

DOI: 10.5748/19CONTECSI/REX/DSC/6960

ANÁLISE DE ALGORITMOS DE RECONHECIMENTO FACIAL: EIGENFACES E DEEPFACES.

Dilermando Piva Jr ; <https://orcid.org/0000-0002-2534-9618>
Fatec Itu / Fatec Sorocaba

Ed Carlos da Silva Pereira ; <https://orcid.org/0000-0003-1875-4137>
Fatec Itu



ANÁLISE DE ALGORITMOS DE RECONHECIMENTO FACIAL: EIGENFACES E DEEPFACES.

ANALYSIS OF FACIAL RECOGNITION ALGORITHMS: EIGENFACES AND DEEPFACES.

RESUMO: O objetivo desse artigo é descrever e analisar o comportamento de dois dos principais algoritmos de reconhecimento facial: Eigenfaces, Deepfaces, evidenciando seus princípios de funcionamento e as tecnologias que foram implementadas na sua criação fazendo com que cada um tenha uma característica única, o Eigenfaces se baseia em cálculos matemáticos geométricos enquanto o Deepfaces tem uma tecnologia mais moderna que simula o funcionamento do cérebro humano para realizar suas análises, realizando testes para verificar seu desempenho de performance e de acurácia. Para a obtenção desses resultados, ambos os algoritmos foram submetidos a um único dataset contendo cerca de 1000 imagens aleatórias que foram derivadas do conjunto de três dataset públicos: Yalefaces, LFW (Labeled Faces in the Wild) e Dogs vs cats, onde 600 imagens contêm faces humanas e 400 de animais subdividido em 200 imagens de cães e 200 imagens de gatos, de forma aleatória algumas imagens contêm escala de tom de cinza e outras são coloridas. Ao analisar os resultados, foi detectado que os algoritmos apresentam uma acurácia superior a 90% no quesito de reconhecimento facial, porém, em tempo de execução de processamento o Deepface foi o que conseguiu fazer a análise de todas as imagens e fazer o reconhecimento em menos tempo, mostrando assim que a tecnologia de redes neurais auxilia o algoritmo a realizar um aprendizado profundo sobre os arquivos que lhe foram entregues de forma mais performática fazendo assim dessa tecnologia uma tendencia a ser mais utilizada para a criação de novos algoritmos.

ABSTRACT: The objective of this article is to describe and analyze the behavior of two of the main facial recognition algorithms: Eigenfaces, Deepfaces, evidencing their working principles and the technologies that were implemented in their creation making with each one having a unique characteristic, Eigenfaces is based on geometric mathematical calculations while Deepfaces has a more modern technology that simulates the functioning of the human brain to perform its analysis, performing tests to verify its performance and accuracy. To obtain these results, both algorithms were submitted to a single dataset containing about 1000 random images that were derived from the set of three public datasets: Yalefaces, LFW (Labeled Faces in the Wild) and Dogs vs cats, where 600 images contain human faces and 400 of animals subdivided into 200 images of dogs and 200 images of cats, randomly some images contain grayscale and others are colored. When analyzing the results, it was detected that the algorithms present an accuracy of more than 90% in terms of facial recognition, however, in the processing runtime, Deepface was the one that managed to analyze all the images and perform the recognition in less time, thus showing that the technology of neural networks helps the algorithm to carry out a deep learning about the files that were delivered to it in a more performative way, thus making this technology a tendency to be more used for the creation of new algorithms.

PALAVRAS-CHAVE: Reconhecimento facial. Algoritmos para reconhecimento facial. Eigenfaces. Deepfaces. Aprendizagem Profunda.

KEYWORD: Facial recognition. Algorithms for facial recognition. Eigenfaces. Deepfaces. Deep Learning.

1 INTRODUÇÃO

Os humanos baseiam-se frequentemente na face para realizar o reconhecimento de indivíduos. Por sua vez, com os avanços das últimas décadas, principalmente na capacidade de poder computacional, permitem aos computadores, atualmente, realizar um reconhecimento similar aos seres humanos e de forma automática. Os primeiros algoritmos, utilizavam modelos geométricos simples e demoravam muito para fazer seus processos, as entradas dos dados eram feitas de forma manual com auxílio de uma pessoa que era responsável por inserir características da face, tais como posição dos olhos, do nariz, da boca. A próxima geração de algoritmos, se baseavam em cálculos algébricos, como por exemplo, o Eigenfaces que utilizava o PCA (Principal Component Analysis ou Análise de Componentes Principais). Em 2014, o Deepfaces e o DeepID, criados pela empresa Facebook alcançaram precisão de ponta quando foram utilizados em um famoso teste de desempenho que foi feito por pesquisadores da Universidade de Massachusetts, utilizando um banco de dados de fotografias de rostos projetado para estudar o problema do reconhecimento irrestrito, conhecido como Labeled Faces in the Wild.

2 METODOLOGIA

Além de cursos, treinamentos e livros, este projeto teve sua base teórica desenvolvida com base em artigos científicos nacionais e internacionais, sempre procurando explicações técnicas e mais profundas de: o que é reconhecimento facial, como é feito e quais técnicas são utilizadas para fazer o reconhecimento de faces em imagens a partir de algoritmos computacionais.

Após a explicação de como funciona a arquitetura desses algoritmos, um teste de desempenho é descrito, onde ambos foram submetidos a um único dataset contendo 1000 (mil) imagens aleatórias derivadas do conjunto de três dataset públicos: Yalefaces, Labeled Faces in the Wild e Dogs vs cats. Foi utilizado como critério de análise os seguintes parâmetros: tamanho do código, tempo de execução e taxa de acerto assim, ao analisar a técnica de implementação e os resultados obtidos, chegasse à conclusão de que os algoritmos que utilizam redes neurais são mais performáticos e tendem a ser os mais utilizados para a criação de novos algoritmos.

3 ALGORITMO EIGENFACES

Turk e Pentland (1991) descrevem o foco do método Eigenfaces como:

[...] método é utilizado para em realizar a detecção e o reconhecimento facial. Sua principal característica é fazer a análise de imagens identificando seus componentes principais. Sua abordagem inicial utiliza a álgebra linear para fazer a representatividade de baixa dimensão das imagens faciais. (TURK e PENTLAND, 1991)

O método Eigenfaces, consiste em fazer a extração de toda informação que seja relevante da imagem facial a ser analisada e fazer a codificação dessa informação o mais eficiente possível para

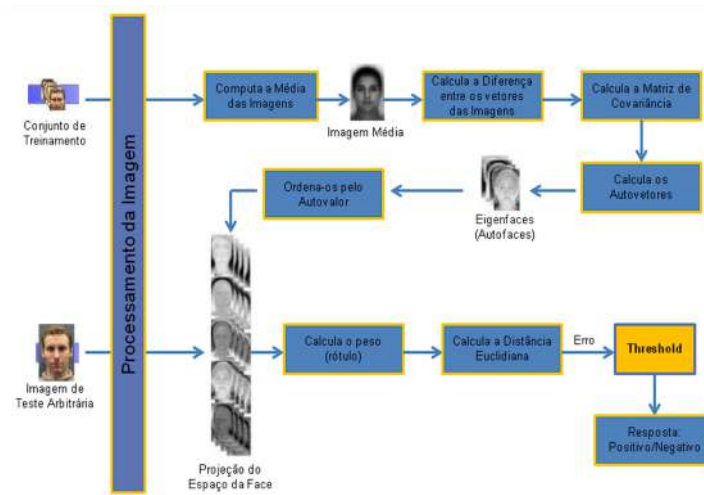
depois fazer a comparação da face codificada com toda uma base de dados composta por faces codificadas de forma semelhante.

O processo em si consiste em pegar uma imagem, fazer toda a sua leitura e comparar com outras imagens armazenadas no banco de dados, a fim de realizar a comparação dos padrões de uma imagem a outra, fazendo assim a confirmação de igualdade. Para tanto, utiliza o método PCA. Resumidamente, o método PCA consiste em pegar um dado ou uma matriz de grande dimensão e reduzir em pequenas partes, procurando preservar o máximo possível seus valores para fazer comparação da similaridade entre elas. (PESSOA, 2019)

Eigenfaces foi criado pensando nas características que a face de uma pessoa possa ter, pois, fazer o reconhecimento de um rosto é semelhante a qualquer outro sistema biométrico onde cada indivíduo tem suas características próprias, como por exemplo, a impressão digital que também tem suas estruturas e características únicas que determinam as características de um único indivíduo.

A Figura 1 apresenta o fluxo de funcionamento do Eigenface de maneira resumida.

Figura 1 - Processamento do Eigenfaces



Fonte: (CORREIA, PITERI, *et al.*, 2014)

4 ALGORITMO DEEPFACES

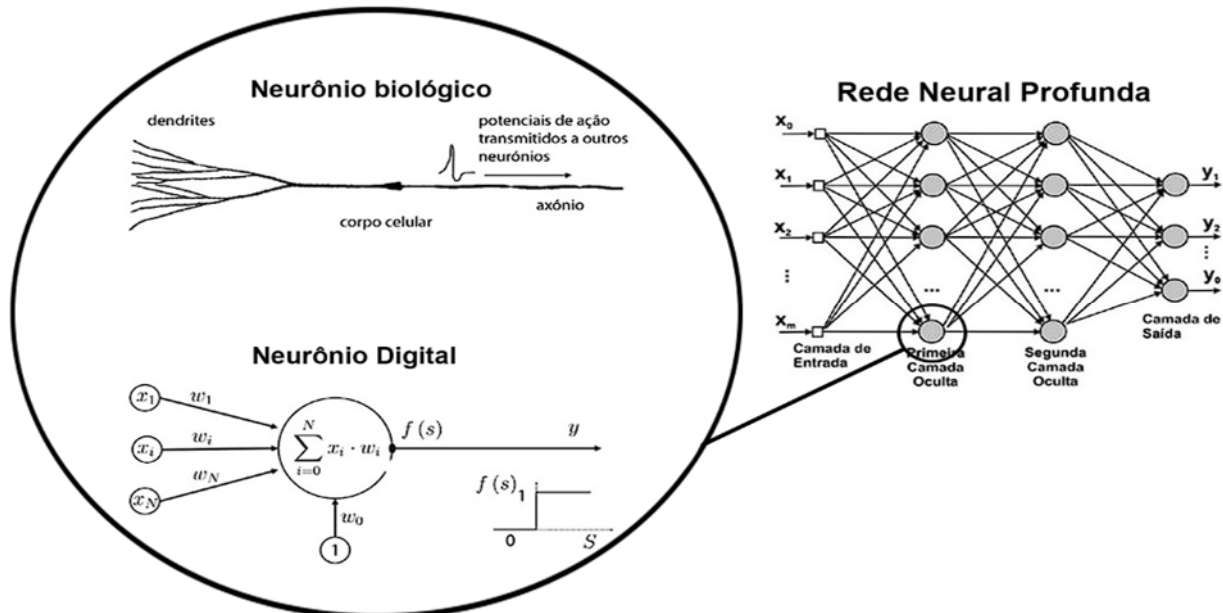
Taigman e Yang descrevem o método DeepFaces como:

[...] um método utilizado para detecção e reconhecimento facial, que se utiliza de uma tecnologia totalmente inovadora para fazer o reconhecimento facial em imagens. Criado pela empresa Facebook, o Deepfaces utiliza redes neurais profundas para fazer as análises e alcançou uma precisão de ponta quando foi utilizado em um famoso teste de desempenho feito por pesquisadores da Universidade de Massachusetts, superando o desempenho humano pela primeira vez. (TAIGMAN, YANG, *et al.*, 2014)

Desde então, técnicas de reconhecimento facial profundo, que aproveitam a arquitetura hierárquica para aprender a representação discriminativa de rostos, melhoraram drasticamente o desempenho dos algoritmos e promoveram vários aplicativos bem-sucedidos no mundo real. Deep Neural Network (DNN), é um conjunto de algoritmos modelados com inspiração no funcionamento do cérebro humano. Cada nó da rede neural foi projetado para atuar como se fosse um neurônio, de forma que consiga reconhecer padrões, agrupando e analisando dados coletados no mundo real,

como por exemplo, imagens, sons, textos ou séries temporais, sempre traduzindo-os em alguma coisa.

Figura 2 - Comparação de neurônio biológico com neurônio digital e exemplo de rede neural profunda

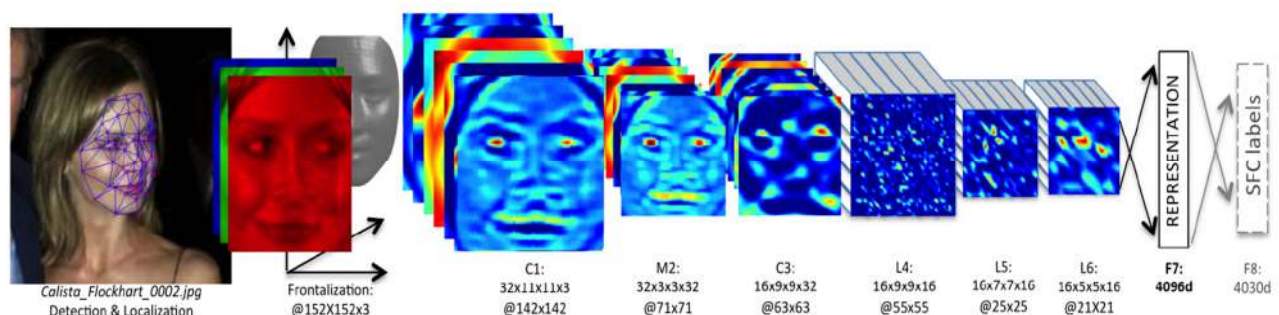


Fonte: adaptado de (FACURE, 2017) e (AGUIAR e AGUIAR, 2022)

As redes neurais geralmente são utilizadas em projetos de reconhecimento e classificação de imagens, uma de suas principais características é a capacidade de aprender com outros exemplos que é conhecido como aprendizagem supervisionada, onde a rede neural é treinada por outros exemplos em grande quantidade, consistindo em pares de entrada-saída. Nas tarefas de reconhecimento de objetos, esse par será a imagem de entrada e a etiqueta correspondente ao nome do objeto.

O treinamento da rede neural é um processo iterativo que reduz o desvio da saída da rede de uma determinada etiqueta correspondente a esta imagem. Esse processo é formado por etapas que é conhecida como épocas de aprendizado. Em cada uma das quais os pesos da rede neural são ajustados aos parâmetros das camadas ocultas da rede.

Figura 3 - Processamento do Deepfaces



Fonte: (TAIGMAN, YANG, *et al.*, 2014)

As redes neurais profundas geralmente consistem em três ou mais camadas: a camada de entrada, a camada oculta e a camada de saída conforme ilustrado na Figura 2; em alguns casos, as camadas de entrada e saída não são levadas em consideração e, em seguida, o número de camadas na rede é calculado pelo número de camadas ocultas.

O Deepfaces, tem uma arquitetura de rede neural profunda muito bem elaborada, onde há n quantidades de camadas responsáveis para executar processos de filtragem nas imagens recebidas onde elas passam por processos de detecção, alinhamento, representação, classificação, e mesmo ao receber uma imagem em 2D faz a conversão para 3D e vice e versa.

5 TESTES DE DESEMPENHO E ANÁLISE

Ao realizar o teste comparativo de desempenho com os dois algoritmos, foi utilizado como critério de análise os seguintes parâmetros: tamanho do código, tempo de execução e taxa de acerto.

Código 1 - Bibliotecas do Eigenfaces

```
import cv2
import os
import numpy as np
import time
```

A implementação do Eigenfaces chegou a 460 linhas que foram divididos em dois arquivos onde um foi programado para fazer o treinamento das imagens e o outro foi programado para fazer o reconhecimento das novas imagens.

Código 2. Método para criar o modelo de treinamento

```
reconhecedor = cv2.face.EigenFaceRecognizer_create()
reconhecedor.read('classificadorEigen.yml')
```

O algoritmo cria um modelo de treinamento com as características de cada imagem. Isso ocorre, pois o algoritmo em questão é focado em reconhecimento facial. Dessa forma, para fazer a detecção da face na imagem foi utilizado o Haarcascade conforme ilustrado no Código 3. (OpenCv, 2021)

Código 3. Método para detecção facial

```
imagemFace = cv2.cvtColor(cv2.imread(caminhoImagem), cv2.COLOR_BGR2GRAY)
imagemFace = cv2.resize(imagemFace, (400, 400))
# Detectar a face na imagem
faceDetect = cv2.CascadeClassifier('C:\\Users\\gog_e\\OneDrive\\Ambiente de Trabalho\\Eigenfaces v2\\venv\\Lib\\site-packages\\cv2\\data\\haarcascade_frontalface_default.xml')
faces = faceDetect.detectMultiScale(imagemFace, scaleFactor=1.1, minNeighbors=4, minSize= (200, 200))
```

Para esses algoritmos foram utilizadas 1000 imagens, sendo 600 com humanos e 400 com animais. Dentre as 600 imagens com face, foram escolhidas 10 de forma aleatória afim de verificar se o modelo conseguiria fazer o reconhecimento.

O segundo algoritmo testado, o Deepfaces, mostra como a implementação de um algoritmo de reconhecimento facial com técnicas de DNN ajudou a evoluir em relação as técnicas antigas, com apenas 165 linhas de código em um único arquivo o algoritmo conseguiu fazer as mesmas coisas que seu antecessor e um pouco mais.

Código 4 - Método de detecção facial do Deepfaces

```
detectors = ["opencv", "ssd", "mtcnn", "dlib", "retinaface"]
face = DeepFace.detectFace(imagem, detector_backend = detectors [4],
enforce_detection = False)
```

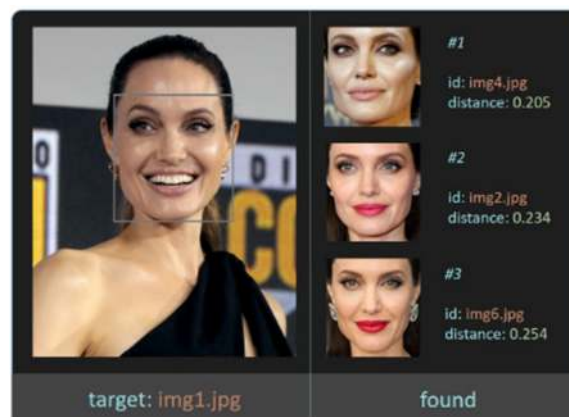
Conforme ilustrado na Figura 4, opencv é o nome dado ao modelo de treinamento que faz a utilização do HaarCascade para fazer a detecção da face assim como mtcnn é feita a utilização da CNN (Convolutional Neural Network ou Redes Neurais Convolucionais) e o dlib é feita a utilização do Yolo (You Only Look Once).

Código 5. Método de reconhecimento facial do Deepfaces

```
def reconhecerFace (foto, directorio):
df = DeepFace.find(img_path = foto, db_path = directorio)
return df
```

Para fazer o reconhecimento facial, basta chamar o método ilustrado no trecho do Código 5, passando o diretório onde contém todas as imagens e passando a imagem que quer obter suas semelhantes.

Figura 4 - Resultado do reconhecimento facial Deepfaces



Fonte: (SERENGIL e OZPINAR, 2020)

Com uma precisão de 100%, o algoritmo conseguiu reconhecer todas as imagens entregues. Identificou todas as imagens em um tempo muito curto. Enquanto o Eigenfaces levou cerca de 50 minutos para fazer o treinamento e mais 10 minutos para fazer o reconhecimento. Deepfaces levou 30 minutos completos para fazer todo o processamento das faces nas imagens e menos de 1 minuto para o reconhecimento de 100% das faces.

Ambos os algoritmos foram testados 5 vezes, onde os dois algoritmos apresentaram os mesmos resultados obtidos no primeiro teste.

Após realizar testes com os algoritmos Eigenfaces e Deepfaces, foi identificado que ambos os algoritmos têm alto desempenho no reconhecimento facial, com precisão acima de 90%, em ambos os casos.

6 CONSIDERAÇÕES FINAIS

Após expor brevemente as características inerentes a cada um dos algoritmos estudados, passou-se a análise objetiva dos algoritmos com uma base de imagens preparadas para tal finalidade. Resgatando o objetivo inicialmente traçado, que era entender e analisar os métodos de implementação que cada algoritmo utiliza para fazer a detecção e reconhecimento facial e verificar a evolução e o que difere em cada implementação, focando na acurácia e no tempo de processamento das imagens que ambos desempenham. Ao finalizar o processo, observa-se que o objetivo inicialmente traçado foi cumprido e que ambos os algoritmos apresentam uma precisão superior a 90%.

Para trabalhos futuros, a gama de algoritmos e imagens podem ser ampliados, com a finalidade de uma maior abrangência e identificação de nichos específicos para utilização de um ou outro algoritmo e também a possibilidade de teste em vários tipos de hardware, verificando se o desempenho do tempo de processamento de imagens de ambos continua conforme descrito relatório.

REFERÊNCIAS

- AGUIAR, D.; AGUIAR, P. Percepções e redes neurais artificiais. **REVISTA DE CIÊNCIA ELEMENTAR**, Porto, v. 10, n. 1, p. 15, Março 2022. <https://doi.org/10.24927/rce2022.004> .
- CORREIA, T. A. et al. Development of an application for security based in face recognition on Android platform. **X Workshop de Visão Computacional - WVC**, Uberlândia, 2014. 138.
- FACURE, M. Repositório de Matheus Facure. **Github**, 2017. Disponível em: <https://matheusfacure.github.io/2017/02/09/rnn-bike-demand/> . Acesso em: 22 Julho 2022.
- GRIGSBY, S. Augmented Cognition: Intelligent Technologies: **12th International Conference. Las Vegas: Springer**, v. 10915, 2018.
- OPENCV. **OpenCv**, 2021. Disponível em: <https://opencv.org/> . Acesso em: 10 Junho 2022.
- PESSOA, W. Perfil Medium. **Medium**, 2019. Disponível em: <https://medium.com/@williangp/reconhecimento-de-padr%C3%B5es-eigenfaces-e4cef8f04919> . Acesso em: 20 Fevereiro 2022.
- SERENGIL, S. I.; OZPINAR,. LightFace: A Hybrid Deep Face Recognition Framework. **Innovations in Intelligent Systems and Applications Conference (ASYU)**, 2020. 1-5.
- TAIGMAN, Y. et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. **IEEE Conference on Computer Vision and Pattern Recognition**, Menlo Park, CA, USA; Tel Aviv, Israel, 2014. 1701-1708.
- TURK, ; PENTLAND, A. Eigenfaces for Recognition. **Journal of Cognitive Neuroscience**, Massachusett, v. 3, n. 1, p. 71-86, 1991. ISSN <https://doi.org/10.1162/jocn.1991.3.1.71> .