

RECONHECIMENTO FACIAL

FACIAL RECOGNITION

Izaque Maciel Cunha, Jeferson Cruz, Thiago Godoy Souza, Marco Antonio Piloto

CUNHA, Izaque Maciel; CRUZ, Jeferson; SOUZA, Thiago Godoy; PILOTO, Marco Antonio. Reconhecimento facial. Revista Tecnológica da FATEC-PR, Edição Especial, p. 46-29, jan/dez, 2019.

RESUMO

Existem vários algoritmos utilizados para a detecção facial, dentre eles podemos citar o do tipo Geométrico, um dos mais utilizados e intuitivos, onde é construído marcos nas imagens (posições), o mesmo é muito eficiente contra mudanças de iluminação, é o mais utilizado no reconhecimento facial. Podemos citar também o Eigenfaces, que utiliza como método o PCA (Principal Component Analysis), nesse método é selecionado apenas os atributos mais importantes da imagem e o restante é descartado.

Dando sequência nos tipos de algoritmos podemos mencionar também o Machine Learning, onde é utilizada uma base de dados para treinamento e teste, submetendo uma série de imagens e o classificador automaticamente vai aprendendo a identificar o que é uma face. Por fim temos o Cascade (Cascada – Viola Jones), nesse tipo são aplicados filtros retangulares, somando pixel a pixel e fazendo a média entre áreas positivas e negativas, o que resulta na detecção das características de cada pixel, assim podendo agrupar o pixel ao padrão que ele pertence, fazendo com que seja possível detectar o objeto desejado.

Palavras chave: Estrutura de Programas. Inovação. Recursos Tecnológicos.

ABSTRACT

There are several algorithms used for facial detection, among them we can mention the Geometric type, one of the most used and intuitive, where landmarks are built in the images (positions), it is very efficient against changes in lighting, it is the most used in facial recognition. We can also mention Eigenfaces, which uses the PCA (Principal Component Analysis) method, in this method only the most important attributes of the image are selected and the rest is discarded.

Following the types of algorithms, we can also mention Machine Learning, where a database is used for training and testing, submitting a series of images and the classifier automatically learns to identify what a face is. Finally we have Cascade (Cascada - Viola Jones), in this type rectangular filters are applied, adding pixel by pixel and averaging positive and negative areas, which results in the detection of the characteristics of each pixel, thus being able to group the pixel at the same time. pattern it belongs to, making it possible to detect the desired object.

Keywords: Program Structure. Innovation. Technological Resources.

1 INTRODUÇÃO

Em uma sociedade onde cada vez mais surge a necessidade de nos preocuparmos com a segurança das informações, torna-se necessário criar obstáculos para combater os ataques ou cyber-ataques. Vindo a esse encontro, intensifica-se a utilização através de

reconhecimento facial, o que em tese, gera uma forma única de acesso seja ele físico ou a uma aplicação.

O desenvolvimento dessa tecnologia busca trazer facilidade e convergência no modo de acesso, pois necessita apenas de um banco de dados com imagens pré-cadastradas para comparação no sistema de reconhecimento facial a fim de controlar e determinar privilégios e quaisquer outras informações demandadas a um usuário, da mesma forma que torna pessoal e intransferível uma responsabilidade a um determinado usuário, diminuindo assim as fraudes.

1.1 OBJETIVO GERAL

Entender como funciona e executar integração de um algoritmo de reconhecimento facial à linguagem Java com câmera IP (Protocolo da internet), arduino e fechadura elétrica.

1.2 OBJETIVOS ESPECÍFICOS

- a) Entender o funcionamento de um algoritmo de reconhecimento facial, ter o conhecimento de como funciona o desenvolvimento de um sistema para reconhecimento facial utilizando a linguagem Java,
- b) Efetuar a integração do mesmo com câmera IP, arduino e fechadura elétrica para abertura automática de portas.
- c) Desenvolver integração entre os sistemas e suas utilidades.

2 JUSTIFICATIVA

Um dos porquês que levaram a execução deste trabalho é o fator de haver pouca documentação e informações em português sobre o assunto abordado. Já pelo cenário acadêmico, este trabalho engloba conceitos que tem importância crucial no desenvolvimento como profissional e de pesquisa. A parte lógica busca conceitos de programação (Java). Passando para o estudo dos algoritmos de reconhecimento facial. Sendo assim, o desenvolvimento deste trabalho intensifica os estudos em torno destas áreas de modo em que por consequência estimula o aprendizado na prática das teorias estudadas em sala de aula.

3 REVISÃO BIBLIOGRÁFICA

3.1 DETECÇÃO X RECONHECIMENTO

Antes de compreendermos como funciona e o que é o reconhecimento facial, devemos

ter bem definido a diferença entre detecção facial e o reconhecimento facial. Ambas as tecnologias pertencem a um sub-ramo da Inteligência Artificial chamado Visão Computacional, o mesmo tem como objetivo simular a capacidade do olho humano. De acordo com Antonio Henrique Dianin (2019), a detecção facial não determina de quem são os rostos que estão em uma foto, apenas detecta se existem rostos na mesma. Abaixo temos algumas imagens exemplo de como se diferencia ambas as tecnologias.



Figura 1: Detecção facial.
Fonte: Udeemy Inc.¹

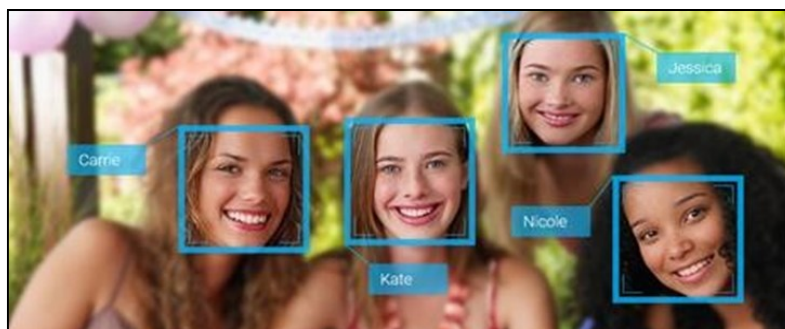


Figura 2: Reconhecimento facial.
Fonte: Udeemy Inc.¹

As figuras acima mostram de forma simples e direta a diferença entre a detecção e o reconhecimento facial, enquanto a detecção tem como objetivo apenas detectar uma face seja qual for ela, o reconhecimento além de detectar, identifica quem é a pessoa.

Para as aplicações de detecção se tem:

- a) contar quantas pessoas estão no ambiente;
- b) obter o tamanho de um objeto;

¹ Disponível em: <https://www.udemy.com/course/deteccao-de-faces-com-java-e-opencv/>

c) detecção de sorrisos em câmeras/ajuste de foco.

Para as aplicações de reconhecimento se tem:

- a) desbloqueio do celular;
- b) sistemas de segurança;
- c) validação em cursos on-line

3.2 RECONHECIMENTO FACIAL

De acordo com Erico Theis Pereira dos Santos (2019), o reconhecimento facial é uma tecnologia que consiste em mapear a imagem do rosto de um indivíduo, gravando as distâncias entre pontos específicos, baseando-se pelos olhos, boca e nariz. Mesmo que cada pessoa possua uma característica única, de forma ampla a composição do rosto de um humano é sempre a mesma: olhos, nariz, boca, sobrancelha e etc, e isso faz com que seja possível ocorrer a primeira fase do reconhecimento, a detecção facial.

Após a captura da imagem, na detecção facial o algoritmo treinado detecta o rosto humano diferenciando-o de qualquer outro objeto, logo em seguida entra a fase seguinte, onde o algoritmo começa a mapear a face detectada. Cada rosto humano possui características singulares, distância entre os olhos, tamanho do nariz, comprimento da boca, bochechas, queixo e assim por diante, e isso faz com que seja criada uma sequência única que é armazenada em um banco de dados. Se o padrão da imagem comparada for igual a que está armazenada no banco de dados através de cálculos feitos pelo algoritmo, o mesmo informa a porcentagem de similaridade e faz a validação.

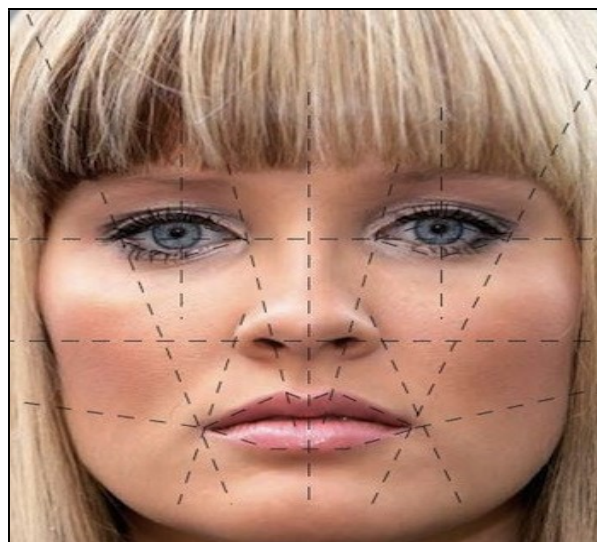


Figura 3: Mapeamento das características da face.

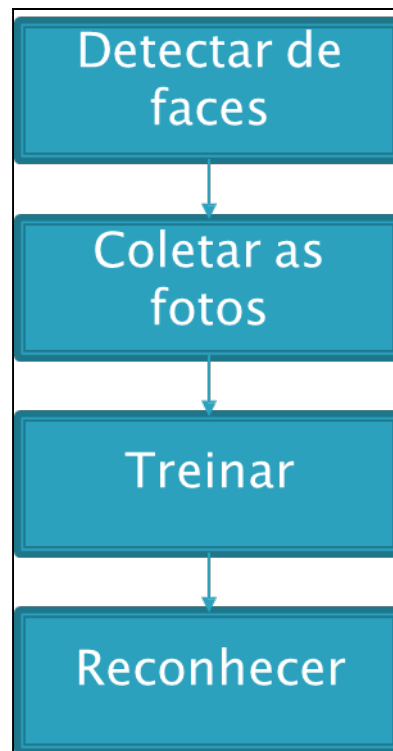
Fonte: Techtudo²

Figura 4: Etapas do reconhecimento facial.
Fonte: Udeemy Inc.³

4 METODOLOGIA UTILIZADA NO DESENVOLVIMENTO

Este trabalho foi desenvolvido por meio de uma pesquisa bibliográfica e a aplicação de uma teoria na prática, seguindo os passos e como foram desenvolvidos conforme destacados a seguir:

- a) Seleção e o estudo da bibliografia;
- b) Levantamento de ferramentas para apoiar no desenvolvimento profissional dos colaboradores em suas habilidades e competências;
- c) Análise comparativa entre a teoria e a prática utilizada no caso de estudo;
- d) Conclusões e considerações.

² Disponível em: www.techtudo.com.br/artigos/noticia/2012/04/como-funciona-o-reconhecimento-facial.html

³ Disponível em: www.udemy.com/reconhecimento-facial-com-java

5 DESENVOLVIMENTO

5.1. LINGUAGEM DE PROGRAMAÇÃO

Para desenvolver o sistema de reconhecimento facial, será utilizado o Java. Java é uma linguagem de programação orientada a objetos e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995, a mesma foi adquirida pela Oracle Corporation em 2008.

5.2 AMBIENTE DE DESENVOLVIMENTO

Será utilizado a IDE *open source* Eclipse para desenvolvimento Java, essa IDE suporta várias outras linguagens como C/C++, PHP, ColdFusion, Python dentre outras e é muito utilizada por desenvolvedores Android.

Para realizar o reconhecimento facial será utilizado a biblioteca OpenCV que possui os algoritmos de reconhecimento facial já preparados para realizar o reconhecimento após serem treinados. Essa biblioteca foi desenvolvida na linguagem C, C++ pela Intel por volta do ano 2000, a mesma é multiplataforma, livre para ser usada no meio acadêmico e comercial para o desenvolvimento de aplicações na área de Visão Computacional, possui licença do tipo BSD.

5.3 BANCO DE DADOS

Neste trabalho será utilizado um banco de dados que armazenará o ID de identificação, o qual será associado a pessoa no reconhecimento junto ao seu nome. Dentre vários bancos de dados será utilizado o gerenciador de banco de dados MySQL que utiliza a linguagem SQL (*Structure Query Language*). Sendo uma das linguagens mais populares para se inserir, acessar e gerenciar o conteúdo armazenado em banco de dados. Também será utilizado para administração do MySQL o aplicativo phpMyAdmin, que trata – se de um aplicativo web de código aberto para desenvolvimento em PHP e gerenciamento do MySQL pela internet.

É uma câmera geralmente de monitoramento que funciona com o protocolo da internet (IP), para receber dados de controle e envio de imagens via internet. Após o desenvolvimento do software, a mesma será integrada ao código para que todo reconhecimento facial ocorra por ela.

É uma fechadura com um dispositivo de destravamento por energia elétrica, são mais eficientes, pois podem ser abertas remotamente sem a necessidade de um chaveamento manual, muito utilizada em estabelecimentos comerciais. A mesma será integrada ao código para destravamento automático de porta por reconhecimento facial.

É uma plataforma de baixo custo voltada para projetos de eletrônica e programação, que possui uma mini placa com mini controladores e portas analógicas e digitais, como se fosse um pequeno computador. O mesmo possui licença hardware livre, ou seja, pode ser modificado e vendido de forma por qualquer um. Será utilizado para intermediar a comunicação entre o código de reconhecimento facial e a fechadura elétrica.

No desenvolvimento do software de reconhecimento facial será utilizado a biblioteca OpenCV em conjunto com a biblioteca JavaCV, onde há os algoritmos Eigenfaces, Fisherfaces e LBPH já prontos para executar o reconhecimento facial efetuando a escolha do mais adequado para o software em desenvolvimento. O JavaCV é um wrapper para utilizar a biblioteca OpenCV com o Java.

5.4 EIGENFACES

Para entendermos melhor o funcionamento do algoritmo, temos que ter em mente que imagens são nada mais que uma matriz de pixels com valores RGB. Nesse algoritmo é feita a extração das informações mais relevantes da imagem capturada, e as mesmas são codificadas para posterior comparação. Essa técnica foi desenvolvida por Sirovich e Kirby (1987), a face é “quebrada” em uma serie de componentes, ou vetores da matriz covariância principal e é analisada pelo PCA (*Principal Component Analysis*) que extrai as principais características. Como exemplo, dado uma imagem $X(x,y)$ em um espaço bidimensional $N \times N$, essa imagem pode ser encarada como um vetor de dimensão N^2 . Uma imagem com 210×210 pixels passa a ser considerado um vetor de dimensão 44.100, ou um ponto no espaço com 44.100 dimensões, são chamados de Eigenvectors. Se o vetor usado for igual ao vetor comparado, ocorre o reconhecimento facial.

Lembrando que o vetor gerado após todos os cálculos e análises feitas pelo algoritmo é chamada de “imagem média”, pois ele soma todas as fotos levando em conta a face como um todo e a reconstrói retirando a média das mesmas, assim recriando uma “nova” face e gerando uma imagem média. A figura 6 exemplifica de forma direta o processo.

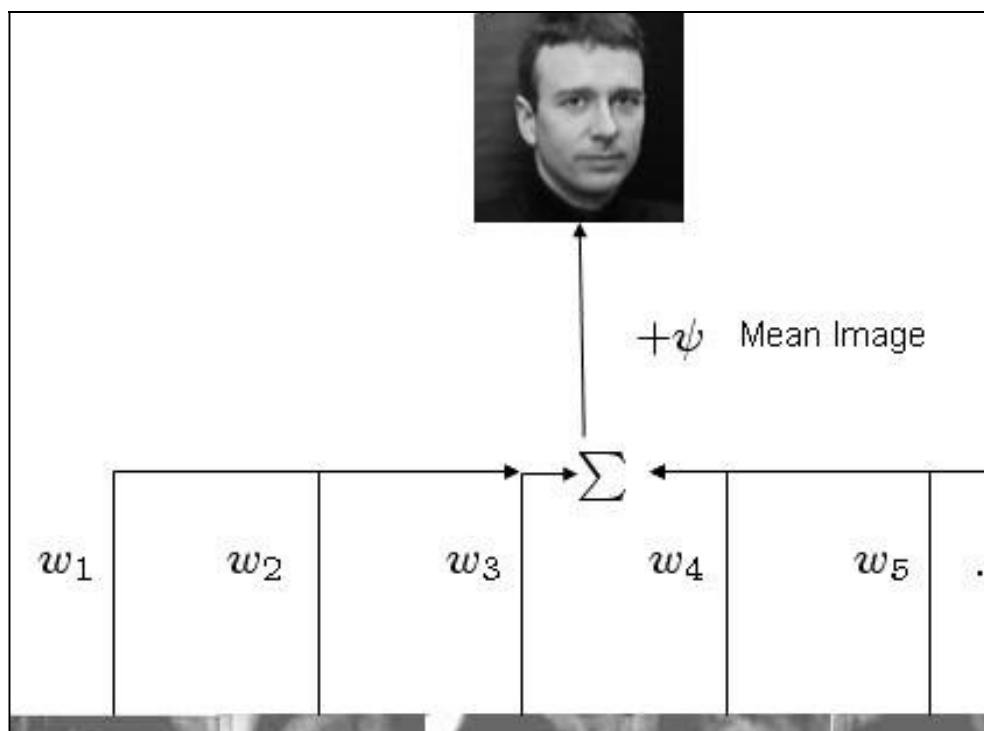


Figura 5: Geração da imagem média com Eigenfaces.
Fonte: ShubhenduTrivedi⁴

Utiliza como base o FLD (*Fisher Linear Discriminant*) que foi proposto por R. A. Fisher na década de 1930, nesse algoritmo é levado em conta como a luz reflete a face e as expressões faciais, onde é feita a maximização das imagens projetadas, a fim de melhorar a relação de classes e a distribuição de interclasses, pois as imagens utilizadas para treino são representadas por classes, onde cada classe específica representa uma pessoa.

Após o algoritmo gerar o vetor da imagem, o mesmo é projetado em um espaço de menor dimensão, para que assim haja menos espaços entre os vetores o que faz com que seja reduzido o espaço entre as características detectadas. A figura 6 abaixo mostra de forma direta o resultado do processamento feito pelo algoritmo, onde o mesmo utiliza a reflexão da luz para capturar as características, armazenando – a no vetor e matrizes das classes e interclasses para posterior reconhecimento facial.

⁴ Disponível em: <https://onionesquereality.wordpress.com/2009/02/11/face-recognition-using-eigenfaces-and-distance-classifiers-a-tutorial/>

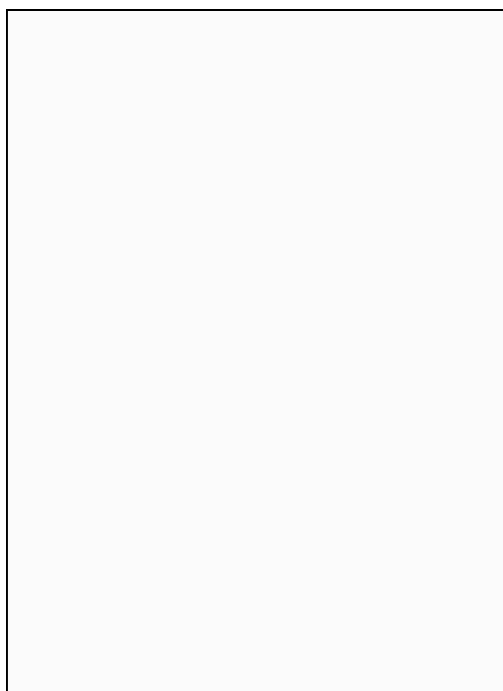
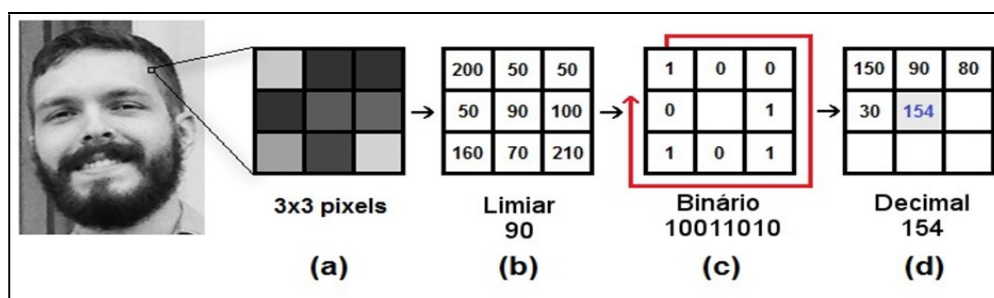


Figura 6: Captura das características refletidas pela luz.
 Fonte: Aleix Martinez (2011), Scholarpedia⁵

O operador *Local Binary Patterns* foi proposto por Ojala, Pietikäinen e Harwood em 1996, para análise de texturas em imagens, entretanto viu – se que a sua utilização poderia ser aplicada em outros campos, como por exemplo, no reconhecimento facial, pois permite a extração de características no processo de classificação de imagens. Neste tipo de algoritmo é geralmente utilizada uma matriz 3x3, onde é inserido um rótulo para os pixels da imagem, o pixel central da matriz é definido como base para os pixels vizinhos restantes.

Abaixo temos uma figura que exemplifica a ocorrência do processo. A primeira parte ocorre conforme figura 8b, onde é atribuída em forma de inteiros a intensidade dos pixels para a região específica da matriz, logo em seguida conforme figura 8c os valores da matriz são alterados para 1 quando o valor de cada célula é igual ou maior que o central e 0 para valor menor que o central da matriz. Assim sendo, ocorre a próxima fase, em que o valor binário obtido é convertido para decimal conforme figura 7.



⁵ Disponível em: www.scholarpedia.org/article/Fisherfaces

Figura 7: Etapas de processamento do LBPH.
Fonte: Prado (2018)

Na figura 8 abaixo temos o resultado do processamento do algoritmo.



Figura 8: Resultado do processamento do LBPH.
Fonte: Prado (2018)

A imagem gerada após o processamento é dividida em regiões organizadas em malha conforme figura 9 abaixo, logo em seguida os padrões de cada região são extraídos de forma individual e gerados um histograma de toda a imagem como ficar figura 9 abaixo. Após o histograma de toda a imagem ter sido gerado, o mesmo é utilizado para realizar o cálculo de similaridade entre os histogramas.

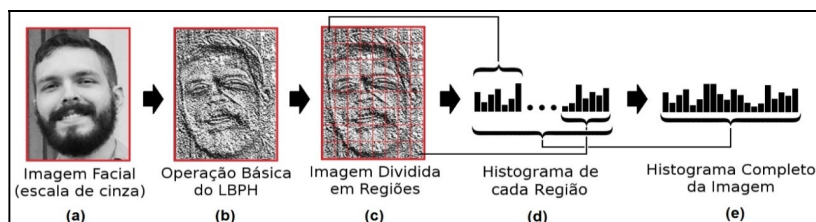


Figura 9: Processo de extração de histogramas do LBPH.
Fonte: Prado (2018)

Os operadores também podem ser do tipo circular, pois permitem utilizar pixels vizinhos de diferentes tamanhos. Para a operação é utilizado a letra P ($P > 0$) para o número de pixels vizinhos, ou seja, a quantidade de amostragens, e R ($R > 0$) que identifica o raio do círculo. A figura 10 abaixo representa como exemplo.

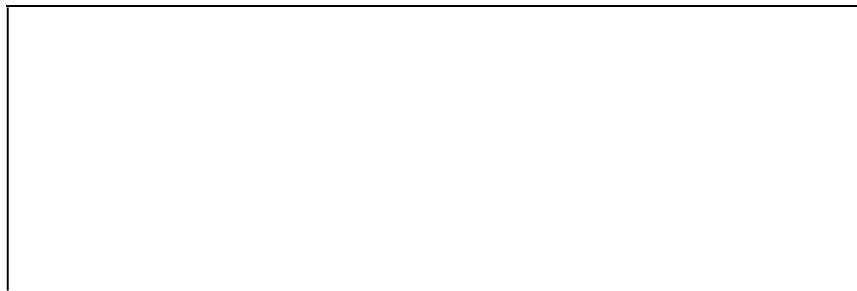


Figura 10: Operadores circulares do LBPH.
Fonte: Prado (2018)

Quanto maior o raio mais vizinho haverá, o que resultará em mais padrões, porém exigirá mais processamento computacional, em contra partida haverá mais detalhes capturados.

6 CONCLUSÕES E RECOMENDAÇÕES

O Código fonte do programa foi baseado no código do curso de reconhecimento facial da Udemy (www.udemy.com/reconhecimento-facial-com-java), que foi estudado pelos integrantes do grupo, todavia a parte de conexão com banco de dados, interface gráfica, integração com arduino e a fechadura elétrica foi produzida a partir do conhecimento dos integrantes do grupo, deve – se salientar também que o reconhecimento facial ocorre somente em nível 2D, assim sendo, pode ser burlado por uma foto da pessoa que está cadastrada no sistema. Com o desenvolvimento do software, estudo dos algoritmos de reconhecimento facial foi possível ter conhecimento a respeito do funcionamento dos mesmos e gerar documentação para os temas estudados, o que torna o trabalho relevante para o meio acadêmico em decorrência de pouca ou nenhuma documentação em português sobre esse tema.

Foi também possível obter dados de comparação entre os algoritmos de reconhecimento facial de forma a obter o melhor algoritmo dentre os três que foram estudados, tabela IV traz esses dados, bem como a experiência empírica que o protótipo do projeto agregou, de forma com que fosse possível visualizar os resultados de forma concreta conforme figura 26 e assim ajudar a identificar falhas no processo prático e corrigi-las, definir indicadores de tolerância e analisar os resultados. Em consequência do desenvolvimento do trabalho e do estudo realizado, em trabalhos futuros pretende – se realizar o reconhecimento facial em nível 3D para aperfeiçoamento do código em função do aumento do índice de acerto de faces detectadas.

REFERÊNCIAS

ARDUÍNO. Disponível em: <https://www.youtube.com/watch?v=sv9dDtYnE1g>. Acesso em: 23 ago 2019.

CCM. **Banco de dados.** Disponível em: <https://br.ccm.net/contents/65-bancos-de-dados>. Acesso em: 11 abr 2019.

CVC. **CS.YALE.** Disponível em: <http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html>. Acesso em: 07 abr 2019.

DIANIN, Antônio Henrique. **Diferenças entre reconhecimento e detecção facial.** Disponível em: <https://www.portaleducacao.com.br/conteudo/artigos/informatica/as-diferencas-entre-deteccao-de-rostos-e-reconhecimento-de-rostos/67656>. Acesso em: 23 ago 2019.

DUQUE, Luciano Henrique. **RSTP.** Disponível em: <https://www.leroymerlin.com.br/fechaduras-eletricas>. Acesso em: 23 ago 2019.

JAVA. **Detecção de Faces com Java e OpenCV.** Disponível em: <https://www.udemy.com/deteccao-de-faces-com-java-e-opencv>. Acesso em: 02 mar 2019.

JAVA. **O que é a Tecnologia Java e por que preciso dela?** Disponível em: https://www.java.com/pt_BR/download/faq/whatis_java.xml. Acesso em: 02 mar 2019.

JUNIOR, Paulo Nazareno Lagoia Fonseca. **UDP.** Disponível em: https://www.teleco.com.br/tutoriais/tutorialpushtotalk2/pagina_2.asp. Acesso em: 23 ago 2019.

MARTINEZ, Aleix. **Fisherfaces.** Disponível em: <http://www.scholarpedia.org/article/Fisherfaces>. Acesso em: 09 maio 2019.

MIRANDA, Willian. **Os 5 bancos de dados mais utilizados do mercado.** Disponível em: <http://aprendaplsql.com/oracle/os-5-bancos-de-dados-mais-utilizados-do-mercado/>. Acesso em: 11 abr 2019.

OLHAR DIGITAL. **Reconhecimento facial: como funciona a tecnologia.** Disponível em: https://olhardigital.com.br/fique_seguro/noticia/reconhecimento_facial_como_funciona_a_tecnologia/16703. Acesso em: 15 mar 2019.

PRADO, Kelvin Salton do. **Comparação de técnicas de reconhecimento facial para identificação de presença em um ambiente real e semicontrolado**. 2018. Dissertação para pós graduação em Sistemas de Informação

Universidade de São Paulo, São Paulo, 2018. Disponível em: https://www.teses.usp.br/teses/disponiveis/100/100131/tde-07012018-222531/publico/Corrigida_Kelvin_Salton.pdf. Acesso em: 07 abr 2019.

PRADO, Kelvin Salton do. **Face Recognition**: Understanding LBPH Algorithm. Disponível em: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>. Acesso em: 07 abr 2019.

PISA, Pedro. **Como funciona o reconhecimento facial**. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/04/como-funciona-o-reconhecimento-facial.html>. Acesso em: 15 mar 2019.

PIETIKÄINEN, Matti. **Local Binary Patterns**. Disponível em: http://www.scholarpedia.org/article/Local_Binary_Patterns. Acesso em: 07 abr 2019.

SANTOS, Erico Theis Pereira dos. **O que é reconhecimento facial**. Disponível em: <https://revistadigitalsecurity.com.br/artigo-seu-rosto-sua-chave-reconhecimento-facial-no-controle-de-acesso/>. Acesso em: 23 ago 2019.

SOUZA, Flavio Lima de. **Classificador Fisherface Fuzzy para o Reconhecimento de faces**. Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/110600/000791000.pdf>. Acesso em: 09 maio 2019.

TECHTUDO. **Câmera IP**. Disponível em: <https://www.techtudo.com.br/dicas-e-tutoriais/noticia/2016/04/camera-de-seguranca-ip-descubra-como-vigiar-sua-casa-pela-internet.html>. Acesso em 23 ago 2019.

TECMUNDO. **Como funcionam os sistemas de reconhecimento facial**. Disponível em: <https://www.tecmundo.com.br/camera-digital/10347-como-funcionam-os-sistemas-de-reconhecimento-facial.htm>. Acesso em: 15 mar 2019.

TELECO. **TCP**. Disponível em: https://www.teleco.com.br/tutoriais/tutorialip/pagina_5.asp. Acesso em 23 ago 2019.

TRIVEDI, Shubhendu. **Face Recognition using eigenfaces and Distances Classifiers: A Tutorial**. Disponível em: <https://onionesquereality.wordpress.com/2009/02/11/face-recognition-using-eigenfaces-and-distance-classifiers-a-tutorial>. Acesso em: 09 maio 2019.

UDEMY. **Reconhecimento Facial com Java**. Disponível em: <https://www.udemy.com/reconhecimento-facial-com-java>. Acesso em: 02 mar 2019.