

CENTRO UNIVERSITÁRIO DINÂMICA DAS CATARATAS

THIAGO CORREA LIMA DA SILVA

**ESTUDO E IMPLEMENTAÇÃO DE UM SISTEMA DE
RECONHECIMENTO DE FACES EM VÍDEO COM OS
ALGORÍTIMOS *EIGENFACES* E *FISHERFACES* NA
LINGUAGEM JAVA**

FOZ DO IGUAÇU

2018

THIAGO CORREA LIMA DA SILVA

ESTUDO E IMPLEMENTAÇÃO DE UM SISTEMA DE
RECONHECIMENTO DE FACES EM VÍDEO COM OS ALGORÍTIMOS
EIGENFACES E *FISHERFACES* NA LINGUAGEM JAVA

Trabalho de conclusão de curso apresentado como
requisito obrigatório para obtenção do título de Ba-
charel em Ciência da Computação do Centro Univer-
sitário Dinâmica das Cataratas.

Orientador: Alessandra Bussador

FOZ DO IGUAÇU

2018

SUMÁRIO

1	INTRODUÇÃO	3
1.1	BREVE HISTÓRIA	3
1.2	PROBLEMÁTICA	4
1.2.1	DETECÇÃO E AQUISIÇÃO DA FACE	4
1.2.2	TREINAMENTO E RECONHECIMENTO DA FACE	6
1.3	OBJETIVOS	8
1.3.1	ORGANIZAÇÃO DO TRABALHO	8
1.3.2	JUSTIFICATIVA	9
2	REVISÃO BIBLIOGRÁFICA	13
2.1	AQUISIÇÃO E PROCESSAMENTO DE IMAGENS E VÍDEO .	13
2.1.1	Imagem e Vídeo Digital	13
2.1.2	Formato PNG	15
2.1.3	Aquisição da Imagem e de Vídeo	15
2.1.4	Processamento da Imagem	16
2.1.4.1	Conversão em Escala de Cinza	17
2.1.4.2	Escalonamento	17
2.1.4.3	Equalização de Histograma	18
2.1.4.4	Segmentação	18
2.1.4.5	Classificação	19
2.2	DETECÇÃO DE FACES COM CARACTERÍSTICAS HAAR . .	19
2.2.1	As Características Haar (<i>Haar Features</i>)	19
2.2.2	Algoritmo Viola-Jones (<i>Haar Cascades Classifier</i>)	20
2.3	<i>EIGENFACES</i> E ANÁLISE DE COMPONENTE PRINCIPAL (ACP)	22
2.3.1	Análise de Componentes Principais (ACP)	25
2.3.2	ACP E <i>Eigenfaces</i>	27
2.3.2.1	Utilizando Eigenfaces como Componente Principal	28
2.3.2.2	O Reconhecimento	30
2.4	CONSIDERAÇÕES FINAIS	33
	REFERÊNCIAS	34

1 INTRODUÇÃO

O reconhecimento de face é um dos campos de pesquisa mais interessantes e importantes nas ultimas duas décadas (CHAO, 2011). Grandes empresas estão na corrida para ver quem desenvolve o melhor algoritmo de reconhecimento de faces, e apenas à pouco tempo, duas delas (Google e Baidu) conseguiram taxas de erros abaixo das apresentadas por seres humanos (que se acera aos 8%) (PORTAL, 2017).

As possibilidades de uso de um sistema de reconhecimento de faces são vastas (desde o uso comercial como empresas que querem identificar seus clientes e oferecer um serviço personalizado ao uso da polícia para o combate criminal e forense). Chega a ser um assunto polêmico em alguns círculos pois podem inferir em invasão de privacidade e privação da liberdade como a conhecemos.

As pesquisas sobre o assunto envolvem conhecimentos de disciplinas como neurociência, psicologia, computação visual, reconhecimento de padrões, processamento de imagens, matemática avançada (CHAO, 2011), por isso apresenta um grande problema computacional. Além do problema de manipulação de imagens, de identificação e reconhecimento das faces, deve-se considerar problemas de persistência destas, a questão de treinamento das faces, velocidade de verificação, de extração de dados de imagens e vídeos (tarefa em que a linguagem Java não tem boa fama) e desempenho em geral. Não houve um grupo de desenvolvedores ou indivíduos capaz de encerrar esse assunto.

1.1 BREVE HISTÓRIA

O reconhecimento automático de faces é um conceito relativamente novo. Desenvolvido na década de 60, o primeiro sistema semi-automático requeria que um administrador do sistema localizasse pontos da imagem como olhos, boca, nariz, etc, e depois um algoritmos comparava as distâncias entre estes pontos em diferentes fotos para dar um resultado de comparação (SCIENCE; (NSTC), 2009). Nos anos 80, algoritmos simples foram criados para automatizar este processo. Em 1988, Kirby e Sirovich aplicaram Análise de Componente Principal, uma técnica padrão de álgebra linear, para o problema de reconhecimento de face. Este foi considerado um marco pois mostrou que menos que 100 valores são requeridos para codificar uma imagem de uma face normalizada e bem posicionada (SCIENCE; (NSTC), 2009). Em 1991, Turk e Pentland descobriu o erro residual das comparações feitas com a técnica *eigenfaces* poderiam ser usadas para detectar faces em uma imagem - uma descoberta que permitiu detecção em tempo-real. Apesar desta abordagem ser imitado a posição da face, qualidade da imagem e fatores de ambiente, ela criou um significant interesse no desenvolvimento de tecnologias automáticas para reconhecimento (SCIENCE; (NSTC), 2009). A tecnologia capturou a atenção da mídia

e do público em janeiro de 2001 no evento *SuperBowl*, que capturou rostos das imagens das câmeras de vigilância e comparou com fotos digitais "3x4" de uma base de dados apresentando rostos similares (SCIENCE; (NSTC), 2009).

Hoje, o reconhecimento de faces está sendo usado no combate à fraude de passaportes e cédulas de identidades onde a foto é padronizada com ambiente controlado, aplicativos de celulares e redes sociais para entretenimento, enquanto que grandes governos e organizações patrocinam, incentivam e desafiam empresas para conseguir o algoritmo ideal de reconhecimento (INTRONA, 2010).

1.2 PROBLEMÁTICA

A face é um objeto 3D que é iluminada por uma grande variedade de fontes de luz e envolvida por um fundo com “dados arbitrários” (inclusive outras faces). Portanto, a aparência que uma face possui quando projetada para um modelo 2D pode variar tremendamente. Na verdade, problemas de iluminação e de segmentação “*foreground-background*” tem sido questões pertinentes no campo da computação gráfica e visual como um todo (JEBARA, 1995). Sendo assim o primeiro problema a enfrentar seria a aquisição de imagem e suas condições.

1.2.1 DETECÇÃO E AQUISIÇÃO DA FACE

Em geral, em uma imagem retirada de vídeo um digital de nível amador, o módulo de identificação da face deve encontrar condições luminosas descontroladas, alta variação de poses, maquiagem, mudanças nos pelos faciais, adoecimento, envelhecimento, oclusões da face por interferência, roupa ou cabelo, enfim, uma incontável gama de variáveis (JEBARA, 1995). De fato, há diversos desafios e fatores chaves que podem significativamente impactar a performance da identificação e reconhecimento da face e os pontos de verificação (*matching scores*).

Na Figura 1, lista-se exemplos de alguns destes desafios em imagens, respectivamente:

- (a) Variação de iluminação;
- (b) Variação de poses ou pontos de visão;
- (c) Envelhecimento;
- (d) Expressões faciais e estilo da face (pelos faciais ou maquiagem);
- (e) Oclusão.

A aquisição de faces em vídeos (diz-se *tracking* ou rastreamento) segue a mesma lógica e tem os mesmo problemas de aquisições de faces estáticas. O rastreamento nada



FIGURA 1 – Exemplos de Variação

FONTE: *The Researchgate.net*

mais é do que a detecção contínua da face em *frames* advindos de um vídeo, com um forte problema adicional de que deve-se manter a usabilidade do sistema em um computador contemporâneo de baixa a média performance. Em outras palavras, o processamento envolvido deve ser eficiente o bastante considerando o tempo de execução do vídeo (*frames* por segundo) e do sistema e tempo de armazenamento e consulta de dados.

A performance da detecção da face é uma questão-chave no processo de reconhecimento, porém considerada bem para poses de faces frontais, deste a década de 90. A detecção de faces pode ser considerada um caso específico da área de detecção de objetos, alcançando confiáveis (ELECTRICAL; IEEE, 1997).

Em um trabalho feito em 2003 por (KIM JOON HYUNG SHIM, 2003) na Universidade de Stanford - Califórnia, utilizaram a técnica *EigenFaces* para detecção, acrescido de algoritmo de identificação de gênero de sexo, com um *Pentium 3* de 700 Mhz e menos

de 500 megas de memória, alcançaram resultados que superam 93% de taxa de sucesso nas condições mais adversas, considerando iluminação e escala (numero grande de faces em uma imagem). Outros *benchmarks* mais recentes (2015) apresentam resultados que superam os 97.2% (YANG PING LUO, 2015).

O problema com estas técnicas relativamente antigas, é que também reconhecem fotos de fotos, ou desenhos de faces como legítimas faces, por vezes até com pontuação bastante para uma verificação de sucesso com uma face real. Outro problema é estas técnicas não funcionam num angulo de perfil, ou de qualquer outro ângulo que não seja o frontal.

Novas técnicas de detecção não-frontais estão sendo implementadas, bem como as de modelagem sub-espacial 3D e comparações com reconhecimento de padrões baseados em aprendizado de máquinas e redes neurais, são fundamentos para os sistemas de reconhecimento de face avançados, capazes de reconhecer não só a face, mas também a estrutura cranial do alvo (BARBU NATHAN LAY, 2015).

Após a face ser identificada e localizada na imagem, para que sirva ao processo de reconhecimento, deve-se recortar esta face da imagem e performar algumas operações gráficas sobre a mesma. Nesta fase são consideradas atividades de corte da face na região localizada, escalamento e correção de rotação, transformar em preto e branco, normalizar a imagem para minimizar as condições de ambiente da foto, deixando a face pronta para a próxima fase do processo: o treinamento.

1.2.2 TREINAMENTO E RECONHECIMENTO DA FACE

Esta fase consiste em manipular a face recortada da imagem ou *frame* e já normalizada e tratada de forma a extrair informações e características desta para que se possa salvar de alguma forma e relacionar estas características com a pessoa ou alvo. Nesta fase é importante aquistar fotos da mesma face mostrando diferentes expressões faciais e situações de luminosidade e posição.

A forma que isso pode ser feito depende totalmente do algoritmo de reconhecimento envolvido. Há várias técnicas e algoritmos de reconhecimento, como (WóJCIK KONRAD GROMASZEK, 2016):

- reconhecimento baseado em redes neurais;
- reconhecimento baseado em processamento 3D;
- reconhecimento baseado em descritores de face;
- reconhecimento baseado em reconstrução;
- reconhecimentos tradicionais ou clássicos, dentre outros;

Este trabalho se focará em dois dos algoritmos clássicos, famosos por serem pioneiros e objetos de muito estudo, testes e documentação: *EngenFaces* e *FisherFaces*. Os algoritmos convencionais podem ser divididos em duas categorias: caracterização holística ou linear, sendo que *EngenFaces* e *FisherFaces* fazem parte da primeira, que fazem parte de outra subdivisão de métodos de projeção linear chamada Análise de Componentes Principais - ACP (ou no inglês, Principal Component Analysis – PCA) (WóJCIK KONRAD GROMASZEK, 2016).

Basicamente, o método ACP consiste em tratar a imagem de uma forma uniforme, coletar características da mesma transformando-as em valores numéricos e disponibilizá-las em um plano cartesiano, que pode ter mais de três dimensões. Este processo visa destacar discrepâncias da face, ou seja, padrões de mudança de constaste, de relevo ou sombreamentos, ou diferença de cores. A transformação destes valores manipulados de volta em imagens, cria os chamados *EngenFaces* (Fases de fantasmas, em Alemão), pois como mostra a Figura 2, é devido a sua estranha aparência (DAVISON, 2013).

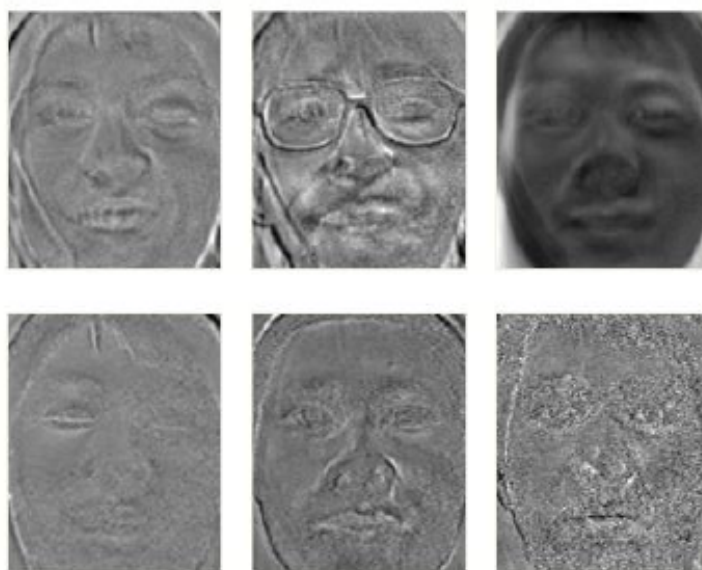


FIGURA 2 – Exemplos de EigenFaces

FONTE: (DAVISON, 2013)

A ideia é quem uma imagem de treinamento pode ser decomposta em uma soma multiplas *eigenfaces* medidas e disponibilizadas em uma sequencia especial.

Uma maneira mais simples de representar o relacionamento entre *eigenfaces* e imagens das faces é quem estas faces são disponibilizadas em um espaço multidimensional, ou um plano cartesiado "n" dimensões onde os eixos do plano são as *eigenfaces* (DAVISON, 2013) como ilustra a Figura 3.

Tendo a face (ou faces) relacionada com o alvo, representado em valores e disponibilizada em um plano cartesiano de “n” dimensões, resta a verificação ou reconhecimento da face contra outra.

Para o reconhecimento da face propriamente dito, o mesmo processo do treinamento deve ser repetido e a face representada e disponibilizada no mesmo plano cartesiano das faces treinadas (DAVISON, 2013). A distância entre as duas faces representadas no plano cartesiano é medida, e basicamente, quando menor a distância, maior é a taxa de reconhecimento. Ou seja, uma distância zero entre as duas faces representadas no plano seria uma correspondência perfeita. A Figura 3 abaixo ilustra esta descrição:

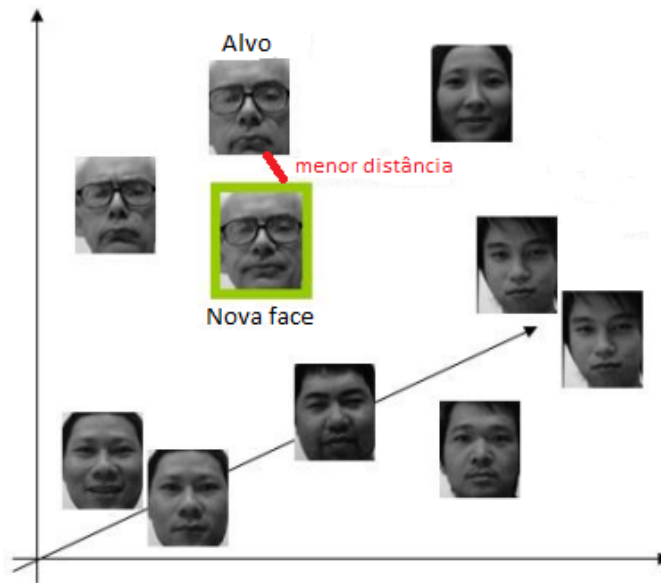


FIGURA 3 – Exemplos de EigenFaces
FONTE: Adaptado de (DAVISON, 2013)

Este plano cartesiano, na prática, pode ser um arquivo binário ou texto, comumente chamado de "*bundle*".

1.3 OBJETIVOS

Este projeto propõe a implementação de um Sistema de Reconhecimento de Faces em Vídeo utilizando dois dos primeiros algoritmos desenvolvidos para este fim: *FisherFaces* e *EigenFaces*.

O algoritmo *EigenFaces* será implementado seguindo as instruções e exemplos de código de (DAVISON, 2013). Quanto ao algoritmo *FishFaces*, será usada a implementação da biblioteca OpenCV. Haverá uma tentativa de fazer com que os dois algoritmos funcionem juntos para que se use os pontos fortes de cada um, eliminando os fracos, e se possível melhorar a taxa de reconhecimento.

1.3.1 ORGANIZAÇÃO DO TRABALHO

O desenvolvimento do sistema, basicamente, se fará na exploração das seguintes fases:

- Análise das bibliotecas e suas versões para melhor escolha;
- Preparação do ambiente;
- Implementação da interface e manipulação de imagens (e vídeos) utilizando componentes Java e as bibliotecas gráficas (para recorte de frames e imagens, aplicação de filtros, mineração de dados da imagem, manipulação de cores, iluminação, normalização, etc);
- Implementação do algoritmo EigenFaces;
- Detecção de face em imagens tiradas com câmera;
- Treinamento / reconhecimento de face na imagem;
- Incrementar a implementação feita acima para o treinamento e reconhecimento em vídeo tirado da câmera em tempo real;
- Integrar os algoritmos de outras implementações;
- Performar uma bateria de testes, e comparar resultados.

O objetivo final é estruturar toda esta pesquisa e o respaldo técnico que desses algoritmos, contemplar problemas tecnológicos da linguagem JAVA e de ambiente de desenvolvimento e a partir disso criar um sistema de reconhecimento de faces em vídeo, analisando seus resultados e disponibilizar a pesquisa e anotações para fins didáticos, finalmente disponibilizando todo o trabalho e o código aberto ao público desenvolvedor e acadêmico.

1.3.2 JUSTIFICATIVA

O reconhecimento de faces é uma das tecnologias mais estudadas e complexas de hoje em dia. As maiores empresas e governos do planeta estão na corrida para se acercarem aos 100% de acerto em seus reconhecimentos e governos e organizações como FBI e NSA investem e desafiam os produtores desta tecnologia (SCIENCE; (NSTC), 2009).

Eis na Tabela 1 algumas das empresas que produzem tecnologias de reconhecimento de ponta:

TABELA 1 – Empresas que produzem tecnologias de reconhecimento de faces

Empresas	Local	website
Acsys Biometrics Corp.	Burlington, Canada	http://www.acsysbiometrics.com
Animetrics, Inc.	Conway, NH	http://www.animetrics.com
Asia Software Ltd.	Almaty, Kazakhstan	http://www.asia-soft.com/frs/en/main
Aurora Computer Services Ltd.	Northampton, United Kingdom	http://www.facerec.com
Bioscrypt [Acquired by L-1 Identity Solutions in 2008; Bioscrypt acquired A4 Vision in 2007]	Toronto, Canada	http://www.bioscrypt.com
C-VIS Computer Vision und Automation GmbH [Acquired by Cross Match Technologies]	Palm Beach Gardens, FL	http://www.crossmatch.com
Carnegie Mellon University	Pittsburgh, PA	http://www.ri.cmu.edu/labs/lab_51.html
Cognitec Systems GmbH	Dresden, Germany	http://www.cognitec-systems.de
Cybula Ltd.	York, United Kingdom	http://www.cybula.com
Diamond Information Systems (DIS)	Beijing, China	http://www.disllc.net
FaceKey Corp.	San Antonio, TX	http://www.facekey.com
Neurotechnology [Formerly Neurotechnologija]	Vilnius, Lithuania	http://www.neurotechnology.com
Neven Vision [Acquired by Google in 2006; Formerly Eyematic Interfaces, Inc.]	Mountain View, CA	http://www.google.com/corporate
New Jersey Institute of Technology (NJIT)	Newark, NJ	http://www.cs.njit.edu/liu/facial_recognition_VPlab/index.html
Nivis, LLC	Atlanta, GA	http://www.nivis.com
Old Dominion University	Norfolk, VA	http://www.lions.odu.edu/org/vlsi/demo/vips.htm
OmniPerception Ltd.	Surrey, United Kingdom	http://www.omniperception.com
Omron	Kyoto, Japan	http://www.omron.com/r_d/coretech/vision
Panvista Limited	Sunderland, United Kingdom	http://www.panvista.co.uk

FONTE: Adaptado de (SCIENCE; (NSTC), 2009)

As pioneiras estão avançando exponencialmente no assuntos e gigantes como a Google e a Baidu já conseguem taxas de erros menores que humanos, com rapidez de

verificação sem precedentes, como mostra a Figura 4.

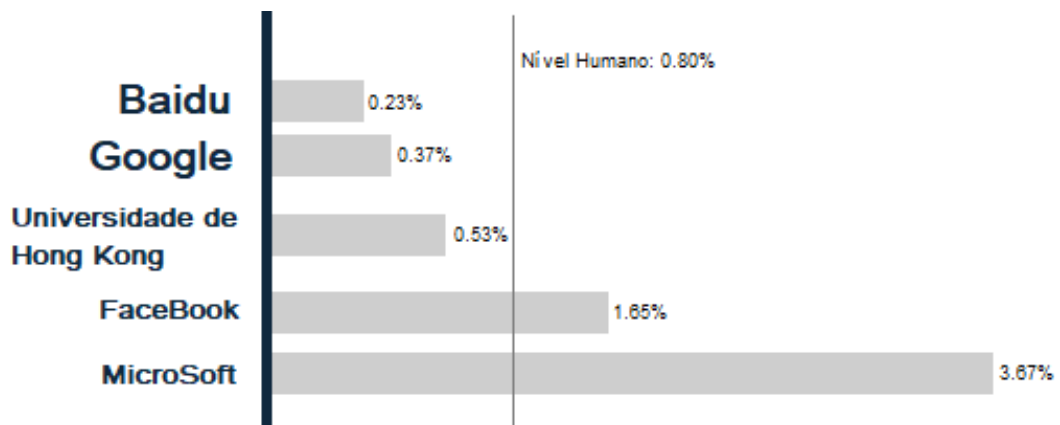


FIGURA 4 – Comparação de sistemas de reconhecimento de faces em percentagem (%) de erros

FONTE: Adaptado de (PORTAL, 2017)

A Figura 4 mostra que a tecnologia da gigante chinesa Baidu está liderando a corrida com apenas 0.23% de taxa de erro, contra 0.37% da empresa americana Google (PORTAL, 2017). Ambas já conseguiram ultrapassar as taxas alcançadas por seres humanos, que como mostra a Figura 4, atinge a taxa de 0.8% de erros. Como mostra (PORTAL, 2017), os humanos em geral não são uma fonte 100% precisa para reconhecimento de faces. Humanos podem facilmente confundir pessoas, ou não reconhecê-las pela idade ou por mudança de aparência ou estilo, ou até esquecê-las.

No final de 2016, a Baidu implantou seu sistema em sua histórica “Cidade de Água”, substituindo totalmente o sistema de cartões e *tickets* com 99% de sucesso (VERGE.COM, 2016), e já estão no processo de implantação em outros parques temáticos. Policiais chi-

neses estão usando óculos de reconhecimento que verificam instantaneamente as faces de turistas e procurados. Com os armazenamentos e reconhecimentos dominando aplicativos de celulares e redes sociais, e a polícia com grande interesse forense no assunto, e outras infinitas possibilidades, obviamente esta tecnologia fará parte do dia a dia do nosso futuro, sendo assim é necessário estudá-la, entendê-la e desenvolvê-la.

2 REVISÃO BIBLIOGRÁFICA

Vários trabalhos serviram como fonte de conhecimento e suporte para o entendimento e desenvolvimento deste tema. Alguns foram considerados mais importantes por abordar detalhadamente o problema de reconhecimento, outros foram buscados para se entender melhor, ou ter outra visão sobre algum detalhes específico do problema.

Trabalhos de autores como (SILVA, 2009) ajudaram a estruturar e organizar o tema por abordar tópicos similares e por possuir riqueza de detalhes no assunto de análise de componentes apresentando uma implementação em *MatLab* (um *software* interativo voltado para o cálculo numérico).

O trabalho de (BALAN, 2009) foi utilizado para se aprofundar no tema de imagens digitais, assim como (GONZALEZ, 2010) que também esclareceu o funcionamento de escalonamento, normalização e mono cromatização de imagens. O autor Dr. Andrew Davison em seus trabalhos, dentre os quais, (DAVISON, 2013) se destacou pela ótima didática facilitando o entendimento de Análise de Componentes e sua relação com *Eigen-Faces*, disponibilizando ainda exemplos de implementação em *JAVA*.

Os trabalhos (SMITH, 2002) e (KITANI,) foram estudados como tutoriais do processo de ACP e será usado posteriormente como guias de implementação do algoritmo.

Nas próximas seções serão explanados os conteúdos destes e de outros trabalhos disponibilizados em uma ordem didática.

2.1 AQUISIÇÃO E PROCESSAMENTO DE IMAGENS E VÍDEO

Com a abrangência dos sistemas de comunicação, com difusão de conhecimentos informações pelos diversos meios, captar, armazenar e processar imagens se tornou necessidade fundamental. Para o reconhecimento de faces em vídeo, o sucesso deste processo é pré-requisito para o funcionamento dos algoritmos.

As sessões a seguir contemplam fundamentos básicos sobre imagem, vídeo, a alguns de seus processamentos necessários para a aplicação dos algoritmos de detecção e reconhecimento de faces.

2.1.1 Imagem e Vídeo Digital

A imagem digital são valores numéricos disponibilizados em uma matriz bidimensional. Basicamente existem dois tipos de imagens: os chamados *rasters* ou *bitmaps*, e vetorizadas. A primeira são representações de cada ponto da imagem com alguma cor usadas geralmente em fotografias, enquanto que o segundo é produzido por plotadores

que recebem os pontos e as distâncias entre eles como parâmetros, considerando retas, curvas, polígonos e etc, sendo assim não perdem sua qualidade quando redimensionados.

Converter uma imagem para o formato digital significa transferir os elementos que a compõem para elementos representativos de cada pequeno fragmento original. O menor elemento da imagem, o *pixel*, é identificado segundo sua intensidade de nível de cinza e as cores correspondentes. Identificados, estes elementos são armazenados por códigos que podem ser reconhecidos pelo dispositivo de visualização e apresentados novamente por um dispositivo de visualização, como um monitor de vídeo ou impressora (BALAN, 2009).

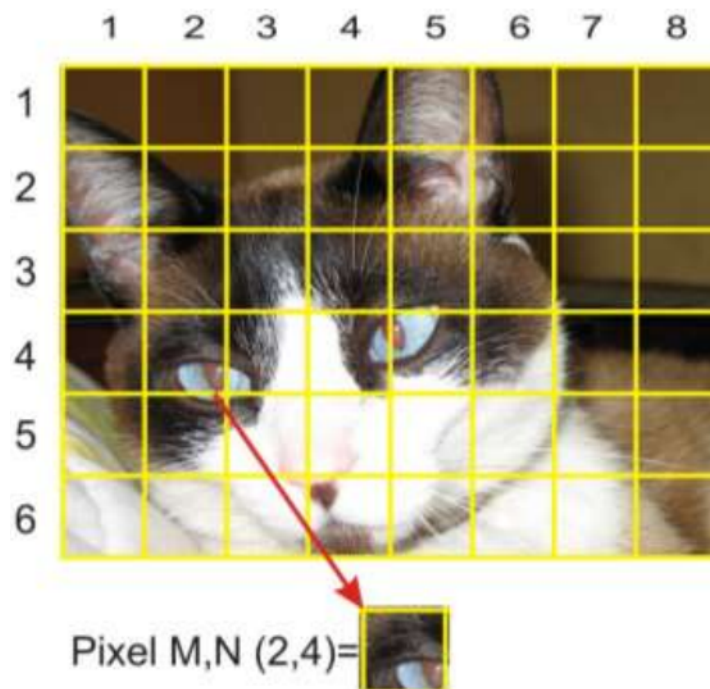


FIGURA 5 – Exemplo de imagem representada como uma matriz (M,N) de pixels, com destaque para o pixel (2,4).

FONTE: (BALAN, 2009)

Uma imagem analógica que é a (representação real da cena, para ser convertida para o formato do processamento computacional deve sofrer uma separação espacial (amostragem) e em amplitude (quantização) (BALAN, 2009).

A amostragem é a divisão do plano x,y em uma grade (ou matriz bi-dimensional) onde x e y serão números inteiros. Os pontos da matriz de são denominados *pixels* (*Picture Elements*), como ilustra a Figura 5. Cada pixel representa uma parte da cena real, desta forma a resolução espacial da imagem é proporcional aos valores de M e N correspondentes na matriz (exemplo Fig.22). Em geral a malha de amostragem, o formato dos pixels (x,y), é retangular, mas pode também ser triangular ou mais complexa. Os valores de cada ponto da matriz, coluna x linha (xy), que identifica um único pixel (M,N0, devem ser escolhidos de forma a respeitar a relação qualidade da imagem x espaço de armazenamento, em função da aplicação para a qual a imagem se destina. Para uma imagem

digital com 256 níveis de cinza o número de bytes ocupados para armazenar a imagem é o produto da linha vezes a coluna da matriz (BALAN, 2009).

Matematicamente, toda imagem monocromática (preto e branco) é um função $f(x,y)$ da intensidade luminosa, em qualquer parte das coordenadas (x,y) , proporcional ao brilho (tons de cinza) da imagem em um determinado ponto. A figura mostra uma imagem e como representamos os eixos x e y no plano cartesiano (GONZALEZ, 2010).

A função $f(x,y)$ é a multiplicação da iluminância $i(x,y)$ (que é a quantidade de luz que incide sobre o objeto) pela refletância $r(x,y)$ (que é fração de luz incidente que o objeto vai refletir ao ponto (x,y)).

Sendo assim, podemos dizer que :

$$f(x,y) = i(x,y) * r(x,y),$$

onde $0 < i(x,y)$ e $0 < r(x,y) < 1$.

Quando se utiliza uma imagem colorida, no padrão RGB por exemplo, deve se usar uma função $f(x,y)$ para cada banda, R(Red), G(Green) e B(Blue) que são as cores primárias(GONZALEZ, 2010).

2.1.2 Formato PNG

O PNG (*Portable Network Graphics* ou "*PNG is Not Giff*") é um formato de representação de imagens do tipo *rasters*. foi desenhado para substituir o formato GIF e o TIFF em certa extensão. O PNG tem 3 vantagens principais: suporta canal alfa (transparência) de forma eficiente, tem a maior gama de profundidade de cores e alta compressão que pode ser regulável. Além disso, o formato é livre enquanto os outros possuem patentes (ROELOFS, 2017).

O trabalho da compressão é justamente retirar essas informações redundantes para diminuir o peso do arquivo. Por exemplo, vamos imaginar um pixel qualquer de uma imagem. Possivelmente, a cor ou o valor desse pixel será igual a de vários outros elementos dentro da mesma imagem. Quando a imagem é comprimida, em vez de ela possuir vários pixels iguais e repetidos, ela guarda apenas um valor desse pixel, que é reproduzido para os outros semelhantes, economizando tempo de carregamento (FARIAS, 2017). Outras técnicas mais complexas também são utilizadas.

Este formato leve e de fácil manipulação foi escolhido neste trabalho para a representação de imagens digitais quando for necessário.

2.1.3 Aquisição da Imagem e de Vídeo

No processo de aquisição de imagens, tradicionalmente, um uma "cena" tridimensional (ou imagem analógica) deve ser capturada por um dispositivo eletrônico, que colhe uma

amostragem convertendo-a para uma imagem digital de duas dimensões. Por exemplo, uma câmera digital captando uma cena 3D e convertendo-a em 2D.

Atualmente o dispositivo de conversão mais usado é a câmera CCD (*charge coupled device*), que é uma matriz de células semi-condutoras fotossensíveis que trabalham com capacitadores, fazendo um armazenamento da carga elétrica proporcional à energia luminosa incidente (GONZALEZ, 2010).

A saída deste processo é a imagem do tipo *raster*, pronta pra ser formatada e compactada por alguma representação (PNG, JPG, BPM, etc).

2.1.4 Processamento da Imagem

O processamento de imagem é todo o processo que tem uma imagem como entrada e saída, tais como fotografia ou quadros de vídeo.

Usa-se para melhorar o aspecto visual de certas feições estruturais para o analista humano ou para performance e para fornecer outros subsídios para a sua interpretação, inclusive gerando produtos que possam ser posteriormente submetidos a outros processamentos (EYMAR LAÉRCIO, 2018).

O processamento de imagens pode ser dividido em 3 fases básicas: pré-processamento, realce e classificação. Pré-processamento refere-se ao processamento inicial de dados brutos para calibração radiométrica da imagem, correção de distorções geométricas e remoção de ruído. Realce visa melhorar a qualidade da imagem, permitindo uma melhor discriminação dos objetos presentes na imagem. Na classificação são atribuídas classes aos objetos presentes na imagem (EYMAR LAÉRCIO, 2018).

Nas próximas seções algumas técnicas de processamento de imagens serão abordadas pela sua importância e sua necessidade de uso em algoritmos de detecção e reconhecimento de faces. São elas:

- Conversão em Escala de Cinza: considerada uma técnica pré-processamento, é usada para transformar a imagem em "preto e branco"(na escala de cinza) , o que torna outras operações de processamento mais rápidas e eficientes;
- Escalonamento: pré-processamento de correção linear geométrica 2D (WANGHEIM, 2017);
- Equalização: uma técnica de realce de contraste (GONZALEZ, 2010)
- Segmentação: recorte da imagem, onde se extraem os objetos relevantes para a aplicação desejada (EYMAR LAÉRCIO, 2018).
- Classificação: uso de classificadores para reconhecer padrões em imagens (DAVISON, 2013)

2.1.4.1 Conversão em Escala de Cinza

Em fotografia, computação e colometria, uma imagem em escala de cinza é uma imagem em que cada valor de pixel é uma única amostra da representação de intensidade de luz (JOHNSON, 2006). Em outras palavras, o valor R, G e B de cada pixel devem ser iguais para se obter uma escala de cinza que varia de preto em sua intensidade mais baixa para o branco absoluto.

A conversão de uma cor para escala de cinza não é algo único. Existem diversas maneiras e possibilidades de inserção de parametros que dariam resultados diferentes. A maneira mais usual é calcular a média dos valores RGB de cada pixel e atribuir o resultado ao pixel novamente. Exemplo:

$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3$, reconhecendo que o algoritmo na verdade seria algo como:

For Each Pixel in Image

Red = Pixel.Red Green = Pixel.Green Blue = Pixel.Blue

Gray = (Red + Green + Blue) / 3

Pixel.Red = Gray Pixel.Green = Gray Pixel.Blue = Gray

Converter a imagem em escala de é necessário para tornar eficientes os subseqüente processos, como equalização e aplicação de classificadores (DAVISON, 2013).

2.1.4.2 Escalonamento

O escalonamento de imagens é uma transformação geométrica 2D. Pontos no Plano xy podem ser escalonados (esticados) por fatores de escala S_x e S_y através de multiplicação (WANGENHEIM, 2017):

$x' = x \cdot S_x$, $y' = y \cdot S_y$. Onde o novo ponto é resultado da multiplicação do ponto pela escala.

No exemplo da figura abaixo $S_x=2$ e $S_y=1$ os valores de x foram escalonados em 2 (S_x) enquanto y foi escalonado em 1 (S_y). Observe o resultado a seguir.:

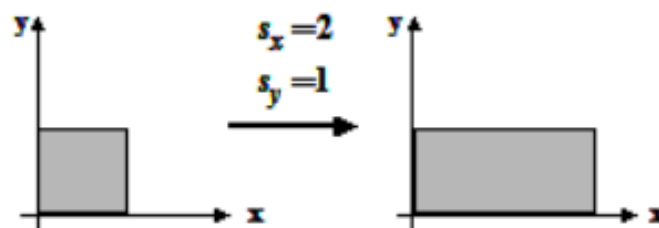


FIGURA 6 – Escalonamento com $S_x=2$ e $S_y=1$.

FONTE: (WANGENHEIM, 2017)

O escalonamento pode causar distorções na imagens se nao for feita linearmente, e perda de qualidade caso os valores sejam alterados de forma a aumentar a imagem. É muito usado para padronização de tamanhos, e se o escalonamento for feito com a subtração de valores pequenos, a qualidade da imagem pode ser melhorada. Outra vantagem de escalonar a imagem subtraindo valores é o fato de posteriores processamentos performarem mais rapidamente, pois a imagem apresenta-se menor (DAVISON, 2013).

2.1.4.3 Equalização de Histograma

O histograma de uma imagem digital é definido como "uma função discreta $h(r_k) = n_k$ onde r_k é o k -ésimo valor de intensidade e n_k é o número de pixels da imagem com intensidade r_k e cujos níveis de intensidade desta imagem estejam no intervalo $[0, L - 1]$." Os Histogramas são a base para várias técnicas de processamento no domínio espacial, onde sua manipulação pode ser utilizada para realce de imagens, além de fornecer dados estatísticos a seu respeito (GONZALEZ, 2010).

Em outras palavras, esta técnica realça o contraste da imagem com o objetivo de melhorar a qualidade da imagem em um ponto de vista humano. Do ponto de vista computacional, a equalização examina o intervalo de valores em escala de cinza (ou outra disposição de cores, dependendo da imagem) e tonifica mais a parte que existe mudança abrupta do branco pro preto (ou cinza escuro). O resultado ;e uma imagem com maior contraste entre áreas sombreadas e iluminadas, e contornos, tornando a detecção de padrões mais fácil posteriormente (DAVISON, 2013). A Figura 7 mostra o resultado depois da aplicação deste procedimento.

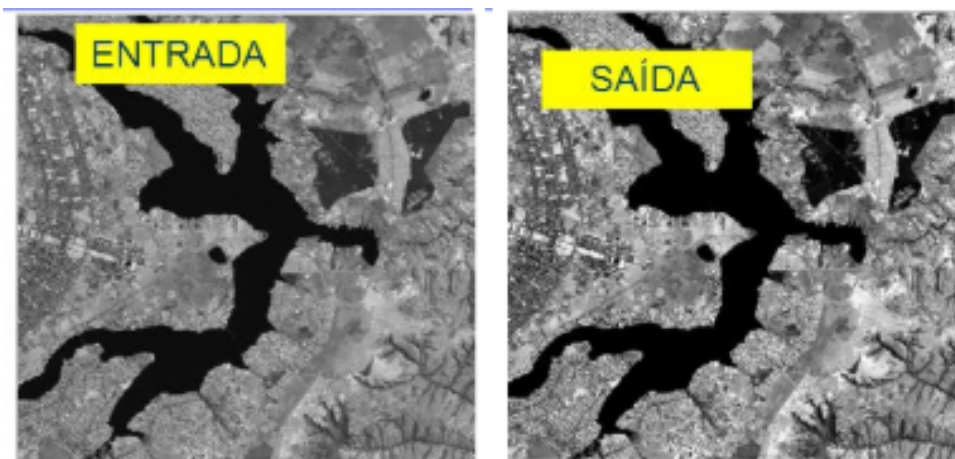


FIGURA 7 – Exemplo de aplicação de equalização de histograma.

2.1.4.4 Segmentação

Em visão computacional, segmentação se refere ao processo recortar uma imagem digital, ou seja, recolher um conjunto de pixels, para simplificar, mudar a representação de uma imagem ou extrair áreas relevantes para facilitar a sua análise.

Como resultado, cada um dos pixels em uma mesma região é similar com referência a alguma característica ou propriedade computacional, tais como cor, intensidade, textura ou continuidade. Este processo é pré-requisito para que reconhecimentos de objetos tenham grandes chances de sucesso (GONZALEZ, 2010).

2.1.4.5 Classificação

Segundo (VELASQUEZ, 2017), classificação de imagem contextual é um tópico de reconhecimento de padrões em Visão Computacional. Chama-se "contextual" pois significa que esta abordagem é focada na proximidade com os pixels e sua relação com o "classificador", (contemplado na seção Subseção 2.2.1). É utilizado efetivamente na detecção de objetos e também pode ser aplicado para a detecção de faces em uma imagem.

2.2 DETECÇÃO DE FACES COM CARACTERÍSTICAS HAAR

Um classificador HAAR pode ser treinado para detectar vastos tipos de objetos rígidos, definidos, como carros, motos ou partes do corpo humano como olhos ou boca, com altas taxas de sucesso. Não é muito eficiente em reconhecer objetos com ramos tipo árvores, mãos ou objetos camuflados ou contendo pouca textura, contorno e sub-regiões que variam em cor e iluminação (DAVISON, 2013).

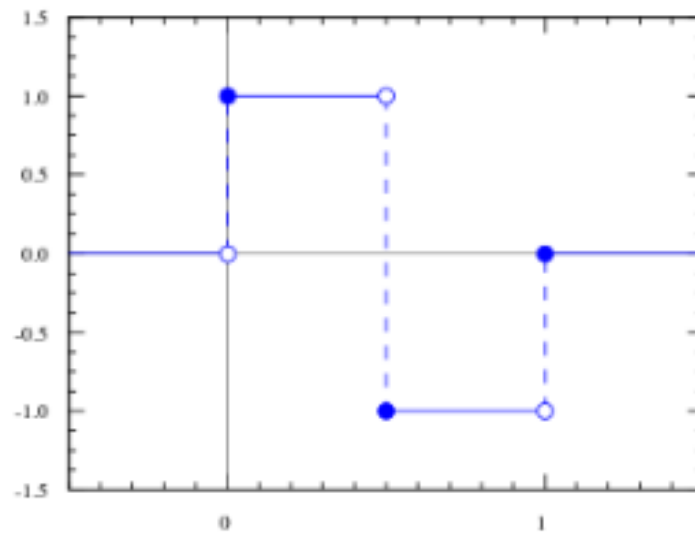
Um classificador bem treinado pode envolver *milhares* de fotos de alta qualidade com imagens positivas. Para a detecção de faces, isto significa imagens de rostos tiradas de perto que devem ter posições similares com muito pouca variação de fundo (*background variation*). Olhos bocas e narizes devem estar na mesma posição em todas as fotos, e estas devem ser do mesmo tamanho. Também é necessário treinar o classificador com um número similar de imagens negativas (isto é, sem rostos).

Existem diversas bibliotecas com classificadores pré-treinados para diferentes objetos, incluindo faces.

2.2.1 As Características Haar (*Haar Features*)

Os classificadores Haar (ou características de haar, ou filtro haar) recebem esse nome pela sua similaridade com a *Haar wavelet* (ou ondulação Haar), que consiste em uma sequência de funções quadradas redimensionadas que formam uma família de ondulações das mais simples possíveis.

Características Haar são basicamente imagens treinadas e usadas para achar objetos similares em outras imagens. Viola e Jones em seu algoritmo *Haar Cascades*, contemplado na seção seguinte, classificadores retangulares costumam a ser usados. Cada classificador pode indicar a existência ou a ausência de uma característica em outra imagem. A maior motivação para o uso de características de um objeto ao invés do uso de

FIGURA 8 – Um exemplo de *Haar wavelet*

pixel, é que a velocidade da análise de uma imagem baseada no conjunto de suas principais características é muito maior do que a análise baseada sobre seus pixels, devido ao fato do número de características ser substancialmente inferior em relação ao número de pixels (MOREIRA, 2008).

A Figura 9 mostra um classificador Haar em ação.

FIGURA 9 – Classificadores Haar representando características relevantes face
FONTE: Biblioteca OpenCV

Estes padrões retangulares podem ser escalonados para que características de diferentes tamanhos na imagem possam ser detectados usando a mesma abordagem.

2.2.2 Algoritmo Viola-Jones (*Haar Cascades Classifier*)

O *framework* de detecção de objetos Viola-Jones, apelidado em inglês de *Haar Cascades Classifier*, ou em português Cascata de Classificadores Haar (MOREIRA, 2008),

foi o primeiro algoritmo de detecção a fornecer detecção de objetos com taxas de sucesso competitivas e em tempo real. Foi proposto em 2001 por Paul Viola e Michael Jones. Apesar de poder detectar uma variedade de classes de objetos, foi desenhada com o objetivo primordial de detectar faces.

Com a utilização das características Haar, o objetivo a ser alcançado é classificar corretamente um dado objeto a partir do conjunto de suas principais características.

De acordo com (DAVISON, 2013), o algoritmo funciona com o uso de integrais que é rápida e eficiente, e ainda é possível reduzir drasticamente o numero de atributos que precisam ser testado para decidir se a imagem contem um objeto (por exemplo uma face). O teste de atributos da imagem são organizados em "cascatas" (também pode ser representado por uma árvore binária), representando o uso das integrais e iterativas. O nó raiz da cascata contém o teste que se provou ser o melhor em encontrar um objeto durante o treino. Se uma imagem não é rejeitada por este teste então a imagem vai para o nó com o segundo melhor teste, e assim por diante. Apenas se a imagem alcançar o fim de todos os testes (ou o fim da cascata) sem ser rejeitado, a imagem certamente deverá conter o objeto característico .

O grande ponto negativo é que este algoritmo também detecta os objetos que não são realmente os objetos que se tinha a intenção de detectar (DAVISON, 2013). Por exemplo, o desenho de um rosto certamente vai ser detectado com uma face original, e este desenho não precisa ser muito bem elaborado. As vezes uma disposição de sombras e regiões luminosas aleatórias podem ser detectadas como objetos característicos .

Segundo (MOREIRA, 2008), os estágios dentro de uma cascata são criados através da combinação de funções de classificadores Haar previamente treinados. O principal objetivo do uso de cascatas é fazer com que seus estágios iniciais descartem um grande numero de regiões que contem o objetivo desejado, e estágios mais avançados sejam melhores em evitar um falso positivo na região analisada.

Na Figura 10 representa-se os N estágios de uma cascata de classificadores. Cada um deles deve descartar ao máximo o numero de regiões da imagem que não contem o objeto, a fim de diminuir a quantidade de processamento de outras sub-áreas da imagem original (MOREIRA, 2008).

O processo para geração da cascata é guiado por um conjunto de metas de detecção e de desempenho. Na prática, uma arquitetura muito simples pode ser usada para produzir uma eficiente cascata de classificadores. O usuário pode informar as taxas mínimas aceitáveis e cada estágio da cascata é treinada junto ao conjunto de características Haar contando com cada vez mais elementos até que se atinjam as taxas de detecção de sucesso e falsos positivos. As taxas são encontradas testando-se o detector corrente sobre um conjunto de validação. Se a taxa de falsos positivos ainda não for atendida, então um novo estágio subsequente é ativado, coletando todas as falsas detecções encontradas

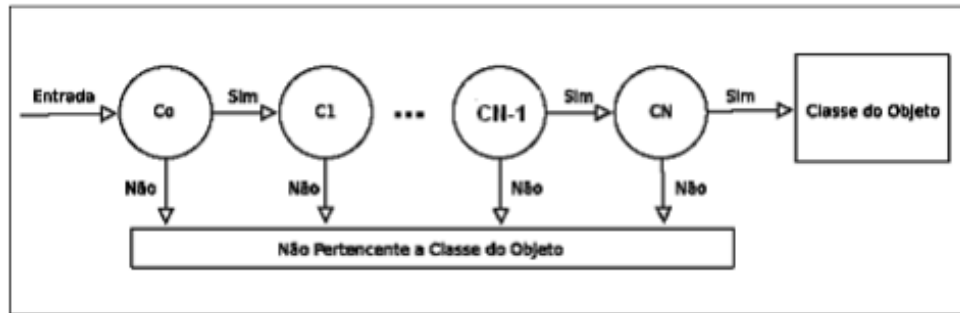


FIGURA 10 – Cascata de classificadores com estágios (C_0, C_1, \dots, C_n)

FONTE: (MOREIRA, 2008)

(MOREIRA, 2008).

Para que o algoritmo possa detectar o objeto de interesse, é necessário que a imagem seja percorrida diversas vezes em vários, assim a cada iteração é analisado uma região de tamanho diferente, chamada **janela de busca** (MOREIRA, 2008).

O usuário pode definir o tamanho desta janela que deve ser maior ou igual ao tamanho das imagens que foram treinadas na cascata, ou seja, se a cascata foi treinada com imagens positivas de 24x24 pixels, o tamanho mínimo da janela de busca deve seguir a mesma resolução (MOREIRA, 2008). As janelas devem funcionar em *thread* (múltiplas localidades), sendo que n janelas devem se deslocar no eixo X da imagem e m no eixo Y. A quantidade de janelas deve ser o suficiente para ocupar todo o espaço da imagem.

Outro valor que o usuário pode informar é número de detecções subsequentes necessários para que a região seja considerada a ter o objeto alvo. Reduzir este valor pode aumentar a velocidade de processamento, mas também aumenta as chances de taxas de falsos positivos (DAVISON, 2013).

O resultado do algoritmo são as posições dos objetos classificados com as características com as quais se recebeu treinamento, podendo ser identificados na imagem e sofrer posteriores processamentos como a segmentação.

2.3 EIGENFACES E ANÁLISE DE COMPONENTE PRINCIPAL (ACP)

O processo de reconhecimento de faces depende de uma fase de treinamento precedente envolvendo imagens faciais com suas identidades associadas. Imagens típicas de treinamento são mostradas na Figura 11.

É importante que as imagens sejam todas colhidas e orientadas de maneira similar, para que as variações entre as imagens sejam causadas por diferenças entre as faces e não por diferenças em plano de fundo ou posição facial. Também deve haver uniformidade na posição, clareamento, tamanho e resolução da imagem. É recomendado incluir várias imagens da mesma face mostrando diferentes expressões, como sorrindo ou abrindo a boca.



FIGURA 11 – Exemplo de imagens típicas de treinamento.

A relação da imagem com o indivíduo pode ser feita de diversas maneiras, uma delas é codificar alguma chave único para o indivíduo junto ao nome da imagem (DAVISON, 2013).

O processo de treinamento cria as *EigenFaces*, como mostra a Figura 2 explicada na Subseção 1.2.2, que são composições das imagens de treinamento que tem a propriedade de acentuar elementos que podem ser distinguidos mais facilmente pelo algoritmo. Uma imagem de face pode gerar várias Eigenfaces. As a imagem de treino da face podem ser representadas por uma sequência de "pesos" ($p_1, p_2, p_3, p_4, p_5, p_6$), como ilustra a figura Figura 12.

O objetivo é que a imagem de treinamento pode ser decomposta para uma soma dos pesos das multiplas engeifaces, sendo todos estes pesos salvos como uma sequência.

Nem todas as eigenfaces são igualmente importantes - algumas pode contar elementos faciais mais importantes para distinguir faces. Isto quer dizer que não é necessário utilizar todas as eigenfaces para reconhecer uma face, o que permite que uma imagem seja representada por uma pequena quantidade de eigenfaces, oque consome menos espaço e torna a execução do algoritmo mais rápida. O ponto negativo é que quando menos pesos, a probabilidade de se perder importantes elementos faciais também aumenta(DAVISON, 2013).

Outra maneira de se entender a relação entre eigenfaces e as imagens é representa-las em um plano cartesiado multidimensional (ou um *eigenspace*), como mostra a ??.

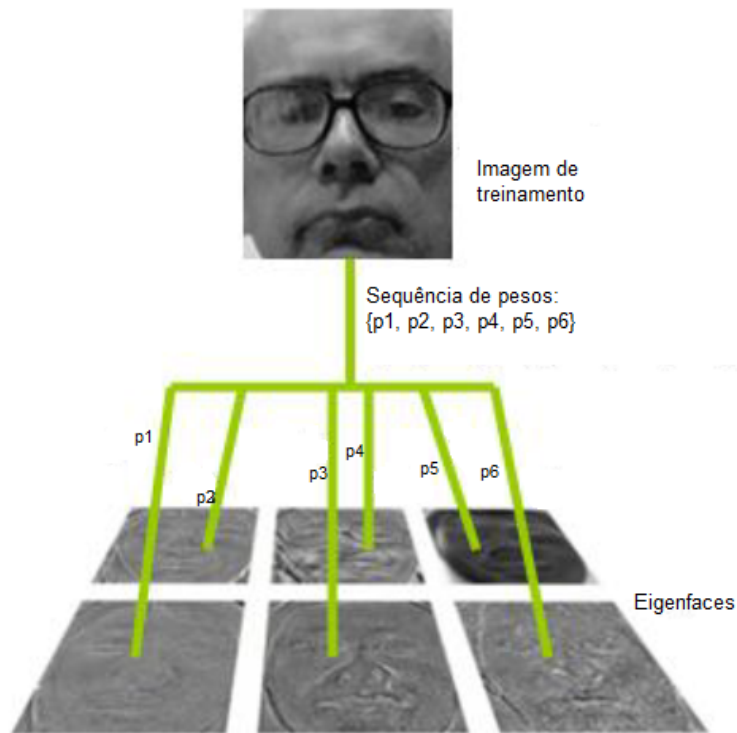


FIGURA 12 – Exemplo de uma imagem de treinamento representada em uma sequência de pesos de engenfaces.

FONTE: Adapdato de (DAVISON, 2013)

Os pesos agora podem ser observados como as coordenadas da imagem de treinamento em um plano cartesiano multidimensional (*eigenspace*). Na Figura 13, existem apenas 3 eixos representando as engenfaces p1, p2 e p3, porém se houvessem 10 engenfaces, haveria 10 eixos neste plano cartesiano.

Após o processo de treinamento das faces vem o processo de reconhecimento. Neste novo processo, uma nova imagem para reconhecimento deve ser decomposta em engenfaces (na mesma maneira da Figura 12). A sequência de pesos resultante é comparada com cada sequência de pesos que foram treinadas previamente, e o nome associado com o as características mais próximas dos engenfaces treinados é associado (DAVISON, 2013)..

Em outras palavras, em termos de *engenspaces*, a nova imagem é posicionada no plano cartesiano com suas próprias coordenadas (pesos). Então uma técnica de medida de distância (usualmente a distância Euclidiana, que será abordada na próxima seção) é utilizada para encontrar e figura de treinamento correspondente mais próxima. Sendo assim, uma distância Zero seria uma correspondência (reconhecimento) perfeita (DAVISON, 2013). A Figura 3 na Subseção 1.2.2 ilustra este processo.

As Engenfaces são geradas utilizando um procedimento matemático chamado Análise de Componentes Principais - ACP (em inglês Principal Components Analysis - PCA). Esta técnica acentua as similaridades e as diferenças nos dados.

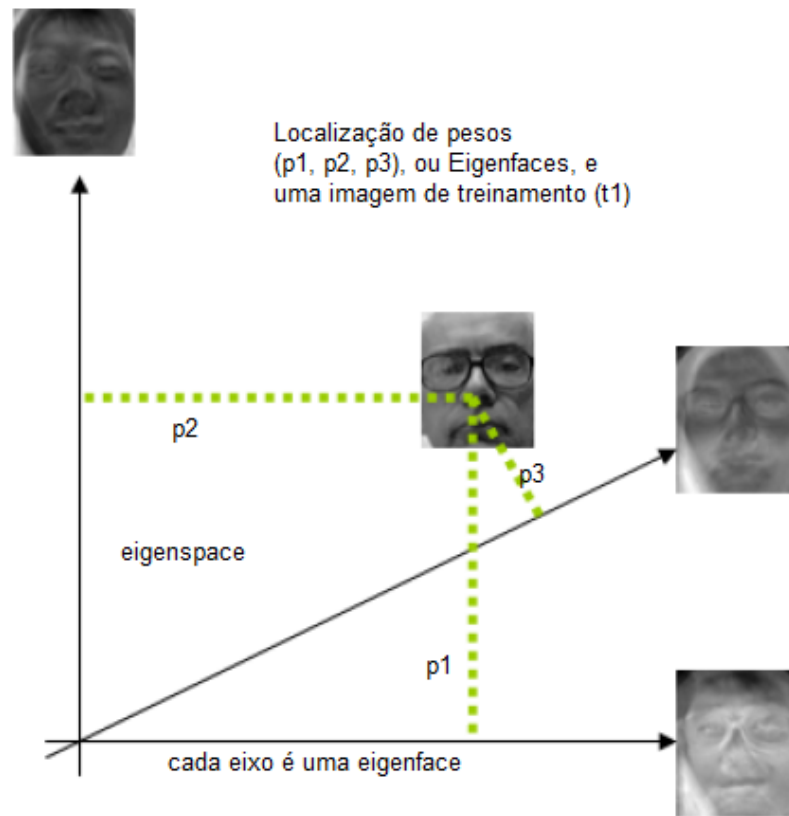


FIGURA 13 – Exemplo de eigenfaces representadas em um plano cartesiano (*eigenspace*) de 3 dimensões.

FONTE: Adaptação de (DAVISON, 2013)

2.3.1 Análise de Componentes Principais (ACP)

Para que esta técnica seja explicada, será utilizado vetores de valores numéricos: o vetor x e y , como mostra a Tabela 2.

TABELA 2 – Dados dos vetores x e y .

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

FONTE: (DAVISON, 2013)

Os dados de x e y poderiam representar a altura de estudantes, preços de frutas, ou valores numéricos extraídos de uma imagem, por exemplo. O objetivo da demonstração

é acentuar as diferenças e similaridades entre os dois vetores de dados em uma abordagem numérica (DAVISON, 2013).

Para tal, medidas estatísticas como média e desvio padrão (uma medida que tem o objetivo de espalhar valores acerca da média). Variância é outra medida de espalhamento, que é igual ao desvio padrão ao quadrado:

$$\text{var}(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})}{(n - 1)}$$

A variável n é o número de itens dos vetores (10 de acordo com a Tabela 2), e \bar{x} é a média dos valores d conjunto de dados.

Contudo, a média e a variância fornecem informação apenas sobre a forma dos dados em um único vetor (por exemplo, e média e a variância do vetor x), sendo que o objetivo é quantificar as diferenças **entre** os dois vetores (DAVISON, 2013). Justamente a covariância é uma medida estatística, generalizada da variância, que compara dois conjuntos de dados (DAVISON, 2013):

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)}$$

Se a fórmula da covariância for aplicada nos dados dos vetores x e y dados na Tabela 2, o resultado será 0.6154. A parte em que se deve estar atendo é o sinal: se for negativo significa que quando um vetor aumenta em dados, o outro diminui. Quando o valor é positivo, que é o caso, então os dois conjuntos de dados aumentam juntas. Para ilustrar este conceito, na Figura 14 plota-se os valores correspondentes dos vetores x e y como pontos em um gráfico (DAVISON, 2013).

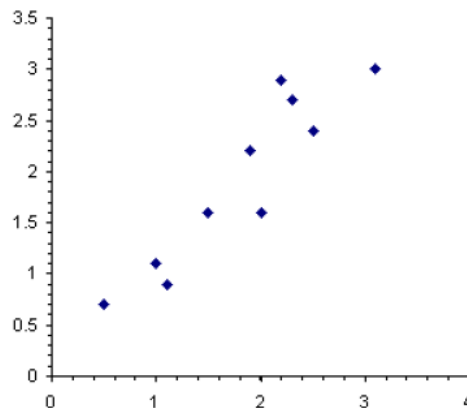


FIGURA 14 – Plotagem dos valores de x e y .

FONTE: Adapdato de (DAVISON, 2013)

Se houverem mais de dois vetores (por exemplo o vetor x , y e z) então a covariância deve ser calculada entra todas os conjuntos de dados, ou seja, calculando $\text{cov}(x, y)$, $\text{cov}(x, z)$ e $\text{cov}(y, z)$.

Não há necessidade de calcular $\text{cov}(y, x)$, $\text{cov}(z, x)$ e $\text{cov}(z, y)$ sendo que a definição da equação garante que $\text{cov}(A, B)$ é igual a $\text{cov}(B, A)$.

A maneira padrão de armazenar a covariância entre múltiplos vetores de dados é por uma matriz onde os conjuntos de dados tornam-se os índices das colunas e linhas. Por exemplo, a covariância para os conjuntos de dados x, y e z se tornariam uma matriz de dimensões 3×3 (DAVISON, 2013):

$$\begin{pmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{pmatrix}$$

Na diagonal principal, pode-se observar que a covariância é entre o conjunto de dados a própria covariância, que é equivalente à variância nestes dados. Podemos provar isto comparando a variância de x ($\text{var}(x)$) e a covariância de x com x ($\text{cov}(x)$). A matriz também apresenta simetria em volta da diagonal principal.

Os conjuntos de dados x e y da Tabela 2 seriam representados em uma matriz de dimensões 2×2 :

$$\begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

A matriz de covariância é um meio útil de demonstrar a relação entre os conjuntos de dados, mas deve-se obter ainda mais informações para que sejam geradas as eigenfaces calculando dois vetores que serão contemplados na seção seguinte: *eigenvectors* e *eigenvalues*.

2.3.2 ACP E Eigenfaces

Para que as eigenfaces sejam geradas, deve-se transformar a covariância ou outros conjuntos de dados chamados de *eigenvectors* e *eigenvalues*.

Segundo (DAVISON, 2013), o *eigenvector* é um vetor de valores ordinário que quando multiplicado por uma dada matriz, muda de magnitude, enquanto que *eigenvalue* é apenas um termo para esta magnitude.

Por exemplo, dada a matriz:

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

O eigenvector para esta matriz é $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$, pois quando multiplicado pela matriz acima, o mesmo vetor é retornado o mesmo vetor é retornado se multiplicado por 4:

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Sendo assim, 4 é o *eigenvalue* para o *eigenvector* $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$.

Os eigenvectores podem ser encontradas apenas para matrizes quadradas, porém nem todas. Para uma dada matriz $n \times n$, se realmente houver eigenvectores, haverá n . A matriz 2×2 usada de exemplo acima tem dois eigenvetores (e seus valores eigenvalues correspondentes).

A relação eigenvalue pode ser escrita matematicamente como:

$$\Lambda \nu = \lambda \nu$$

onde Λ é a matriz quadrada, ν é um eigenvetor para A e o valor λ é um eigenvalue.

Para usa-los no algoritmo PCA, os eigenvetores devem ser normalizados para que todos eles tenham a mesma unidade de tamanho. Isto é, o eigenvetor acima, $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$, tem a unidade de tamanho "falso" de $\sqrt{(3^2 + 2^2)} = \sqrt{13}$. Se o vetor for dividido por este valor resultara em um tamanho de unidade (DAVISON, 2013):

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \div \sqrt{13} = \begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$$

Se recuperarmos a matriz de covariância que geramos acima:

$$\begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

pode-se observar que, com é uma matriz 2×2 , possui dois eigenvetores e eigenvalues:

$$\begin{pmatrix} -0.7352 \\ 0.6779 \end{pmatrix} \text{ e } 0.049$$

$$\begin{pmatrix} 0.6779 \\ 0.7352 \end{pmatrix} \text{ e } 1.284$$

Ambos os eigenvetores possuem unidade de tamanho.

2.3.2.1 Utilizando Eigenfaces como Componente Principal

Observando novamente a plotagem dos vetores x e y contemplados na Figura 14, se a média destes conjuntos de dados (\bar{x} e \bar{y}) for subtraído de deus valores os dados serão

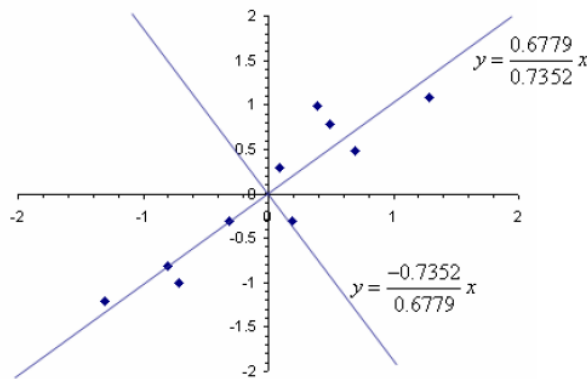


FIGURA 15 – Plotagem normalizada dos dados de x e y com eigeenvetores.

FONTE: (DAVISON, 2013)

normalizados. Isto também quer dizer que as linhas que representam os vetores x e y são transladados para o centro como mostra a Figura 15.

Os dois engenvetores pode ser adicionados ao gráfico como linhas convertendo os vetores em equações. $\begin{pmatrix} -0.7352 \\ 0.6779 \end{pmatrix}$ torna-se $y = \frac{-0.7352}{0.6779}x$ e $\begin{pmatrix} 0.6779 \\ 0.7352 \end{pmatrix}$ torna-se $y = \frac{0.6779}{0.7352}x$.

Os dados normalizados e as duas equações são ilustradas na Figura 15 como linhas azuis.

Observa-se que na Figura 15 demonstra como os eigeenvectores acentuam as relações entre os conjuntos de dados- os vetores indicam como os conjuntos de dados se espalham nas coordenadas do espaço cartesiano. Este espalhamento facilita a distinção dos dados (DAVISON, 2013).

Dos dois eigeenvetores marcados em azul, a linha $\frac{0.6779}{0.7352}x$ é a mais útil, pois os dados estão mais espalhados acerca do trajeto desta linha. Isto pode ser confirmado numericamente observando o eigenvalue associado com seu eigeenvetor: o eigenvalue (1.284) é um melhor indicador de espalhamento pois é o maior dentre os dois eigenvalues.

A outra linha, $\frac{-0.7352}{0.6779}x$, contribui com a informação demonstrando como os dados são espaçados para direita ou esquerda do eigeenvetor principal (a linha $\frac{0.6779}{0.7352}x$). Entretanto, trata-se de um componente menos importante no espalhamento de dados, assim como indica seu eigenvalue (0.049) (DAVISON, 2013).

O eigeenvetor com o maior eigenvalue (no caso a linha $\frac{0.6779}{0.7352}x$, ou o vetor $\begin{pmatrix} 0.6779 \\ 0.7352 \end{pmatrix}$) é chamado de **componente principal** do conjunto de dados. Isto é, os eigeenvetores e eigenvalues são efetivamente utilizados para performar uma análise de componente principal nos conjuntos de dados (DAVISON, 2013).

Todos os eigeenvetores extraídos da matriz são perpendiculares. Isto significa que é possível rotacionar (talvez refletir) os dados para que os eigeenvetores tornem-se alinhados

com os eixos. Se este componente principal $\begin{pmatrix} 0.6779 \\ 0.7352 \end{pmatrix}$ for rotacionado e refletido para que seja alinhado com o eixo y, resulta-se em uma plotagem mostrada na Figura 16:

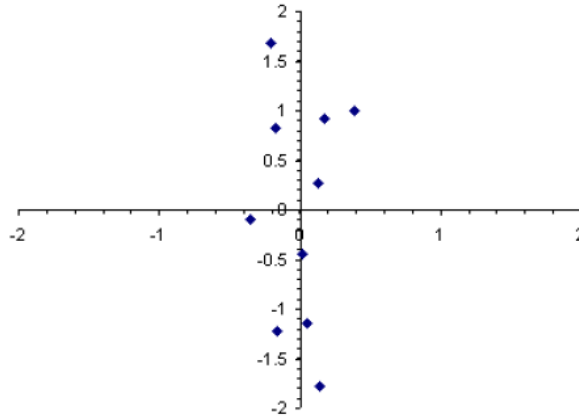


FIGURA 16 – Versão rotacionada e refletida da Figura 15

FONTE: (DAVISON, 2013)

Observando a Figura 16, o ponto mais próximo da origem se situa na coordenada (0.13, 0.27). Pode-se afirmar que também que este ponto tem uma sequência de pesos relativos a cada eigenvetor, no caso, a sequência 0.13, 0.27.

Tanto a denotação de coordenadas quanto a de pesos pode ser usada para representar as eigenfaces - como contemplado anteriormente, na Figura 12 foi dado o exemplo em que uma imagem é uma sequência de pesos de seis eigenfaces, enquanto que, alternativamente, na Figura 13 é representada em coordenadas de um plano cartesiano de 3 dimensões, sendo que cada eixo é definido por uma eigenface (ou eigenvetor).

2.3.2.2 O Reconhecimento

O processo de reconhecimento consiste em recolher novos conjuntos de dados (no caso, uma nova face), e utilizando os dados processados até agora e representados na Figura 16, deve-se comparar os dois novos conjuntos de dados (a nova face e os eigenfaces ou eigenvetores) fazendo uso de medidas de distância.

Existem algumas técnicas de medidas de distância, dentre elas, há uma simples chamada **Distância Euclidiana**. Esta medida nada é do que a distância mínima entre dois pontos no espaço (uma linha reta). O resultado desta medida é então utilizada para calcular a taxa de sucesso de uma comparação, sendo assim uma distância nula representaria uma equivalência perfeita.

Contudo, em aplicações em vida-real, deve-se reduzir o tamanho dos dados, pois existem informações irrelevantes e a tarefa de reconhecimento pode ser executada mais rapidamente enquanto retém informações suficientes para distinguir entre os pontos quando comparados com os novos conjuntos de dados.

A técnica consiste em reter todas as coordenadas dos dados, mas reduzir a dimensionalidade do eigenspace, isto é, subtrair o número de eixos do plano cartesiano. Este processo é equivalente a subtrair algumas eigenfaces, mas como citado anteriormente, os eigenvalues de menor intensidade não tem influência no espalhamentos dos dados no plano, sendo pouco significativos para o reconhecimento, e portanto podem ser removidos.

Portanto, os dados mostrados na Figura 16 possuem os seguintes eigenvetores e eigenvalores:

$$\begin{pmatrix} -0.7352 \\ 0.6779 \end{pmatrix} \text{ e } 0.049$$

$$\begin{pmatrix} 0.6779 \\ 0.7352 \end{pmatrix} \text{ e } 1.284$$

Observando que primeiro eigenvetor foi rotacionado em direção ao eixo x, e o segundo eigenvetor em direção ao eixo y, se o primeiro eigenvetor que contribui muito pouco para a informação de espalhamento (devido ao seu pequeno eigenvalue) for descartado junto ao seu eixo, os dados deste eigenvetor são projetados para o eixo y, resultando na plotagem mostrada na



FIGURA 17 – Dados da Figura 16 projetados em direção ao eixo y.

FONTE: (DAVISON, 2013)

Apesar dos eixos serem eliminados, seus dados ainda estão presentes e espalhados o suficiente para que o novo conjunto de dados seja comparado com o uso da distância euclidiana.

Considerando que um novo conjunto de dados adicionado com os valores do vetor $z = (2.511, 2.411)$, e estes conjuntos passarem pelos mesmos processos que os vetores x e y , e por fim normalizado e projetado assim como mostra a Figura 15 e Figura 16, plotando-o no mesmo espaço resulta na Figura 18.

A Figura 18 é a mesma que a Figura 16 com a adição de um novo ponto vermelho que indica a coordenada do novo conjunto de dados do vetor z , posicionada logo acima

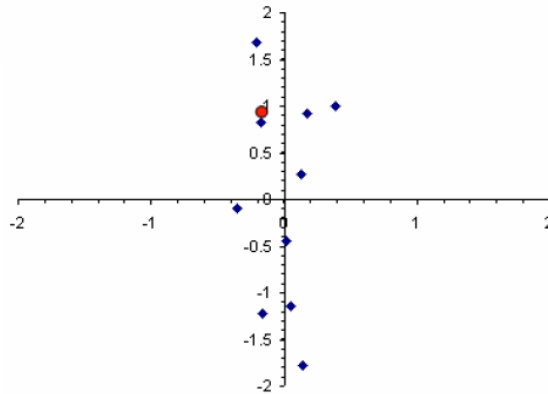


FIGURA 18 – Plotagem dos novos dados com os dados de treino transformados.

FONTE: (DAVISON, 2013)

do ponto $(-0.1751, 0.8280)$, que por sua vez começou como um ponto de treinamento de pesos $(2.5, 2.4)$, informada nos conjuntos de dados x e y . Isto quer dizer que o novo conjunto de dados z $(2.511, 2.411)$ possui a menor distância euclidiana com o conjunto de treinamento $(2.5, 2.4)$.

A real vantagem dos eigenvetores é quando a dimensionalidade do espaço (eigenspace) torna-se extremamente grande. No presente exemplo foi utilizado uma dimensionalidade dois, enquanto que em aplicações reais com imagem os eixos podem aumentar para valores como 40 mil (DAVISON, 2013).

Com tamanha dimensão e, portanto, quantidade de eixos, como citado anteriormente pode-se eliminar alguns eixos dependendo de seus valores eigenvalues (se forem pequenos, podem ser considerados dispensáveis). Se por uma hipótese isto for feito neste exemplo incluindo o novo conjunto dados z , a plotagem do resultado seria como na Figura 19 abaixo:



FIGURA 19 – Plotagem dos conjuntos de dados x , y e z , transformados e projetados no eixo do Componente Principal

FONTE: (DAVISON, 2013)

A Figura 19 é similar à Figura 17 com exceção do ponto vermelho (novo dado). Pode-se observar nesta figura o problema potencial em se reduzir a dimensionalidade: a probabilidade de erros aumenta tornando difícil de decidir qual ponto de treinamento é

que possui a menor distância euclidiana. Outro fator da equação da distância euclidiana é que se utiliza de um parâmetro para definir quando dois pontos estão perto. Se um valor alto é definido então diversos pontos podem estar dentro do alcance aceitável do novo dado (DAVISON, 2013).

2.4 CONSIDERAÇÕES FINAIS

REFERÊNCIAS

- BALAN, W. C. A imagem digital. **Espaço para apoio didático do Curso de Comunicação social - Radialismo**, 2009. Disponível em: <<http://willians.pro.br/textos/>>. Acesso em: 05.2018.
- BARBU NATHAN LAY, G. G. A. Face detection with a 3d model. 2015. Disponível em: <<https://pdfs.semanticscholar.org/2bbf/047993f08a608ce4ca22720d374d4cf76f98.pdf>>.
- CHAO, W.-L. **Face Recognition**. Dissertação (Mestrado) — National Taiwan University, GICE, Taiwan, Maio 2011.
- DAVISON, D. A. In: _____. **Java Prog. Techniques for Games**. Vienna: Universidade Prince of Songkla, Tailândia, 2013. Disponível em: <<http://fivedots.coe.psu.ac.th/~ad/index.html>>.
- ELECTRICAL, I. of; IEEE, E. E. Face recognition: eigenface, elastic matching, and neural nets. **Proceedings of the IEEE (Volume: 85, Issue: 9, Sep 1997)**, 1997. Disponível em: <<http://ieeexplore.ieee.org/document/628712/>>.
- EYMAR LAÉRCIO, R. Teoria : Processamento de imagens. **Instituto Nacional de Pesquisas Espaciais - INPE**, 2018. Disponível em: <<http://www.dpi.inpe.br/spring/teoria/realce/realce.htm>>. Acesso em: 05.2018.
- FARIAS, F. Como compactar imagens sem perder qualidade e aumentar a velocidade de seu site. **Resultados Digitais**, 2017. Disponível em: <<https://resultadosdigitais.com.br/blog/compactar-imagens-sem-perder-qualidade/>>. Acesso em: 05.2018.
- GONZALEZ, R. C. W. R. C. In: _____. **Processamento Digital de Imagens 3ªed**. São Paulo: Pearson Prentice Hall, 2010.
- INTRONA, H. N. L. D. Facial recognition technology - a survey of polycy and implementation issues. 2010. Disponível em: <https://www.nyu.edu/projects/nissenbaum/papers/facial_recognition_report.pdf>.
- JEBARA, T. S. **3D Pode Estimation and Normalization for Face Recognition**. Dissertação (Mestrado) — McGill University, Columbia University, Montréal, Québec - Canada, 1995. Disponível em: <<http://www.cs.columbia.edu/~jebara/htmlpapers/UTHESES/node7.html>>.
- JOHNSON, S. In: _____. **Stephen Johnson on Digital Photography**. [S.l.]: O'Reilly Media, Incorporated, 2006.
- KIM JOON HYUNG SHIM, J. Y. I. **Final Project - Face Detection Project**. Dissertação (Mestrado) — Stanford University, Stanford, California, 2003. Disponível em: <https://web.stanford.edu/class/ee368/Project_03/Project/reports/ee368group02.pdf>.
- KITANI, C. E. T. E. C. Um tutorial sobre análise de componentes principais para o reconhecimento automático de faces. **Departamento de Engenharia Elétrica**, Centro Universitário da FEI. Disponível em: <http://fei.edu.br/~cet/Tutorial_ReconhecimentoFaces.pdf>. Acesso em: 05.2018.

MOREIRA, G. C. G. **Reconhecedor de Objetos em Vídeos Digitais para Aplicações Interativas**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro - PUCRJ, Rio de Janeiro, 2008.

PORTAL, S. T. S. Face recognition software comparison as recognition. **Digital Economy Compass** **2017**, 2017. Disponível em: <https://www.slideshare.net/statista_com/statista-digital-economy-compass-2017>.

ROELOFS, G. A basic introduction to png features. **PNG LIB**, 2017. Disponível em: <<http://www.libpng.org/pub/png/pngintro.html>>. Acesso em: 05.2018.

SCIENCE, N.; (NSTC), T. C. Face recognition. **National Science and Technology Consil (NSTC)**, National Science and Technology Consil (NSTC), p. 92–99, 2009. Disponível em: <https://www.fbi.gov/file-repository/about-us-cjis-fingerprints_biometrics-biometric-center-of->. Acesso em: 04.2018.

SILVA, G. N. da. **Estudo das Técnicas PCA (Análise de Componentes Principais) e Auto-faces Aplicadas ao Reconhecimento de Faces Humanas**. Dissertação (Mestrado) — Centro Universitário Eurípides Soares - UNIVEM, Marília, 2009.

SMITH, L. I. A tutorial on principal components analysis. 2002. Disponível em: <http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf>. Acesso em: 05.2018.

VELASQUEZ, H. C. Advanced vision module. **The University of Edinburgh**, 2017. Disponível em: <<http://www.inf.ed.ac.uk/teaching/courses/av/index.html>>. Acesso em: 05.2018.

VERGE.COM, T. • baidu swaps tickets for facial recognition in historic chinese 'water town'. novembro 2016. Disponível em: <<https://www.theverge.com/2016/11/17/13663084/baidu-facial-recognition-park-wuzhen>>.

WANGENHEIM, D. A. von. Transformações geométricas 2d e coordenadas homogêneas. **Image Processing and Computer Graphics Lab - LAPiX**, Universidade Federal de Santa Catarina – UFSC, 2017. Disponível em: <<http://www.lapix.ufsc.br/ensino/computacao-grafica/transformacoes-geometricas-2d-e-coordenadas-homogeneas>>. Acesso em: 05.2018.

WÓJCIK KONRAD GROMASZEK, M. J. W. Face recognition: Issues, methods and alternative applications. 2016. Disponível em: <<https://www.intechopen.com/books/face-recognition-semisupervised-classification-subspace-projection-and-evaluation-methods/face-recognition-issues-methods-and-alternative-applications>>.

YANG PING LUO, C. C. L. X. T. S. Wider face: A face recognition benchmark. 2015. Disponível em: <<https://arxiv.org/pdf/1511.06523.pdf>>.

