



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CAMPUS PAU DOS FERROS**

TALISON FERNANDES COSTA

**RECONHECIMENTO FACIAL COM
HAAR CASCADE E KNN**

PAU DOS FERROS

2018

TALISON FERNANDES COSTA

**RECONHECIMENTO FACIAL COM
HAAR CASCADE E KNN**

Monografia (Graduação em Engenharia de Computação) submetido à Coordenação do Curso de Bacharelado em Engenharia de Computação, da Universidade Federal Rural do Semi-Árido, como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Prof. Msc. Marco Diego Aurelio Mesquita

PAU DOS FERROS

2018

TALISON FERNANDES COSTA

RECONHECIMENTO FACIAL COM HAAR CASCADE E KNN

Monografia (Graduação em Engenharia de Computação) submetido à Coordenação do Curso de Bacharelado em Engenharia de Computação, da Universidade Federal Rural do Semi-Árido, como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Computação

Aprovada em 20/10/2017

BANCA EXAMINADORA

Prof. Msc. Marco Diego Aurelio Mesquita (Orientador)
Universidade Federal Rural do Semi-Árido

Prof. Dr. Rodrigo Soares Semente
Universidade Federal Rural do Semi-Árido

Prof. Dr.^a Náthalee Cavalcanti de Almeida Lima
Universidade Federal Rural do Semi-Árido

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

F838r Fernandes Costa, Talison.
Reconhecimento Facial com Haar Cascade e Knn /
Talison Fernandes Costa. - 2017.
53 f. : il.

Orientador: Marco Diego Aurelio Mesquita.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Engenharia de
Computação, 2017.

1. Reconhecimento de padrões. 2. Detecção de
face. 3. KNN. 4. Haar Cascade. I. Diego Aurelio
Mesquita, Marco , orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

*Aos Meus Pais,
Tios, Noiva, Amigos
e Professores!*

AGRADECIMENTOS

Agradeço primeiramente a Deus, ser não explicado que de alguma maneira nos força a continuar insistindo. A minha família que diante das limitações sempre mim apoiaram. A minha noiva, Wellida Damares, que por várias vezes tem sido compreensível nas minhas muitas ausências. Ao meu orientador professor Marco Diego que tem me ajudado e ensinado a fazer ciência, agradeço pela compreensão, pela amizade, pelas críticas, pelo incentivo e pela disponibilidade. Para mim, além de um grande mestre um exemplo de humildade e caráter. A todos aqueles que compõem a UFERSA onde diretamente ou indiretamente contribuí para a realização deste trabalho, o meu muito obrigado.

Não poderia deixar de citar algumas pessoas que ao longo desta caminhada tem sido meus grandes apoiadores: Geraldo Fernandes (meu pai), Auxiliadora Maria da Costa (minha mãe), tia Doroteia Fernandes, Manoel Nobrega Fernandes Junior (grande amigo), Maria Francisca Soares (futura sogra) e Edilson José Soares (futuro sogro). Além destes, meus irmãos Tiago, Gracinha, Tassio, Talismar, Matheus, e Debora. Por fim, estendo este agradecimento a minha família de modo geral e especificamente ao meu tio Francisco Fernandes, que de modo especial tem me ajudado nessa caminhada do conhecimento e acreditado em mim.

RESUMO

Este trabalho traz uma proposta de um algoritmo de reconhecimento facial simples baseada nas principais abordagens de detecção e reconhecimento de padrões discutido na literatura. Inicialmente, é feita uma revisão geral dos principais métodos de detecção de face e reconhecimento de padrões, destacando as vantagens e desvantagens de cada uma delas. A análise de cada metodologia ajudou a decidir quais estratégias utilizar para projetar um algoritmo simples que forneça bons resultados. Dentro do escopo de detecção de face, foi utilizado uma abordagem em cascada denominada Haar Cascade. Para a etapa de reconhecimento de padrões, propomos um algoritmo baseado em K-NN. Apesar de simples, o K-NN apresenta alguns problemas de decisão que é corrigido inserindo critérios menos genéricos que na abordagem tradicional. Os resultados mostraram que apesar da simplicidade do classificador é possível reconhecer faces com alta probabilidade de acerto.

Palavras-chave: Reconhecimento de padrões. Detecção de face. K-NN. Haar Cascade.

ABSTRACT

This work presents a proposal for a simple facial recognition algorithm in the main approaches to detection and pattern recognition discussed in the literature. Initially, it performed a general review of the main methods of face detection and pattern recognition, highlighting the advantages and disadvantages of each. An analysis of each methodology helps a process for the application of a simple algorithm that provides good results. Within the scope of face detection, a cascade approach (Haar Cascade) was discussed in the work of Viola and Jones. For a pattern recognition stage, we propose an algorithm based on knn. Although simple, K-NN has some decision problems and is corrected by inserting less generic criteria than in the traditional approach. The results showed that despite the simplicity of the classifier it is possible to recognize faces with high probability of acceptance.

Keywords: Pattern recognition. Face detection. K-NN. Haar Cascade.

LISTA DE TABELAS

1	Resposta para faces presente do conjunto de treinamento.	p. 44
2	Resposta para faces não presentes no conjunto de treinamento.	p. 45
3	Resposta com limiar de 0.06	p. 45

LISTA DE FIGURAS

1	Projeção horizontal e vertical de uma imagem com rosto.	p. 18
2	Funcionamento do algoritmo de detecção de face baseado em RNA, (ROWLEY, 1996).	p. 23
3	Resultado da aplicação sucessiva em três níveis de decomposição, da transformada wavelet utilizando a base de Haar. (a) Imagem original, (b) remanescente do primeiro nível, (c) remanescente do segundo nível e (d) remanescente do terceiro nível. Para perceber o efeito da aplicação da transformada wavelet na resolução, o tamanho das imagens foi aumentado.	p. 26
4	A soma dos pixels dentro do retângulo D pode ser calculada a partir das quatro matrizes de referência. O valor da imagem integral no ponto 1 é a soma dos pixels dentro de retângulo A . No ponto 2 é $A + B$, em 3 $A + C$, finalmente em 4 $A + B + C + D$ ou $D = 4 + 1 - (3 + 2)$, (VIOLA e JONES, 2001).	p. 26
5	Características em imagens integrais.	p. 27
6	A primeira e segunda característica selecionada pelo AdaBoost, (VIOLA e JONES, 2001)	p. 29
7	K vizinhos mais próximos.	p. 34
8	Fluxo de execução do sistema de reconhecimento facial.	p. 37
9	Descritores faciais.	p. 37
10	Conjunto de imagens de treinamento.	p. 42
11	Conjunto de imagens de teste.	p. 43
12	Curvas com a média das distâncias mostradas nas Tabelas 1 e 2.	p. 46
13	Curvas com a média das distâncias mostradas nas Tabelas 1 e 2 para K de 1 até 5.	p. 47
14	Curvas de distâncias para as 5 primeiras imagens de teste.	p. 48

LISTA DE SIGLAS

RNA	Rede Neural Artificial
K-NN	K Nearest Neighbor
IA	Inteligência Artificial
CCD	Charge-Coupled Device
KLT	Karhunen-Loève Transform
DCT	Discrete Cosseno Transform
PCA	Principal Component Analysis
MOC	Modelo Oculito de Markov
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis

SUMÁRIO

1	INTRODUÇÃO	p. 13
1.1	Visão computacional	p. 13
1.2	Visão geral do trabalho	p. 14
1.3	Objetivos e Justificativa	p. 15
2	DETECÇÃO DE FACE COM HAAR CASCADE	p. 16
2.1	Técnicas de detecção de face	p. 16
2.1.1	Métodos Baseados em Conhecimento	p. 17
2.1.2	Métodos Baseados em Características Invariantes	p. 18
2.1.3	Métodos baseados em <i>Templates</i>	p. 21
2.1.4	Métodos Baseados em Aparência	p. 21
2.1.4.1	<i>Eigenfaces</i>	p. 21
2.1.4.2	Redes Neurais Artificiais - RNA	p. 23
2.1.4.3	Modelos Ocultos de Markov	p. 24
2.2	Haar Cascade	p. 25
2.2.1	Características	p. 27
2.2.2	Função de detecção de face	p. 28
2.2.3	Atenuação em cascata	p. 29
2.2.4	Algoritmo de treinamento do classificador	p. 30
3	RECONHECIMENTO DE PADRÕES EM IMAGENS COM K- NN	p. 32
3.1	<i>K-Nearest Neighbors</i>	p. 33

4 SISTEMA DE RECONHECIMENTO FACIAL COM HAAR CASCADE E K-NN	p. 36
4.1 K-NN clássico	p. 37
4.2 K-NN melhorado	p. 38
5 RESULTADOS E DISCUSSÕES	p. 41
5.1 Resultados	p. 42
6 CONCLUSÃO E TRABALHOS FUTUROS	p. 49
REFERÊNCIAS	p. 51

1 INTRODUÇÃO

Nos últimos anos a inteligência artificial (IA) tem evoluído no estudo e simulação da inteligência humana, (LOPES, 2012). Esse campo da computação tem ganhado espaço nos diversos seguimentos da indústria, seja na área de jogos digitais ou sofisticados sistemas de controle automático. No campo da medicina, os métodos de aprendizado de máquina aliados ao conhecimento do especialista tem permitido a criação de ferramentas de diagnóstico, auxiliando os profissionais nas tomadas de decisão. Na área jurídica, aplicações com a mesma finalidade já são consumidas por advogados e juristas. Aplicações com processamento de linguagens natural e visão computacional tem permitido grandes avanços tecnológicos na maneira como interagimos com as máquinas.

De maneira geral, a inteligência artificial tem evoluído na direção de simulação dos aspectos humanos, tais como cognição, aprendizado e emoção.

Em particular, o campo de visão computacional tem possibilitado a criação do carro autônomo e sistemas de controle realimentado por imagens, (LOZANO, 2012). Neste trabalho, será tratado um sistema de reconhecimento facial baseado em Haar Cascade (método de detecção rápida de objetos) e K-NN (algoritmo de reconhecimento de padrões), projetado para servir de base para o desenvolvimento de um interfone inteligente.

1.1 Visão computacional

Resumidamente, a visão computacional está além da captura de imagens, envolve um processo cognitivo de reconhecimento dos elementos da imagem, (BACKES, 2016). No projeto de um sistema de visão computacional, alguns aspectos devem ser considerados, são eles:

1. **Captura de imagem** - trata dos mecanismos de formação da imagem digital, incluindo desde o CCD (*charge-coupled device*) até a imagem.

2. **Processamento de imagem** - às vezes, antes de partir para os passos seguintes, algum pré-processamento é necessário. Tais processamentos podem ser aplicados para finalidades como redução de ruído, realce de característica, melhoramento de contraste e outras. Basicamente as técnicas usadas nessa fase destinam-se a projeto de filtros.
3. **Segmentação** - depois que a imagem foi capturada e processada, o objeto de interesse na imagem deve ser definido para a escolha da técnica que calculará as regiões da imagem que contém o objeto de estudo.
4. **Extração de características** - detectado o objeto de estudo na imagem, um conjunto de métricas deve ser projetada para quantificar o objeto estudado. Essas métricas formam um conjunto de dados denominado vetor característico.
5. **Reconhecimento de padrões** - os dados do vetor característico precisam ser classificados em algum grupo previamente cadastrado para a definição do resultado final que é uma conclusão interpretativa das informações extraídas da imagem.

1.2 Visão geral do trabalho

Nos capítulos seguintes serão tratados em detalhes algumas técnicas relacionados aos aspectos citados que foram utilizados no sistema estudado. Será dado maior ênfase nas etapas de segmentação, extração de características e reconhecimento de padrões.

No Capítulo 2 é dada uma abordagem superficial de vários métodos de detecção de faces com ênfase em detecção de face baseada em Haar Cascade, uma vez que este último foi a abordagem utilizada no trabalho desenvolvido.

No Capítulo 3 teremos uma introdução ilustrando como o método K-NN funciona e como utilizá-lo para detecção de padrões. O Capítulo 4 trata de ilustrar como combinar o que foi visto anteriormente (Haar Cascade e K-NN) para reconhecimento de faces. Também, no Capítulo 4 é mostrado um problema ao se utilizar K-NN na forma clássica e uma solução é apresentada.

Em seguida, o Capítulo 5 apresenta os resultados obtidos com a solução desenvolvida em um conjunto de testes. Detalhes dos testes feitos e justificativa para o conjunto escolhido são dados.

Enfim, no Capítulo 6 uma conclusão é apresentada e uma lista de melhorias é dada

como possibilidade de trabalhos futuros.

1.3 Objetivos e Justificativa

A proposta de desenvolver um sistema de visão computacional partiu da ideia de criar um sistema de porteiro eletrônico (interfone) capaz de reconhecer pessoas e ativar comandos de abertura de portas mediante a leitura da face de alguém cadastrado numa base de dados interna.

Diante da necessidade apresentada, o objetivo geral desse trabalho é buscar maneiras de projetar e construir um software capaz de fazer reconhecimento facial executando em um microcomputador como a *raspberry pi*. Para isso, foi proposto uma melhoria para o algoritmo de reconhecimento de padrão K-NN combinado com um técnica de detecção em cascata denominada Haar Cascade.

A justificativa para o projeto apresentado está diretamente relacionado a aplicação pretendida. Como deseja-se um processamento interno de imagens para detecção e reconhecimento, a implementação deste sistema constitui o primeiro passo para atingir o resultado esperado que é a construção do porteiro eletrônico com reconhecimento facial.

2 DETECÇÃO DE FACE COM HAAR CASCADE

Antes de apresentar Haar Cascade, é importante conhecermos alguns métodos usados para reconhecimento de faces. Será dada uma visão geral de **Métodos Baseados em Conhecimento**, **Métodos Baseados em Características Invariantes**, **Métodos Baseados em *Templates*** e **Métodos Baseados na Aparência**.

2.1 Técnicas de detecção de face

Conforme descrita em Lopes (2012), as principais técnicas de detecção de face estão subdivididas nas seguintes categorias:

- **Métodos Baseados em Conhecimento:** a codificação do programa de detecção depende da definição do que é uma face. Geralmente essa definição está baseada em atributos geométricos da face codificado em forma de regras.
- **Métodos Baseados em Características Invariantes:** implementam regras de detecção independente da orientação da face. Tais técnicas utilizam segmentação de pele e modelagem estatística da textura da face humana.
- **Métodos Baseados em *Templates*:** dependem de um *template* (modelo) para descrever de forma genérica o objeto a ser detectado, esse *template* é representado por uma família de curvas que modelam as características geométricas.
- **Métodos Baseados na Aparência:** necessitam de conhecimento prévio sobre a característica a ser detectada. Tal conhecimento é extraído de um conjunto de imagens que fará parte do conjunto de treinamento de algum algoritmo de aprendizagem de máquina. A abordagem de aprendizado pode ser baseado em Redes Neurais, *Eigenfaces* ou Modelos Ocultos de Markov.

Uma visão um pouco menos superficial dos métodos é necessária. Em especial, deve ser visto como cada método funciona e suas vantagens e desvantagens precisam ser apresentadas. Para tanto, nas próximas sub-seções, são dados mais detalhes sobre os métodos previamente listados.

2.1.1 Métodos Baseados em Conhecimento

Os métodos baseados em conhecimento dependem do conhecimento prévio do problema para que o projetista defina parâmetros únicos da face. É sabido que faces humanas possuem dois olhos, um nariz e uma boca, e esses elementos se encontram distribuídos de maneira relativamente previsível em uma imagem de um rosto.

Há desvantagens inerente ao conjunto de regras. Se as regras forem muito gerais, a taxa de falsos positivos é alta, (LOPES, 2012). Em contrapartida, se forem muito específicos pode ser ineficaz se a face não satisfizer a todo o conjunto de regras.

O uso de regras hierárquicas é uma maneira de fazer detecção de face baseado em conhecimento, (YANG *et al.*, 2001). No primeiro nível as regras tem como objetivo detectar os possíveis elementos candidatos da face extraídos da imagem por um algoritmo de janela deslizante. O segundo nível contém descrições dos elementos da face como olhos, boca e nariz. No seu trabalho, um conjunto de regras de distribuição de cores é usada no algoritmo de janela deslizante para identificar possíveis regiões que contém face.

Uma segunda abordagem é utilizar projeções verticais e horizontais da imagem definidas como, (Kotropoulos, 1997):

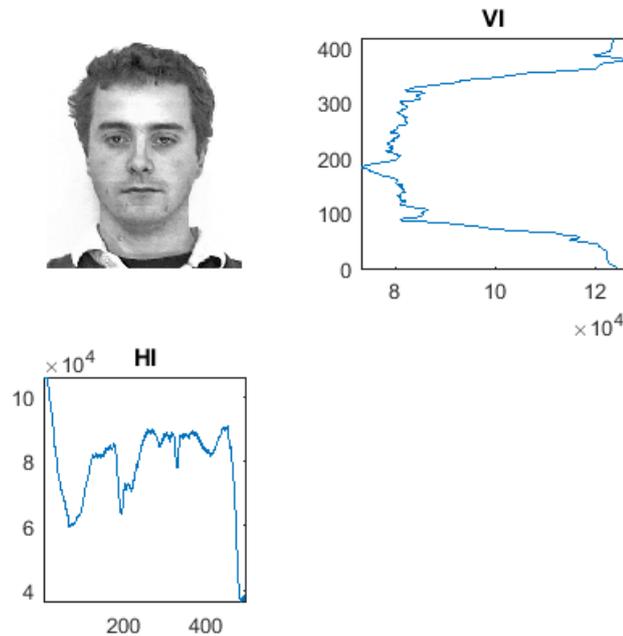
$$HI(x) = \sum_{y=1}^n I(x, y)$$

$$VI(x) = \sum_{x=1}^m I(x, y)$$

considerando uma imagem de tamanho $n \times m$. Analisando-se os mínimos e máximos locais da projeção é possível detectar a face e as regiões dos olhos, boca e nariz. O problema dessa abordagem é que os critérios ficam comprometidos em imagens obtidas com fundo complexo. Na Figura 1 é mostrado um exemplo da aplicação desse método. As curvas VI e HI compõem a informação de análise para a detecção.

A principal vantagem dos métodos baseados em conhecimento, é a relativa facilidade de definir critérios que podem ser implementados num detector, conforme mostrado nas

Figura 1: Projeção horizontal e vertical de uma imagem com rosto.



três abordagens comentadas onde o conhecimento do objeto é suficiente. A desvantagem é que tais critérios podem ser complicados de implementar ou insuficientes. Em geral, para objetos com poucas características, esta pode ser uma abordagem interessante.

2.1.2 Métodos Baseados em Características Invariantes

Os métodos baseados em características invariantes derivam da capacidade humana de identificar objetos independentemente do ponto de vista. Em relação a detecção de face humana, a cor da pele e a textura da face são as principais características invariantes, (YANG *et al.*, 2001). A análise pode ser realizada com pouca informação usando imagem em tons de cinza ou com muita informação usando imagens coloridas.

Constata-se que a cor da pele independente de suas variações (branca, negra, amarela, etc), tende a formar um *cluster* no espaço de cores, podendo ser modelada por uma distribuição gaussiana, (WANG, 1999). Essa característica pode ser explorada para detectar a presença de uma face em uma região colorida e classificar regiões com face e regiões sem face.

Buscar padrão de cor numa imagem RGB (sistema de cor aditivo formado por vermelho (Red), verde (Green) e azul (Blue)) é problemático porque este sistema de cor é bastante dependente da luminosidade do ambiente, (BUHIYAN, 2003). Um solução é uti-

lizar a seguinte transformação para o padrão de cor YIQ (espaço de cor usado no padrão NTSC):

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.320 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Como a cor da pele tende a se aglomerar numa região do espaço, um ponto mínimo (*threshold*) determinado empiricamente, separa regiões que contém pele das demais. O *threshold* baseado no padrão YIQ é dado por,

$$(60 < Y < 200) \wedge (20 < I < 50) \quad (2.1)$$

Após a aplicação da Eq. 2.1 tem-se os *pixels* que possivelmente fazem parte da pele humana. Para localizar a face toma-se a região com a maior quantidade de *pixels* conectados.

O sucesso ou fracasso das técnicas de segmentação baseadas na cor da pele estão ligados à maneira que a cor é representada. Devido o problema da representação em RGB, outros espaços de cores foram propostos, a saber: o YCbCr (*luminance-chrominance*) e o YES (Y representa luminosidade e E e S as componentes de cor). A principal vantagem do YES é a redução da variação de cor (*chrominance*) e principalmente a facilidade de calcular os valores de E e S a partir de uma imagem RGB fazendo somente deslocamento de bit, o que reflete positivamente no desempenho dos algoritmos, (SABER *et al.*, 1998).

Outra característica invariante é a textura dos objetos. A utilização de textura se deu com a tentativa de responder a pergunta: como pode o ser humano reconhecer um objeto que ora apresenta um forma e ora apresenta outra forma complemente diferente, como roupa aberta e dobrada? Da mesma maneira, a face humana muda drasticamente quando vista de frente ou de perfil. Observa-se que a textura é independente do ponto de vista e pode ser modelada como uma distribuição probabilística, ou seja, ela é gerada a partir de um processo estocástico que fornece algum padrão a partir de permutações aleatórias, (RIKERT, 2003). Para analisar esses padrões, o autor utiliza a transformada *wavelets* guiada, que permite analisar a imagem em diferentes escalas e orientações.

A hipótese de Rikert (2003) é que imagens do mesmo objeto obtidas de pontos de vistas diferentes possuem distribuições de probabilidade similares da intensidade da cor da pele na imagem, por possuir a mesma textura. A aplicação da transformada *wavelets* resulta em um conjunto de coeficientes independentes entre si na mesma escala, mas não

independentes quando aplicada em diferentes escalas. Para capturar essa dependência, Rikert utilizou um conjunto de vetores denominado *parent vectors*, que é o resultado da resposta de filtros aplicados em diferentes escalas. Após a obtenção dos *parent vectors*, esses vetores são agrupados em classes através da aplicação de um algoritmo de agrupamento para posterior construção do modelo gaussiano misto que generalize a distribuição. Para saber a qual classe uma imagem de entrada do detector pertence, é feito um cálculo de probabilidade do *parent vector* da imagem de entrada em relação às classes, e selecionado a classe de maior probabilidade. A principal desvantagem dessa abordagem é que para uma taxa de falsos positivos baixa, a quantidade de classes deve ser maior que 1000, tornando o processamento demorado.

Uma abordagem similar é discutida no trabalho de Luo (2000), com a diferença de que a geração do vetor característico não é feito no domínio de *pixel*, mais sim no domínio DCT (*Discrete Cosine Transform*), que segundo o autor é mais eficiente em termos computacionais. Com essa abordagem, o problema de detecção de face é tratada como um problema de classificação de vetores unidimensionais. Com o objetivo de tornar a técnica capaz de tratar imagens em várias resoluções, os modelos de faces e não faces foram criados em 6 escalas diferentes. O vetor característico de uma imagem de entrada é agrupada em 6 grupos de distribuição gaussiana através da aplicação do algoritmo *K-means*, mas com o cálculo de distância baseado na distância gaussiana logarítmica.

Na abordagem de Rikert, a aplicação da transformada *wavelet* não modifica a característica do domínio. Já a aplicação da DCT depende de cálculos de transformação direta e inversa mais complexos. Apesar de reduzir a quantidade de classes, os teste realizados não foram superiores a abordagem direta de *pixel*. Apesar de não ter obtido um desempenho melhor, esse método foi o primeiro a trabalhar em um domínio DCT.

Nas abordagens mencionadas observamos que apesar da vantagem de poder detectar objetos independente da posição, não existe garantia de que o objeto detectado seja o objeto de interesse, isto é, utilizando a cor ou textura como característica, qualquer objeto semelhante a uma face pode ser detectado. Se o detector foi escrito para encontrar face humana, então não há garantia mínima de que algum objeto detectado seja uma face. Além disso, o domínio de representação da imagem influencia diretamente no desempenho e eficiência dos algoritmos. Por exemplo, utilizando o sistema de cor RGB a iluminação e ruído corrompe qualquer valor de *threshold* estabelecido para segmentar faces em imagens coloridas.

2.1.3 Métodos baseados em *Templates*

Uma técnica clássica de detectar objetos é procurar pelo mesmo dentro da imagem e testar se ele corresponde a um modelo prévio de sua forma. A morfologia matemática do objeto é descrita em termos de seus componentes geométricos básicos, como círculos, quadrados ou triângulos. Essa técnica é denominada *template*. A técnica consiste em encontrar correspondência entre o objeto procurado e seu modelo. Essa correspondência pode ser dada em termos de área, perímetro, quantidade de arestas ou vértices. No caso da detecção de faces, o modelo mais utilizado é aquele que trata a face como uma elipse, (LOPES, 2012). A procura pelo modelo descritor pode ser feita através de heurística ou algoritmos genéticos, (WON *et al.*, 2004). Qualquer objeto que pode ser representado por meio de formas, pode ser detectado usando *templates*.

A definição de um modelo matemático do objeto é o pre-requisito fundamental dessa abordagem. Assim, objetos com forma matemática bem definida (exemplo: bola, mesa, cadeira, porta, etc...) podem ser facilmente detectados utilizando seu modelo. Em contrapartida, objetos complexos impõem limitações na modelagem de algoritmos de detecção por *template*, uma vez que os modelos descritores são aproximações do modelo real. A principal vantagem, é que como a forma é a característica necessária desse método, ferramentas de detecção de bordas, operações em conjuntos (complemento, diferença, união e intersecção) e operações morfológicas (dilatação, erosão, abertura e fechamento), são suficientes, o que torna a técnica mais robusta em relação a iluminação e ruído.

2.1.4 Métodos Baseados em Aparência

Os métodos baseados em aparência, não utilizam qualquer conhecimento prévio ou características empíricas do objeto a ser detectado. Nessa classe de algoritmos surgem os conceitos de **aprendizado** e **treinamento**, uma vez que as informações necessárias para detectar o objeto desejado são retiradas do conjunto de imagens. Portanto, as técnicas apresentadas a seguir representam internamente, por meio de técnicas de aprendizado de máquina, o objeto de interesse.

2.1.4.1 *Eigenfaces*

O método de *eigenfaces* é baseado na transformada de Karhunen-Loeve (KLT) e PCA (*Principal Component Analysis*), cuja motivação principal é encontrar um espaço vetorial que melhor descreve um objeto utilizando poucas dimensões, (TURK *et al.*, 1991).

Conforme observado por Pentland e Turk (1991), as imagens de faces não se encontram distribuídas de forma aleatória em um espaço de alta dimensionalidade, portanto podem ser descritas em um espaço de dimensão menor. Uma maneira de se obter essa descrição é aplicando-se a transformada KLT com o objetivo de encontrar os vetores que melhor descrevem a distribuição de imagens dentro do espaço de imagens inteiro. Tais vetores são denominados *eigenfaces*.

Para encontrar o conjunto de vetores (base do espaço vetorial n -dimensional) que produzem um espaço de dimensão menor, parte-se da análise de componente principal. A determinação da nova base é feita da seguinte maneira: considerando $I(x, y)$ uma imagem de duas dimensões $N \times N$ ou N^2 , a partir do conjunto de testes $I = \{I_1, I_2, \dots, I_M\}$, encontra-se a face média f_m do conjunto definida por:

$$f_{m_j} = \frac{1}{M} \sum_{i=1}^M I_i^j \quad (2.2)$$

onde I_i^j refere-se ao pixel j da imagem i . Cada face difere da média por,

$$\phi_i = I_i - f_m \quad (2.3)$$

O conjunto I é submetido ao PCA que encontra um conjunto de M vetores ortogonais u_k e seus respectivos *eigenvalues* λ_k . Os vetores u_k e as escalares λ_k são os autovetores e autovalores, respectivamente, da matriz de covariância definida por:

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (2.4)$$

onde $A = \phi_1, \phi_2, \dots, \phi_M$. A dimensão de A é N^2 , o que torna o cálculo dos N^2 autovalores e autovetores uma tarefa dispendiosa para imagens de 128×128 *pixels*. Felizmente, utilizando KLT, existe uma maneira de calcular os autovetores através da resolução de uma matriz de tamanho M e realizar-se combinações lineares dos vetores resultantes, (TURK *et al.*, 1991). Tal análise permite resolução dos cálculos da ordem de N^2 para M .

Os autovalores associado a cada autovetor nos permite organizar cada autovetor de acordo com a sua utilidade na caracterização das variações entre as imagens. Aproveitando-se do fato de que as faces não mudam radicalmente quando projetadas no espaço de *eigenfaces*, mas quando um objeto arbitrário é projetado neste mesmo espaço ele muda sua aparência drasticamente, pode-se utilizar esta técnica para detectar faces apenas calculando a distância do objeto projetado em relação as classes de face projetadas e através

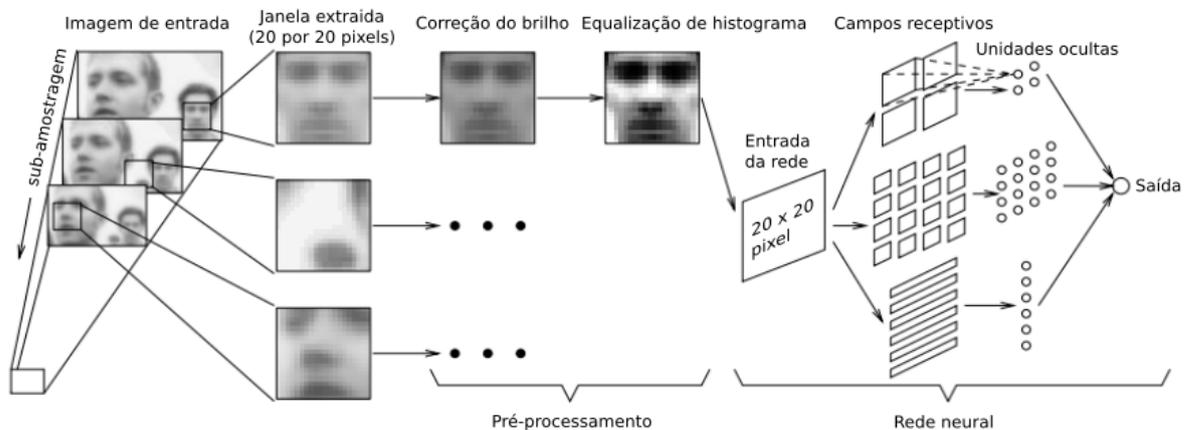
de um *threshold* verificar se este é uma face ou não.

2.1.4.2 Redes Neurais Artificiais - RNA

As redes neurais artificiais são modelos matemáticos inspirados no modelo do neurônio biológico com o objetivo de emular o processo cognitivo do cérebro humano. A RNA é usada nas tarefas de classificação e reconhecimento de padrões. No caso de detecção de face, usa-se RNA para diferenciar imagens ou regiões da imagem que possivelmente contenham uma face daquelas que não contém. Ou seja, a função da rede é classificar as imagens em duas classes: face e não-face.

Um sistema de detecção de faces baseado em RNA é mostrado resumidamente na Figura 2. Nesse sistema uma imagem de entrada é percorrida por um filtro que consiste de uma janela de 20×20 que extrai regiões onde podem existir faces. Logo após a extração da possível região de face, ela é submetida à equalização de histograma e normalização da luz. Se a face for maior que o tamanho da janela, a imagem é sub-amostrada repetidamente por um fator de 1.2 e o filtro é aplicado em cada subamostragem. Cada bloco da imagem depois de extraído e pré-processado é enviado para uma rede neural que decidirá se nele existe ou não uma face. A rede é composta de várias camadas com arquiteturas diferentes chamadas de campos receptivos. Cada camada avalia uma parte de imagem com a finalidade de detectarem características que podem ser úteis para determinar a presença da face. A rede apresenta como resposta um único valor que determina ou não a existência da face, (ROWLEY, 1996).

Figura 2: Funcionamento do algoritmo de detecção de face baseado em RNA, (ROWLEY, 1996).



Um teste realizado com uma rede possuindo 78 neurônios intermediários e 4357 co-

nexões apresentou uma taxa de acerto na ordem de 92.7%, (ROWLEY, 1996). A estratégia descrita aqui é eficaz apenas na detecção de faces frontais ou com leves variações na rotação. Diante desta limitação, uma extensão dessa técnica é torna-la robusta quanto a rotações de face. A modificação consiste na utilização de outra rede neural chamada rede roteadora (*router network*) que é responsável por determinar o ângulo de rotação da face.

2.1.4.3 Modelos Ocultos de Markov

Uma terceira abordagem para detectar faces é usando Modelos Ocultos de Markov (MOM). Historicamente, o MOM tem sido uma ferramenta útil no reconhecimento de fala, (LOPES, 2012) . Tais modelos possuem como principal característica a capacidade de caracterização de propriedades estatísticas de um sinal. O MOM define um conjunto de estados, símbolos e transições. Na tese de Ara (1996) encontra-se uma descrição completa da aplicação do MOM na construção de um sistema de detecção de face. A utilização do MOM permite construir sistemas de detecção e reconhecimento robusto quanto a diferença de escala e variações na orientação e expressões da face, (ARA, 1996).

Diferente das outras técnicas abordadas, os métodos baseados em aparência apresentados, não utilizam características das imagens (como cor ou textura) ou características dos objetos (como elementos definidos a partir do conhecimento e morfologia matemática). Antes, quantifica tais características por meio de transformações e às representam em espaços genéricos.

A abordagem por *eigenfaces* depende que várias operações sejam realizadas para a definição de um espaço que descreve o objeto alvo da detecção. De maneira análoga, o MOM depende da análise dos parâmetros observáveis para se ter os parâmetros ocultos. Apesar de apresentarem bons resultados, as duas sugestões anteriores não é tão dinâmica quanto a utilização de RNA. Enquanto as primeiras necessitam de ajustes analíticos da função de detecção, com RNA pode-se utilizar algoritmos de treinamento e aprendizado para que a rede se auto ajuste. Esta última abordagem é encontrada na maioria dos trabalhos atuais relacionado a detecção e reconhecimento de objetos e também é empregada na ferramenta de detecção aplicada neste trabalho.

2.2 Haar Cascade

A semelhança observada nas técnicas de detecção de face revisadas nesse capítulo é que todas apresentam limitações de escopo e desvantagens em casos específicos. Métodos baseados em conhecimento, características invariantes ou *templates* exigem um conhecimento do objeto a ser detectado para ajustar os parâmetros do detector. Já a abordagem baseada em aparência transfere essa exigência para o próprio algoritmo de detecção (RNA) onde técnicas de aprendizado de máquina são aplicadas para ajustar os parâmetros.

Uma abordagem baseada em cascata é apresentada no artigo *Rapid object detection using a boosted cascade of simple features* (Detecção rápida de objetos usando uma cascata de características simples), (VIOLA e JONES, 2001). Em seu trabalho são introduzidas três contribuições: um nova representação de imagem chamada de **imagem integral**, um algoritmo de aprendizado baseado em AdaBoost e a combinação de classificadores com nível de detalhe e complexidade crescente. A abordagem em cascata permite que os métodos apresentados sejam aplicados de maneira mais inteligente, contribuindo para uma alta taxa de detecção. No seu trabalho, Viola e Jones conseguiu detectar faces a uma taxa de 15 quadros por segundo em uma máquina com um processador Intel Pentium III de 700 MHz.

O processo idealizado por Viola e Jones trabalha com características ao invés de intensidade dos *pixels*. Tais características são obtidas a partir do remanescente das aplicações sucessivas da transformada *wavelet* com a base de Haar. Na Figura 3 é mostrado um exemplo da decomposição em três níveis de uma imagem contendo uma face. Observe-se que na terceira decomposição, o remanescente da imagem contém somente 3 tons. Nesse nível, o valor dos pixel é usado como característica, conforme será melhor explicado mais adiante. Lembrando que a aplicação desse procedimento para obter o remanescente acontece dentro de uma janela deslizante ao longo da imagem para gerar caracterizadores de todas as regiões.

Para obter rapidamente as características em muitas escalas, foi introduzido o conceito de imagem integral. A ideia é representar uma região da imagem por um valor dado a partir da soma dos *pixels* da região. Matematicamente, a imagem integral pode ser escrita como segue:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.5)$$

onde $ii(x, y)$ é a imagem integral e $i(x, y)$ a imagem original. Na Figura 4 está ilustrado

Figura 3: Resultado da aplicação sucessiva em três níveis de decomposição, da transformada wavelet utilizando a base de Haar. (a) Imagem original, (b) remanescente do primeiro nível, (c) remanescente do segundo nível e (d) remanescente do terceiro nível. Para perceber o efeito da aplicação da transformada wavelet na resolução, o tamanho das imagens foi aumentado.

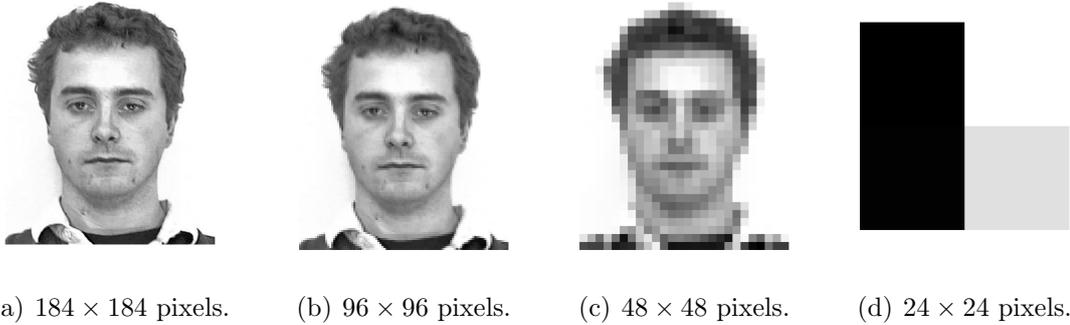
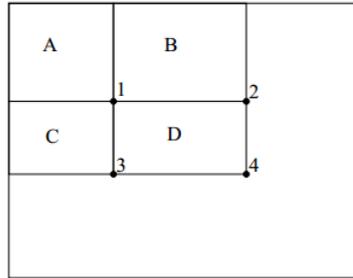


Figura 4: A soma dos pixels dentro do retângulo D pode ser calculada a partir das quatro matrizes de referência. O valor da imagem integral no ponto 1 é a soma dos pixels dentro de retângulo A . No ponto 2 é $A + B$, em 3 $A + C$, finalmente em 4 $A + B + C + D$ ou $D = 4 + 1 - (3 + 2)$, (VIOLA e JONES, 2001).



como ocorre a formação da imagem integral. Os valores de x e y são incrementados e cada incremento define um novo ponto da imagem integral. Observa-se que a soma dentro de D pode ser calculada como $4 + 1 - (3 + 2)$. Assim, o valor da soma dos pixels de uma região da imagem pode ser obtida a partir da imagem integral. O termo **imagem integral** vem do fato da soma ser cumulativa, semelhante a operação de integração. Usando os seguintes pares de recorrência,

$$s(x, y) = s(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

a soma cumulativa da linha pode ser calculada rapidamente em um único passo sobre a imagem original, considerando que $s(x, -1) = 0$ e $ii(-1, y) = 0$.

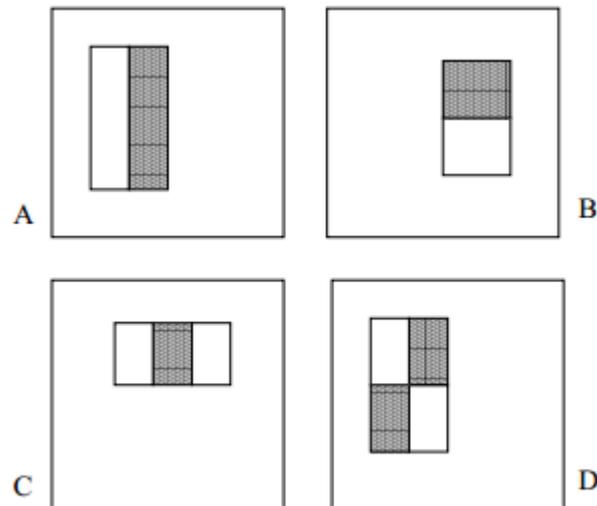
2.2.1 Características

Existe muitas motivações para o uso de características ao invés de *pixels* diretamente. A principal delas é que as características codificam o conhecimento do domínio, que é difícil de aprender usando uma quantidade finita de dados de treinamento. Outra vantagem é o ganho de desempenho conseguido com a imagem integral.

Especificamente, são usados três tipos de características baseados na distribuição dos retângulos da imagem integral mostrada na Fig. 5. São elas:

1. **Características de dois retângulos:** diferença da soma dos *pixels* dentro de cada retângulo
2. **Característica de três retângulos:** soma dos *pixels* dos dois retângulos externos subtraído da soma dos *pixels* do retângulo interno.
3. **Características de quatro retângulos:** diferença da soma dos *pixels* das diagonais.

Figura 5: Características em imagens integrais.



Dado que a resolução base do detector é 24×24 , tamanho escolhido empiricamente por Viola e Jones (2001), o conjunto de retângulos característicos, considerando as sobreposições, pode chegar a 180000 numa imagem de 300×600 pixels, justificando ainda mais o uso de *wavelets*.

Apesar de os retângulos característicos parecerem bastante primitivos em relação a outras abordagens de representação de imagem, eles provém uma forma simples para uso de aprendizado de máquina.

2.2.2 Função de detecção de face

Dado o conjunto de características e o conjunto de treinamento com imagens positivas e negativas, isto é, imagens que contém face e não-face, alguma abordagem de aprendizado de máquina pode ser usada para encontrar a função de classificação. Lembrando que o classificador simplesmente informará se na imagem existe ou não uma face.

Conforme dito na seção anterior, a quantidade de características pode chegar a 180000. Viola e Jones (2001) percebeu que algumas características eram mais decisivas que outras. Assim, o principal desafio para uma detecção rápida é escolher as características mais fortes, isto é, aquelas que se não encontradas garante que o objeto a ser detectado não existe na região analisada, caso contrário, a região que contém tal característica é recursivamente analisada utilizando características mais detalhadas.

Para selecionar as características, Papageorgiou *et al* (1996) propôs um esquema baseado na variância das características. Ele demonstrou bons resultados selecionando 37 características de um total de 1734. Roth *et al* (1997) propôs a seleção de características baseada no algoritmo de perceptron exponencial. O processo de aprendizagem convergiu para uma solução onde muitos dos pesos eram nulos. Mesmo assim, uma grande quantidade de características são retidas, contribuindo para um baixo desempenho do detector.

No sistema proposto por Viola e Jones (2001), uma variação do AdaBoost (algoritmo de aprendizado de máquina usado para melhorar a performance de classificadores simples) é usado para selecionar um pequeno número de características e treinar o classificador. Ele percebeu que utilizando as características selecionadas nas primeiras iterações do AdaBoost, o detector apresentava uma taxa de erro entre 0.1 e 0.3. Já usando as características selecionadas depois de muitas iterações, o detector respondia com uma taxa de erro entre 0.4 e 0.5. Baseado nesta observação, a abordagem proposta consiste em um algoritmo de aprendizado fraco (que seleciona poucas características) usado para selecionar retângulos característicos simples que melhor separam exemplos positivos dos negativos.

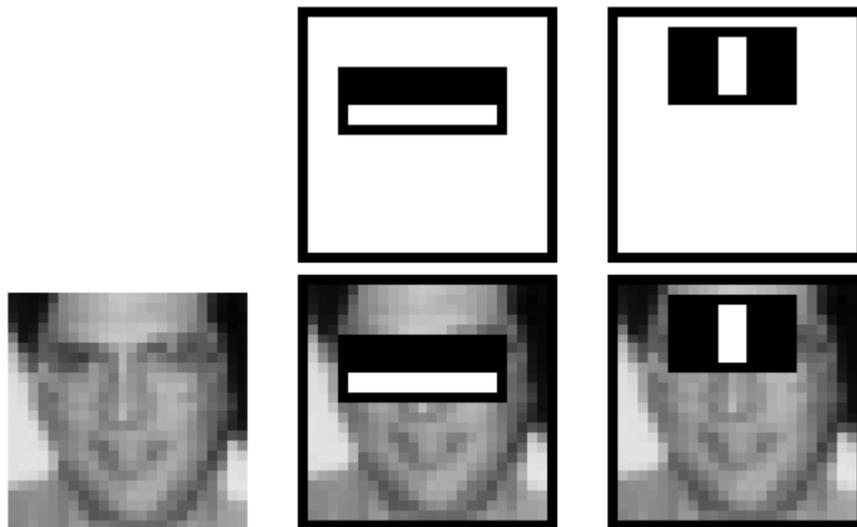
Tal metodologia é útil pois descarta no nível de menor detalhes a maioria das características da imagem. Para cada característica, o aprendizado fraco determina um *threshold* ótimo para a função de classificação. Portanto, um classificador fraco $h_i(x)$ consiste de uma característica f_j , um *threshold* θ_j e um par p_j indicando a direção do sinal da inequação:

$$h_j(x) = \begin{cases} 1 & \text{se } p_i f_i(x) < p_j \theta_j \\ 0 & \text{caso contrário} \end{cases} \quad (2.6)$$

Experimentos iniciais demonstraram que um classificador de face frontal construído com 200 características apresenta um desempenho de 95%, isto é, das 100 faces testadas, o classificador conseguiu identificar 95, (VIOLA e JONES, 2001). Esse resultado apesar de bom implica num maior processamento.

Para a tarefa de detecção, a característica inicial selecionada pelo AdaBoost deve ser significativa e de fácil interpretação. A primeira característica foca na propriedade de que a região dos olhos é mais escura que a região do nariz e bochecha, Fig. 6. Essa característica dá uma medida da diferença de intensidade entre a região dos olhos e sua vizinhança. A segunda característica selecionada depende da propriedade de que os olhos são mais escuros do que a ponta do nariz.

Figura 6: A primeira e segunda característica selecionada pelo AdaBoost, (VIOLA e JONES, 2001) .



As demais características são definidas pelos algoritmos de aprendizado, sendo em sua maioria de difícil interpretação.

2.2.3 Atenuação em cascata

Como com a utilização de poucas características fortes, a maioria das sub-janelas já são descartadas no menor nível de detalhe, a aplicação de um classificador mais complexo

(que usa um maior número de características) somente nas regiões potenciais, evita processamento desnecessário com alta probabilidade, isto é, forte evidência que na região de busca não existe o objeto de interesse. Esse fato induz a ideia de usar uma cascata de classificadores. Esta abordagem é denominada Haar Cascade.

A forma geral do processo de detecção é de uma árvore de decisão. Um resultado positivo do primeiro classificador dispara um sinal para um segundo classificador que foi ajustado para alcançar uma alta taxa de detecção. Um resultado positivo do segundo classificador dispara um terceiro e assim por diante. Lembrando que os classificadores mais seletivos processam apenas janelas definidas pelo classificador mais fraco. A visão chave é que pequenos classificadores podem ser construídos para rejeitar muitas das sub-janelas negativas enquanto detecta quase todas as instâncias positivas.

Os estágios da cascata são construídos treinando o classificador usando AdaBoost e ajustando o *threshold* para minimizar falsos negativos. No primeiro nível é usado somente duas características. Se o resultado de uma sub-janela for positivo, esta passa para os classificadores com maior poder de reconhecimento. Caso contrário, a sub-janela é rejeitada e o processo é finalizado na sub-janela.

Um detector em cascata completo tem 38 estágios com 6000 características, (VIOLA e JONES, 2001). Mesmo assim o processamento em cascata é rápido devido a estrutura inteligente de agrupamento de características.

2.2.4 Algoritmo de treinamento do classificador

O algoritmo de treinamento do classificador baseado em AdaBoost está desenhado a seguir. O conjunto de entrada são as imagens exemplos acompanhado de uma constante y que define se o exemplo contém face (positivo) ou não (negativo). A saída é a função de classificação melhorada. O processo inicia definindo o valor de peso de cada exemplo. O peso dos exemplos positivos é função da quantidade de entrada positiva, analogamente, o peso dos exemplos negativos é função da quantidade de entrada negativa.

A iteração inicia normalizando os pesos utilizando alguma distribuição de probabilidade. Na sequência, uma característica é selecionada e aplicada no classificador para avaliar todo o conjunto de treinamento, ao fim, o erro é calculado. Depois de obter o erro dos j classificadores, escolhe-se o classificador de menor erro. O passo seguinte consiste em atualizar os pesos em função do erro.

Nas primeiras iterações as características mais gerais são atribuídas ao classificador.

Após muitas iterações, as características mais específicas são selecionadas com maior probabilidade.

Algoritmo 1: Algoritmo AdaBoost para treinar o classificador.

Entrada: Conjunto de imagens $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, onde $y_n = 0, 1$ para exemplos positivo e negativo, respectivamente.

Saída: Função de classificação, $h(x)$.

início

Inicializar os pesos $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$, onde m e l são os números de exemplos negativos e positivos, respectivamente.

para t de 1 até T **faça**

Normalizar os pesos,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

onde w_t é uma distribuição de probabilidade.

Para cada característica j , treinar o classificador h_j com a restrição de usar uma única característica, e depois avaliar o erro em relação a w_t ,

$$\epsilon_j \leftarrow \sum_i w_i |h_j(x_i) - y_i|.$$

Escolher o classificador h_t que diminui o erro ϵ_t .

Atualizar os pesos,

$$w_{t+1,i} \leftarrow w_{t,i} \beta_t^{1-\epsilon_i}$$

onde $\epsilon_i = 0$ se x_i for classificado corretamente, $\epsilon_i = 1$ caso contrário, e

$$\beta_t = \frac{\epsilon_t}{1-\epsilon_t}.$$

Depois de t ciclos, o classificador final é,

$$\begin{aligned} h(x) &= 1 \text{ se } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ &= 0 \text{ caso contrário} \end{aligned}$$

onde $\alpha_t = \log \frac{1}{\beta_t}$.

fim

fim

3 RECONHECIMENTO DE PADRÕES EM IMAGENS COM K-NN

No capítulo anterior foi visto que a saída do detector é um conjunto numérico que foi denominado de **vetor característico**. A última etapa do sistema desenvolvido, consiste na classificação dos vetores característicos. Essa fase é denominada de **reconhecimento de padrão**.

Existem diversas técnicas para agrupar um dado vetor característico numa classe. Backes e Junior (2016) agrupa essas técnicas em 4 classes, conforme discutido a seguir:

- **Classificadores elementares** - baseiam-se exclusivamente em métricas de distâncias. As principais são: K-vizinhos mais próximos (K-NN- do inglês, *K-nearest neighbors*) e classificador de protótipo mais próximo.
- **Classificadores Bayesianos** - fundamentam-se no teorema de Bayes para estimar a probabilidade de um dado vetor característico A pertencer a uma classe B segundo sua distribuição de probabilidade. É uma categoria de classificador estatístico. Dentro dessa categoria podemos destacar a análise linear discriminante (LDA), análise quadrática discriminante (QDA) e classificadores *Naive Bayes*.
- **Agrupamento** - quando não existe um supervisor que atribua uma classe a cada amostra de uma base de dados, é possível analisar a disposição espacial dos vetores característicos em termos de coesão em grupos (*clusters*) e de separabilidade entre diferentes grupos, e, desse modo, inferir a quantidade de classes e quais amostras pertencem a cada um dos grupos. Existem vários algoritmos de agrupamento, dentre eles podemos destacar o dendrograma e *K-means* (K-média).
- **Redes Neurais Artificiais (RNA)** - nasceu do trabalho de McCulloch e Pitts (1943) sobre neurônios artificiais que tentam simular as propriedades do neurônio

biológico. Uma rede neural é um conjunto de neurônios artificiais conectados, onde a quantidade de entrada da rede corresponde ao número de elementos do vetor característico e o número de saídas à quantidade de classes. Para que a classificação ocorra corretamente, os pesos devem estar ajustados. Para obter uma rede com os pesos ajustados, podemos usar uma rede com alguma regra de aprendizado. Uma delas é a rede de Perceptron criada em 1958 por Rosenblatt.

Dentre as abordagens de classificação citados, considerando que o vetor característico de uma face possui uma classe bem definida, isto é, dado o conjunto de imagem da mesma face esse conjunto de dados já define a classe que representa a face treinada, então a abordagem de clusterização não é adequada, uma vez que o agrupamento de dados para definir uma classe não é necessário. Como os dados não apresentam uma variabilidade aleatória, a abordagem de Bayes também não é aplicável.

Uma característica dos classificadores elementares é que eles são lineares (a saída é proporcional a alguma métrica de distância), ideal para aplicar em conjunto de dados que apresenta comportamento linear. Diferente disso, as RNA's podem ser generalizadas para classificar conjuntos com distribuição espacial não linear (a saída depende dos pesos, que não necessariamente estão inter-relacionados), isso pode ser visto em (RUSSEL, 2010).

Devido à características dos dados usado nesse trabalho apresentar uma região de separação linear, propomos verificar a possibilidade de reconhecimento facial usando algum classificador elementar. Pra este trabalho, foi escolhido o K-NN.

3.1 *K-Nearest Neighbors*

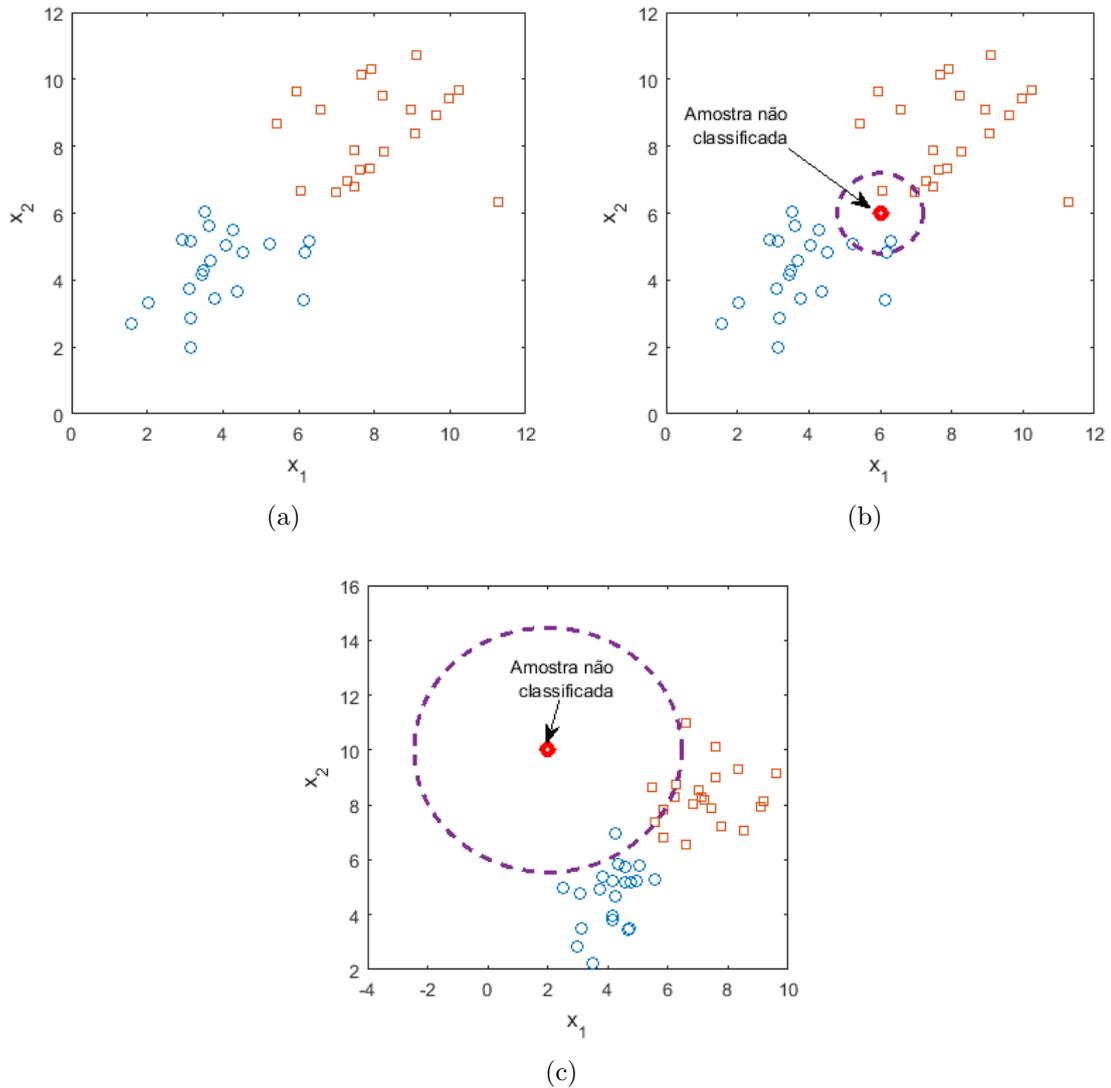
O algoritmo do classificador K-NN baseia-se na seguinte regra: **atribua a uma amostra desconhecida a classe das K amostras que estejam mais próximas** usando alguma métrica de distância como euclidiana, mahalanobis, manhattan, cyteblock, etc. De maneira genérica pode-se escrever o cálculo de distância usando a distância de Minkowski ou norma L_P , definida como,

$$L_P(x_j, x_q) = \left(\sum_i |x_{j,i} - x_{q,i}| \right)^{1/P} \quad (3.1)$$

onde x_j é a amostra não classificada e x_q o conjunto de treinamento. Com $P = 2$ temos a distância euclidiana, para $P = 1$ a distância Manhattan.

A distância Euclidiana é usada para mensurar propriedades similares como altura,

Figura 7: K vizinhos mais próximos.



largura ou profundidade, já a distância Manhattan é usada se elas não são similares, como eras, peso, gênero ou paciente.

Depois de obter as distâncias do objeto desconhecido em relação aos objetos do conjunto de treinamento, a classe dominante entre as K amostras será determinada como a classe do objeto desconhecido. Caso haja empate, algum critério de decisão deve ser adotado. Uma alternativa é que as amostras mais próximas do objeto decida a classe. Na Figura 7 é mostrado um exemplo da aplicação do K-NN a um conjunto de dados x descritos por suas componentes x_1 e x_2 . Na Figura 7a os pontos circulares pertencem à classe **A** e os pontos quadrados à classe **B**. Considerando uma amostra não classificada $x = (6, 6)$, para $K = 5$ a classe da amostra será aquela que tiver o maior número de exemplos no conjunto de treinamento dentre os 5 vizinhos mais próximos. A Figura 7b

mostra os 5 vizinhos mais próximos confinados dentro do círculo tracejado. Portanto, utilizando K-NN conclui-se que a amostra de teste pertence à classe **B**. Na Figura 7c observamos que apesar da amostra desconhecida $x = (2, 10)$ está distante de ambos os grupos, o K-NN tenta atribuir a classe **B**.

O principal inconveniente do K-NN para o problema em questão é que a amostra sempre será associada a alguma classe, mesmo que os K vizinhos mais próximos estejam relativamente longe da amostra. O problema de o classificador K-NN sempre sugerir uma classe para a amostra de teste o torna inadequado para reconhecimento facial. Por exemplo, pode-se treinar o classificador com faces de um conjunto de pessoas e testá-lo verificando a classificação da face de uma pessoa com a qual o algoritmo não foi treinado. Espera-se que, nesse caso, o classificador não reconheça a face testada, porém, como o K-NN sempre produz uma classificação, o resultado obtido será necessariamente errado.

Utilizar o K-NN para classificar elementos fora de suas classes conhecidas produz uma classificação errônea quando, idealmente, o algoritmo deveria retornar algum aviso de que não é possível classificar confiavelmente a amostra testada. Foi observado que nesses casos, alguns sintomas de classificação errada podem ser detectados. Esses sintomas são: a amostra testada é mapeada para uma região com vizinhos muito distantes ou a amostra testada tem vizinhos mais próximos de classes distintas. Essa constatação foi utilizada para desenvolver um melhoramento do algoritmo K-NN a fim de torná-lo adequado para reconhecimento de faces.

Outro fator que pode influenciar no resultado é o uso dos dados brutos. Se os dados não estiverem normalizados, isto é, todos escritos em relação a mesma escala, uma mudança na escala de uma das dimensões afeta o resultado. Para evitar esse tipo de erro uma solução simples é aplicar alguma forma de normalização. Além disso, com poucas amostras de treinamento, a classificação pode apresentar muito falso negativo, tornando o classificador ruim. No exemplo mostrado, a diferença entre a quantidade de vizinhos da classe A e B foi de apenas uma unidade devido à amostra de teste estar na região de transição. Assim, para garantir bons resultados com K-NN é ideal que o conjunto de treinamento possua muitas amostras. Conforme discutido em Russel (2010), em espaço de poucas dimensões com um conjunto de treinamento suficiente, o K-NN apresenta bons resultados. No entanto, quando a quantidade de dimensões aumenta, obter amostras para o conjunto de treinamento se torna difícil, tornando o K-NN pouco assertivo.

Dadas as limitações do K-NN e os resultados observados, tratou-se de se desenvolver uma maneira de contornar-se os problemas encontrados.

4 SISTEMA DE RECONHECIMENTO FACIAL COM HAAR CASCADE E K-NN

O sistema de visão computacional desenvolvido nesse trabalho segue o fluxo de atividade mostrado na Figura 8. O processo inicia com a captura da imagem. A imagem pode ser capturada por alguma câmera em 'tempo real', ou pode está armazenada em algum arquivo. Por simplicidade, após a captura, a etapa de processamento foi omitida, devido os testes terem ocorrido com imagens obtidas em laboratório com baixo ruído. As imagens de testes foram obtidas do banco de imagens *Psychological Image Collection at Stirling (PICS)*, cujo os direitos de imagem permite o uso para fins científicos. A etapa seguinte consiste em detectar a presença de face na imagem, para isso, foi usado Haar Cascade. A aplicação de Haar Cascade gera um conjunto de marcas faciais incluindo as regiões dos olhos, nariz, boca e íris, conforme mostrado na Figura 9. Com essas marcações, podemos determinar o vetor característico. Para compor o vetor característico, foram escolhidas quatro métricas: D_o distância entre os olhos; H_o altura dos olhos em relação ao limite inferior da face; H_n altura do nariz e H_b altura da boca, tais métricas também estão ilustradas na Figura. 9. Para normalizar esse conjunto de dados, a largura da base b do retângulo da face, foi escolhido como métrica de referência, lembrando que as regiões das marcações varia com o tamanho da face. Por fim, dado que o vetor característico já foi gerado, a última etapa deve definir a qual classe a face de teste pertence. Note que “definir a qual classe a face de teste pertence” equivale a, em termos simples, dizer a que pessoa aquele rosto pertence.

No caso do sistema de visão computacional desenvolvido, cada face distinta define uma amostra e um conjunto de faces de uma mesma pessoa determina uma classe. Assim, várias imagens de faces de uma mesma pessoa compõem o conjunto de treinamento da classe. Dessa forma, adicionar uma nova classe ao sistema consiste em cadastrar a face

Figura 8: Fluxo de execução do sistema de reconhecimento facial.

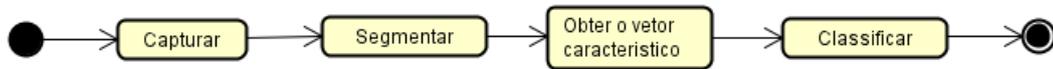
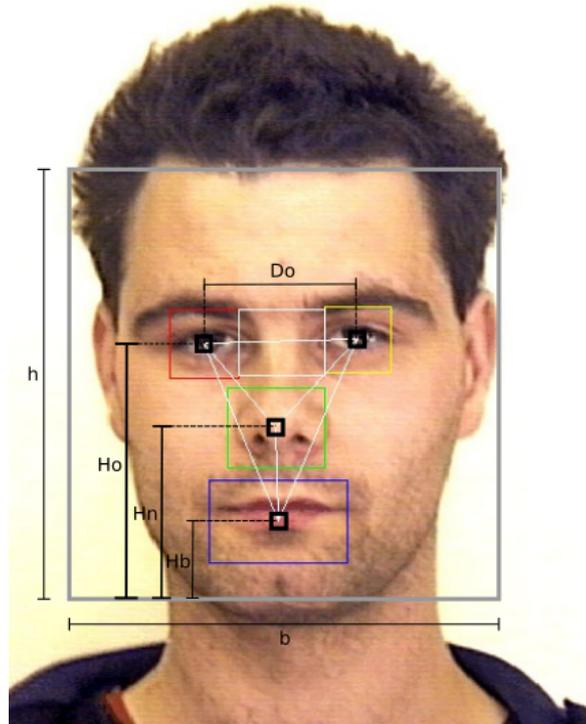


Figura 9: Descritores faciais.



de uma nova pessoa não conhecida pelo sistema.

Matematicamente, utilizando as métricas projetadas, cada face é descrita por um vetor de quatro dimensões $x_j = (D_o, H_o, H_n, H_b)$. O conjunto de todas as faces do conjunto de treinamento é representado pelo vetor $X_q = x_1, x_2, \dots, x_N$, onde N é número de faces cadastradas e X_q o vetor característico de cada face.

4.1 K-NN clássico

O algoritmo K-NN clássico, semelhante ao apresentado no capítulo anterior, pode ser usado na etapa de reconhecimento facial. Como o sistema mantém um conjunto de treinamento com cada uma das faces, a etapa de reconhecer equivale à tarefa de detectar um padrão já conhecido. Com K-NN, isso se traduz em encontrar as faces mais semelhantes à face de teste. A descrição completa do K-NN clássico aplicado a reconhecimento facial é mostrada a seguir.

Algoritmo 2: Classificador de face com K-NN clássico

Entrada: x_j , X_q e K
Saída: $S \leftarrow$ classe de maior ocorrência

início
 $N \leftarrow$ tamanho de X_q
para i de 1 até N **faça**

 Calcular a distância euclidiana de x_j em relação aos vetores de X_q :

$$D[i] = \sqrt{\sum_k (x_{j,k} - X_{q,k}[i])^2}$$

 $O \leftarrow$ ordenar(X_q, D)

 $S \leftarrow O[1 : K]$
fim
fim

A entrada é formada pelo vetor característico x_j da imagem de teste, o conjunto de vetores característicos X_q das imagens do conjunto de treinamento e o valor de K . Para cada face do conjunto de treinamento, a equação de distância euclidiana é aplicada em relação a face de teste. Na sequência, as faces do conjunto de treinamento são ordenadas tomando como referência o vetor de distância. Finalmente são selecionadas as K amostras mais próximas do vetor O , a classe de maior ocorrência desse subconjunto é atribuída como sendo a classe da imagem de teste.

Essa abordagem pressupõe que o sistema já havia sido treinado para reconhecer a face de teste, uma vez que alguma classe será atribuída. Dessa maneira não há como garantir que a resposta gerada esteja correta. Idealmente, se o conjunto de treinamento contém alguma face igual à face de teste, a essa face a distância Euclidiana será nula, nesse cenário, pode-se garantir que o resultado encontrado corresponde a classe da imagem de teste. Devido à variabilidade das variáveis envolvidas no processo de captura, dificilmente uma imagem será exatamente igual a outra. Portanto, para evitar uma resposta falsa é necessário definir um valor de limiar V_{th} pelo qual, todas as faces com distância abaixo dele, pertencem à classe da imagem de teste.

4.2 K-NN melhorado

O problema elucidado na seção anterior é devido ao critério genérico de atribuição de classe do K-NN clássico. Uma modificação desse algoritmo permite tornar o K-NN

mais assertivo. A solução consiste em definir um valor de *threshold* V_{th} para atribuir a uma imagem de teste, alguma classe já treinada. Essa modificação está explicitada no algoritmo a seguir.

Algoritmo 3: Classificador de face com K-NN melhorado

Entrada: x_j , X_q , K e V_{th}

Saída: classe dos elementos do conjunto de faces F

início

$N \leftarrow$ tamanho de X_q

para i de 1 até N **faça**

 Calcular a distância euclidiana de x_j em relação aos vetores de X_q :

$$D[i] = \sqrt{\sum_k (x_{j,k} - X_{q,k}[i])^2}$$

$O \leftarrow$ ordenar(X_q, D)

$S \leftarrow O[1 : K]$

$n \leftarrow 1$

$F \leftarrow [\emptyset]$

para i de 1 até K **faça**

se $D[i] \leq V_{th}$ **então**

$F[n] = S[i]$

$n++$

fim

fim

se $F \neq \emptyset$ e o número de classes em $F = 1$ **então**

return classe($F[1]$)

fim

senão

return face desconhecida

fim

fim

fim

A modificação realizada, consiste em testar, antes de atribuir qualquer classe, se os valores das K amostras do conjunto ordenado está abaixo de V_{th} . Se estiver, um novo vetor de faces F será povoado com as possíveis respostas. Se F não foi povoado, significa que não existe nenhuma imagem no conjunto de treinamento que descreve a classe da face de teste. Caso contrário, se o número de classes distintas em F for unitário, então,

a classe de qualquer elemento de F corresponde a classe da face de teste. Se existir mais de uma classe, não será atribuída nenhuma classe. O valor de V_{th} foi definido a partir da média das distâncias, isso será melhor explicado no capítulo seguinte onde será falado dos teste para encontrar esse *threshold*.

Resumidamente, o algoritmo desenvolvido considerada apenas elementos do conjunto de treinamento dentro de um determinado raio da amostra de teste e considera que a classificação não pode ser feita se os vizinhos mais próximos encontrados não pertencerem à mesma classe.

O que se segue no trabalho a partir deste ponto foi verificar quão boa essa modificação é para a aplicação desejada.

5 RESULTADOS E DISCUSSÕES

Para a implementação do sistema de reconhecimento facial projetado nesse trabalho, foi utilizado a linguagem **python** por prover mecanismos simples que agilizam alguns passos devido à existência de bibliotecas auxiliares como **opencv** (biblioteca de processamento de imagem), **numpy** (biblioteca de processamento matricial) e **matplotlib** (biblioteca para gerar gráficos).

A biblioteca *opencv* já implementa a metodologia de detecção em cascata de objetos por meio do método *CascadeClassifier*, bastando apenas fornecer a ele as características que devem ser usadas pelo detector para identificar o objeto de interesse. Tais características foram obtidas utilizando AdaBoost, conforme explicado no Cap. 2. Os dados encontrados com AdaBoost são armazenados em um arquivo XML. Este arquivo é o parâmetro de configuração do detector implementado no *opencv*. Para os testes realizados nesse trabalho, foram utilizados 4 arquivos XML, o primeiro contém as características para detectar a face, o segundo os olhos, o terceiro o nariz e o último a boca.

Seguindo o fluxo de atividade descrito no Capítulo 4, os dados de teste e treinamento foram extraídos do banco de dados de imagens, em seguida, utilizando *opencv* a região da face, olhos, nariz e boca foram identificadas. Com essas regiões detectadas foi gerado o vetor característico e finalmente esses dados foram aplicados no classificador K-NN modificado apresentado neste trabalho.

Para os testar o algoritmo de reconhecimento proposto, foram selecionados 40 imagens. Esse grupo contém a face de 4 pessoas, o conjunto de imagens de teste de cada face é composta por 10 imagens. De cada pessoa foram selecionadas aleatoriamente 5 imagens para o conjunto de treinamento e 5 para o conjunto de teste. A Figura 10 mostra o conjunto de imagens de treinamento e a Figura 11 mostra o conjunto de imagens de teste.

Dois testes foram realizados, o primeiro aplicando o algoritmo K-NN clássico e o segundo usando o algoritmo K-NN modificado. Os resultados de cada teste serão apresentados e discutidos a seguir.

Figura 10: Conjunto de imagens de treinamento.



5.1 Resultados

A Tabela 1 mostra a resposta do classificador para um $K = 2$. Do total de 20 imagens de testes, apenas 4 apresentaram resultados incorretos, totalizando 80% de acerto. O resultado apresentado, segue a mesma sequência das imagens mostradas na Figura 11. Na mesma tabela também é mostrado a média do módulo vetor distância para as K imagens.

Para verificar o desempenho do classificador para imagens que não fazem parte do conjunto de treinamento, foram incluídas no conjunto de testes 4 novas faces. O resultado é mostrado na Tabela 2.

A Figura 12 mostra o comportamento da média das distâncias para o conjunto de teste cuja as faces possuem amostras no conjunto de treinamento (Figura 12a) e para

Figura 11: Conjunto de imagens de teste.



faces que não possuem amostras no conjunto de treinamento (Figura 12b). Na Figura 12c as curvas sobrepostas das médias das faces treinadas e não treinadas, respectivamente, indicam que existe um limiar de separação entre as faces conhecidas (treinadas) e não conhecidas pelo sistema.

Analisando as curvas da Figura 12 observa-se que a média mínima das faces não treinadas é 0.0683. As curvas também mostram que ao utilizar-se esse valor como o limiar, podemos ter imagens do conjunto de treinamento que não serão reconhecidas.

Utilizando-se 0.06 como valor de limiar, isto é, $V_{th} = 0.06$ e executando-se o teste novamente usando o K-NN modificado, obtém-se os resultados apresentados na Tabela 3. Nota-se que o conjunto de respostas não desconhecida foi reduzido em 60%, em contrapartida, nenhuma face foi classificada de forma errada.

Para avaliar a influência de K nos resultados já apresentados, outro teste realizado

Tabela 1: Resposta para faces presente do conjunto de treinamento.

Imagem de teste	Resposta do classificador	Resposta correta	Média
barry15.jpg	['barry16.jpg', 'barry6.jpg']	CORRETO	0.0699
barry17.jpg	['barry16.jpg', 'barry18.jpg']	CORRETO	0.0762
barry2.jpg	['barry3.jpg', 'barry18.jpg']	CORRETO	0.0658
barry4.jpg	['barry3.jpg', 'barry6.jpg']	CORRETO	0.0394
barry7.jpg	['barry3.jpg', 'lisa2.jpg']	INCORRETO	0.0866
jenni1.jpg	['jenni2.jpg', 'jenni7.jpg']	CORRETO	0.0761
jenni15.jpg	['jenni16.jpg', 'stephen18.jpg']	INCORRETO	0.1004
jenni17.jpg	['jenni16.jpg', 'stephen18.jpg']	INCORRETO	0.0936
jenni3.jpg	['jenni2.jpg', 'jenni7.jpg']	CORRETO	0.0698
jenni6.jpg	['jenni7.jpg', 'jenni4.jpg']	CORRETO	0.0555
lisa1.jpg	['lisa2.jpg', 'lisa4.jpg']	CORRETO	0.0282
lisa11.jpg	['lisa2.jpg', 'lisa7.jpg']	CORRETO	0.1219
lisa17.jpg	['lisa15.jpg', 'lisa18.jpg']	CORRETO	0.0532
lisa3.jpg	['lisa4.jpg', 'lisa2.jpg']	CORRETO	0.0342
lisa5.jpg	['lisa4.jpg', 'lisa7.jpg']	CORRETO	0.0526
stephen11.jpg	['stephen3.jpg', 'stephen7.jpg']	CORRETO	0.0513
stephen16.jpg	['stephen15.jpg', 'barry18.jpg']	INCORRETO	0.0432
stephen2.jpg	['stephen3.jpg', 'stephen5.jpg']	CORRETO	0.0825
stephen4.jpg	['stephen3.jpg', 'stephen7.jpg']	CORRETO	0.0702
stephen6.jpg	['stephen7.jpg', 'stephen5.jpg']	CORRETO	0.0269

foi refazer a mesma análise com diferentes valores de K . Na Figura 13 são mostrados 5 resultados para $K = [1, 5]$. Conforme esperado, o valor da média aumenta proporcional a K , mantendo o comportamento médio. Esse comportamento é visto na Figura 13a para $K = 1$ até a Figura 13e para $K = 5$. Logo, percebe-se que o valor de K não influi diretamente na assertividade dos resultados obtidos uma vez que não houve separação das regiões de intersecção.

Matematicamente foi dito que para faces de uma mesma pessoa, as distâncias entre o vetor de teste e o conjunto de treinamento seriam menores. Essa propriedade pode ser percebida geometricamente construindo um gráfico com as medidas de distâncias de uma face de teste em relação a todo o conjunto de treinamento contendo todas as faces. O resultado é mostrado na Figura 14. As curvas mostram o resultado das distâncias para as 5 primeiras faces do conjunto de teste. Como para essas faces existem 5 faces de treinamento, observa-se que a região mais baixa da curva são os 5 primeiros pontos. Essa região corresponde exatamente as imagens do conjunto de treinamento que contém o rosto da mesma pessoa cujo rosto está presente na imagem de teste.

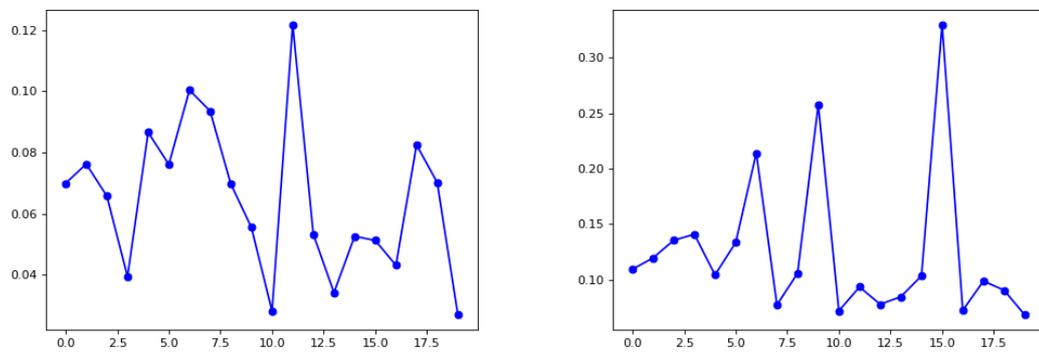
Tabela 2: Resposta para faces não presentes no conjunto de treinamento.

Imagem de teste	Resposta do classificador	Resposta correta	Média
adrian1.jpg	['lisa15.jpg', 'lisa18.jpg']	INCORRETO	0.1094
annanena1.jpg	['jenni7.jpg', 'jenni4.jpg']	INCORRETO	0.1196
anon-one1.jpg	['lisa2.jpg', 'lisa4.jpg']	INCORRETO	0.1355
catherine11.jpg	['stephen3.jpg', 'barry6.jpg']	INCORRETO	0.1409
hin4.jpg	['stephen3.jpg', 'stephen5.jpg']	INCORRETO	0.1043
jsheenan1.jpg	['stephen3.jpg', 'stephen5.jpg']	INCORRETO	0.1334
kirsty2.jpg	['jenni2.jpg', 'lisa18.jpg']	INCORRETO	0.2142
marie1.jpg	['lisa18.jpg', 'lisa15.jpg']	INCORRETO	0.0773
marie11.jpg	['lisa18.jpg', 'jenni2.jpg']	INCORRETO	0.1059
miranda2.jpg	['barry6.jpg', 'barry3.jpg']	INCORRETO	0.2576
neil2.jpg	['barry16.jpg', 'barry18.jpg']	INCORRETO	0.0719
olive1.jpg	['jenni4.jpg', 'jenni7.jpg']	INCORRETO	0.0934
paul1.jpg	['barry6.jpg', 'barry16.jpg']	INCORRETO	0.0778
peter1.jpg	['lisa2.jpg', 'lisa4.jpg']	INCORRETO	0.0846
peter11.jpg	['barry6.jpg', 'barry3.jpg']	INCORRETO	0.1035
ruth1.jpg	['jenni2.jpg', 'lisa15.jpg']	INCORRETO	0.3301
terry-johnstone1.jpg	['barry16.jpg', 'barry6.jpg']	INCORRETO	0.0723
tracy1.jpg	['lisa7.jpg', 'lisa2.jpg']	INCORRETO	0.0988
trevor1.jpg	['lisa2.jpg', 'lisa7.jpg']	INCORRETO	0.0905
trevor3.jpg	['lisa2.jpg', 'lisa7.jpg']	INCORRETO	0.0683

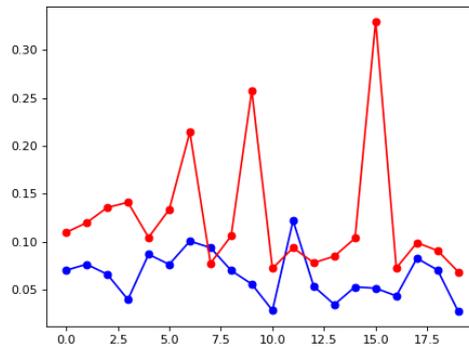
Tabela 3: Resposta com limiar de 0.06

Imagem de teste	K faces possíveis	Resposta do classificador	Média
barry15.jpg	['barry16.jpg', 'barry6.jpg']	DESCONHECIDA	0.0699
barry2.jpg	['barry3.jpg', 'barry18.jpg']	DESCONHECIDA	0.0658
barry4.jpg	['barry3.jpg', 'barry6.jpg']	barry	0.0394
jenni3.jpg	['jenni2.jpg', 'jenni7.jpg']	DESCONHECIDA	0.0698
jenni6.jpg	['jenni7.jpg', 'jenni4.jpg']	jenni	0.0555
lisa1.jpg	['lisa2.jpg', 'lisa4.jpg']	lisa	0.0282
lisa17.jpg	['lisa15.jpg', 'lisa18.jpg']	lisa	0.0532
lisa3.jpg	['lisa4.jpg', 'lisa2.jpg']	lisa	0.0342
lisa5.jpg	['lisa4.jpg', 'lisa7.jpg']	lisa	0.0526
stephen11.jpg	['stephen3.jpg', 'stephen7.jpg']	stephen	0.0513
stephen16.jpg	['stephen15.jpg', 'barry18.jpg']	DESCONHECIDA	0.0432
stephen6.jpg	['stephen7.jpg', 'stephen5.jpg']	stephen	0.0269
trevor3.jpg	['lisa2.jpg', 'lisa7.jpg']	DESCONHECIDA	0.0683

Figura 12: Curvas com a média das distâncias mostradas nas Tabelas 1 e 2.

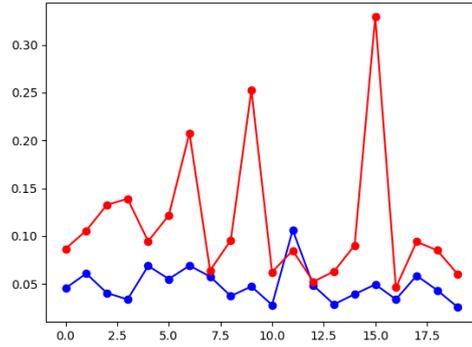


(a) Média das K distâncias das faces de teste treinadas. (b) Média das K distâncias das faces de teste não treinadas.

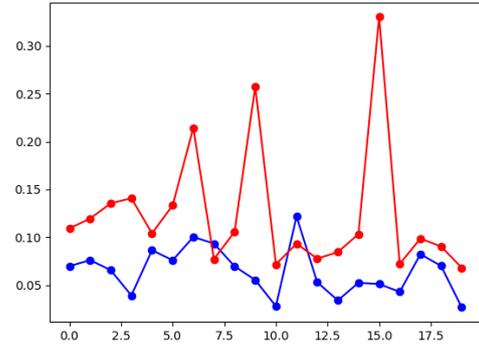


(c) Média das K distâncias das faces treinadas e não treinadas.

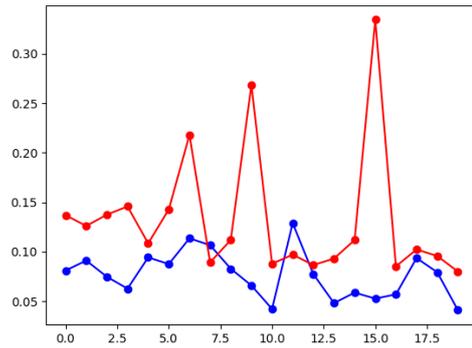
Figura 13: Curvas com a média das distâncias mostradas nas Tabelas 1 e 2 para K de 1 até 5.



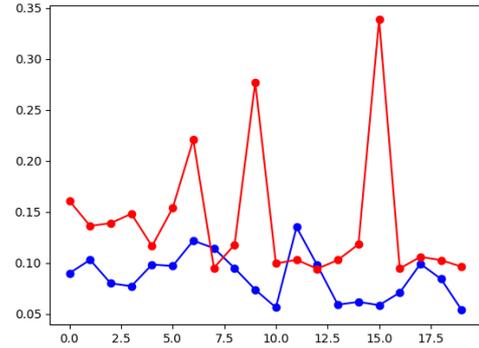
(a) $K=1$



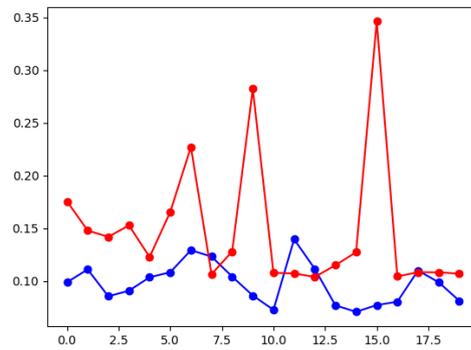
(b) $K=2$



(c) $K=3$

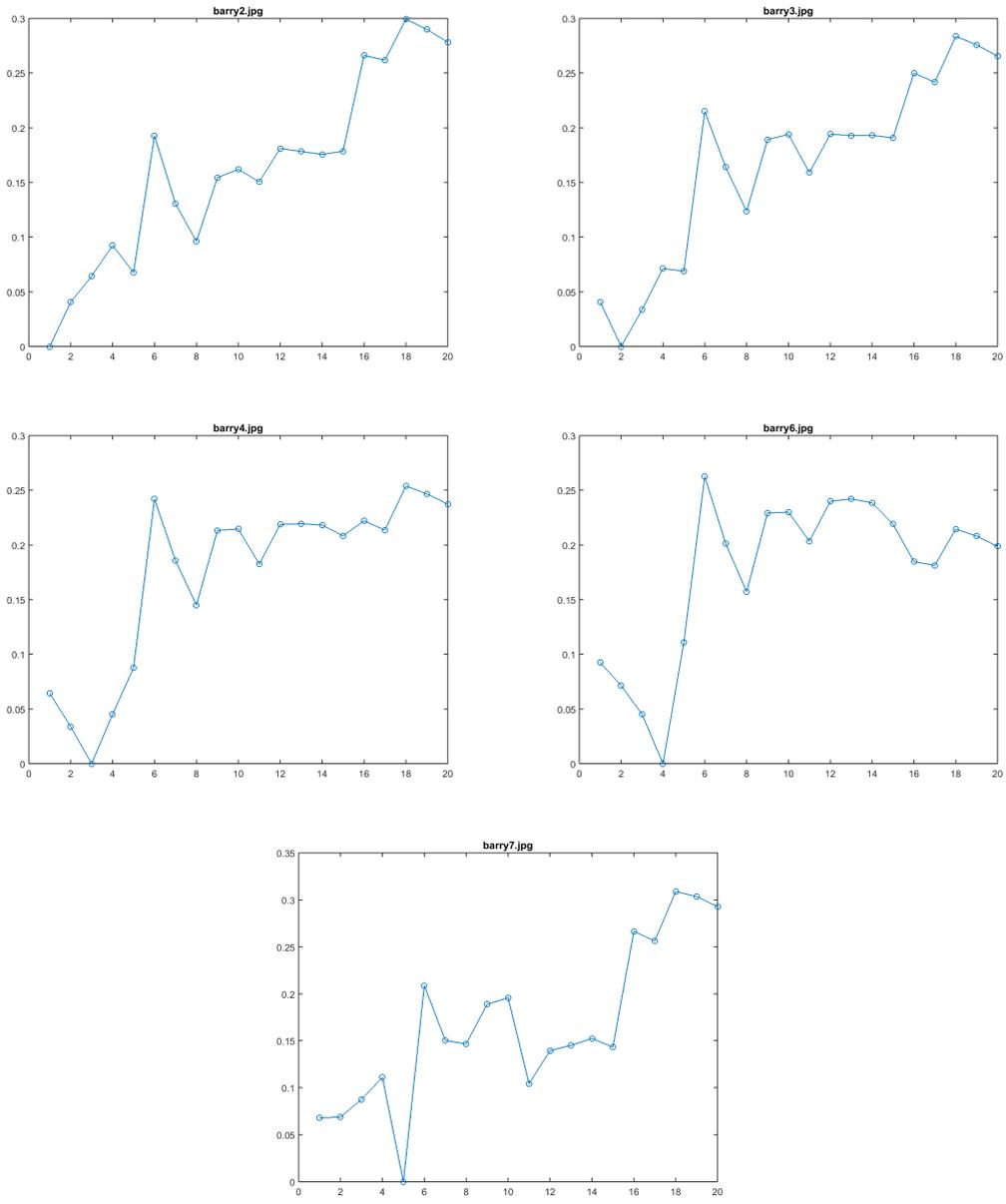


(d) $K=4$



(e) $K=5$

Figura 14: Curvas de distâncias para as 5 primeiras imagens de teste.



6 CONCLUSÃO E TRABALHOS FUTUROS

Foi mostrado que um sistema de visão computacional depende de um conjunto de ferramentas matemáticas que descreve desde como uma imagem é formada (captura), como ela pode ser melhorada (filtragem), como partes dela podem ser selecionadas (segmentação), como extrair informações quantitativas da imagem ou de objetos presentes nela (extração de característica) e de como usar esses dados para programar um sistema de reconhecimento. Dependendo do ambiente que as imagens são capturadas, a etapa de filtragem pode ser omitida.

Portanto, a construção de um sistema de visão computacional específico para reconhecer faces de pessoas, resulta da combinação de algoritmos de processamento de imagem aplicados à detecção de face, combinado com técnicas de reconhecimento de padrões. O sistema apresentado, utilizou Haar Cascade na etapa de detecção e K-NN na fase de reconhecimento de padrões. A escolha dessas técnicas foi motivada pela robustez do detector em cascata e a simplicidade do K-NN.

Os resultados obtidos demonstraram que apesar da simplicidade do K-NN, com poucas modificações é possível construir um classificador com uma taxa de acerto bastante elevada. Com o K-NN clássico, a taxa de acerto depende do conjunto de treinamento e teste. No teste execução, a taxa de erro foi de 20%. Para que a resposta independa do conjunto de treinamento, isto é, garantir que se não houver exemplos no conjunto de treinamento da face testada, o classificador responda como face desconhecida, foi necessário melhorar o K-NN inserindo critérios menos genéricos de seleção dos vizinhos mais próximos. Sob essa abordagem discutida no texto, a taxa de acerto pode chegar a 100%.

O sistema de reconhecimento facial proposto pode ser aplicado em sistema de segurança, equipamentos de acesso residencial ou qualquer outra aplicação que necessite de buscar informações por meio de imagens. Para trabalhos futuros, pretendemos experimentar o sistema implementado capturando imagens direto de uma câmera em ambientes

controlado e não controlado e aplicá-lo na construção de um sistema de monitoramento e equipamentos de controle de acesso a locais reservados, como o porteiro eletrônico. Outra sugestão de melhoria é criar uma forma automática de definir o valor de *threshold* usado na abordagem K-NN melhorada e testar outras técnicas de reconhecimento de padrões como o uso de RNA.

REFERÊNCIAS

- LOPES, E. Costa. Detecção de Faces e Características Faciais. Pós-Graduação em Ciência da Computação - PUCRS, 2012.
- LOZANO, P. Tomás. Autonomous robot vehicles. Springer Science Business Media, 2012.
- BACKES, A. Ricardo; JUNIOR, Jarbas J. M. de Sá. Introdução à visão computacional usando Matlab. Editora Alta Books, 2016.
- YAN, Ming-Hsuan; DAVID J. Kriegman; NARENDA Ahuja. "Detecting faces in images: A survey." IEEE Transactions on pattern analysis and machine intelligence 24.1 (2002): 34-58..
- KOTROPOULOS, Constantine; IOANNIS Pitas. "Rule-based face detection in frontal views." Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on. Vol. 4. IEEE, 1997.
- WANG, Jian-Gang; ERIC, Sung. "Frontal-view face detection and facial feature extraction using color and morphological operations." Pattern recognition letters 20.10 (1999): 1053-1068.
- BHUIYAN, M. A., et al. "Face detection and facial feature localization for human-machine interface." NII journal 5.1 (2003): 25-39.
- SABER, Eli; TEKALP, A. Murat. "Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions." Pattern Recognition Letters 19.8 (1998): 669-680.
- RICKERT, Thomas Dale. Texture-based statistical models for object detection in natural images. Diss. Massachusetts Institute of Technology, 1999.
- LUO, Huitao; ELEFThERIADIS, Alexandros. "On face detection in the compressed domain." Proceedings of the eighth ACM international conference on Multimedia. ACM, 2000.
- ALATTAR, Adnan M.; SARAH A. Rajala. "Facial features localization in front view head and shoulders images." Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on. Vol. 6. IEEE, 1999.
- LEE, Hyong Woo, et al. "Automatic face and facial features detection." Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on. Vol. 1. IEEE, 2001.

- VEZHNEVETS, Vladimir. "Face and facial feature tracking for natural human-computer interface." International Conference on Computer Graphics between Europe and Asia (GraphiCon-2002). 2002.
- WONG, Kwok-Wai; LAM, Kin-Man; SIU, Wan-Chi. "An efficient algorithm for human face detection and facial feature extraction under different conditions." Pattern Recognition 34.10 (2001): 1993-2004.
- TURK, Matthew A.; PENTLAND, Alex P. "Face recognition using eigenfaces." Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE, 1991.
- KIRBY, Michael; SIROVICH, Lawrence. "Application of the Karhunen-Loeve procedure for the characterization of human faces." IEEE Transactions on Pattern analysis and Machine intelligence 12.1 (1990): 103-108.
- ROWLEY H. A. S. Baluja. Neural Network-Based Face Detection, Computer Vision and Pattern Recognition, 1996.
- ROWLEY, H. A. S. Baluja; KANADE, Takeo. Rotation Invariant Neural Network-Based Face Detection, IEEE Proceedings on Computer Vision and Pattern Recognition, 1998.
- ROWLEY, H. A. S. Baluja, KANADE, Takeo. "Neural network-based face detection." IEEE Transactions on pattern analysis and machine intelligence 20.1 (1998): 23-38.
- VIOLA, Paul; JONES, Michael. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.
- RUSSEL. Stuart; NORVIG. Peter. Artificial Intelligence - A Modern Approach. 3^a Ed. Editora Pearson, 2010.
- Psychological Image Collection at Stirling (PICS). Disponível em <http://pics.psych.stir.ac.uk/>. [Último acesso ocorreu em setembro de 2017].