com base no algoritmo de rastreamento EMD (*Earth mover's distance*) com pontos SURF (*Speeded up robust features*). Como o processo de extração de característica é geralmente muito lento, foi utilizado o CUDA (*Compute Unified Device Architecture*) para acelerar esse processo.

Outras técnicas também são encontradas na literatura, como segmentação por cor (WANG; ZHANG; MIAO, 2013). O trabalho proposto utiliza a semântica da cor como característica para segmentação da imagem, ou seja, a cor é utilizada para realizar a classificação apenas da parte superior do corpo. No trabalho foi estimado uma quantidade de classes com valor fixo, que possui sua variação de acordo com a semântica (pele, cabelo, roupa e fundo). E para classificação foi utilizado classificador SMV com *kernel* linear, algoritmo utilizado comumente no estado da arte deste tipo de problema.

Um outro trabalho tem como principal objetivo a detecção de cabeças a partir de recursos de histograma de movimento (HSU; GUBBI; PALANISWAMI, 2015). Também são introduzindo novas ideias, que são baseadas em uma arquitetura multinível de pirâmide e por Histograma de fronteira de movimento (MBH) e Histograma Orientado a Fluxo Óptico (HOOF). Para a classificação com o intuito de distinguir uma cabeça humana de características similares são utilizados dois estágios de SVM, e um mapeamento de *kernel* é utilizado para característica do histograma de movimentos com o *kernels* Bhattacharyya que possui características simétricas e invariantes.

A forma também é uma característica importante, como no trabalho de Ardiyanto (ARDIYANTO; MIURA, 2014), que utiliza uma abordagem que emprega hierarquicamente modelos baseados em mínimos quadrados parciais de características de gradiente e textura em conjunto com um classificador *Random Forest*. No trabalho também é desenvolvido o UKF (*Unscented Kalman Filter*) que tem como objetivo fazer uma

estimativa do movimento humano. Sendo essa uma estratégia diferente das demais apregadas anteriormente. A orientação do corpo humano é algo bastante importante para resolução do problema neste trabalho, pois com a utilização de tais padrões é possível aprender o comportamento de grupos de pessoas e tentar prever os próximos passos até um determinado limite.

Alguns descritores já mencionados, HOG e LBP, são bastantes utilizados na maioria dos trabalhos e outros como Hu e Zernike, mesmo que Zernike são tidos como mais acurados que os momentos de Hu (SABHARA; LEE; LIM, 2013), e SIFT (*scale-invariant feature transform*) (LOWE, 1999) também são usados.

O SVM não linear melhorado IKSVM (MAJI; BERG; MALIK, 2008) foi utilizado para classificar de objetos no trabalho de Van de Sande (SANDE et al., 2011). Podemos ainda citar a SVM como classificador de pedestres no trabalho de Gonzalez et al. (GONZÁLEZ et al., 2015). O algoritmo *Random Forests* também foi utilizado para classificação de pedestres no trabalho de Marin et al. (MARIN et al., 2013). No domínio da classificação cabeças de pessoas, temos como exemplo a classificação de ombros e cabeça utilizando a SVM no trabalho de Tu et al. (TU; ZHANG; HAO, 2013) e de cabeça no trabalho de Xie et al. (XIE; DANG; TONG, 2012). Kępski e Kwolek (KEPSKI; KWOLEK, 2014) utilizaram o KNN (*K – Nearest Neighbors*) para identificar a pessoa rastreada através de sua cabeça. Guan e Huang (GUAN; HUANG, 2015) utilizaram o algoritmo *Adaboost* em cascata para treinar exemplos positivos e negativos de cabeças.

O uso de descritores de característica é motivado pelo fato de que as imagens são representadas de uma forma adequada para o processamento computacional (GONZALEZ; WOODS, 2010) e posteriormente a etapa de reconhecimento ou classificação atribui um rótulo a um objeto com base nos dados que são a representação obtida por seus descritores (GONZALEZ; WOODS, 2010).

As técnicas citadas têm utilizado contextos específicos para a solução de detecção de pessoas e foram desenvolvidas visando o uso a partir de um hardware considerado o mínimo. O objetivo do trabalho é identificar pessoas a partir de imagens de praia, através da avaliação de descritores de imagens e classificadores. E para reduzir a dimensionalidade do vetor serão aplicadas técnicas geralmente utilizadas na literatura que podem encontrar padrões em dados de alta dimensionalidade (SMITH, 2002) sem perder uma quantidade significativa de informação.

1.2 Objetivos

Objetivo Geral:

O trabalho proposto tem por objetivo propor um método para classificar banhistas em imagens de praia.

Objetivos Específicos:

- Selecionar uma ou mais técnicas de extração de características para identificar um banhista.
- Avaliar algoritmos na tarefa de classificar banhistas em imagens de praias.
- Realizar busca por parâmetros que melhorem os resultados do classificador.
- Aplicar técnicas que tem como objetivo reduzir a dimensionalidade do vetor de características.

1.3 Estrutura do trabalho

O Capítulo 2 desse trabalho apresentará uma introdução sobre as técnicas utilizadas, no contexto envolvendo processamento de imagens, descritores e classificadores. O Capítulo 3 apresenta o conceito de utilização de descritores de características e classificadores para classificação de banhista na faixa segura de praia. O Capítulo 4 apresenta todo o material e a metodologia aplica nos experimentos. O Capítulo 5 descreve os resultados obtidos para cada experimento realizado. E o Capítulo 6, a conclusão e os trabalhos futuros.

2 Processamento de Imagem

Uma imagem digital pode ser vista como uma função bidimensional, representados por valores finitos e discretos na forma $f(x, y)$ (GONZALEZ; WOODS, 2010). Para gerar uma imagem a partir de um processo físico, todos os valores relacionados a representação da imagem são proporcionais a energia irradiada por uma fonte real. A saída comum dos sensores é representada por uma forma de onda de tensão contínua, cujo os valores e comportamento estão relacionados ao fenômeno físico que está sendo captado pelos sensores. Para criar uma imagem digital precisamos converter tais valores contínuos para um formato digital. Neste Capítulo será feita uma introdução aos conceitos básicos que estão ligados ao processo de representação, espaço de cores, conversão entre espaços, os descritores, que apresentam como os aspectos da imagem podem ser representados a partir de um conjunto de operações, e os classificadores.

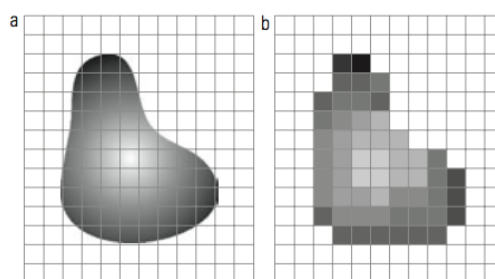
2.1 Introdução

Como já mencionado, uma imagem digital pode ser expressa como uma função bidimensional com a forma $f(x, y)$. Um valor das coordenadas espacial (x, y) representa a amplitude, sendo um valor escalar positivo e seu significado físico é determinado pela origem da imagem. Para criar a imagem digital, os valores contínuos gerados pelo sensor são convertidos e tal processo envolve a amostragem e quantização. Amostragem, o nome dado ao processo de digitalizar os valores das coordenadas da imagem, pode ser entendido como tamanho ou quantidade de coordenadas utilizada para representar a imagem capturada ou imagem analógica e cada componente que constituem a imagem resultante são chamados de *pixel*. E quantização é o nome dado ao processo de digitalizar os valores das amplitudes, valores esses obtidos durante o processo de aquisição de imagem e são os valores contínuos captado pelo sensor (GONZALEZ; WOODS, 2010).

No computador, a imagem pode ser representada por uma matriz de dimensões $I \times J$, onde esse tamanho é gerado no processo de amostragem, e em uma dada coordenada é encontrado um valor, que foi gerador a partir do mapeamento durante o processo de quantização. O valor encontrado para cada posição da matriz está limitado de acordo com os limites escolhidos durante a quantização. Para representar uma imagem preta e branca é utilizado dois valores que podem ser representados como zero e um, o zero sendo a ausência de cor (preto) e um sendo a presença de cor (branco).

A Figura 3 apresenta o resultado obtido após o mapeamento ocorrido durante o processo de quantização e amostragem. É possível ver que após o processo a imagem pode ficar distorcida ou embaçada e tal fato é advindo dos limites para a representação da imagem digital. Ao se aumentar a resolução da imagem a mesma pode apresentar uma forma mais próxima da original.

Figura 3 – (a) Imagem projetada em uma matriz. (b) Imagem resultante após o processo de quantização e amostragem.



Fonte – (GONZALEZ; WOODS, 2010)

2.2 Espaço de cores

A utilização das cores, do ponto de vista do processamento de imagens pode ser influenciado por diversos fatores. A cor pode ser uma característica poderosa para encontrar um objeto em uma cena, tanto para os sistemas computacionais como para os seres humanos. E dependendo da atividade pretendida, um espaço de cor pode representar melhor um dado ou pode tornar mais fácil a comparação entre duas cores distintas. Outros espaços de cores podem ser mais preferíveis a outros para um nicho específico como impressoras que utiliza o espaço CMY (*cyan*, *magenta* e *yellow*) ou os monitores com o espaço RGB (*red*, *green* e *blue*) (GONZALEZ; WOODS, 2010).

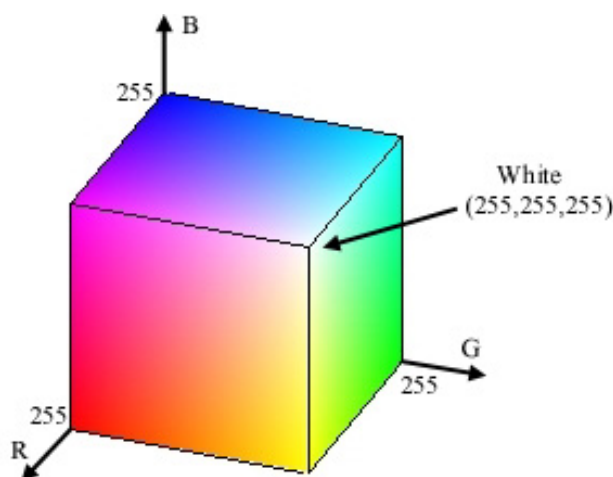
O objetivo de um espaço de cores (podendo ser chamado também de modelos de cores ou sistemas de cores) é a simplificar a especificação de cores em uma forma comum ou padronizada. E pode ser resumido em uma forma de especificação de coordenadas e um subespaço, tal que cada ponto representa uma única cor. Um espaço de cor pode ser orientado especificamente para uma determinada área, um exemplo que temos são orientados para hardware e manipulação de cores. Cada espaço pode ter uma representação diferente e limites diferentes (GONZALEZ; WOODS, 2010).

2.2.1 Modelo RGB de cores

Esse modelo é representado por 3 componentes espectrais primários sendo: vermelho, verde e azul. Todo o conjunto de cores formadas nesse espaço pode ser

visto como um cubo, os três valores primários ficam nos vértices. A representação de uma cor nesse modelo é feita utilizando um sistema de coordenadas cartesianas. Ao se observar o cubo alguns fatos interessantes são vistos, a origem é onde está a cor preta e o ponto mais distante dela é o branco e toda a diagonal do cubo, onde os valores dos três componentes são iguais fica a escala de cinza (GONZALEZ; WOODS, 2010). O cubo RGB é apresentado na Figura 4.

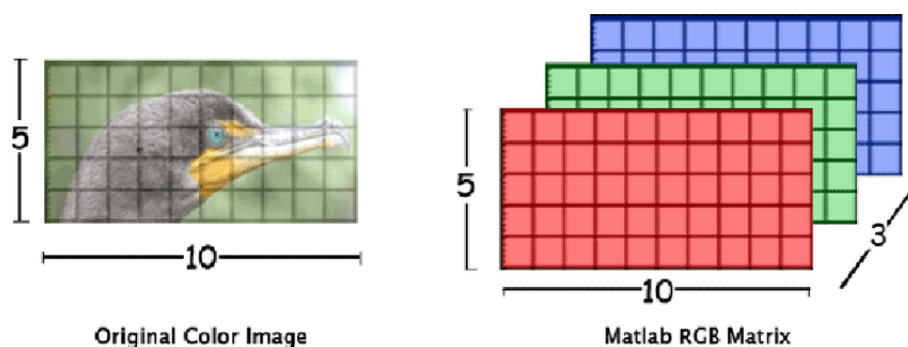
Figura 4 – Cubo RGB.



Fonte – <http://radio.feld.cvut.cz/matlab/toolbox/images/color4.html>

As imagens que utilizam essa representação possuem três componentes, um para cada cor primária. E quando essas componentes são combinadas em uma tela são formadas as cores compostas. Para representar uma cor em um determinado espaço é utilizada uma determinada quantidade de bits e essa quantidade é chamada de profundidade de *pixel* a Figura 5 mostra um exemplo.

Figura 5 – Imagem em RGB.



Fonte – <https://www.researchgate.net/figure/Color-image-representation-and-RGB-matrix/fig15282798184>

Uma imagem utilizando o modelo RGB pode ser decomposta em uma imagem

para cada canal, que representa uma cor, e cada *pixel* em uma imagem utiliza 8 bits. Logo, para cada *pixel* no modelo RGB, a combinação dos três canais ou trio de valores, possui uma profundidade de 24 bits. Nesse modelo de cores, temos que cada canal pode variar de 0 até 255 (256 ou 2^8 tons), como são 3 canais de cores e cada um com 8 bits temos um total de combinações de $(2^8)^3 = 16.777.216$.

2.2.2 Conversão de RGB para tons de cinza

Os descritores de característica em sua maioria, por padrão, utilizam a imagem em tons de cinza e motivando a conversão nesses casos. A conversão (OPENCV, 2015) é feita de forma simples, o resultado final é dado pela formula 2.1:

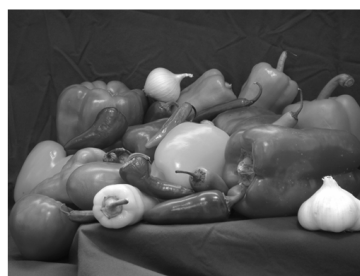
$$Y = 0.299.R + 0.587.G + 0.114.B \quad (2.1)$$

O resultado final é uma soma ponderada de todas as cores, RGB, e cada *pixel* é multiplicado por um valor de acordo com a importância do canal específico. A Figura 6 mostra uma imagem após o processo de conversão.

Figura 6 – Imagem após o processo de conversão para tons de cinza, (a) imagem em RGB e em (b) imagem convertida para tons de cinza.



(a) Imagem em RGB



(b) Imagem em tons de cinza

Fonte – https://www.mathworks.com/help/matlab/ref/rgb2gray.html?s_tid=srchtitle

2.3 Vizinhança de um pixel

Algumas das operações e descritores, que serão apresentado posteriormente, utilizam operações de vizinhanças de um *pixel*. Como já foi mencionado um *pixel* em uma imagem é expressado como $f(x, y)$. A vizinhança de um *pixel* é comumente utilizada em determinadas operações envolvendo processamento digital de imagens.

Inicialmente estamos focado em um único *pixel* em uma posição qualquer e esse *pixel* está em uma coordenada (x, y) . Os vizinhos verticais e horizontais desse

pixel se encontram em uma *vizinhança-4* (GONZALEZ; WOODS, 2010) que pode ser expressa por $N_4(p)$, onde p é o *pixel* em questão. Para encontrar a posição do conjunto de *pixel* que estão em uma *vizinhança-4* utilizamos os valores abaixo:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad (2.2)$$

Já para encontrar os vizinhos diagonais de um *pixel* devemos utilizar os valores abaixo:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1) \quad (2.3)$$

essa *vizinhança* é expressa por $N_d(p)$ (GONZALEZ; WOODS, 2010) onde p é o *pixel* analisado. A combinação da vizinhança vertical e horizontal acrescida da *vizinhança* diagonal é chamada de *vizinhança-8* (GONZALEZ; WOODS, 2010) que pode ser expressa como $N_8(p)$.

2.4 Operações lógicas

Como será mostrado na Seção 4, apresentada posteriormente, no projeto foram utilizadas algumas operações lógicas envolvendo imagens digitais, as operações foram E (AND) e OU (OR).

Considerando duas imagens A e B, imagens binárias, temos que cada *pixel* é representado por 1 (UM), representando a cor branca, e 0 (ZERO), representando a cor preta. A operação OU entre essas duas imagens resulta em 1 (UM), no *pixel* na imagem resultante, apenas quando uma das imagens possuir a cor branca e 0 (ZERO) apenas no caso de nenhum conter o valor 1 (UM).

Ainda levando em consideração o conjunto de imagens citadas no parágrafo anterior, temos que a operação E vai resultar, na imagem resultante, no valor 1 (UM) apenas no caso onde o *pixel* nas duas imagens tenha a cor branca e 0 (ZERO) em qualquer outro caso. Na Figura 7 apresenta a tabela completa com as operações lógicas.

Figura 7 – Operações lógicas.

Entrada		Saída	
A	B	OU	E
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Fonte – O autor

Para o caso de operações envolvendo imagens em tons de cinza a operação passa a ser feita bit a bit, ou seja, o valor 7 (SETE) é representado como sendo 000 0111, levando em consideração que os valores possíveis da imagem são de 0 até 255, e o valor 5 (CINCO) é representado como 0000 0101. Ao se aplicar a operação OU sobre o dois valores indicados temos como saída o valor 7 (SETE) e ao se aplicar a operação E temos como saída o valor 5 (CINCO).

2.5 Descritores de características

Como já mencionado anteriormente, a partir da aplicação de descritores é possível obter valores que representam formas, curvas ou medidas estatísticas da imagem. De uma forma simples, um descritor pode ser visto como um algoritmo que quando aplicado em uma imagem tem como resultado valores que são considerados informações relevantes e úteis para um determinado tipo de problema ou representação desse problema.

2.5.1 Momentos de Hu

Momentos em uma imagem são utilizados devido as suas características e seu poder de representação. Também podendo ser chamados de momentos estatísticos e são utilizados ou avaliados quando se deseja realizar alguma atividade visando o reconhecimento de objetos. Seu poder vem do fato que os valores gerados pelos momentos e os valores das derivadas descrevem a distribuição dos pontos contidos na imagem ou região dela.

Em 1962, um trabalho [HU \(1962\)](#) apresentou um conjunto de equações que apresentaram uma característica muito importante para o problema de reconhecimento de objetos, os momentos são invariantes a escala, rotação e translação. Abaixo são apresentados o conjunto de sete equações conhecidas como momentos de Hu:

$$I_1 = \eta_{20} + \eta_{02} \quad (2.4)$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (2.5)$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.6)$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (2.7)$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (2.8)$$

$$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.9)$$

$$+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \quad (2.10)$$

$$- (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

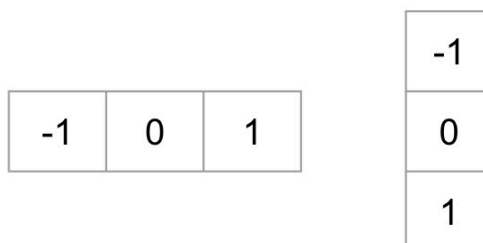
2.5.2 HOG (*Histogram of Oriented Gradients*)

A principal ideia desse descritor é que uma imagem pode ser representada pelo seu gradiente ou direções das bordas. Para obter um vetor de características HOG é necessário efetuar as seguintes etapas:

- Calcular os gradientes de uma imagem em tons de cinza, para tal são utilizadas as máscaras derivadas discretas, tanto no eixo x (horizontal) como no eixo y (vertical);
- Dividir a imagem em células de um tamanho específico, que é o agrupamento de um conjunto de pixels. As células são janelas da imagem original com dimensões de $V \times V$ *pixels* e os blocos são vistos como uma junção de $Nc \times Nc$ células gerando uma total de $(Nc \times V) \times (Nc \times V)$ *pixels*. Quando o tamanho da célula aumenta é possível obter informações de maior escala. E quando se reduz o tamanho da célula, obtém-se informações mais detalhadas.
- Extrair o histograma de orientação do gradiente para cada célula, que consiste no agrupamento das direções das arestas num histograma com Nb dimensões onde tal processo é feito agrupando células em blocos, repedindo este processo para todos os blocos.
- Agrupar as células dos histogramas em blocos e normalizar cada bloco, que são a energia total dos histogramas de grupos de células. Blocos de menor tamanho aumentam a atenuação para variações de iluminação local.
- O vetor de características final é obtido concatenando os histogramas de cada célula.

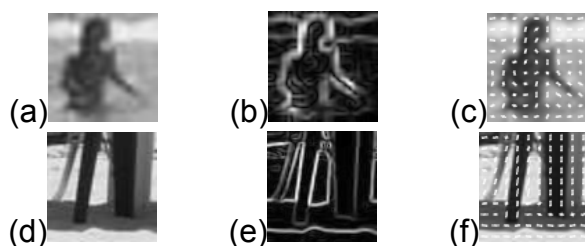
A Figura 9 mostra o resultado obtido ao se aplicar a máscara, Figura 8, onde na esquerda é exibido o valor absoluto do gradiente x, na imagem central o valor absoluto do gradiente y e na direita a magnitude do gradiente.

Figura 8 – Máscara de gradientes utilizadas no descritor HOG.



Fonte – <https://www.learnopencv.com/histogram-of-oriented-gradients/>

Figura 9 – Uma amostra positiva (a) e negativa (d) da base de dados com os respectivos gradientes, magnitude (b) (e) e orientação (d) (f).



Fonte – O autor.

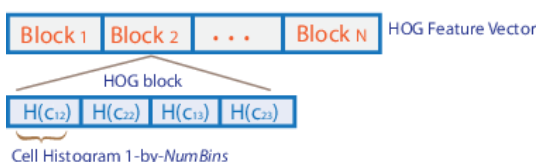
Já a Figura 10 mostra como é realizado o agrupamento de células em blocos e a Figura 11 mostra como são gerados os histograma a partir do conjunto de blocos.

Figura 10 – Conjunto de células do descritor HOG.



Fonte – <https://www.mathworks.com/help/vision/ref/extrachogfeatures.html>

Figura 11 – Geração os histogramas para o conjunto de blocos.

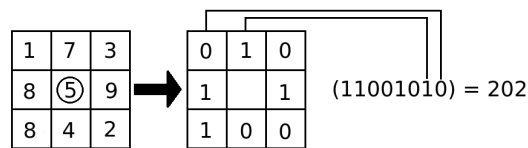


Fonte – <https://www.mathworks.com/help/vision/ref/extrachogfeatures.html>

2.5.3 LBP (*Local binary patterns*)

Inicialmente proposto por Ojala *et al.* (OJALA; PIETIKÄINEN; HARWOOD, 1996), esse descritor de característica foi aplicado com sucesso para diferentes problemas que envolvem algum reconhecimento, como detecção de faces e também de pessoas. Para extrair características de uma imagem o processo utilizado é semelhante ao mostrado na Figura 12.

Figura 12 – Um exemplo de características LBP.



Fonte – O autor

Uma característica LBP, levando em consideração um determinado pixel em uma coordenada (x, y) , em uma imagem, pode ser calculado utilizando um conjunto de P amostras, espaçadas igualmente em um raio R , e com o centro no *pixel* (x, y) . O *pixel* central é fixado e seu valor é então comparado com todos os *pixels* da vizinhança, o resultado é codificado como um rotulo binário, 1 (UM) quando o valor for maior que o *pixel* central e 0 (ZERO) caso contrário. Posteriormente esses valores binários são convertidos para um valor inteiro. Para converter o conjunto de valores, obtidos anteriormente, é feita uma multiplicação dos valores obtidos por uma conjunto de valores assim como mostrado abaixo:

$$LBP_{RN}(x, y) = \sum_{k=0}^{N-1} S(gc - gp)2^k, \quad (2.11)$$

$$S(gc - gp) = \begin{cases} 1, & \text{se } gp \geq gc \\ 0, & \text{se } gp < gc \end{cases}$$

onde gc é o pixel central e gp são os *pixels* vizinhos. E é utilizando essa fórmula que são obtidos os valores inteiros (AWAD; HASSABALLAH, 2016). Um padrão LBP é considerado uniforme quando o padrão binário tiver no máximo duas transições, transições estas de 0 (ZERO) para 1 (UM) ou de 1 (UM) para 0 (ZERO). Quando o histograma é criado todos os padrões não uniformes são atribuídos em um único espaço, enquanto os uniformes possuem um compartimento para cada padrão.

2.5.4 Características de Gabor

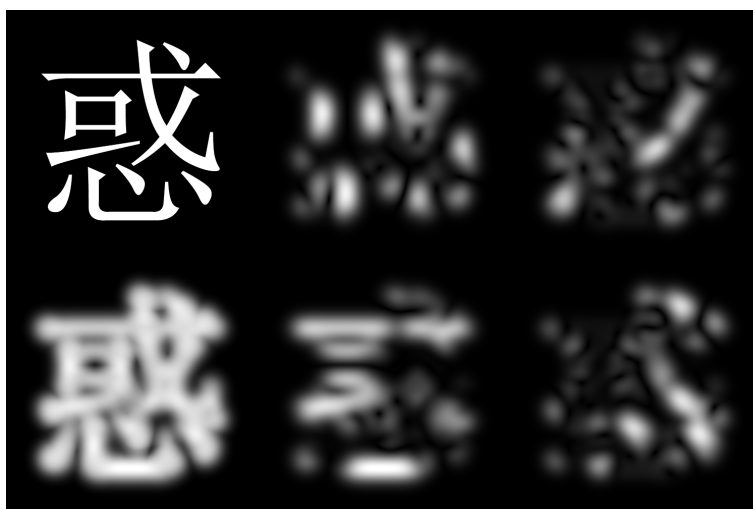
Essas características são baseadas no filtro de Gabor(JAIN; FARROKHNIA, 1991). As características de Gabor, e outros recursos baseado no mesmo, já são uti-

lizados há bastante tempo, seu uso pode ser motivado por propriedades importantes como à invariância à iluminação, rotação, escala e translação (KAMARAINEN; KYRKI; KALVIAINEN, 2006). E tais recurso já foram utilizados para o reconhecimento de conhecimento de caracteres (WANG; DING; LIU, 2005) e reconhecimento de expressões faciais (DENG et al., 2005). Um filtro de Gabor 2D de uma imagem tem os seguintes parâmetros:

- Tamanho da janela do filtro (w);
- Desvio padrão gaussiano (σ);
- Orientação da normal para as linhas paralelas da função de Gabor (θ);
- Comprimento de onda do fator sinusoidal(λ);
- Razão de aspecto espacial (γ);
- Fase (ψ).

Variando os parâmetros acima é possível melhorar as propriedades visuais de uma imagem, como por exemplo a frequência espacial e a orientação. O vetor característica de Gabor é obtido através da aplicação de um filtro sobre a imagem e transformando a resposta em um vetor unidimensional. Desta forma, o vetor resultante é do mesmo tamanho da imagem original.

Figura 13 – Resultado da aplicação do filtro de Gabor em uma imagem.



Fonte – https://en.wikipedia.org/wiki/Gabor_filter

2.5.5 Momentos de Zernike

Os polinômios de Zernike foram primeiramente propostos em 1934 por Frits Zernike. Estes momentos são conhecidos por serem invariantes à rotação e podem ser modificados para serem estáveis a escala e translação (TEH; CHIN, 1988).

O momento complexo 2D de ordem n e repetição m , sobre uma função de intensidade de imagem $f(x, y)$, é

$$Z_{n,m} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) V_{nm}(x, y)^* \quad (2.12)$$

$n - |m|$ até, $|m| \leq n$

onde $x^2 + y^2 \leq 1$. As coordenadas da imagem original são dimensionadas para um disco unitário $x^2 + y^2 = 1$, e $*$ denota o complexo conjugado. O polinômio de Zernike $V_{nm}(x, y)$ é definido como

$$V_{nm}(x, y) = V_{nm}(r, \theta) = R_{nm}(r) e^{jm\theta}$$

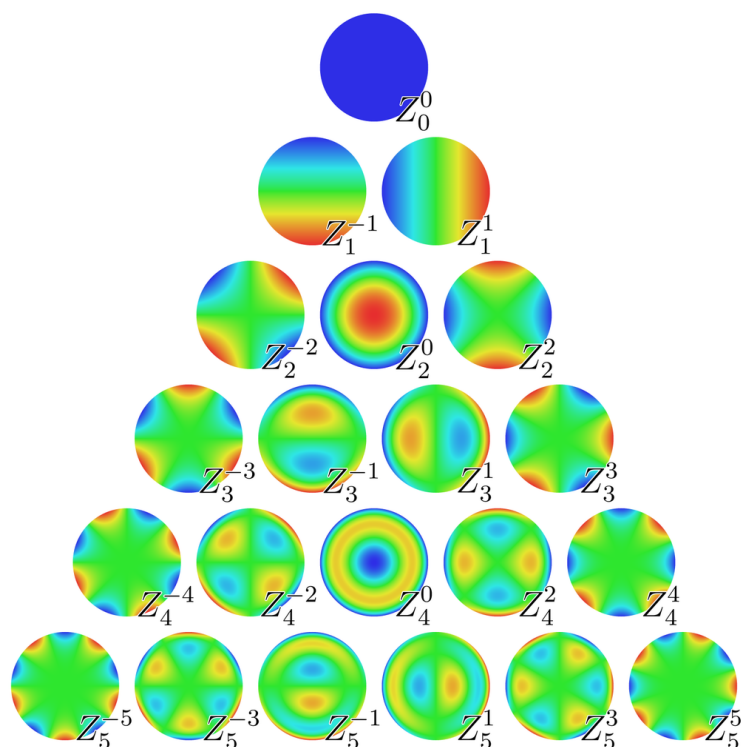
$$R_{nm}(r) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} r^{n-2s} \quad (2.13)$$

Onde j é a unidade imaginária e $0 \leq r \leq 1$. A transformação da imagem normalizada (x, y) para o domínio do círculo unitário (r, θ) é dado por

$$r = \sqrt{x^2 + y^2}, \quad \theta = \tan^{-1} \left(\frac{y}{x} \right) \quad (2.14)$$

Esses momentos são utilizados para descrever textura e formato das imagens.

Figura 14 – Polinômios de Zernike.

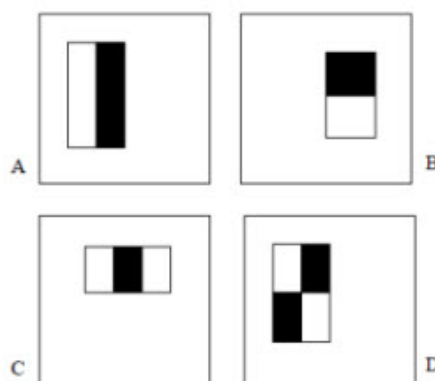


Fonte – https://en.wikipedia.org/wiki/Zernike_polynomials

2.5.6 Características de Haar

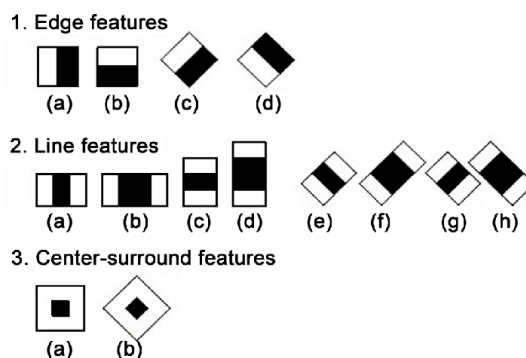
Características de Haar-like foram melhoradas em (LIENHART; MAYDT, 2002), inicialmente eram utilizados apenas quatro tipos de retângulos (VIOLA; JONES, 2001). Para extração de característica desse descritor o processo é realizado com o uso de máscaras retangulares, estas máscaras possuem uma região branca e outra região preta, o resultado é obtido após realizar a subtração entre a soma dos valores dos *pixels* da região branca e a soma dos *pixels* região preta. Tal operação mostra a variação luminosa que ocorre entre regiões da imagem, Figura 15 e 16 mostram os dois tipos de filtro, o melhorado e o primeiro criado.

Figura 15 – Retângulos utilizados na versão antiga das características de Haar.



Fonte – (VIOLA; JONES, 2001)

Figura 16 – Retângulos utilizados na versão atual das características de Haar.



Fonte – (LIENHART; MAYDT, 2002)

2.6 Classificação de Banhista

Os classificadores são utilizados geralmente quando temos amostras de entrada e queremos saber a que classe ou grupo uma determinada amostra pertence. Em tarefas de classificação, a meta é encontrar uma função a partir dos dados de treinamento que possa ser utilizada para prever um rótulo ou valor que caracterize um novo exemplo, com base nos valores de seus atributos de entrada (CARVALHO et al., 2011). Neste trabalho foram utilizados três classificadores: floresta randômica, máquinas de vetores de suporte e cascata de classificadores, que serão explanados de forma mais clara adiante e também a técnica análise de componentes que é utilizada para a redução de dimensionalidade dos dados PCA (*Principal Component Analysis*)

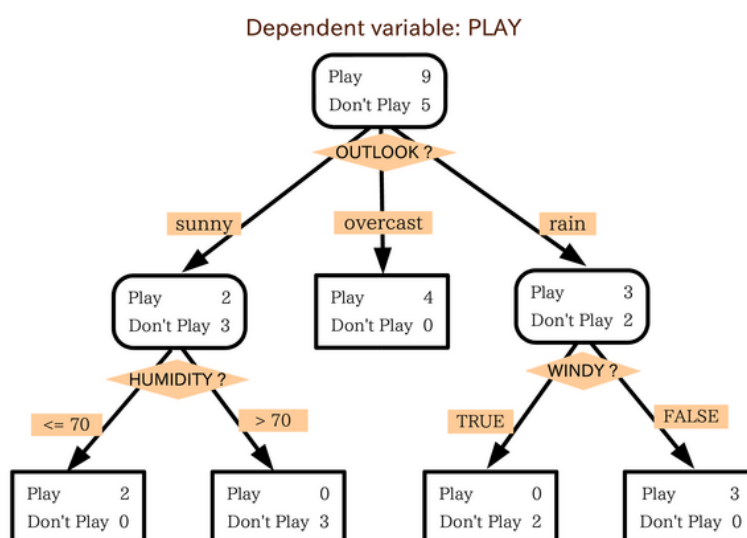
2.6.1 Floresta Randômica (*Random forest*)

Floresta randômica é um classificador que possui seu treinamento supervisionado, e para classificar utiliza árvores de decisão, que utiliza um modelo para constru-

ção desse tipo de classificador (SUTHAHARAN, 2016).

Uma árvore de decisão usa o modelo de dividir e conquistar para solucionar um problema de decisão. A instância do problema é geralmente dividida em problemas mais simples, de forma que uma avaliação de uma condição pode solucionar a instância menor daquele problema. No final, as instâncias menores são combinadas, para formar a árvore, e assim o problema é solucionado. A principal força dessa abordagem vem pela sua capacidade de dividir a instância do problema em instâncias menores ou subespaço, e cada instância menor pode ser resolvida utilizando diferentes modelos (CARVALHO et al., 2011).

Figura 17 – Árvore de decisão.



Fonte – [http : //artedosdados.blogspot.com/2014/11/uma-arvore-de-decisao-e-predicao-dos.html](http://artedosdados.blogspot.com/2014/11/uma-arvore-de-decisao-e-predicao-dos.html)

Uma árvore de decisão, como mostrada na Figura 17, também pode ser como um grafo acíclico direcionado em que cada nó ou é um nó de divisão, podendo ter dois ou mais sucessores, ou um nó de folha. Nesse contexto o nó folha é rotulado como uma função e são consideradas apenas valores da variável alvo nos exemplos que chegam em um nó folha. Já um nó de divisão contém um teste condicional usando como base os valores do atributo. Geralmente essas condições só levam em conta um único atributo e valores no domínio desse atributo (CARVALHO et al., 2011).

Para escolher os atributos que devem ser usados como “filtro” é utilizada a entropia. Entropia mede a aleatoriedade de uma variável aleatória. Para cada nó analisado no processo de decisão, o atributo que mais reduz a aleatoriedade da variável alvo será o escolhido para dividir os dados e em alguns casos atributos podem não ser utilizados pois não tem valor bastante significativo (CARVALHO et al., 2011).

A floresta aleatória pode ser definida pelas seguintes etapas (SUTHAHARAN, 2016):

- Gerar, através de uma reamostragem, um conjunto de amostras retiradas da amostra original utilizando uma seleção aleatória com reposição, seus atributos também são escolhidos de forma aleatória.
- Desenvolver para cada amostra, no conjunto obtido anteriormente, uma árvore de decisão.
- Desenvolver uma árvore única que é constituída pelas árvores obtidas na etapa anterior.

2.6.2 Máquina de vetores de suporte (*support vector machine* – SVM)

A Máquina de vetores de suporte (*support vector machines* – SVM) tem sua base na teoria do aprendizado estatístico, que foi desenvolvida por Vapnik (1995), que estabelece uma série de princípios, que se seguidos, obtém classificadores de boa capacidade de generalização (CARVALHO et al., 2011).

Há diversos tipos de SVM, dentre eles a SVM de *kernel* linear e SVM de *kernel* não linear. Ao se utilizar um *kernel* diferente, a divisão do hiperplano é modificada de acordo com o *kernel* escolhido.

Uma SVM linear tenta definir uma fronteira linear usando dados que são linearmente separáveis. Utilizando um conjunto X , como conjunto de treinamento, e $Y = \{-1, 1\}$ que seriam os rótulos da classe de cada elemento de X representado como y . O conjunto X é considerado linearmente separável se é possível dividir os objetos, sendo classificados por -1 e 1 , com um hiperplano. A Equação 2.15 mostra a equação do hiperplano.

$$h(x) = w \cdot x + b \quad (2.15)$$

onde $w \cdot x$ é o produto escalar entre os vetores \mathbf{w} e \mathbf{x} , \mathbf{w} e X é o vetor normal ao hiperplano descrito e $\frac{b}{\|\mathbf{w}\|}$ corresponde à distância do hiperplano em relação a à origem, com $b \in \mathfrak{R}$.

A Equação 2.15 pode ser usada para dividir o conjunto em duas regiões de forma que quando um objeto pertence a primeira região a função retorna -1 e 1 quando de outra região. Uma função sinal $g(x) = \text{sgn}(h(x))$ pode ser empregada para obtenção dos classificadores, mostrada na Equação 2.16:

$$g(x) = \text{sgn}(\text{sgn}(h(x))) = \begin{cases} +1 & \text{se } w \cdot x + b > 0 \\ -1 & \text{se } w \cdot x + b < 0 \end{cases} \quad (2.16)$$

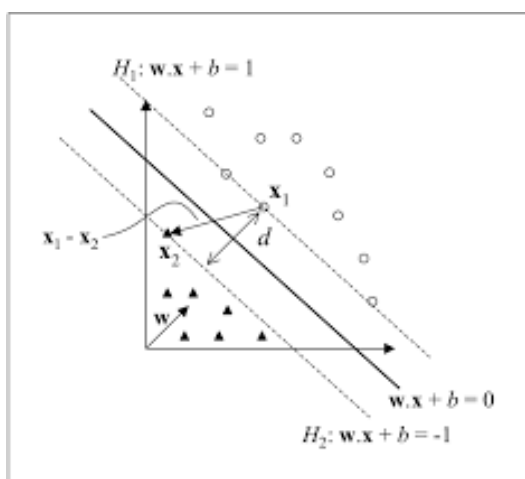
Usando a função $h(x)$ é possível obter um número infinito de hiperplanos equivalentes. Define-se o hiperplano canônico em relação ao conjunto X como aquele em que w e b são escalados de forma satisfazer a Equação 2.17:

$$|w \cdot x_i + b| = 1. \tag{2.17}$$

Essa forma implica na Equação 2.18 é uma forma resumida.

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall (x_i, y_i) \in X \tag{2.18}$$

Figura 18 – Ilustração de hiperplanos canônicos e separador.



Fonte – (LORENA; CARVALHO, 2007)

Utilizando manipulações algébricas simples usando os pontos sobre os hiperplanos h_1 e h_2 , como mostrado na Figura 18, permitem deduzir que a maximização da margem que separa objetos de cada classe pode ser obtido aplica a minimização de $\|w\|$ e por fim temos o seguinte problema de otimização:

$$\underset{w,b}{\text{Minimizar}} \frac{1}{2} \|w\|^2 \tag{2.19}$$

Como a função objetivo é convexa e os pontos satisfazem as restrições formam um conjunto convexo, esse problema possui um único mínimo global e esse tipo de problema pode ser solucionados utilizando introduzindo uma função lagrangiana. Após a aplicação dessa função e derivando L relação a b e w e igualando a 0 e aplicando as devidas substituições temos:

$$\underset{\alpha}{\text{Maximizar}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{2.20}$$

$$\text{Com as restrições : } \begin{cases} \alpha_i \geq 0, \forall i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (2.21)$$

No caso para classificar dados que não são linearmente separáveis, a função do hiperplano é modificada, assim seu produto é modificado para uma função de *kernel*, um exemplo comum é o uso de função de função gaussiana de base radial (AGGARWAL, 2014).

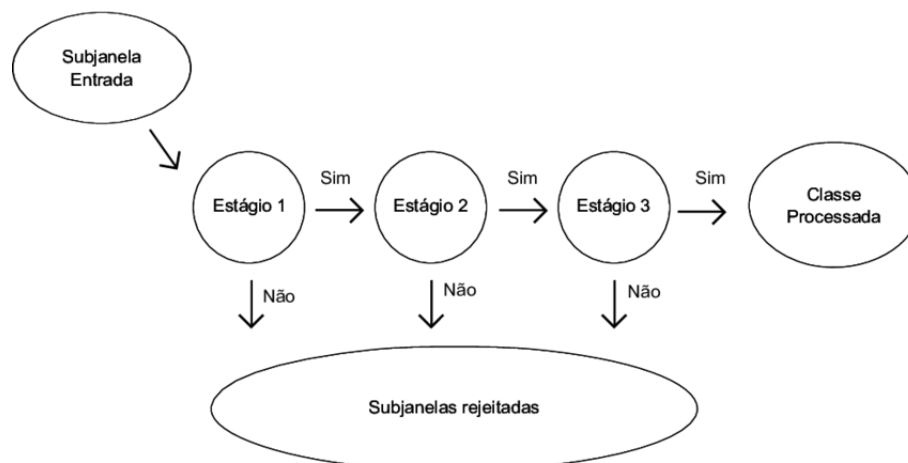
2.6.3 Cascata de classificadores (*Train cascade*)

Os classificadores em cascata podem ser vistos como uma árvore degenerada, o conceito de cascata vem da ideia que ao se analisar um novo exemplo o mesmo será avaliado para primeiro classificador da cascata, o considerado mais fraco, e em caso se o objeto for classificado como verdadeiro será passado para o próximo classificador, que é mais “forte” que classificador anterior.

A principal ideia desse é fazer com que os estágios iniciais do classificador descartem o uma grande quantidade de regiões que não contém o objeto que deve ser detectado, tal etapa visa diminuir qualquer processamento de áreas que não são regiões de interesse.

Para o desenvolvimento da cascata é necessário um conjunto de metas de detecção e desempenho. Parâmetros como o número de cascata deve ser grande o suficiente para garantir uma elevada taxa de detecção de objetos e gerar uma baixa ocorrência de falsos positivos. A Figura 19 mostra como é o funcionamento de uma cascata de classificadores.

Figura 19 – Classificador em cascata.



Fonte – <https://www.researchgate.net/figure/Figura-44-Cascata-de-Classificadores-adaptado-de-Viola-e-Jones-3fig9305766897>

2.6.4 Análise de componentes principais - PCA

Análise de componentes principais (PCA) é aplicado quando se deseja reduzir a dimensionalidade dos dados de entrada ou usados na classificação. Para se obter o resultado esperado, primeiro os dados devem ser organizados utilizando um matriz $M \times N$, onde as linhas representam o número de observações e as colunas os valores para cada observação, posteriormente é obtida a variância para cada uma das observações, por fim, deve ser aplicada a decomposição em autovalores dessa matriz de variância (DAWSON, 2016).

O que o PCA realmente obtêm é o quanto uma variável da amostra contribui para a obtenção de uma classificação dos dados. Ao se aplicar essa técnica, para uma base de dados onde a maioria das variáveis da uma amostra não representa uma real “quantitatividade” ela pode ser escolhida depois da seleção, pois esse método busca um valor mínimo de representatividade para a base (SHLENS, 2014).

O valor da representatividade escolhido é indicado como uma variável de entrada, então o conjunto de parâmetros obtido é aquele cujo a representatividade é maior ou igual ao valor tido como entrada. O vetor obtido como resposta pode possuir o tamanho igual ou menor que a base dados, dependendo das limitações do algoritmo e característica dos dados.

3 Classificação de banhista

A detecção de objetos ou pessoas, na maioria dos casos, exige um processo de segmentação de imagens, com o objetivo de reduzir a quantidade de objetos para a classificação, seguida de uma etapa de extração de informação, que transforma as regiões de interesse em informação mais simples e útil para o algoritmo de classificação, e por fim a etapa de classificação, que a partir das informações extraídas tenta classificar a que classe do problema o objeto fornece tem a maior semelhança.

Esse trabalho tem como foco a etapa de extração de características e classificação, visto que, mesmo utilizando formas mais simples de geração de regiões de interesse, como o uso de uma janela deslizante, pode classificar a área que foi apresentada como entrada. Para usar utilizar a combinação de descritores e classificador corretamente, deve ser adicionada uma etapa de segmentação ou técnica que forneça regiões de interesse para a etapa de extração de característica e tal etapa não foi utilizada neste trabalho. Na Subseção 3.1 serão apresentados características do problema e a escolha dos descritores e Subseção 3.2 a escolha dos classificadores.

3.1 Extração de características

A extração de característica, utilizando descritores, é uma etapa fundamental para que um banhista seja classificado corretamente pelo classificador. Neste trabalho, o descritor deve ter baixo custo computacional, visando o uso em uma máquina comum e sem auxílio de processamento em GPU. Os descritores escolhidos para avaliação foram: HOG, LBP, momentos de Hu, momentos de Zernike e características de Gabor. Cada um desses descritores possui características importantes, tais como, invariância a escala, rotação e invariância a mudanças na iluminação, que quando combinados podem se tornar mais robustos para o problema avaliado. A escolha desses descritores se deu pelo fato de seu uso em trabalhos clássicos da literatura ou trabalhos mais recentes envolvendo a tarefa de detecção de objetos ou para detecção de pessoas.

Como apresentado na Seção 2, alguns dos descritores requerem parâmetros de entrada, de forma que a mudança desses parâmetros podem sofrer modificações de acordo com características do problema analisado. Na Seção 4 serão apresentadas quais as variações e quais métodos foram aplicados para selecionar cada um desses parâmetros e resultado obtido. Dois problemas básicos que podem ser citados é o espaço de cor e tamanho da imagem analisada.

Figura 20 – Processo de extração de características.



Fonte – O autor.

Para cada um desses descritores, o processo de extração de características é obtido de forma bastante semelhante, como mostrado na Figura 20. O algoritmo recebe um recorte da figura original obtido por uma etapa de segmentação, que deve ser redimensionado para um tamanho padrão, de 25 *pixels* de altura por 29 *pixels* de largura, e quando necessários, parâmetros de entrada que pode variar de acordo com o descritor. Depois da etapa inicial são obtidos os valores que representam a imagem de entrada, tais valores são obtidos como vetor de escalares e para cada tipo de descritor ocorre uma variação no seu tamanho, como serão apresentados na Seção 4.

Figura 21 – Imagem da praia de Boa Viagem.



Fonte – O autor.

A Figura 21 foi adquirida para a base de dados e a partir dessa imagem é possível

observar a variação de distância que cada pessoa se encontra com relação a ponto de captura da imagem. No caso, podemos ver como o problema de escala e a posição de cada banhista, onde o mesmo pode ser encontrado de pé, com uma inclinação para direita ou esquerda ou deitado. E a iluminação pode variar decorrendo do momento do dia da captura ou devido a fatores climáticos.

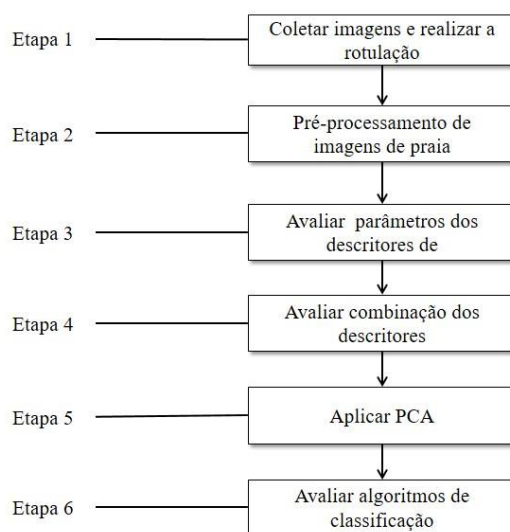
3.2 Classificação de Banhista

Os classificadores podem ser escolhidos para um determinado uso de acordo com as características do problema e isso reflete substancialmente no custo computacional e na qualidade da detecção ou classificação dos objetos analisados. De acordo com a separação espacial dos dados, um classificador linear como o perceptron ou SVM com *kernel* linear podem resolver o problema e possuem processamento rápido. Porém, geralmente produzem resultados inferiores aos algoritmos não lineares. Já classificadores como SVM com *kernel* não linear e *Random Forest* são geralmente utilizados devido aos bons resultados e são algoritmos relativamente mais rápidos que outros algoritmos não lineares.

Neste trabalho, o tempo é um fator importante e devido a tal motivo devemos avaliar classificadores que atendem a tal requisito, tais como SVM e *Random Forest*, como apresentado na Seção 4. Um fator que influenciou bastante na escolha desse classificadores foi apresentada em (FERNÁNDEZ-DELGADO et al., 2014), e com base com outros trabalhos de detecção de pessoas apresentados na Seção 1. E como citado na Seção 2, para se obter melhores resultados foi aplicada a técnica PCA, que como já mencionado reduz a dimensionalidade dos dados de entrada para os classificadores, como pode ser visto na Seção 2

A Figura 22 apresenta algumas das etapas realizadas nesse trabalho.

Figura 22 – Fluxograma do trabalho.



Fonte – O autor.

1. Coletar imagens e realizar a rotulação. A rotulação de banhistas é necessária para avaliação dos algoritmos para teste de validação do modelo do classificador, sendo ela a responsável por analisar seus resultados.
2. Pré-processamento de imagens de praia: Realizar o processamento necessário visando o funcionamento correto do descritor, para o projeto as imagens foram convertidas para tons de cinza.
3. Avaliar parâmetros dos algoritmos de extração de características, efetuando variação de parâmetros, o tempo necessário para realizar a extração e aplicação das técnicas para reduzir a dimensionalidade do vetor resultante. Os descritores selecionados foram: HOG, LBP, momentos de Hu, momentos de Zernike e filtro de Gabor.
4. Realizar a combinação de descritores 2 a 2.
5. Aplica a técnica de redução de dimensionalidade PCA.
6. Avaliar algoritmos de Classificação. Os classificadores selecionados foram: SVM com *kernel* linear, SVM com *kernel* radial e *Random forest*.

O objetivo deste projeto é desenvolver um estudo sobre classificação de banhistas em imagens de praias e tem como principal motivação que nenhum outro estudo,

segundo o conhecimento dos autor, apresentou um sistema ou analise sobre esse tipo de classificação visando mitigar o problema de risco de afogamento ou de alguma outra situação perigosa na água.

4 Avaliação experimental

Nesta Seção serão apresentadas as métricas, base de dados, seleção de parâmetros e metodologia dos experimentos que foram utilizadas para avaliação e comparação e seleção dos algoritmos utilizados na Seção 5. Os experimentos visam selecionar a melhor configuração, com relação ao conjunto de parâmetros, e a melhor combinação, visando avaliar a taxa de reconhecimento dado o conjunto de descritores e classificadores utilizados e também avaliando o resultado com a aplicação e sem aplicação da técnica PCA, de redução de dimensionalidade.

4.1 Base de dados

A base de dados foi obtida a partir de imagens capturadas na praia de Boa Viagem em 2016 e 2017, essas imagens foram obtidas a partir de postos de guarda-vidas. A base de dados consiste de mais de 150 imagens de resolução de 1083×720 e foram capturadas durante diferentes momentos do dia. Após a aquisição das imagens, toda a base de dados foi devidamente rotulada utilizando a ferramenta online *LabelMe* (RUSSELL et al., 2008). A Figura 23 mostra como é o resultado da imagem após a rotulação, que consiste em criar um polígono na área ao redor da pessoa na imagem.

Figura 23 – Imagem rotulada.



Fonte – O autor.

Após a rotulação da imagem, foi gerado um arquivo com extensão *xml*, que contém o conjunto de marcações para cada um desses polígonos e utilizando esse arquivo é possível criar uma máscara binária, como mostrado na Figura 24, para extração das regiões de interesse na imagem. Com a utilização da máscara é efetuada uma operação e (*and*) com a imagem original em tons de cinza, e a partir dessa operação é extraído um recorte retangular contendo a região circunscrita para cada um dos polígonos criado na etapa de rotulação. Para classe negativa o processo é semelhante mas utiliza um imagem binária gerada por uma imagem diferente da original e dentro dos recortes gerados ainda são escolhidos elementos com alta textura, imagens com apenas água e areia são descartadas.

Figura 24 – Imagem binária.



Fonte – O autor.

A base de dados utilizada para avaliação dos algoritmos é composta por um conjunto de segmentos (recortes) extraídos a partir da imagem previamente rotuladas. No total foram obtidos 2000 segmentos, 1000 contendo pessoas, que podem estar dentro ou fora da água, e 1000 contendo objetos com textura alta ou formas que podem ser confundidas com pessoas durante a etapa de segmentação ou geração de regiões de interesse. Ainda na classe negativa, que não contém pessoas, podem ser encontradas extensões de recifes, cadeiras, pássaros, entre outros objetos encontrados nas imagens. A Figura 25 mostra algumas das imagens que pertencem ao conjunto avaliado.

Figura 25 – Um exemplo de imagens positivas (a) (b) (c) e negativas (d) (e) (f).



Fonte – O autor.

4.2 Seleção de parâmetros

Como já mencionado anteriormente, a maioria dos descritores e classificadores possuem alguns parâmetros de entrada esses por sua vez influenciam no resultado final. Para selecionar esses valores foram experimentos em uma base de dados separada, que foi obtida inicialmente nas primeiras análises do trabalho, essa base de dados possui um total de 688 segmentos, sendo 392 da classe positiva e 296 da classe negativa, a classe positiva pode ser entendida como blocos da imagem onde as pessoas aparecem, podendo ser de corpo inteiro até ombros e cabeça.

O primeiro parâmetro obtido foi o tamanho padrão que deveria ser utilizado para o redimensionamento dos fragmentos, este foi calculado utilizando a média do tamanho dos fragmentos e com base nesse resultado o valor obtido foi 25 *pixels* de altura por 29 *pixels* de largura. Este valor foi utilizado para todos os experimentos e a partir de ponto foi utilizado como valor padrão.

Para obter o valor dos parâmetros dos descritores foi escolhido o classificador *Random Forest*, cada configuração foi avaliada 50 vezes utilizando a seleção aleatória de 80% da base de dados como treinamento e o restante para teste. A partir desse experimento, os melhores parâmetros foram selecionados com base na taxa média obtida de detecção do classificador e descritor correspondente. O descritor momentos de Hu foi o único que não participou da análise pois não tinha parâmetros de entrada como os demais avaliados.

4.2.1 Experimento P

O experimento P visa encontrar os parâmetros, para cada um dos descritores, visando obter a melhor taxa de reconhecimento para os dados avaliados. Estes parâmetros foram selecionados experimentalmente em uma base de dados separada, contendo 392 imagens da classe positiva e 296 imagens da classe negativa. Cada configuração foi avaliada 50 vezes através da seleção aleatória de 80% da base de dados para treinamento e o restante para teste. Os melhores parâmetros foram selecionados com base na taxa média de detecção do classificador com o descritor correspondente e o classificador utilizado nesse experimento foi o *Random Forest*.

4.2.2 Experimento P-1

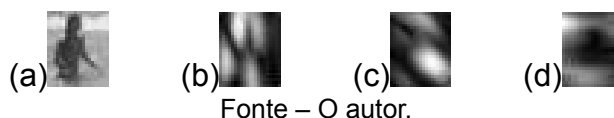
O experimento P-1 visa encontrar os melhores parâmetros para o descritor características de Gabor. Para o descritor em questão os parâmetros que foram escolhidos para a avaliação foram o w e θ . A variação dos valores é indicada abaixo:

- w : 2, 5, 10, 20, 40, 80 e 120 *pixels*.

- θ : 0° , 45° e 90° .

Com a variação dos parâmetros acima, as propriedades visuais de uma imagem, como frequência espacial e orientação, pode ser aprimorado, conforme ilustrado na Figura 26.

Figura 26 – Filtro de Gabor com janelas 5×5 aplicada (a) com três diferentes orientações: 0° (b), 45° (c) e 90° (d).



4.2.3 Experimento P-2

O experimento P-2 visa encontrar os melhores parâmetros para o descritor momentos de Zernike. O parâmetro escolhido para avaliação do Momentos de Zernike foi ordem (n) que com os seguintes valores: 5, 10, 15, 20, 25 e 40.

4.2.4 Experimento P-3

O experimento P-3 visa encontrar os melhores parâmetros para o descritor HOG. Para a avaliação do HOG o parâmetros escolhidos foram: tamanho da célula e tamanho do bloco. E a variação dos valores desses parâmetros pode ser mostrada abaixo:

- Tamanho da célula: 4×4 , 8×8 e 16×16 .
- Tamanho do bloco: 1×1 , 2×2 , 3×3 e 4×4 .

4.2.5 Experimento P-4

O experimento P-4 visa encontrar os melhores parâmetros para o descritor LBP. Como já mencionado anteriormente devido ao tamanho reduzidos das imagens, 25×29 pixels, raio e vizinhança foram fixados em 1 e 8 respectivamente. O parâmetros escolhidos para a avaliação foram: tamanho da célula, tipo de normalização e tipo de interpolação de pixels vizinhos

- Tamanho da célula: 4×4 , 8×8 e 16×16 .
- Normalização do histograma: nenhuma e L2.
- Interpolação de pixels vizinhos: linear e mais próximo.

4.2.6 Combinação de descritores e classificadores

O experimento do tipo C visa avaliar a combinação de descritores com diferentes classificadores, dado que o experimento P já obteve quais parâmetros devem ser utilizados para cada um dos classificadores. A base de dados utilizada contém um total de 2000 imagens, sendo 1000 da classe positiva e 1000 da classe negativa. Os resultados apresentados são a média entre 50 repetições com separação aleatória do conjunto de dados, onde 80% é treinamento (800 imagens por classe) e 20% de teste (200 imagens por classe). Cada combinação de características é testada com classificadores *Random Forest* e SVM (*kernel* linear e radial).

Para os experimentos realizados os vetores de característica de cada descritor foram combinados dois a dois, usando os parâmetros que obtiveram melhor resultados nos experimentos de tipo P.

O conjunto de descritores utilizados para o experimento C-1, C-2 e C-3 foram: momentos de Hu, momentos de Zernike, Gabor, HOG e LBP. O descritor, com nome encontrado na linha, é combinado com o outro encontrado na coluna. Assim as diagonais indicam apenas o descritor sem aplicar nenhuma combinação. Os parâmetro utilizando são os obtidos nos experimentos de tipo P.

No experimento C-4 foi avaliado utilizando o algoritmo do OpenCv que fornece um pequeno conjunto de *scripts*, disponível na documentação do OpenCV ([OPENCV, 2018](#)), para criar um classificador em cascada utilizando o descritor LBP ou Haar.

4.2.7 Experimento C-1

O experimento C-1 consistem em utilizar os melhores resultados de cada um dos experimentos P e utilizar o classificador Floresta aleatória.

Configuração:

PCA: `weka.filters.unsupervised.attribute.PrincipalComponents -R 0.95 -A 5 -M -1`

Random Forest: `weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1`

4.2.8 Experimento C-2

O experimento C-2 visa avaliar os resultados obtidos da combinação dos descritores em pares, utilizando os parâmetros que obtiveram o melhor resultado nos experimentos do tipo P, com o classificador SVM com *kernel* linear e também avalia o resultado obtido com o uso do PCA.

Configuração:

PCA: `weka.filters.unsupervised.attribute.PrincipalComponents -R 0.95 -A 5 -M -1`
SVM *kernel* linear: `weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1`

4.2.9 Experimento C-3

O experimento C-3 visa avaliar os resultados obtidos da combinação dos descritores em pares, utilizando os parâmetros que obtiveram o melhor resultado nos experimentos do tipo P, com o classificador SVM com *kernel* radial e também avalia o resultado obtido com o uso do PCA.

Configuração:

PCA: `weka.filters.unsupervised.attribute.PrincipalComponents -R 0.95 -A 5 -M -1`
SVM *kernel* radial: `weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1`

4.2.10 Experimento C-4

O experimento C-4 consistem em utilizar os classificadores já implementados pelo OPENCV para detecção de objetos, será utilizado o classificador em cascata utilizando os descritores de características de Haar e LBP. Os parâmetros utilizados são os mesmo escolhidos para os outros testes do tipo C, quantidade de iterações, tamanho da base de dados. A quantidade de classificadores em cascata é indicada como parâmetro, foram avaliados utilizando os valores entre 7 e 10, mas caso o classificador obtenha um resultado satisfatório de classificação o treinamento irá convergir e assim encerrar o algoritmo com uma quantidade menor de classificadores que a indicada.

4.3 Ambiente experimental

O ambiente utilizando para análise das imagens do trabalho foram desenvolvidos utilizando a linguagem de programação Python 3.6, o ambiente de desenvolvimento MatLab® e para classificação o Weka (HALL et al., 2009). Abaixo segue o que foi utilizado para cada etapa da pesquisa:

- Para dividir a base de dados, em treino e teste, visando o desenvolvimento do modelo do classificador no OpenCv (*Open Source Computer Vision Library*) (BRADSKI et al., 2000) foi utilizada a biblioteca Scikit-learn. Essa biblioteca é gratuita e possui soluções de aprendizagem de máquina para Python (PEDREGOSA et al., 2011).

- Para criar o modelo para classificação, utilizando o descritor de haar e LBP, foi utilizada a biblioteca OpenCv que código aberto e oferece uma ampla gama de algoritmos de visão computacional e aprendizagem de máquina.
- O ambiente MatLab[®] foi utilizado para gerar os arquivos com extensão csv utilizado para avaliação e seleção de parâmetros dos descritores utilizados, todos os algoritmos utilizados são facilmente encontrados distribuídos pela própria comunidade do MatLab[®].
- Na avaliação dos classificadores, *Random Forest* e SVM, e com e sem a aplicação do PCA foi utilizado o software Weka (*Waikato Environment for Knowledge Analysis*) (HALL et al., 2009) que possui uma vasta gama de algoritmos prontos para o uso, como os classificadores utilizados nesse trabalho.

4.4 Métricas de validação

Para mediar a exatidão dos dados classificados, as seguintes medições foram aplicadas:

- Medida de acurácia: esse valor é dado em forma de porcentagem e indica qual a quantidade de objetos que foi corretamente classificada. A Fórmula 4.4 mostra como calcular esse valor.

$$a = \frac{VP + VN}{TE} \quad (4.1)$$

onde VP é Verdadeiro Positivo e representa um objeto da classe positiva que foi classificado corretamente, VN é Verdadeiro Negativo e representa um objeto da classe negativa que foi classificado corretamente e somatório desses dois resultados é dividido pelo número total de exemplos.

- Tempo de extração médio que indica o tempo gasto para extração de característica de cada descritor, esse tempo foi obtido para extração característica da base de dados de 2000 imagens.

Os parâmetros do PCA são apresentados na configuração dos experimentos do tipo C. Para os experimentos também foi utilizado o teste-t (ZIMMERMAN; ZUMBO, 1993), esse teste é comumente usado para determinar se as médias de dois grupos são iguais entre si.

5 Resultados

Este Capítulo mostra os resultados obtidos durante os experimentos apresentados na Seção 4, e será dividido em quatro Seções. A primeira Seção mostra os resultados obtidos no experimento do tipo P, que foi executado com o objetivo de avaliar parâmetros para os descritores de característica. A segunda Seção apresenta os resultados obtidos nos experimentos do tipo C, esses experimentos têm como objetivo avaliar, dado que os parâmetros já foram escolhidos, a melhor combinação entre descritores e classificadores utilizados neste trabalho. Os resultados apresentados nessa Seção utilizaram o descritores: Momentos de Hu, Momentos de Zernike, Gabor, LBP, HoG e Haar. Os classificadores avaliados foram: SVM, *Random Forest* e cascata de classificadores. A terceira Seção apresenta o tempo de extração médio de cada descritor, após a seleção de parâmetros, e a quarta Seção apresenta uma discussão sobre os resultados.

5.1 Resultados do experimentos do tipo P

Está Seção apresenta os resultados obtidos durante a seleção de parâmetros dos descritores. Os parâmetros escolhidos para a avaliação são apresentados e logo em seguida seus resultados. Todos os resultados apresentados são as médias obtidas após 50 execuções, como mencionado no Seção 4 para o experimentos do tipo P.

5.1.1 Características de Gabor

Para o descritor em questão os parâmetros que foram escolhidos para a avaliação foram o w e θ . A variação dos valores é indicada abaixo:

- w : 2, 5, 10, 20, 40, 80 e 120 *pixels*.
- θ : 0° , 45° e 90° .

A Tabela 1 apresenta todos os resultados obtidos para cada uma das configurações do parâmetros. Para esse experimento o melhor resultado obtido foi a taxa de reconhecimento de 85,14% foi obtida com $w = 40$ e $\theta = 45^\circ$.

Tabela 1 – Experimento P-1. Taxa de reconhecimento (%) durante a seleção do parâmetro do filtro de Gabor (orientação e tamanho da janela).

	2	5	10	20	40	80	120
0°	79,22	76,33	76,68	80,50	70,36	64,90	64,46
45°	79,38	78,72	77,82	81,82	85,14	79,44	71,88
90°	71,55	76,43	74,70	67,77	63,60	62,03	61,79

5.1.2 Momentos de Zernike

O parâmetro escolhido para avaliação do momentos de Zernike foi ordem (n) que com os seguintes valores: 5, 10, 15, 20, 25 e 40. Como pode ser visto pela Tabela 2, a taxa de reconhecimento de 87,42% foi obtida quando a $n = 10$, onde n é a ordem.

Tabela 2 – Experimento P-2. Taxa de reconhecimento para diferentes ordens do momento de Zernike.

Ordem:	5	10	15	20	25	40
Acurácia (%):	87,04	87,42	86,97	86,27	85,80	84,38

5.1.3 HOG

Para a avaliação do HOG o parâmetros escolhidos foram: tamanho da célula e tamanho do bloco. E a variação dos valores desses parâmetros pode ser mostrada abaixo:

- Tamanho da célula: 4×4 , 8×8 e 16×16 .
- Tamanho do bloco: 1×1 , 2×2 , 3×3 e 4×4 .

As taxas de reconhecimento associadas com os parâmetros escolhidos são apresentadas na Tabela 3. Devido ao tamanho reduzidos das imagens, resolução de (29×25) , algumas combinações de células e blocos não foram possíveis, deixando espaços em branco na tabela. Os melhores resultados foram alcançados com células 8×8 e blocos 2×2 .

Tabela 3 – Experimento P-3. Taxa de reconhecimento (%) para diferentes tamanhos de células (linhas) e blocos (colunas) para o descritor HOG.

	1x1	2x2	3x3	4x4
4x4	78,22	79,77	79,29	81,01
8x8	79,57	81,39		
16x16	69,36			

5.1.4 LBP

Na escolha desses parâmetros, foi levado em conta o tamanho reduzidos das imagens que era de 25 *pixels* de altura por 29 *pixels* de largura , assim, o raio e vizinhança foram fixados em 1 e 8 respectivamente. O parâmetros escolhidos para a avaliação foram: tamanho da célula, tipo de normalização e tipo de interpolação de *pixels* vizinhos

- Tamanho da célula: 4×4, 8×8 e 16×16.
- Normalização do histograma: nenhuma e L2.
- Interpolação de *pixels* vizinhos: linear e mais próximo.

Como pode ser observado na Tabela 4, a melhor configuração do descritor LBP foi formada por células 8×8 sem normalização e interpolação para o pixel mais próximo. Além disso, a taxa de reconhecimento varia principalmente pelo tamanho da célula.

Tabela 4 – Experimento P-4. Taxa de reconhecimento (%) para diferentes parâmetros do LBP.

Normalização:	L2	Nenhum	L2	Nenhum
Interpolação:	Linear	Linear	Mais próximo	Mais próximo
célula 4x4:	87,56	87,32	87,88	87,79
célula 8x8:	89,54	89,19	89,77	89,82
célula 16x16:	84,41	84,42	84,41	85,05

5.1.5 Tamanho do vetores de características

Após a etapa de seleção de parâmetros, o método de extração de características obteve os vetores de características que serão as entradas dos classificadores:

1. **Hu** – Vetor de características com 7 escalares.
2. **Zernike** – Vetor de características com 66 escalares, correspondente as características de Zernike de ordem 10.
3. **Gabor** – Vetor de características com 725 escalares, obtidos através da convolução de uma imagem com o filtro de Gabor. O filtro é formado por uma janela de 40×40 pixels com inclinação de 45°. Outros parâmetros deste filtro são mantidos com seus valores padrões.
4. **HOG** – Vetor de características com 144 escalares são obtidos através da divisão da imagem em células 8×8 e agrupando histogramas em blocos de 2×2 para normalização.

5. **LBP** – Vetor de características com 531 escalares, obtidos considerando 8-vizinhança e células 8×8 .

O tamanho de cada vetor de característica acima e a sua combinação de pares são resumidos na Tabela 5.

Tabela 5 – Tamanho dos vetores de característica e suas combinações em pares

	Hu	Zernike	Gabor	HOG	LBP
Hu	7				
Zernike	73	66			
Gabor	732	791	725		
HOG	151	210	869	144	
LBP	538	597	1256	675	531

5.2 Resultados do experimentos do tipo C

Nesta seção serão apresentados os resultados obtidos nos experimentos de classificação, Tipo C, de cada combinação de descritores, levando em conta apenas os parâmetros que obtiveram o melhor resultado nos experimentos do tipo P. São analisados os classificadores Floresta Aleatória e SVM (*kernel* linear e radial). Também, será avaliada a aplicação da técnica de redução de característica PCA. Para cada experimento, a taxa de reconhecimento com os dois classificadores acima é apresentada com e sem o uso do PCA nas Tabelas 6, 7, 8, 9, 10 e 11. Foi realizado um teste-T estatístico com o intervalo de confiança de 95%. O teste estatístico é usado para comparar o melhor resultado numérico em cada tabela com os demais resultados pertencentes a mesma tabela. Os resultados destacados, em negrito, são estatisticamente melhores do que outras taxas de acurácias na mesma Tabela.

5.2.1 Floresta aleatória

Esse experimento utiliza cada um dos descritores, de forma individual e combinados em pares, com o classificador Floresta aleatória. O melhor resultado obtido nesse experimento foi de 88,03% com a combinação dos descritores HOG e LBP.

Tabela 6 – Experimento C-1. Taxa de reconhecimento (%) na base de testes usando classificador Floresta aleatória.

	Hu	Zernike	Gabor	HOG	LBP
Hu	65,61				
Zernike	78,94	75,59			
Gabor	73,26	74,14	72,68		
HOG	81,22	82,82	80,86	80,89	
LBP	85,70	86,44	85,42	88,03	85,66

O próximo resultado é do experimento que utiliza cada um dos descritores, de forma individual e combinados em pares, com o classificador Floresta aleatória com aplicação da técnica PCA. O melhor resultado obtido nesse experimento foi de 84,49% com a combinação dos descritores Gabor e LBP.

Tabela 7 – Experimento C-2. Taxa de reconhecimento (%) na base de testes usando classificador *Random Forest* e PCA

	Hu	Zernike	Gabor	HOG	LBP
Hu	63,19				
Zernike	73,95	70,97			
Gabor	54,52	76,84	54,67		
HOG	80,63	81,96	81,95	80,68	
LBP	82,56	81,01	84,49	83,08	82,32

5.2.2 SVM com *kernel* linear

Esse experimento utiliza cada um dos descritores, de forma individual e combinados em pares, com o classificador SVM com *kernel* linear. O melhor resultado obtido nesse experimento foi de 84,42% com a combinação dos descritores Zernike e HOG seguindo da combinação Zernike e LBP com 83,80%.

Tabela 8 – Experimento C-2. Taxa de reconhecimento (%) na base de testes usando classificador SVM com *kernel* linear

	Hu	Zernike	Gabor	HOG	LBP
Hu	50,39				
Zernike	77,64	78,22			
Gabor	67,02	68,95	67,13		
HOG	80,04	84,42	66,64	80,04	
LBP	82,13	83,80	75,02	82,59	82,13

A Tabela 9 resultados para o experimento que utiliza cada um dos descritores, de forma individual e combinados em pares, com o classificador SVM com *kernel* linear utilizando o PCA. O melhor resultado obtido nesse experimento foi de 85,14% com a combinação dos descritores HOG e LBP.

Tabela 9 – Experimento C-2. Taxa de reconhecimento (%) na base de testes usando classificador SVM com *kernel* linear e PCA

	Hu	Zernike	Gabor	HOG	LBP
Hu	55,36				
Zernike	75,31	66,97			
Gabor	54,88	74,50	54,74		
HOG	79,82	81,56	82,26	79,7	
LBP	82,43	82,7	84,58	85,14	82,03

5.2.3 SVM com *kernel* radial

Esse experimento utiliza cada um dos descritores, de forma individual e combinados em pares, com o classificador SVM com *kernel* radial. O melhor resultado obtido nesse experimento foi de 88,94% com a combinação dos descritores HOG e LBP como mostrado na Tabela 10.

Tabela 10 – Experimento C-3. Taxa de reconhecimento (%) na base de testes usando classificador SVM com *kernel* radial

	Hu	Zernike	Gabor	HOG	LBP
Hu	50,38				
Zernike	55,92	55,28			
Gabor	50,00	50,00	50,00		
HOG	74,72	63,23	50,00	74,92	
LBP	88,59	72,62	50,00	88,94	88,54

A Tabela 11 mostra os resultados para o experimento que utiliza cada um dos descritores, de forma individual e combinados em pares, com o classificador SVM com *kernel* radial e aplicando o PCA. O melhor resultado obtido nesse experimento foi de 90,31% com a combinação dos descritores HOG e LBP.

Tabela 11 – Experimento C-3. Taxa de reconhecimento (%) na base de testes usando classificador SVM com *kernel* radial e PCA

	Hu	Zernike	Gabor	HOG	LBP
Hu	61,91				
Zernike	77,48	74,00			
Gabor	55,26	69,86	55,29		
HOG	83,19	86,68	76,19	83,44	
LBP	88,46	89,18	86,89	90,31	88,44

5.2.4 Classificadores em cascata

Para esse experimento o melhor resultado obtido foi o de 50,70% , como pode ser visto na Tabela 12, para o descritor de características de Haar. O máximo de cascatas gerada foi de 4 estágios de cascata para o descritor LBP e 6 para o descritor de Haar.

5.3 Tempo de extração médio

O tempo de extração de cada uma das características de cada descritor é apresentado na Tabela 13. Essa tabela apresenta em sua diagonal o tempo médio de execução, na extração de informações, de um único descritor de característica. Para esse

Tabela 12 – Experimento C-4. Taxa de reconhecimento (%) na base de testes usando classificador em cascata

LBP	Haar
48,75	50,70

experimento foram utilizadas as imagens da base dados, como já mencionado anteriormente essas imagens possuem resolução de 29×25 pixels e estão em tons de cinza e utilizando os parâmetros que obtiveram melhor resultado no experimentos do tipo P. Tal etapa foi executada em computador com processador i3 e 4GB de memória ram.

Tabela 13 – Tempo de extração médio de características em milissegundos de uma imagem em tons de cinza com dimensões de 29×25 pixels

	Hu	Zernike	Gabor	HOG	LBP
Hu	0,5				
Zernike	19,5	19,0			
Gabor	3,6	22,1	3,1		
HOG	1,6	20,1	4,3	1,2	
LBP	1,3	19,8	3,9	2,0	0,8

5.4 Discussão

Inicialmente foram realizados experimentos com o intuito de selecionar os parâmetros que melhor obtivessem resultados, taxa de classificação com a utilização de um classificador, para tal foram escolhidos os 5 descritores de características, como já mencionando na seção anterior, e adicionalmente o descritor de característica de Haar e LBP, utilizando o classificador em cascata do OpenCv.

Durante a essa etapa foi observado quais os parâmetros obtiveram os melhores resultados, como mostrados nas Tabelas de experimentos do tipo P.

Os resultados obtidos mostram que o descritor LBP apresentou melhor resultado para uma com o tamanho da célula de 8×8 mostrando que essa combinação não requeria um ajuste mais fino ou o maior como os demais avaliados. Já com relação ao tipo de interpolação de pixel, a que melhor apresentou o melhor resultado foi a mais próximo esse método tem uma precisão menor mas pode ser calculado mais rápido, reduzindo ainda mais o tempo sem a aplicação da normalização, que por interpolação linear.

Para capturar informações espaciais em grande escala, o tamanho das células deve ser aumentado. Para a resolução das imagens analisadas, o tamanho de 8×8 se mostrou o mais próximo para o formato dos objetos da classe positiva. Quando o

tamanho de bloco é grande a capacidade de suprimir alterações de iluminação locais é reduzida assim a configuração 2×2 obteve o melhor resultado.

Quanto maior a ordem mais a quantidade de momentos, com isso o tempo para computação também tende a crescer. Ao se obter um bom resultado com a ordem utilizando um valor menor significa um baixo custo computação para o calculo dos momentos.

Para o filtro de Gabor, o melhor resultado obtido possui a configuração com o comprimento de onda de 40 e orientação de 45°

Para os experimentos utilizando o classificador em cascata do OpenCv os resultados tiveram um desempenho abaixo da média se comparado com os demais classificadores utilizados. Mas tal fato pode ter sido causado devido a configuração da base, no trabalho original (VIOLA; JONES, 2001) somava um total de mais de 20 mil imagens, divididas entre a classe positiva e negativa, e contava com 38 elementos em sua cascata de classificadores. Outro fator que deve ser observado é que para a base de dados utilizada o treinamento do classificador em cascata obteve 4 estágios de cascata para o descritor LBP e 6 para o descritor de Haar, mesmo quando foi indicado que o valor mínimo sendo 7 ou mais para o número de cascatas. Durante o treinamento desse classificador, utilizando o descritor LBP o treinamento foi relativamente rápido, com uma média de 15 segundos, enquanto o com características de Haar obteve uma média de 9 minutos. Com uma base de dados maior, o que consequentemente poderia fazer o número de cascatas aumentar, o treinamento poderia durar de dias até semanas.

Com base nos resultados apresentados, pode ser observado que todo os resultados que obtiveram uma maior taxa de classificação utilizava uma combinação dos descritores HOG ou LBP, tal fato indica que para esse tipo de problema o uso desses descritores possuem o maior poder de representatividade.

Na maioria dos experimentos realizados, o uso do PCA melhorou o resultado da classificação, tal fato mostrou que para a configuração e condições utilizadas, o uso de tal técnica apresenta uma vantagem significativa. E como pode ser observado, os experimentos mostraram que o classificador SVM e utilizando *kernel* radial em conjunto com as características obtidas a partir descritores HOG e LBP com redução de características utilizando PCA, apresentaram os melhores resultados, obtendo 90,31% de acurácia.

Como já mencionado na Seção 2, cada um desses descritores possuem certas características ou invariâncias que quando combinadas se tornam mais robustas para descrever de forma mais precisas características relacionada a esse tipo de classificação. Um fator de grande importância que pode ser observado foi o resultado que

obteve a melhor combinação, a combinação de descritores HOG e LBP, também são os mais rápidos, nos experimentos realizados os foi obtida uma média de extração de 2ms por imagem. Outro fato que deve ser observado é que tal combinação ser torna uma boa escolha para detecção de tempo real, devido a não influência da variação de luminosidade. A SVM se mostrou o classificador que melhor resultado para esse tipo de problema e o uso do PCA também melhorou os resultados encontrado para esse classificado.

6 Conclusão e trabalhos futuros

Este trabalho aborda a detecção de banhistas a partir de imagens de praia e para tal, foi feita uma análise do uso de descritores de características em imagens candidatas a serem banhistas ou não. Os descritores estudados foram combinados em pares. O vetor de características formado pelos descritores individuais e suas combinações foram utilizados como entrada para os classificadores SVM, com *kernel* linear e radial, e Floresta aleatória. Por fim, comparamos os resultados com o classificador clássico de pessoas proposto por Viola Jones (VIOLA; JONES, 2001).

Para esse trabalho ser concluído foi necessária a criação de uma base de dados com imagens de praia. Como esse problema ainda não tinha sido avaliado ou estudado na literatura, essa foi a primeira base criada para a partir da mesma criar avaliações de algoritmos e outras análises que geralmente são utilizadas em tarefas de detecção e classificação de banhistas.

No trabalho o melhor resultado, taxa de reconhecimento média, é de 90,31% com 1,43% de desvio padrão. Isto foi obtido com o classificador SVM com kernel radial e utilizando o PCA e com a utilizando a combinação de descritores HOG e LBP.

A partir dos experimentos realizado ficou constatado que o uso da combinação de descritores e classificadores possui resultados promissores para a classificação de banhista em imagens de praia. O classificador SVM se mostrou um dos melhores para esse tipo de classificação e os descritores, HOG e LBP, apresentam um tempo de extração baixo, mostrando atender os requisitos do projeto com relação ao custo computacional. Sendo assim essa abordagem se mostrou inovadoras pois nenhum outro projeto lida com esse tipo de classificação. O PCA, técnica utilizada para reduzir a dimensionalidade do vetor de características também apresentou melhorias significativas no resultado.

6.1 Trabalhos Futuros

Para trabalhos futuros, uma ideia inicial seria avaliar como o pré-processamento das imagens poderia influenciar no resultado final e se tal modificação traz consigo alguma melhoria. Também é interessante incluir uma análise com uma base de dados maior, dividindo entre dias de sol, nublados e com chuva.

Outro ponto de interesse seria a adição de mais classificadores ao conjunto analisado e também a avaliação de técnicas como aprendizado profundo. Tal motivação é causada pelo uso em grande escala dessa técnica associada a as tarefas de

classificação de forma bastante abrangente.

Por fim, adicionar um conjunto maior de análises estatísticas para os dados avaliados, dessa forma será obtido uma visão ainda mais precisa com relação aos dados e o comportamento mais detalhado dos classificadores. Fazer uma análise multi-objetivo fazendo um contraponto entre maior acurácia, falso positivos e tempo de processamento.

6.2 Contribuições

Esse trabalho surgiu como um projeto de iniciação científica, com o mesmo orientador, e no decorrer do tempo foi mostrando resultados promissores. No ano de 2017 os dados obtidos no trabalho foram apresentados na JEPEX (Jornada de Ensino, Pesquisa e Extensão) realizada na UFRPE. Ainda no mesmo ano foi apresentado um trabalho (SILVA et al., 2017) no 29th ICTAI (*International Conference on Tools with Artificial Intelligence*) que foi realizada em Boston nos Estados Unidos e teve um trabalho publicado no IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) e outro artigo (GOMES; GONÇALVES, 2018) no XIII WVC (Workshop de Visão Computacional) realizado na Universidade Federal do Rio Grande do Norte (UFRN) e teve um artigo publicado nos anais do evento.

Referências

- AGGARWAL, C. C. *Data classification: algorithms and applications*. [S.l.]: CRC Press, 2014. Citado na página 37.
- ARDIYANTO, I.; MIURA, J. Partial least squares-based human upper body orientation estimation with combined detection and tracking. *Image and Vision Computing*, Elsevier, v. 32, n. 11, p. 904–915, 2014. Citado 2 vezes nas páginas 16 e 18.
- AWAD, A. I.; HASSABALLAH, M. *Image feature detectors and descriptors: foundations and applications*. [S.l.]: Springer, 2016. v. 630. Citado na página 29.
- BIRCHFIELD, S. Elliptical head tracking using intensity gradients and color histograms. In: IEEE. *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. [S.l.], 1998. p. 232–237. Citado na página 17.
- BRADSKI, G. et al. The opencv library (2000). *Dr. Dobb's Journal of Software Tools*, 2000. Citado na página 50.
- CARVALHO, A. et al. Inteligência artificial—uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, 2011. Citado 3 vezes nas páginas 33, 34 e 35.
- CHANG, C.-Y.; FU, S.-Y. Image classification using a module rbf neural network. In: IEEE. *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on*. [S.l.], 2006. v. 2, p. 270–273. Citado na página 15.
- CHANG, Y.-W.; LIN, C.-J. Feature ranking using linear svm. In: *Causation and Prediction Challenge*. [S.l.: s.n.], 2008. p. 53–64. Citado na página 15.
- DAWSON, J. *Analysing quantitative survey data for business and management students*. [S.l.]: Sage, 2016. Citado na página 38.
- DENG, H.-B. et al. A new facial expression recognition method based on local gabor filter bank and pca plus lda. *International Journal of Information Technology*, v. 11, n. 11, p. 86–96, 2005. Citado na página 30.
- ESHEL, R.; MOSES, Y. Homography based multiple camera detection and tracking of people in a dense crowd. In: IEEE. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.], 2008. p. 1–8. Citado na página 17.
- FERNÁNDEZ-DELGADO, M. et al. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 3133–3181, 2014. Citado na página 41.
- GOMES, R. B.; GONÇALVES, L. M. G. Xiii workshop de visão computacional: anais do 13º workshop de visão computacional realizado na ufrn. Editora da UFRN, 2018. Citado na página 62.
- GONZÁLEZ, A. et al. 3d-guided multiscale sliding window for pedestrian detection. In: *lbPRIA*. [S.l.: s.n.], 2015. p. 560–568. Citado na página 19.

GONZALEZ, R. C.; WOODS, R. *Processamento digital de imagens. tradução: Cristina Yamagami e Leonardo Piamonte*. [S.l.]: São Paulo: Pearson Prentice Hall, 2010. Citado 5 vezes nas páginas 19, 21, 22, 23 e 25.

GUAN, Y.; HUANG, Y. Multi-pose human head detection and tracking boosted by efficient human head validation using ellipse detection. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 37, p. 181–193, 2015. Citado na página 19.

HALL, M. et al. The WEKA data mining software: an update. *SIGKDD Explorations*, v. 11, n. 1, p. 10–18, 2009. Citado 2 vezes nas páginas 50 e 51.

HSU, F.-C.; GUBBI, J.; PALANISWAMI, M. Head detection using motion features and multi level pyramid architecture. *Computer Vision and Image Understanding*, Elsevier, v. 137, p. 38–49, 2015. Citado 2 vezes nas páginas 16 e 18.

HU, M.-K. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, IEEE, v. 8, n. 2, p. 179–187, 1962. Citado na página 26.

HU, R. et al. Robust head-shoulder detection using a two-stage cascade framework. In: IEEE. *Pattern Recognition (ICPR), 2014 22nd International Conference on*. [S.l.], 2014. p. 2796–2801. Citado na página 18.

JAIN, A. K.; FARROKHNIYA, F. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, Elsevier, v. 24, n. 12, p. 1167–1186, 1991. Citado na página 29.

KAMARAINEN, J.-K.; KYRKI, V.; KALVIAINEN, H. Invariance properties of gabor filter-based features-overview and applications. *IEEE Transactions on image processing*, IEEE, v. 15, n. 5, p. 1088–1099, 2006. Citado na página 30.

KEPSKI, M.; KWOLEK, B. Person detection and head tracking to detect falls in depth maps. In: *ICCVG*. [S.l.: s.n.], 2014. p. 324–331. Citado na página 19.

LIENHART, R.; MAYDT, J. An extended set of haar-like features for rapid object detection. In: IEEE. *Image Processing. 2002. Proceedings. 2002 International Conference on*. [S.l.], 2002. v. 1, p. I–I. Citado 2 vezes nas páginas 32 e 33.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado na página 36.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. [S.l.], 1999. v. 2, p. 1150–1157. Citado na página 19.

MAJI, S.; BERG, A. C.; MALIK, J. Classification using intersection kernel support vector machines is efficient. In: IEEE. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.], 2008. p. 1–8. Citado 2 vezes nas páginas 15 e 19.

MARIN, J. et al. Random forests of local experts for pedestrian detection. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 2592–2599. Citado 2 vezes nas páginas 15 e 19.

OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, Elsevier, v. 29, n. 1, p. 51–59, 1996. Citado na página 29.

OPENCV. *OpenCV: Color conversions*. 2015. Acesso em: 15/05/2018. Disponível em: <http://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html>. Citado na página 24.

OPENCV. *Cascade Classifier Training*. 2018. Citado na página 49.

PE, G. *Comitê que monitora incidentes com tubarões investiga relato de ataque no Recife*. 2017. Disponível em <<http://g1.globo.com/pernambuco/noticia/comite-que-monitora-incidentes-com-tubaroes-investiga-relato-de-ataque-no-recife.ghtml>>. Acessado em 02/07/2017. Citado na página 13.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 50.

PERNAMBUCO, D. de. *Ataques de tubarão: 25 anos de medo nas praias*. 2017. Acesso em: 04/08/2018. Disponível em: <http://www.diariodepernambuco.com.br/app/noticia/vida-urbana/2017/11/21/interna_vidaurbana,731668/ataques-de-tubarao-25-anos-de-medo-nas-praias.shtml>. Citado na página 13.

ROUGIER, C. et al. Monocular 3d head tracking to detect falls of elderly people. In: IEEE. *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. [S.I.], 2006. p. 6384–6387. Citado na página 18.

RUSSELL, B. C. et al. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, Springer, v. 77, n. 1, p. 157–173, 2008. Citado na página 44.

SABHARA, R. K.; LEE, C.-P.; LIM, K.-M. Comparative study of hu moments and zernike moments in object recognition. *SmartCR*, v. 3, n. 3, p. 166–173, 2013. Citado na página 19.

SANDE, K. E. Van de et al. Segmentation as selective search for object recognition. In: IEEE. *Computer Vision (ICCV), 2011 IEEE International Conference on*. [S.I.], 2011. p. 1879–1886. Citado na página 19.

SHLENS, J. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014. Citado na página 38.

SILVA, R. L. da et al. Detecting people from beach images. In: IEEE. *Tools with Artificial Intelligence (ICTAI), 2017 IEEE 29th International Conference on*. [S.I.], 2017. p. 636–643. Citado na página 62.

SMITH, L. I. *A tutorial on principal components analysis*. [S.I.], 2002. Citado na página 19.

SUTHAHARAN, S. Machine learning models and algorithms for big data classification. *Integr. Ser. Inf. Syst*, Springer, v. 36, p. 1–12, 2016. Citado na página 34.

- TEH, C.-H.; CHIN, R. T. On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 10, n. 4, p. 496–513, 1988. Citado na página 31.
- TU, J.; ZHANG, C.; HAO, P. Robust real-time attention-based head-shoulder detection for video surveillance. In: IEEE. *Image Processing (ICIP), 2013 20th IEEE International Conference on*. [S.l.], 2013. p. 3340–3344. Citado 2 vezes nas páginas 17 e 19.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.], 2001. v. 1, p. I–I. Citado 4 vezes nas páginas 32, 33, 59 e 61.
- VIOLA, P.; JONES, M. J.; SNOW, D. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, Springer, v. 63, n. 2, p. 153–161, 2005. Citado na página 16.
- WANG, S.; ZHANG, J.; MIAO, Z. A new edge feature for head-shoulder detection. In: IEEE. *Image Processing (ICIP), 2013 20th IEEE International Conference on*. [S.l.], 2013. p. 2822–2826. Citado na página 18.
- WANG, X.; DING, X.; LIU, C. Gabor filters-based feature extraction for character recognition. *Pattern recognition*, Elsevier, v. 38, n. 3, p. 369–379, 2005. Citado na página 30.
- XIE, D.; DANG, L.; TONG, R. Video based head detection and tracking surveillance system. In: IEEE. *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*. [S.l.], 2012. p. 2832–2836. Citado 3 vezes nas páginas 16, 18 e 19.
- ZENG, C.; MA, H. Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In: IEEE. *Pattern Recognition (ICPR), 2010 20th International Conference on*. [S.l.], 2010. p. 2069–2072. Citado 2 vezes nas páginas 15 e 17.
- ZIMMERMAN, D. W.; ZUMBO, B. D. Rank transformations and the power of the student t test and welch t'test for non-normal populations with unequal variances. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, Canadian Psychological Association, v. 47, n. 3, p. 523, 1993. Citado na página 51.