http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2021.v13.n2.e354
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 10/03/2021 Revisado: 27/05/2021 Aceito:01/08/2021

ANÁLISE DE MÉTODOS DE DETECÇÃO E RECONHECIMENTO DE FACES UTILIZANDO VISÃO COMPUTACIONAL E ALGORITMOS DE APRENDIZADO DE MÁQUINA

ANALYSIS OF FACE DETECTION AND RECOGNITION METHODS USING COMPUTER VISION AND MACHINE LEARNING ALGORITHMS

Lucas José da Costa[1], Thiago Luz de Sousa[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Danillo Roberto Pereira[1], Almir Olivette Artero[2], Marco Antonio Piteri[2]

[1]Faculdade de Informática de Presidente Prudente, Unoeste - Universidade do Oeste Paulista, Presidente Prudente
lucasjcosta.98@gmail.com, thiagotls58@gmail.com, chico@unoeste.br, llalmeida@unoeste.br, danilopereira@unoeste.br
[2]Faculdade de Ciências e Tecnologia, UNESP - Universidade Estadual Paulista
Departamento de Matemática de Computação, Presidente Prudente
almir.artero@unesp.br, marco.piteri@unesp.br

RESUMO – O avanço da tecnologia das últimas décadas tem proporcionado muitas facilidades para a humanidade em várias aplicações, e a tecnologia de reconhecimento facial é uma delas. Existem vários problemas a serem resolvidos para se realizar o reconhecimento de faces a partir de imagens digitais, como variação de iluminação do ambiente, mudança das características físicas do rosto e resolução das imagens utilizadas. Este trabalho buscou realizar uma análise comparativa entre alguns dos métodos de detecção e reconhecimento facial, assim como o tempo de execução dos mesmos. Foram utilizados os algoritmos de reconhecimento facial Eigenface, Fisherface e LBPH em conjunto com o algoritmo de detecção facial Haar Cascade, todos da biblioteca OpenCV. Também foi explorado o uso de uma rede neural CNN para reconhecimento facial em conjunto com o algoritmo de detecção facial HOG, estes da biblioteca Dlib. O trabalho almejou, além de analisar os algoritmos com relação a taxas de acertos, fatores como grau de confiabilidade e tempo de execução também foram considerados.
Palavras-chave: Detecção facial, reconhecimento de face, visão computacional, aprendizado de máquina.

ABSTRACT – The advancement in technology in recent decades has provided many facilities for humanity in various applications, and facial recognition technology is one of them. There are several problems to be solved to perform face recognition from digital images, such as varying ambient lighting, changing the face physical characteristics and resolution of the images used. This work aimed to perform a comparative analysis between some of the detection and facial recognition methods, as well as their execution time. We use the Eigenface, Fisherface and LBPH facial recognition algorithms in conjunction with the Haar Cascade face detection algorithm, all from the OpenCV library. We also explored the use of CNN neural network for facial recognition in conjunction with the HOG facial detection algorithm, these from the Dlib library. The work aimed, besides analyzing the algorithms in relation to hit rates, factors such as reliability and execution time were also considered.

Keywords: Facial detection, face recognition, computer vision, machine learning

1. INTRODUÇÃO

Nos últimos anos é muito comum encontrar o uso de tecnologias de biometria para diversos fins e não há dúvidas de que isso está trazendo cada vez mais comodidade, conforto e segurança. Uma pesquisa da Visa, em parceria com a AYTM Market Research, apontou que 89% dos brasileiros preferem a biometria ao invés do uso de senhas, e que é mais fácil de utilizar e mais seguro (MIRANDA, 2017). O que anteriormente poderia ser classificado como uma tarefa extremamente desafiadora e complexa, a detecção de faces e de suas características como olhos, nariz, boca e, até mesmo, derivar emoções de suas formas, tornou-se uma tarefa possível de ser resolvida com o uso de técnicas de *Deep Learning*, e bibliotecas de processamento digital de imagens e visão computacional, como a biblioteca OpenCV (PRADO, 2017a).

Os estudos na área de visão computacional vêm proporcionando inúmeros benefícios para sociedade. Entre esses benefícios, estão a melhoria em sistemas de segurança (tecnologias de detecção facial estão sendo utilizadas em aplicações de reconhecimento de face para automatizar o processo de autenticação, buscando substituir o uso de senhas e impressões digitais), novas aplicações de entretenimento (jogos e outras atividades relacionadas), e em diversos outros processos que envolvam autenticação. Muitas empresas vêm cada vez mais investindo em tecnologias de reconhecimento facial para autenticar seus processos digitais e aprimorar serviços, entre elas Nubank, 99, Loggi, Banco Original, Neon e Gol (BALAN, 2019).

O Departamento Estadual de Trânsito do Rio Grande do Norte (Detran) iniciou o processo de validação facial nos serviços referentes à habilitação, renovação de CNH, alteração de categoria, inclusão de atividade remunerada e outros que exigem o sistema de biometria. A medida traz mais segurança no procedimento de captura de dados, possibilitando o combate a possíveis fraudes. O procedimento consiste em submeter o usuário a uma análise computadorizada da sua imagem, que é comparada com as informações fotográficas que constam no sistema de dados do Órgão gerador, apontando a aproximação da veracidade da imagem avaliada (PORTAL, 2019).

Um dos principais grupos de cosméticos do país, o Boticário, idealiza o desenvolvimento de uma ferramenta de identificação por reconhecimento facial. O objetivo é identificar o cliente quando ele entra na loja, para que a consultora possa fazer uma abordagem mais assertiva (ÉPOCA NEGÓCIOS, 2019).

Este trabalho contribui com uma análise comparativa entre os métodos de detecção e reconhecimento facial mais comumente utilizados na literatura, em relação ao tempo de execução e desempenho deles. Foram utilizados os algoritmos de reconhecimento facial Eigenface, Fisherface e LBPH em conjunto com o algoritmo de detecção facial Haar Cascade, todos da biblioteca OpenCV. Também foi explorada a rede neural CNN (*Convolutional Neural Networks*) para reconhecimento facial em conjunto com o algoritmo de detecção facial HOG, estes da biblioteca Dlib. Os resultados são apresentados comparando *datasets* de imagens que foram utilizados para treinamento, os parâmetros aplicados, taxa de acerto e o tempo de treinamento dos mesmos. São apresentadas tabelas comparativas de desempenho de cada algoritmo e enfatizando aquele que melhor se destaca nos testes.

As demais seções deste trabalho estão organizadas da seguinte maneira: na Seção 2 são apresentados alguns trabalhos relacionados; na Seção 3 é apresentada a metodologia e os algoritmos utilizados para análise comparativa; na Seção 4 são apresentados os experimentos realizados e os resultados obtidos; por fim, a Seção 6 apresenta as considerações finais do trabalho.

2. TRABALHOS RELACIONADOS

Hirakuri (2003) apresentou um sistema que adapta o *Neocognitron* para reconhecimento de atributos faciais pré-definidos: olho direito, olho esquerdo, nariz e lábios. O *Neocognitron* é uma rede neural hierárquica proposto por *Kunihiko Fukushima* em 1979, na qual a informação dada na camada de entrada é

processada passo a passo em cada nível da multicamada. Esse sistema é composto por duas redes neurais: Rede Neural para Detecção de Pontos de Controle (RNDPC) e Rede Neural para Reconhecimento Facial (RNRF). São duas etapas, sendo que na primeira é recebido um padrão (imagem) de entrada e é realizado o pré-processamento do mesmo, seguido da extração de pontos de controle pela rede RNDPC, sendo três níveis de detecção: bordas, linhas e pontos extremos. Na segunda etapa é realizada a extração de atributos (treinamento) e reconhecimento facial pela rede RNRF. Para realização dos testes e análise de resultados, o autor construiu um banco de imagens com 210 imagens de faces de sete pessoas, obtidas a partir de uma máquina fotográfica digital Kodak 210 A, com resolução de 1152x864 pixels. A partir do banco de imagens foi gerado um banco de imagens de atributos de faces utilizando o software de edição de imagens *Adobe Photoshop* 7.0. Foram extraídos atributos (lábios, nariz, olho esquerdo e olho direito) utilizados no processo de reconhecimento.

Em Correia et al. (2014) foi desenvolvida uma aplicação de segurança baseada em reconhecimento facial para plataforma Android utilizando Eigenfaces e LBP (*Local Binary Patterns*). As funcionalidades associadas à aplicação desenvolvida se concentram em dois módulos – o de segurança e o de reconhecimento facial. Na metodologia de segurança foi implementado um mecanismo de bloqueio onde o usuário define previamente quais aplicações deseja submeter. Quando ele ativa a segurança, um serviço em *background* é disparado, o qual detecta se alguma aplicação foi iniciada e analisa se ela faz parte ou não da lista de bloqueio. Se fizer parte, o bloqueio é disparado pelo serviço, que é a tela de reconhecimento facial, caso contrário nenhuma ação é lançada. A metodologia de reconhecimento facial foi dividida nas seguintes etapas: aquisição da imagem, detecção da face na imagem, pré-processamento da imagem, extração de características definidas e, por fim, a classificação. A partir da captura da imagem através da API (*Application Programming Interface*) disponibilizada pelo Android, o próximo passo foi efetuar a detecção da face na mesma. Na etapa de detecção da face, os autores utilizaram ferramentas do Android, que aplicam técnicas baseadas na segmentação da pele para efetuar a detecção. A etapa de pré-processamento é crucial para o sucesso da abordagem de reconhecimento facial, pois nesse procedimento o objetivo é conservar o padrão do objeto eliminando os fatores que dificultam a extração de suas características. O módulo de reconhecimento facial realizou a extração e seleção de características da face, efetuando a sua classificação. No desenvolvimento realizado foram empregadas duas metodologias holísticas, Eigenfaces e LBP (*Local Binary Patterns*), que são disponibilizadas pela biblioteca OpenCV. A aplicação desenvolvida foi escrita na linguagem de programação Java, mas as funções de pré-processamento da imagem e métodos de reconhecimento facial foram escritas na linguagem de baixo nível C/C++, para ter um ganho no poder de processamento e aproveitamento da memória do dispositivo, que executa todo o fluxo da aplicação, sendo que a interface de ambos módulos é realizada através das ferramentas disponibilizadas pelo NDK (*Native Development Kit*). Todas as imagens utilizadas nos experimentos realizados são frontais, porém, foram capturadas em ambientes comuns dos usuários, onde os fatores externos não são controlados, como orientação, distância e condições de iluminação variáveis, em diferentes locais e períodos do dia.

No trabalho de Petri et al. (2013) foi proposta uma metodologia de detecção de faces em imagens digitais utilizando algumas das técnicas já existentes e consolidadas na área de detecção facial, buscando otimizar o processo que ainda tem muito espaço para isso. No trabalho foram escolhidas as abordagens de detecção de cor de pele humana, seguida de detecção de borda Canny e transformação Hough para identificar regiões circulares da imagem. Dada uma imagem de entrada, inicialmente é transformada do espaço de cor de RGB para HSV, na intenção de obter maior desempenho na detecção dos pixels de pele humana. Em seguida, são realizadas operações de segmentação da imagem com o objetivo de facilitar a classificação dos pixels como pele ou não-pele através dos limiares dos canais HSV. Na sequência foram realizadas algumas transformações na imagem com compensação de iluminação e transformação para tons de cinza da imagem original, seguido de uma operação lógica (AND) entre a imagem monocromática e a segmentada, para posterior detecção de bordas utilizando o algoritmo de Canny. Em função da possibilidade de diminuir a ocorrência de falsos positivos, a

transformada de Hough foi utilizada para encontrar círculos (associados a olhos humanos) aumentando a consistência da metodologia proposta. O último passo foi apresentar a imagem original de entrada com um retângulo indicando a localização da face humana. O trabalho foi desenvolvido utilizando a linguagem de programação C/C++, utilizando imagens do *dataset* CVL Face Database composta por 798 imagens.

3. METODOLOGIA

Neste trabalho foi realizada uma análise comparativa de alguns dos principais algoritmos de detecção e reconhecimento facial, tendo como critérios o tempo de execução do algoritmo e a taxa de acerto.

Foram utilizados os algoritmos de reconhecimento facial Eigenfaces (TURK; PENTLAND, 1991), Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) e LBPH (*Local Binary Patterns Histograms*) (STEKAS; HEUVEL, 2016) em conjunto com o algoritmo de detecção facial Haar Cascade (VIOLA; JONES, 2001), todos da biblioteca OpenCV (OPENCV, 2018). Assim como o uso da rede neural CNN (*Convolutional Neural Networks*) (YAN et al., 2017) para reconhecimento facial em conjunto com o algoritmo de detecção facial HOG (*Histograms of Oriented Gradients*) (TRIGGS; DALAL, 2005), estes da biblioteca Dlib (DLIB, 2019). A linguagem de programação escolhida para implementação dos algoritmos foi a linguagem Python.

Foram utilizados os *datasets* de imagens de faces públicos Caltech (WEBER, 1999), Georgia Tech (GEORGIA TECH, 1999) e Yale (UCSD COMPUTER VISION, 1997). O Caltech possui 445 imagens, em que 393 imagens foram utilizadas para treinamento e 52 imagens para teste. O Georgia Tech possui 750 imagens, em que 650 imagens foram utilizadas para treinamento, e 100 imagens utilizadas para testes de reconhecimento. O *dataset* Yale possui 165 imagens, sendo que 130 imagens foram utilizadas para treinamento dos algoritmos e 30 utilizadas para testes de reconhecimento.

A Figura 2 apresenta uma imagem colorida da face de uma pessoa do *dataset* Caltech. A Figura 3 apresenta a imagem de uma pessoa do *dataset* Georgia Tech de um ângulo mais afastado e com objetos no fundo. A Figura 4 apresenta uma amostra de imagem contendo a face de uma pessoa do *dataset* Yale.

Figura 2. Imagem de uma pessoa do *dataset* Caltech.



Fonte: (WEBER, 1999).

Figura 3. Imagem de uma pessoa do *dataset* Georgia Tech.



Fonte: (GEORGIA TECH, 1999).

Figura 4. Imagem da face de uma pessoa do *dataset* Yale.



Fonte: (UCSD COMPUTER VISION, 1997).

Os algoritmos de reconhecimento facial avaliados foram executados em um computador com processador Intel Core I5 de 3ª geração, 8GB de memória RAM e SSD 240 GB.

3.1. Métodos

As análises comparativas dos algoritmos de reconhecimento facial realizadas neste trabalho seguiram algumas etapas:

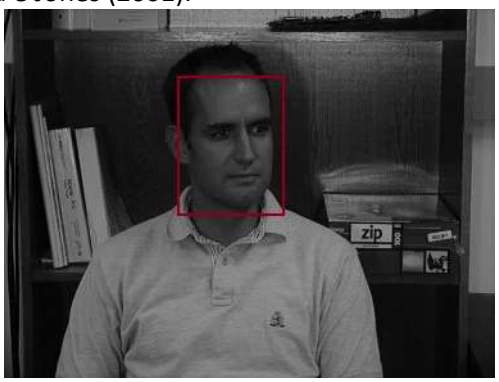
1. Levantamento de *datasets* de imagens com faces.
2. Aplicação de filtros nas imagens.
3. Extração das características das faces de treinamento.

4. Testes de reconhecimento com dados obtidos no treinamento.

Na primeira etapa, foram realizadas buscas de *datasets* públicos com imagens de pessoas. Para obter maior confiabilidade nos testes de reconhecimento, os critérios de escolha dos *datasets* foram: quantidade de imagens disponíveis; quantidade de pessoas diferentes; e variação das expressões faciais no conjunto de imagens de cada pessoa.

Na segunda etapa, as imagens foram convertidas para a escala de cinza, reduzindo a quantidade de dados da imagem e tornando o processo de detecção de faces mais rápido. Para detectar faces na imagem, foi utilizado o algoritmo de Viola e Jones (AdaBoost + Haar Cascade) (VIOLA; JONES, 2001) disponível na biblioteca OpenCV e o algoritmo HOG (*Histogram of Oriented Gradients*) (TRIGGS; DALAL, 2005) disponível na biblioteca Dlib. A Figura 5, apresenta um exemplo contendo uma imagem do *dataset* Georgia Tech após a conversão para escala de cinza com um retângulo vermelho indicando a região da face encontrada pelo algoritmo de detecção de faces de Viola e Jones.

Figura 5. Detecção de face utilizando o algoritmo de Viola e Jones (2001).



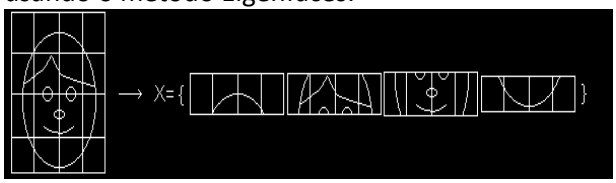
Fonte: Autor (2019).

Na terceira etapa, foram utilizadas apenas as áreas das imagens que contêm as faces das pessoas, previamente detectadas pelo algoritmo de detecção facial, para em seguida efetuar o treinamento. Nesse processo, foram extraídas as principais características das faces utilizando os algoritmos de reconhecimento facial Eigenfaces, Fisherfaces, LBPH (*Local Binary Patterns Histograms*) e CNN (*Convolutional Neural Networks*). As características da imagem, juntamente com os valores dos parâmetros utilizados e um identificador único de cada pessoa foram gravados em um arquivo de treinamento para ser utilizado posteriormente nos testes de reconhecimento.

Na quarta e última etapa, foram realizados os testes de reconhecimento facial nas imagens que foram previamente separadas para os testes. As imagens de teste foram convertidas para a escala de cinza e detectadas as faces. Em seguida, os testes de reconhecimento das faces foram realizados utilizando os algoritmos de reconhecimento, e os arquivos gerados na etapa de treinamento, validando se o identificador da pessoa predito pelo algoritmo é o mesmo da pessoa que está sendo submetida ao reconhecimento, caracterizando este processo como aprendizado de máquina supervisionado.

O método Eigenfaces (TURK; PENTLAND, 1991) necessita de várias imagens de uma mesma pessoa de diferentes ângulos e condições de iluminação para fazer o treinamento. Inicialmente, foi feita a redução da dimensionalidade de cada imagem da pessoa, extraindo as principais características e eliminando as partes que são desnecessárias para que o treinamento utilize menos recursos computacionais. Esse processo é feito pelo algoritmo PCA (*Principal Component Analysis*) (KAUR; HIMANSHI, 2015). Em seguida é feita uma projeção linear (Figura 6) formando um conjunto de imagens denominado de Eigenvector ou Eigenfaces que contêm as principais características de todas as imagens da pessoa, destacando as características que mais variam de uma imagem para outra, a fim de garantir uma maior fidelidade para o reconhecimento. A iluminação da imagem é um elemento importante, pois ela é considerada como uma característica de variação na face, fazendo com que esse método seja sensível à luminosidade. Finalmente, é feita uma combinação das imagens do conjunto Eigenfaces gerando uma imagem média que melhor representa a pessoa, sendo projetado no espaço de faces. Na fase de reconhecimento, a nova imagem é projetada no espaço de faces, extraindo as principais características e gerando o Eigenfaces. Posteriormente, é calculada a distância da nova imagem para cada uma das imagens do treinamento, e é classificada de acordo com a imagem que possui a menor distância. A Figura 6 ilustra uma projeção linear das principais características da face realizado pelo Eigenfaces.

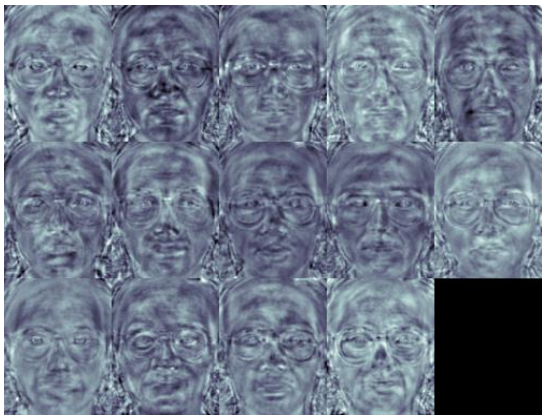
Figura 6. Representação da projeção linear dos componentes principais de uma imagem de face usando o método Eigenfaces.



Fonte: (KHAN; ALIZAI, 2006).

O método Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997) é considerado uma evolução do método Eigenfaces. A fase de treinamento também necessita de um número muito grande de imagens positivas (imagens com face) e negativas (imagens sem face), em que cada pessoa é considerada uma classe. Esse método utiliza o algoritmo de aprendizado LDA (*Linear Discriminant Analysis*) (CHELALI; DJERADI; DJERADI, 2009) para fazer a redução da dimensionalidade da imagem, extraindo as características mais relevantes. O algoritmo LDA faz a projeção linear das imagens formando um conjunto chamado Fisherfaces (Figura 7), porém com a proposta de maximizar a dispersão entre diferentes classes. Para isso, o método faz a extração das características separadamente, ou seja, em cada imagem do conjunto de uma classe. O reconhecimento de uma nova imagem é feito projetando-a no espaço de faces, em que são extraídas as principais características e gerado o Fisherfaces. Posteriormente é calculada a distância da nova imagem para as classes do treinamento, e então é classificada, de acordo com a imagem que possui a menor distância. A Figura 7 ilustra uma projeção linear das principais características da face realizado pelo Fisherfaces.

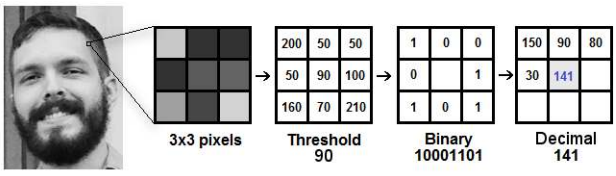
Figura 7. Exemplo da projeção linear das principais características da imagem usando Fisherfaces de uma pessoa do dataset Yale Facedatabase.



Fonte: (OPENCV, 2018).

LBPH (*Local Binary Patterns Histograms*) (STEKAS; HEUVEL, 2016) é um algoritmo usado para reconhecimento facial, que se baseia no operador binário local. Dada uma determinada matriz de pixels dentro da imagem, utiliza-se o valor do pixel central para fazer uma comparação com os pixels vizinhos, resultando em "1" para os pixels com valor maior e "0" para os pixels com valor menor (Figura 8). Assim, ao final de todas as comparações tem-se um número binário, este número é então convertido para decimal para posteriormente formar um histograma. No treinamento deste método, a imagem é dividida em blocos, e para cada bloco são feitas várias comparações em todos os pixels, gerando assim vários valores decimais como resultado. Com esses valores, é gerado um histograma para o bloco, que contém a quantidade de cada valor resultante das comparações. Por fim, é feita uma combinação do histograma de cada bloco da imagem gerando um histograma final, que será utilizado para representar essa imagem. O LBPH se caracteriza por ser bastante eficiente no que diz respeito à confiabilidade do reconhecimento de imagens. Isso porque ele consegue reconhecer faces sob condições de iluminação variadas, visto que, aumentar ou diminuir a iluminação da imagem faz com que os valores dos pixels se alterem proporcionalmente, gerando o mesmo resultado quando é feita as comparações dos pixels. A Figura 8 ilustra a comparação do pixel central com os pixels vizinhos na matriz de pixels de uma determinada região da face na imagem realizado pelo LBPH.

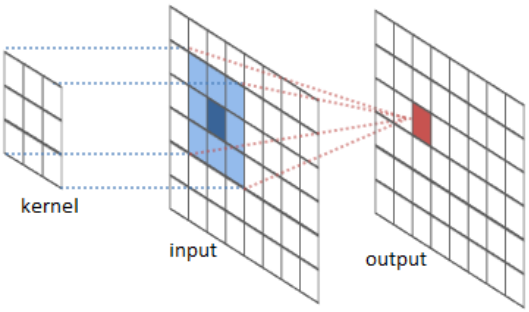
Figura 8. Comparação do pixel central com os pixels vizinhos na matriz dentro de um bloco da imagem.



Fonte: (PRADO, 2017b).

A CNN (*Convolutional Neural Networks*) (YAN et al., 2017) é uma técnica de aprendizado de máquina que busca simular o processo biológico do funcionamento do córtex visual dos animais através das várias camadas conectadas da rede. Nos últimos anos vem sendo amplamente utilizada em processamento de imagens digitais em virtude do alto nível de precisão obtido com a redução da

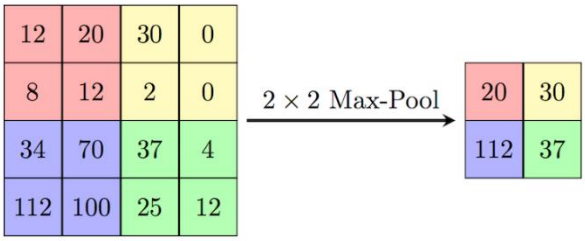
dimensionalidade da imagem, mas sem perder as características essenciais da imagem original. A CNN é composta pelas camadas de convolução e agrupamento para extração do mapa de características, seguidas pelas camadas totalmente conectadas e por último são gerados os classificadores. A camada de convolução utiliza núcleos de convolução (*kernels*) para construir o mapa de características da imagem. Nessa etapa de convolução, a matriz do *kernel* é aplicada sobre as regiões da imagem multiplicando os pesos pelos pixels da imagem (Figura 9). Ao fim desta etapa tem-se uma nova matriz com uma dimensionalidade menor chamada mapa de características. No entanto, mesmo após a aplicação da camada de convolução o mapa de características ainda está com uma dimensionalidade grande, para resolver este problema é aplicado uma camada de *pooling* que reduz ainda mais a dimensionalidade do mapa de características. Max-Pooling (NAGI et al., 2011) é o método mais utilizado neste processo, onde a matriz é dividida em regiões extraindo o maior valor de cada uma delas (Figura 10). As camadas totalmente conectadas contêm neurônios conectados com os neurônios das outras camadas, a camada de entrada recebe o volume de entrada do mapa de características obtido anteriormente e o transforma em um vetor *N* dimensional, sendo que o *N* é o número de classes gerada pela CNN.



Fonte: (COLAH´S BLOG, 2014).

Figura 9. Processo de convolução da CNN utilizando *kernel*. A matriz do *kernel* é aplicada sobre a imagem para reduzir sua dimensionalidade.

Figura 10. Redução da dimensionalidade na camada de *pooling* da CNN utilizando o método Max-Pooling, onde é escolhido o maior valor de cada região da matriz.



Fonte: (STACKEXCHANGE, 2018).

4. EXPERIMENTOS

Para treinamento do algoritmo Eigenfaces, os parâmetros utilizados foram *num_components* e *threshold*. Os valores apresentados na Tabela 1 são os que apresentaram os melhores resultados, definidos empiricamente a partir de várias combinações, utilizando os três *datasets* públicos de imagens com faces (Caltech, Georgia Tech e Yale).

Tabela 1. Parâmetros definidos empiricamente utilizados no algoritmo Eigenfaces com os resultados.

Dataset	Num components	threshold	%Acerto	Segundos
Caltech	388	10000	67,31	18,88
GeorgiaTech	600	13000	70,00	19,56
Yale	45	20000	83,33	2,71

Fonte: Autor (2019).

O algoritmo Eigenface obteve 83,33% de acerto com o *dataset* Yale, 70,00% com Georgia Tech, e 67,31% com Caltech, com tempos de execução de 2,71, 19,56 e 18,88 segundos respectivamente.

Para treinamento do algoritmo Fisherfaces os parâmetros utilizados são os

mesmos do anterior Eigenfaces *num_components* e *threshold*, porém com outros valores, também definidos empiricamente. Os valores apresentados na Tabela 2 são os que apresentaram os melhores resultados.

Tabela 2. Parâmetros utilizados no algoritmo Fisherfaces, definidos empiricamente, com os melhores resultados.

Dataset	Num components	threshold	%Acerto	Segundos
Caltech	25	2000	80,77	1,82
GeorgiaTech	30	5000	56,00	2,37
Yale	9	12000	96,67	1,44

Fonte: Autor (2019).

O Fisherface obteve 96,67% de acerto com o *dataset* Yale, 80,77% com Caltech, e 56,00% com Georgia Tech, com tempos de execução de 1,44, 1,82 e 2,37 segundos respectivamente.

O algoritmo LBPH (*Local Binary Patterns Histograms*) utiliza os parâmetros *radius, neighbors, grid_x, grid_y* e *threshold* para realizar os treinamentos. Os melhores valores encontrados, também de forma empírica, estão na Tabela 3.

Tabela 3. Parâmetros utilizados no algoritmo LBPH.

Dataset	Radius	Neighbors	Grid_X	Grid_Y	Threshold	% Acertos	Segundos
Caltech	3	4	8	8	80	88,46	1,51
GeorgiaTech	1	5	3	3	50	85,00	1,88
Yale	4	8	6	6	40	73,33	1,69

Fonte: Autor (2019).

O LBPH obteve 88,46% de acerto com o *dataset* Caltech, 85,00% com Georgia Tech, e 73,33% com Yale, com tempos de execução de 1,51, 1,88 e 1,69 segundos respectivamente.

Para treinamento da rede neural CNN (*Convolutional Neural Networks*), foram utilizados dois arquivos para auxiliar no processo de detecção e reconhecimento facial. O primeiro deles é o *shape_predictor_68_face_landmarks.dat*, este arquivo está disponível na biblioteca do Dlib e possui um descritor de pontos faciais (KAZEMI; SULLIVAN, 2014) que identifica as principais regiões do rosto humano como a sobrancelhas, olhos, nariz, boca e a mandíbula. Para obter os descritores das faces, foi utilizado o segundo arquivo *dlib_face_recognition_resnet_model_v1*.dat também disponível na biblioteca do Dlib, que possui uma rede ResNet com 29 camadas de convolução. É essencialmente uma versão da rede ResNet-34 (HE et al., 2016) com algumas camadas removidas e o número de filtros por camada reduzido pela metade.

Os melhores valores encontrados com o uso da rede neural CNN estão na Tabela 4.

Tabela 4. Resultados obtidos com o algoritmo CNN.

Dataset	%Acerto	Segundos
Caltech	100,00	10,52
GeorgiaTech	100,00	16,19
Yale	93,33	2,48

Fonte: Autor (2019).

Os resultados obtidos com a rede neural CNN foram de 100% de acerto com os *datasets*

Caltech e Georgia Tech com tempo de execução de 10,52 e 16,19 segundos respectivamente, e 93,33% de acerto e tempo de 2,48 segundos com o *dataset* Yale.

As Tabelas 5, 6 e 7 apresentam o desempenho (porcentagens no reconhecimento e tempos de execução) dos algoritmos de reconhecimento facial utilizados em cada *dataset* ordenados pela porcentagem de acerto.

Tabela 5. Desempenho dos algoritmos no *dataset* Caltech.

Caltech		
Algoritmo	**% Acerto**	**Tempo (segundos)**
CNN	100,00	10,52
LBPH	88,46	1,51
Fisherfaces	80,77	1,82
Eigenfaces	67,30	18,88

Fonte: Autor (2019).

Tabela 6. Desempenho dos algoritmos no *dataset* Georgia Tech.

Georgia Tech		
Algoritmo	**% Acerto**	**Tempo (segundos)**
CNN	100,00	16,19
LBPH	85,00	1,88
Eigenfaces	70,00	19,56
Fisherfaces	56,00	2,37

Fonte: Autor (2019).

Tabela 7. Desempenho dos algoritmos no *dataset* Yale.

Yale		
Algoritmo	**% Acerto**	**Tempo (segundos)**
Fisherfaces	96,67	1,44
CNN	93,33	2,48
Eigenfaces	83,33	2,71
LBPH	73,33	1,69

Fonte: Autor (2019).

5. CONSIDERAÇÕES FINAIS

Embora a rede neural CNN tenha apresentado mais acertos que os demais métodos, ainda existem outros fatores que devem ser levados em consideração, como grau de confiabilidade e tempo de execução, para escolha de qual algoritmo será utilizado.

Na área de visão computacional, já foram feitas várias pesquisas para desenvolvimento de algoritmos para detecção e reconhecimento de faces. Este trabalho contribui com a análise de eficiência de alguns destes métodos, acrescentando que a escolha entre eles será a que melhor atende os requisitos necessários.

REFERÊNCIAS

BALAN, J. As vantagens e os perigos do reconhecimento facial. **Meio&Mensagem**, 2019. Disponível em: https://www.meioemensagem.com.br/home/midia/2019/07/30/reconhecimento-facial-quando-vale-dar-a-cara-a-tapa.html. Acesso em: 17 nov. 2019.

BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 19, n. 7, p. 711-720, jul. 1997. https://doi.org/10.1109/34.598228

CHELALI, F. Z.; DJERADI, A.; DJERADI, R. Linear discriminant analysis for face recognition. *In*: INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS, Ouarzazate, Morocco, p. 1-10, 2009. https://doi.org/10.1109/MMCS.2009.5256630

COLAH´S BLOG, C. Understanding Convolutions. 2014. Disponível em: http://colah.github.io/posts/2014-07-Understanding-Convolutions. Acesso em: 31 dez. 2019.

CORREIA, T. A.; PITERI, M. A.; ARTERO, A. O.; SILVA, F. A.; PEREIRA, D. R. Development of an application for security based in face recognition on Android platform. *In*: WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2014), 10., 2014, Uberlândia, MG. **Proceedings [...].**.2014. p. 137-142,

DLIB. Dlib Files. Dlib. Disponível em: http://dlib.net/files/. Acesso em: 20 mai. 2019.

ÉPOCA NEGÓCIOS, E. **Boticário vai ter ferramentas de reconhecimento facial.** 2019. Disponível em: https://epocanegocios.globo.com/Tecnologia/noticia/2019/11/epoca-negocios-artur-grynbaum-vamos-ter-ferramentas-de-reconhecimento-facial.html. Acesso em: 20 nov. 2019.

GEORGIA TECH, Georgia Tech face database. Georgia Tech, 1999. Disponível em: http://www.anefian.com/research/face_reco.htm. Acesso em: 11 fev. 2019.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep Residual Learning for Image Recognition. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), Las Vegas, NV, USA, p. 770 - 778, 2016. https://doi.org/10.1109/CVPR.2016.90

HIRAKURI, M. H. **Aplicação da Rede Neural Neocognitron para Reconhecimento de Atributos Faciais**. 2003. 100 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de São Carlos. São Carlos, 2003.

KAUR, R.; HIMANSHI, E. Face recognition using Principal Component Analysis. In: IEEE INTERNATIONAL ADVANCE COMPUTING CONFERENCE (IACC), Banglore India, 2015. p. 585-589https://doi.org/10.1109/IADCC.2015.7154774

KAZEMI, V.; SULLIVAN, J. One millisecond face alignment with an ensemble of regression trees. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, Columbus, OH, USA, 2014. p. 1867-1874 https://doi.org/10.1109/CVPR.2014.241

KHAN, A. S.; ALIZAI, L. K. Introduction to Face Detection Using Eigenfaces. In: INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES, Peshawar, Pakistan, 2006. p. 128-132. https://doi.org/10.1109/ICET.2006.335908

WEBER, M. Caltech face database. Caltech, 1999. Disponível em: http://www.vision.caltech.edu/html-files/archive.html. Acesso em: 11 fev. 2019.

MIRANDA, J. **Brasileiros preferem biometria a senhas.** Reconhecimento Facial, 2017. Disponível em: http://reconhecimentofacial.com.br/2018/02/14/brasileiros-preferem-biometria-a-senhas/. Acesso em: 04 set. 2018.

NAGI, J; DUCATELLE, F.; DI CARO, G. A.; CIREŞAN, D.; MEIER, U.; GIUSTI, A.; NAGI, F.; SCHMIDHUBER, J.; GAMBARDELLA, L. M. Max-Pooling Convolutional Neural Networks for Vision-Based Hand Gesture Recognition. *In*: IEEE INTERNATIONAL CONFERENCE ON SIGNAL AND IMAGE PROCESSING APPLICATIONS (ICSIPA), 2011, Kuala Lumpur, Malaysia, p. 342 - 347, 2011. https://doi.org/10.1109/ICSIPA.2011.6144164

OPENCV. Face Recognition with OpenCV. OpenCV. Disponível em: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#fisherfaces-in-opencv. Acesso em: 12 set. 2018.

PETRI, R. A. L.; PITERI, M. A.; ARTERO, A. O.; SILVA, F. A. Digital Image Face Detection. *In*: WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2013), 9., Rio de Janeiro, RJ. **Proceedings [...].**, 2013.

PORTAL, A. **Detran implanta reconhecimento facial na habilitação de condutores**. Portal no Ar, 2019. Disponível em: https://portalnoar.com.br/detran-implanta-reconhecimento-facial-na-habilitacao-de-condutores. Acesso em: 20 nov. 2019.

PRADO, E. O interesse pelo reconhecimento facial da IA. Convergência Digital, 2017a. Disponível em:

http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&UserActiveTemplate=mobile,site&infoid=46223&sid=15. Acesso em: 04 set. 2018.

PRADO, K. Face Recognition: Understanding LBPH Algorithm. Towards Data Science, 2017b. Disponível em: https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b. Acesso em: 12 set. 2018.

STACKEXCHANGE, S. Implement the Max-Pooling operation from Convolutional Neural Networks. 2018. Disponível em: https://codegolf.stackexchange.com/questions/195348/implement-the-max-pooling-operation-from-convolutional-neural-networks. Acesso em: 31 dez. 2019.

STEKAS, N.; HEUVEL, D. Face recognition using Local Binary Patterns Histograms (LBPH) on an FPGA-based System on Chip (SoC). In: 2016 IEEE INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM WORKSHOPS (IPDPSW), 2016. **Proceedings**... Chicago: IEEE, p. 300-304, 2016. https://doi.org/10.1109/IPDPSW.2016.67

TRIGGS, B.; DALAL, N. Histograms of Oriented Gradients for Human Detection. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR'05). **Proceedings [...]**, , 2005. p. 886-893

TURK, M.A.; PENTLAND, A.P. Face Recognition Using Eigenfaces. *In*: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1991. **Proceedings [**...], Maui: IEEE, p. 586-591, 1991.

UCSD COMPUTER VISION, Yale Face Database. UCSD Computer Vision, 1997. Disponível em: http://vision.ucsd.edu/content/yale-face-database. Acesso em: 11 fev. 2019.

VIOLA, P.; JONES, M. Rapid Object Detection using a Boosted Cascade of Simple Features. *In*: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2001. **Proceedings**... Kauai: IEEE, p. 511-518, 2001.

YAN, K.; HUANG, S.; SONG, Y.; LIU, W.; FAN, N. Face Recognition Based On Convolution Neural Network. *In*: CHINESE CONTROL CONFERENCE (CCC), 36th Dalian, China, , 2017. p. 4077 - 4081https://doi.org/10.23919/ChiCC.2017.8027997.