

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

LUCAS RICARDO ORSO

**ESTUDO DE CASO DE UM SISTEMA DE RECONHECIMENTO FACIAL
UTILIZANDO O *OPENFACE* PARA IDENTIFICAÇÃO DE FACES EM BANCOS DE
IMAGENS**

**CRICIÚMA
2018**

LUCAS RICARDO ORSO

**ESTUDO DE CASO DE UM SISTEMA DE RECONHECIMENTO FACIAL
UTILIZANDO O *OPENFACE* PARA IDENTIFICAÇÃO DE FACES EM BANCOS DE
IMAGENS**

Trabalho de Conclusão de Curso,
apresentado para obtenção do grau de
Bacharel no Curso de Ciência da
Computação da Universidade do Extremo
Sul Catarinense, UNESC.

Orientador: Prof. Me. Paulo João Martins.

CRICIÚMA

2018

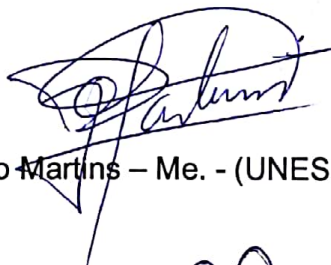
LUCAS RICARDO ORSO

**ESTUDO DE CASO DE UM SISTEMA DE RECONHECIMENTO FACIAL
UTILIZANDO O OPENFACE PARA IDENTIFICAÇÃO DE FACES EM BANCOS DE
IMAGENS**

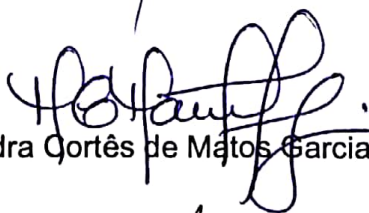
Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Segurança da Informação.

Criciúma, 27 de novembro de 2018.

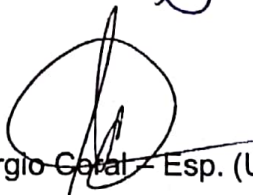
BANCA EXAMINADORA



Prof. Paulo João Martins – Me. - (UNESC) - Orientador



Profª. Merisandra Cortês de Matos Garcia – Dra. - (UNESC)



Prof. Sergio Cerat – Esp. (UNESC)

Dedico este trabalho ao meus pais, Maurício (in memoriam) e Neiva, por instituir todos os valores que achavam importantes e por me tornarem na pessoa que sou hoje.

AGRADECIMENTOS

A minha mãe, por toda a sua ajuda e precaução, e seu incentivo em momentos difíceis.

A minha namorada Indiani, pela sua profunda paciência em me aturar, e ainda por toda sua ajuda durante minha jornada acadêmica.

Ao meu orientador Paulo, pelas horas dedicadas, pelo compartilhamento de seus conhecimentos, e pelo empenho dedicado à elaboração deste trabalho.

A todos os amigos que fiz durante a graduação, que me acompanharam ao longo de minha jornada acadêmica.

A todos os professores do departamento Ciência da Computação da UNESC que fizeram parte de jornada acadêmica, a qual proporcionou meu crescimento científico, profissional e pessoal.

**“Qualquer tecnologia suficientemente
avançada é equivalente à mágica”**

Arthur C. Clarke

RESUMO

Há na atualidade um aumento na utilização de computadores, para fins profissionais, pessoais, informações gerais e restritas. Dessa forma, há uma necessidade na segurança em geral, requerendo um meio de autenticação prática, rápida e segura. Devido a isso, uma alternativa para a segurança é o reconhecimento facial, que vem ganhando atenção em diferentes áreas, por ser menos invasiva e prover as informações biométricas, apresentando características únicas de cada indivíduo. Não obstante, existem várias bibliotecas de reconhecimento facial disponíveis, dentre elas o *OpenFace*, biblioteca gratuita que oferece uma rede neural artificial treinada com mais de 500 mil faces, que contribui com aqueles que têm interesse na área de reconhecimento facial. Considerando a utilização do reconhecimento facial, este trabalho tem como objetivo avaliar o classificador *Support Vector Machines* utilizado pela biblioteca *OpenFace* na classificação das imagens através do método *Cross-Validation*. Inicialmente, foi realizada uma pesquisa bibliográfica acerca dos métodos, padrões e detecção do reconhecimento facial existente na literatura. Para replicar os resultados do *OpenFace* utilizou-se o banco de imagens *Labeled Faces in the Wild*. Foi utilizado duas bases de imagens, o MS-CELEB-1M e o FEI DATABASE, onde com a RNA do *OpenFace* foram extraídas as características de cada imagem, para posteriormente criar um classificador *Support Vector Machines*. Depois disso, o classificador foi avaliado pelo método de validação cruzada. Os resultados da acurácia do *OpenFace* na identificação de faces, quando utilizado a base de dados LFW coincidiram com os resultados existentes sobre o *OpenFace*. Os resultados na validação cruzada do SVM demonstraram acurácia similar entre as bases MS-CELEB-1M e FEI DATABASE, com 0.99% e 0.89%, respectivamente. Em suma, pode-se concluir que os resultados desta pesquisa demonstraram-se satisfatórios quanto a utilização da biblioteca *OpenFace*, o que reforça os resultados existentes na literatura de que a utilização da inteligência artificial para os problemas encontrados na área de reconhecimento facial é uma opção a ser estudada e considerada a sua utilização.

Palavras-chave: Biometria. Reconhecimento facial. *OpenFace*. Inteligência Artificial.

ABSTRACT

There is now an increase in the use of computers, for professional, personal use, general and restricted use. In this way, there is a need for security in general, requiring a practical, fast and secure means of authentication. Because of this, an alternative to security is facial recognition, which has been gaining attention in different areas, being less invasive and providing biometric information, presenting unique characteristics of each individual. Nevertheless, there are several facial recognition libraries available, among them OpenFace, a free library that offers a trained artificial neural network with more than 500 thousand faces, which contributes to those who are interested in facial recognition. Considering the use of facial recognition, this work aims to evaluate the Support Vector Machines classifier used by the OpenFace library in the classification of images through the Cross-Validation method. Initially, a bibliographical research was done on the methods, patterns and detection of facial recognition in the literature. To replicate the results of OpenFace was used the image bank Labeled Faces in the Wild. Two image bases, the MS-CELEB-1M and the FEI DATABASE, were used, where the characteristics of each image were extracted with the OpenFace RNA to create a Support Vector Machines classifier. After that, the classifier was evaluated by the cross-validation method. The results of OpenFace's accuracy in identifying faces when using the LFW database coincided with the existing OpenFace results. The cross-validation results of the SVM demonstrated similar accuracy between the MS-CELEB-1M and FEI DATABASE bases, with 0.99% and 0.89%, respectively. In summary, it can be concluded that the results of this research have been satisfactory in terms of the use of the OpenFace library, which reinforces the results in the literature that the use of artificial intelligence for use in the problems found in facial recognition is a option to be studied and considered for application.

Keywords: Biometrics. Facial recognition. OpenFace. Artificial intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de um Sistema de Reconhecimento de Faces (SRF).	20
Figura 2 - Fluxograma de SRF baseado no Eigenfaces.	22
Figura 3 - Estrutura do projeto do <i>OpenFace</i>	26
Figura 4 - Bibliotecas utilizadas no <i>OpenFace</i>	27
Figura 5 - Linguagem de programação Python e Lua, respectivamente.	28
Figura 6 - Diferença entre máquina virtual e containers.	29
Figura 7 - Exemplo hiperplano separador ótimo.	36
Figura 8 - Exemplo <i>StratifoldKfold</i>	37
Figura 9 - Demonstração de Matriz de confusão.	38
Figura 10 - Modelo de um neurônio artificial.	40
Figura 11 - Fluxograma experimental.	44
Figura 12 - Estrutura de diretórios em subdiretórios diferentes.	46
Figura 13 - Algoritmo para criar subpastas na FEI DATABASE.	47
Figura 14 - Exemplo de pré-processamento.	48
Figura 15 - Comando usado para gerar as representações.	49
Figura 16 - Demonstração do código fonte.	50
Figura 17 - Comando para executar o algoritmo.	50
Figura 18 - Reprodução do experimento.	51
Figura 19 - Acurácia do <i>OpenFaces</i> no teste LFW.	52
Figura 20 - Resultado t-SNE MS-CELEB-1M.	57
Figura 21 - Resultado t-SNE FEI DATABASE.	58
Figura 22 - Amostra da matriz de confusão para cada classe.	59
Figura 23 - Características semelhantes entre as classes.	59

LISTA DE TABELAS

Tabela 1 - Diferenças entre máquina virtual e <i>Docker</i>	30
Tabela 2 - Performance da rede neural.	49
Tabela 3 - Pós processamento dos dados.....	53
Tabela 4 - Matriz de representações.....	54
Tabela 5 - Resultados da validação cruzada.	54

LISTA DE ABREVIATURAS E SIGLAS

ACP	Análise de Componentes Principais
AD	Árvore de decisão
AI	<i>Artificial Intelligence</i>
AM	Aprendizado de máquina
CNN	<i>Convolutional Neural Network</i>
DNN	<i>Deep Neural Network</i>
HOG	<i>Histogram of Oriented Gradients</i>
ICA	<i>Independent Component Analysis</i>
IJCTA	<i>International Journal of Computer Technology and Applications</i>
LFW	<i>Labeled Faces in the Wild</i>
MLP	<i>Multi-Layer Perceptron</i>
NFL	<i>National Football League</i>
OVR	<i>One Vs. Rest</i>
RL	Regressão Logística
SRF	Sistema de Reconhecimento Facial
SVM	<i>Support Vector Machines</i>

SUMÁRIO

1 INTRODUÇÃO	13
1.1 OBJETIVO GERAL	15
1.2 OBJETIVOS ESPECÍFICOS	15
1.3 JUSTIFICATIVA	15
1.4 ESTRUTURA DO TRABALHO	17
2 REFERENCIAL TEÓRICO	18
2.1 RECONHECIMENTO FACIAL	18
2.1.1 Métodos de reconhecimento facial	21
2.1.1.1 Métodos holísticos.....	22
2.1.2.1 Métodos baseados em características.....	23
2.1.3.1 Métodos híbridos.....	23
2.2 OPENFACE.....	24
2.2.1 DOCKER	29
2.3 INTELIGÊNCIA ARTIFICIAL	30
2.3.1 Aprendizado de máquina	31
2.3.1.1 <i>Scikit-learn</i>	34
2.3.1.2 Máquinas de Vetores de Suporte.....	35
2.3.1.3 Validação cruzada.....	36
2.3.1.4 Matriz de confusão	37
2.3.2 Redes Neurais Artificiais	39
3 TRABALHOS CORRELATOS	41
3.1 UMA COMPARAÇÃO DOS ALGORITMOS DO RECONHECIMENTO FACIAL	41
3.2 SISTEMA DE RECONHECIMENTO FACIAL.....	42
3.3 DETECÇÃO E RECONHECIMENTO FACIAL POR MEIO DE APRENDIZADO DE MÁQUINA.....	42
3.4 COMPARAÇÃO DE REDES NEURAIIS DE RECONHECIMENTO FACIAL	43
4 ACURÁCIA DO SVM NO RECONHECIMENTO DE FACES EM BASES DE IMAGES	44
5 RESULTADOS E DISCUSSÃO	51
5.1 REPRODUÇÃO DO EXPERIMENTO DE ACURÁCIA NA BASE LFW	51
5.2 VALIDAÇÃO CRUZADA COM DUAS BASES DE IMAGENS	53

5.3 TESTES ADICIONAS.....	56
6 CONCLUSÃO	60
REFERÊNCIAS.....	61
APÊNDICE (S)	67
APÊNDICE A – ARTIGO	68

1 INTRODUÇÃO

Estudos relacionados à sistemas de reconhecimento facial estão em evidência atualmente, devido a possibilidade desses sistemas em reconhecer rostos a partir de vídeos ou imagens, tornando-se cada vez mais uma realidade a ser utilizada. Ainda, a visão computacional proporciona a construção de um sistema de visão artificial que se assemelha a capacidade do sistema visual humano, no qual, nos últimos anos houve significativos avanços, em razão do desenvolvimento de equipamentos computacionais mais rápidos (LOPES, 2016).

A área da visão computacional engloba a análise de imagens, e tem como princípio diferenciar os objetos de interesse, sem a intervenção humana (PEDRINI; SCHWARTZ, 2007 apud RODRIGUES, 2015). Visto isso, a visão computacional surgiu para facilitar a resolução de problemas mais complexos e busca imitar a cognição e habilidade do ser humano em tomar decisões de acordo com informações contidas na imagem (RODRIGUES, 2015).

Quando se discute sobre visão computacional, normalmente são estipulados dois níveis de abstração: processamento de imagens (baixo nível) e análise de imagens (alto nível) (GONZALEZ; WOODS, 2006; MARQUES FILHO; VIEIRA NETO, 1999). “O processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio de computador” (LOPES; MARIO; SILVA FILHO, 2012, p.29). Isso envolve segmentação de imagem, descrição, representação e reconhecimento de objetos (GONZALEZ; WOODS, 2006).

A área de reconhecimento facial está presente em nosso dia a dia, como por exemplo, em sistemas de segurança e entretenimento (SORTE, 2011; VINODHINI et al., 2016). Segundo Bombardelli (2015), devido a popularização das redes sociais originou-se uma grande quantidade de dados, simultaneamente com o aumento da capacidade de processamento de computadores e com novas técnicas no processamento de imagens e no aprendizado de máquina, e que influenciaram na precisão dos métodos de reconhecimentos facial.

O método de reconhecimento facial é realizado por meio de imagens e filmagens de câmera de segurança (PRAZERES, 2010; TRAMONTIN, 2016). O

mesmo, não é considerado um dos mais precisos no quesito de autenticação, devido as variações de poses, expressões faciais e iluminação. Ainda assim, é um método que possui suas vantagens, pois por não ser um método intrusivo, não é necessário que o usuário interaja ou toque em algum dispositivo por um período, propicia a identificação do indivíduo sem a autorização do mesmo (MENEZES, 2009; PRAZERES 2010).

Porém, as câmeras estão se tornando presentes em todos os lugares, em relação a Internet das coisas, e podem usar tecnologias de reconhecimento facial para melhorar o contexto de reconhecimento facial. Entretanto, há uma grande lacuna de precisão entre os sistemas de reconhecimento facial disponíveis ao público e ao privado, de última geração (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

Há poucos projetos comerciais que utilizam esse método, pelo fato de ainda ter pouco conhecimento sobre reconhecimento facial e por ser uma ferramenta de alto valor, limitando o uso à um pequeno grupo de empresas. Muitos projetos buscam um aprimoramento e uma aplicação precisa de reconhecimento facial, para reforçar a segurança de um determinado ambiente (NUNES et al, 2016).

Na área de segurança vêm sendo muito utilizado o uso da biometria, favorecendo empresas e sistemas públicos que são restritos quanto ao acesso de pessoas, bem como o uso em investigações criminais e policiais, tornando-se assim uma contribuição válida nessa área (NUNES et al, 2016).

Segundo Amos, Ludwiczuk e Satyanarayanan (2016), atualmente a maioria das técnicas de reconhecimento facial utilizam inteligência artificial, a qual tem o intuito de aumentar a precisão de um sistema de reconhecimento facial. Uma dessas técnicas é o Máquina de Vetores de Suporte (do inglês *Support Vector Machines* – SVM), que são conjunto de métodos de aprendizado de máquina, que são usados para classificação (PEDREGOSA et al., 2011, tradução nossa).

Toda via, a métrica acurácia é amplamente utilizada para avaliar a eficácia de um sistema de reconhecimento facial. Por isso, busca-se um maior conhecimento sobre qual o melhor método e eficiência de reconhecimento facial.

1.1 OBJETIVO GERAL

Avaliar o SVM utilizado pelo *OpenFace* na classificação das imagens através do método *Cross-Validation*.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

- a) descrever os diversos métodos de detecção e reconhecimento de faces existentes;
- b) verificar os conceitos sobre a análise de comportamento facial;
- c) elencar os padrões faciais que servem para identificar rostos;
- d) utilizar no mínimo duas bases de imagens na classificação do SVM.
- e) avaliar o SVM utilizando a validação cruzada com o método *K-Fold* para verificar a acurácia.

1.3 JUSTIFICATIVA

Nos últimos anos a área científica que estuda o reconhecimento de padrões, tem gerado o desenvolvimento e aprimoramento de técnicas com diferentes finalidades e aplicações em diversas áreas como na Engenharia, Medicina, Astronomia e Segurança (THEODORIDIS; KOUTROUMBAS, 2003, tradução nossa).

A área de reconhecimento facial é considerada promissora abrangendo várias disciplinas, como o processamento de imagem, visão computacional, reconhecimento de padrões, computação gráfica e inteligência artificial. Desde o início dessa área, deu-se crescentes pesquisas e com isso surgiram diversas técnicas e abordagens para a resolução do problema do reconhecimento de faces (CHELLAPPA; WILSON; SIROHEY, 1995).

Houve um crescimento e interesse na área de reconhecimento de padrões devido as aplicações que, além de serem desafiantes, exigem cada vez mais poder computacional (BIANCHI, 2006).

O reconhecimento facial vem emergindo como uma área efetiva em pesquisas no reconhecimento de padrões. Esta tecnologia está presente em diversas aplicações tanto comerciais quanto legais. Estas aplicações vão desde comparação de imagens estáticas encontradas em passaporte, carteira de habilitação, carteira de identidade e foto de arquivo policial, como à comparação de imagens em tempo real em câmeras de vigilância (BIANCHI, 2006).

Na área de sistemas de segurança, a identificação de criminosos usando reconhecimento facial caracteriza-se como uma importante utilização. Isso se justifica pela crescente sensação de insegurança, um dos principais fatores enfrentados pela sociedade atual, trazida pela cultura do medo. O aumento da violência, representado por crimes patrimoniais diversos, homicídios, dentre outros, constitui motivação significativa para inserção de câmeras de vigilância, não apenas em ambientes privados, mas também em espaços de circulação pública (SILVA NETO, 2016). Bem como autenticação para sistemas biométricos em passaportes e fichas criminais, sistemas de segurança e controles de acesso, monitoramento de multidões em estações, shoppings, criação de retratos falados, exploração, comparação de técnicas para estes tipos de aplicação.

Aplicações em reconhecimento facial simultaneamente com sistemas de vídeo monitoramento já existem, pode ser um instrumento eficaz no combate à criminalidade, no que se refere a identificação e localização de criminosos (NUNES et al., 2016).

Uma alternativa para a segurança é a utilização de técnicas de reconhecimento facial, por esta ser pouco invasiva e prover as informações biométricas apresentando as características únicas de cada indivíduo. Com isso, setores da sociedade debatem soluções para reduzir os danos sofridos, com o aumento da criminalidade. Por parte das empresas e pessoas, idealiza-se investimentos em tecnologia na área de segurança, como por exemplo: alarmes, equipamentos de rastreamento, sistemas de monitoramento e assim por diante (NUNES et al., 2016).

Para a exploração do reconhecimento facial móvel e transitório existe a biblioteca *OpenFace*, onde os desenvolvedores oferecem uma rede neural treinada com mais de 500 mil faces (AMOS, LUDWICZUK; SATYANARAYANAN, 2016,

tradução nossa). Ainda, por se tratar de uma biblioteca disponível para todos os interessados em estudá-la, pode-se estudar a sua implementação e verificar o seu funcionamento.

Para isso, vale ressaltar que a acurácia de um sistema está relacionada à sua capacidade de reconhecer corretamente os indivíduos pela sua face.

Contudo, o reconhecimento facial pode ser uma contribuição válida na área de segurança da informação. Por isso, busca-se um maior conhecimento sobre técnicas de reconhecimento facial.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é dividido em seis capítulos, que tratam os conceitos de reconhecimento facial com o uso de redes neurais.

O primeiro capítulo é elaborado uma introdução, identificando os principais motivos de se usar o reconhecimento facial para identificação de indivíduos no contexto de segurança, constando ainda dos objetivos desta pesquisa assim como a sua justificativa.

No segundo capítulo é feito um levantamento bibliográfico acerca do reconhecimento facial, bem como as técnicas utilizadas para realizar tal função dando ênfase no uso da inteligência artificial para realizar tal tarefa.

Já no terceiro são apresentados os trabalhos correlatos, que abordam pesquisas relativas a modalidade de reconhecimento facial e redes neurais.

A metodologia de como foi realizado esta pesquisa encontra-se no quarto capítulo, no qual demonstram as etapas e processos realizados.

O quinto capítulo é onde encontra-se os resultados que foram obtidos durante a pesquisa, bem como uma discussão dos mesmos.

E por fim, no sexto capítulo são apresentadas as conclusões relativas ao que se observou e feito um encerramento dos assuntos abordados nesta pesquisa. Concluindo assim, com sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

As inúmeras aplicações práticas do reconhecimento facial e seus desafios o tornaram uma área de pesquisa popular em visão computacional. Isto é evidenciado por muitos algoritmos de reconhecimento de face desenvolvidos ao longo das últimas três décadas e, nos últimos anos, o aparecimento de um número considerável de sistemas de reconhecimento de face comerciais (SILVA NETO, 2016).

Este capítulo apresenta uma revisão de literatura sobre o reconhecimento facial com destaque para a contextualização e estado da arte, assim como algumas técnicas a serem utilizadas nesta monografia.

2.1 RECONHECIMENTO FACIAL

A concepção de identificação ou reconhecimento facial foi apresentada em 1960 (NUNES et al., 2016). Entre os anos de 1964 e 1965, os pesquisadores Woodrow Wilson Bledsoe, Helen Chan Wolf e Charles Bisson desenvolveram o primeiro sistema semiautomático de reconhecimento facial humano por meio da utilização do computador (GALVIS TRANLAVIÑA, 2015 apud NUNES et al., 2016).

Nos anos 70, Goldstein, Harmon e Lesk (1971), utilizaram-se da identificação de 21 características físicas específicas, como espessura dos lábios e a cor do cabelo para aprimorar o reconhecimento facial, porém até então a identificação destas características permanecia de forma manual.

Em 1987, Sirovich e Kirby aplicaram o princípio da Análise de Componentes Principais (ACP) ao problema do reconhecimento da face. Isto mostrou que eram necessários menos de uma centena de valores para codificar com exatidão uma imagem da face adequadamente normalizada e alinhada (NSTC, 2006, tradução nossa).

Já no ano de 1991, Turk e Pentland descobriram que com a utilização do *Eigenfaces*, o erro residual podia ser utilizado para detectar faces em imagens. Esta descoberta permitiu a criação de sistemas de reconhecimento facial em tempo real (NSTC, 2006, tradução nossa).

Contudo, foi no ano de 2001 que a tecnologia de reconhecimento facial ganhou atenção do público e da mídia, durante uma partida de *Super Bowl* da *National Football League* (NFL), em que as câmeras de vigilância foram utilizadas para capturar imagens, os quais foram comparadas com um banco de dados de imagens (MONTIBELER; FERNANDES, 2012).

Posteriormente a demonstração, deu-se início a uma importante análise de como utilizar a tecnologia para corresponder as necessidades nacionais do país, mas de modo a considerar as preocupações sociais e de privacidade pública. Na atualidade, a tecnologia de reconhecimento facial está sendo usada para combater fraudes de passaportes, controle de acesso à ambientes restritos, identificar pessoas desaparecidas e minimizar a fraude de benefícios (MONTIBELER; FERNANDES, 2012; NSTC, 2006, tradução nossa).

Foram realizados estudos mais detalhados, nos passar dos últimos anos, isso devido à necessidade de encontrar meios de reconhecer indivíduos através de suas características (VALLEJO; NEIRA, 2005).

Neste sentido, o reconhecimento facial se dá pela possibilidade de identificar pessoas por meio de suas características. O *Eigenfaces* mapeia as características da face de uma pessoa em um espaço multidimensional, onde assim os computadores podem realizar buscas em bases de dados e verificar em tempo real um para um ou um para muitos, em um processamento de uma fração de segundos (TURK; PENTALAND, 1991, tradução nossa).

Para se reconhecer uma face, deve-se primeiro detectá-la, portanto, a primeira etapa de um sistema de reconhecimento facial é a detecção de uma face em uma imagem de entrada. Por isso a detecção de imagens está diretamente relacionada ao reconhecimento facial, e em seguida é realizada a comparação com os modelos conhecidos pelo sistema (SUNG et al., 1994 apud OLIVEIRA, 2006).

A detecção da face é um fator importante, pois é a primeira etapa necessária para qualquer sistema de reconhecimento facial. A maioria das técnicas utilizadas admite uma detecção ideal, conseqüentemente a capacidade do sistema depende da precisão da etapa de detecção (OMAIA, 2009). O ser humano pode reconhecer milhares de faces no decorrer de sua vida. Em relação aos familiares, mesmo após anos de afastamento o ser humano é capaz de reconhecer seus

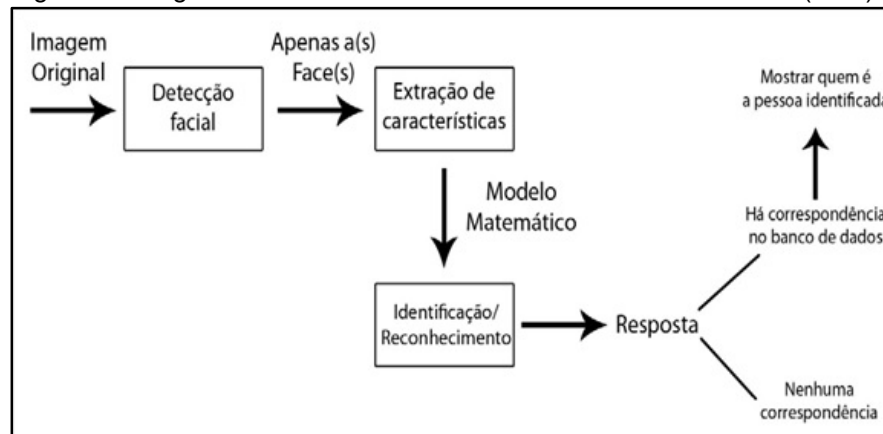
familiares instantaneamente. Ainda que as grandes mudanças no visual de uma pessoa como as expressões, envelhecimento, óculos, barba, cabelo e assim por diante esta capacidade do ser humano é altamente robusta. Apesar do reconhecimento facial ser uma tarefa simples e fácil para o ser humano, para um computador pode ser complicado e difícil (AGARWAL et al., 2010, tradução nossa).

A identificação consiste em padrões de características faciais, como a distância dos olhos, o formato da boca e do rosto (SILVA; CINTRA, 2015), sendo o reconhecimento facial um dos mais utilizados (PRAZERES, 2010; TRAMONTIN, 2016).

Em geral, o processo de reconhecer faces compreende os seguintes passos: aquisição de imagens, detecção de faces, segmentação e extração de características faciais, e finalmente a classificação de uma face (SANTANA; GOMES; SANTOS, 2014).

O processo de reconhecimento facial é composto por três processos distintos (figura 1): registro, verificação e identificação biométrica. Esses processos se diferenciam na forma de determinar a identidade de um indivíduo (SILVA; CINTRA, 2015).

Figura 1 - Diagrama de um Sistema de Reconhecimento de Faces (SRF)



Fonte: Braga (2013)

2.1.1 Métodos de reconhecimento facial

O primeiro passo para qualquer Sistema de Reconhecimento de Faces (SRF) é detectar os locais nas imagens onde as faces estão presentes (YANG; KRIEGMAN; AHUJA, 2002; LOPES, 2016). Porém, detectar a face em uma imagem é uma tarefa desafiadora, uma vez que há variabilidade na escala, orientação (*up-right*), localização e pose (perfil, frontal) (YANG; KRIEGMAN; AHUJA, 2002). Ainda, as expressões faciais, oclusão e iluminação também podem alterar a aparência geral dos rostos (YANG; KRIEGMAN; AHUJA, 2002).

Segundo Lopes (2016, p. 9), “a vantagem de se detectar a face, em um primeiro momento, é que após esta fase a procura pelas características fica limitada apenas em uma determinada região da imagem”. Este mesmo autor, ressalta que as técnicas para detectar faces apresentam alguns problemas comuns, conforme:

- Pose: as imagens de face variam de acordo com a posição da câmera que registrou a imagem;
- Expressão Facial: a expressão da face influencia diretamente na aparência da imagem de face;
- Presença de Elementos Estruturais: a presença de elementos como barba, bigode e óculos que podem modificar as características em termos de tamanho e luminosidade;
- Ocultação: no caso de imagens feitas em ambientes não controlados as faces podem aparecer, parcial ou totalmente sobrepostas, por objetos ou até mesmo por outras faces (Lopes, 2016, p. 9).

No início dos anos 70, o reconhecimento facial foi debatido como um problema de reconhecimento de padrão 2D (HANSEN, 1971 apud PARMAR; MEHTA, 2013). As distâncias medidas entre pontos importantes são necessárias para reconhecer faces, como por exemplo, medir a distância entre olhos e até os diferentes ângulos que compõem as características faciais (PARMAR; MEHTA, 2013). Porém, também é importante que os SRF sejam totalmente automáticos (PARMAR; MEHTA, 2013). Contudo, o reconhecimento facial é um problema bastante desafiador e interessante, que atraiu pesquisadores de diferentes áreas e contextos, como a psicologia, redes neurais, reconhecimento de padrões, computação gráfica e visão computacional (PARMAR; MEHTA, 2013).

São usados alguns métodos para a identificação do reconhecimento facial, como os métodos holísticos, métodos baseados em características e métodos híbridos (PARMAR; MEHTA, 2013).

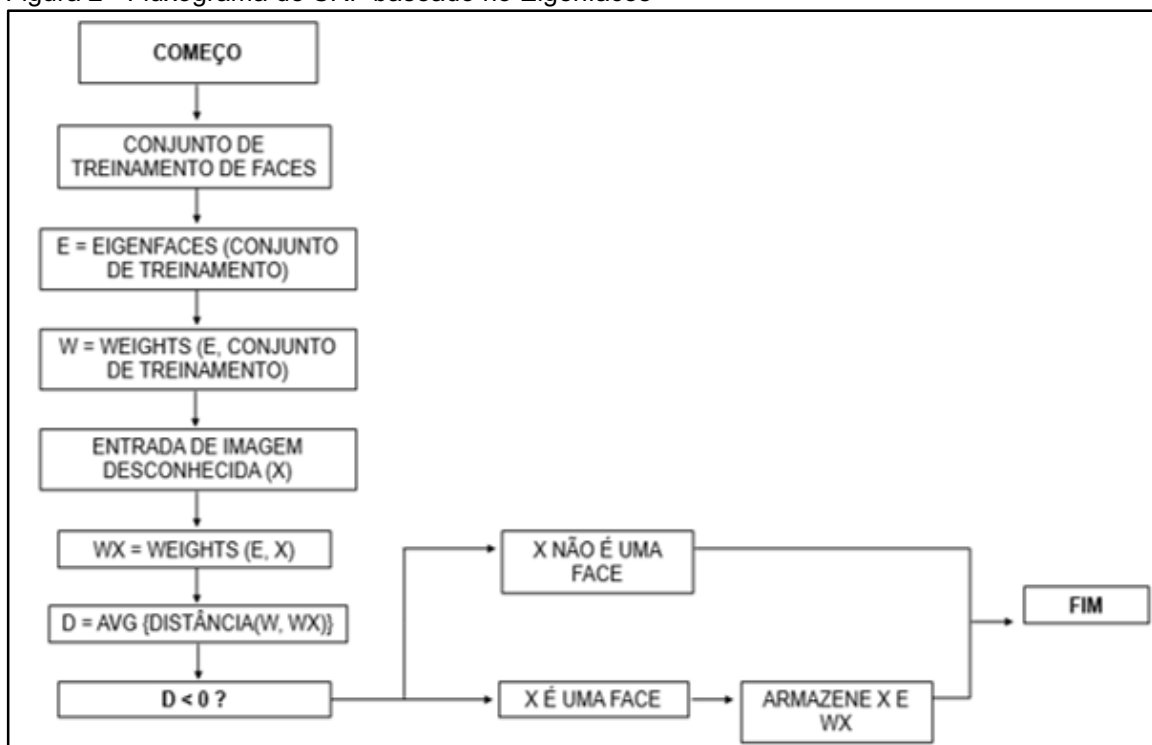
2.1.1.1 Métodos holísticos

Nesta abordagem, utiliza-se a imagem completa da face como dados de entrada no sistema de captura de face. Um dos melhores exemplos de métodos holísticos é o *Eigenfaces* (TURK; PENTLAND, 1991).

“O reconhecimento facial é feito por meio de métricas, como a distância euclidiana, distância de Manhattan ou distância de Mahalanobis, que determinam a similaridade entre as imagens dos indivíduos” (SILVA; CINTRA, 2015, p. 3). Exemplos de métodos que utilizam a abordagem holística são o *Eigenfaces*, *Independent Component Analysis* (ICA) e Redes Neurais Artificiais (RNA) (PARMAR; MEHTA, 2013; SILVA; CINTRA, 2015).

No fluxograma da figura 2 demonstram-se os diferentes estágios em um sistema de reconhecimento baseado no *Eigenfaces*.

Figura 2 - Fluxograma de SRF baseado no Eigenfaces



Fonte: Adaptado de Parmar e Mehta (2013)

A primeira parte do estágio para reconhecimento facial consiste na introdução de imagens em um banco de dados, onde são treinadas para comparar e criar suas próprias formas. Em seguida, faz-se a criação das próprias formas. São

feitas auto definições a partir de características dos rostos, como alinhamento da boca e olhos, que serão redimensionados para conter o mesmo tamanho. Então, as *Eigenfaces* podem ser extraídas das imagens pela ferramenta matemática Análise de Componentes Principais (PCA) (PARMAR; MEHTA, 2013).

Após a criação das *Eigenfaces*, as imagens são apresentadas como um vetor de pesos (PARMAR; MEHTA, 2013). Quando um peso de imagem não é conhecido, é comparado com os pesos que estão inseridos no sistema. Ainda, quando o peso de determinada imagem de entrada for superior a um limite, não será identificado. A identificação da imagem de entrada é feita quando a imagem no banco de dados cujos pesos são os mais próximos dos pesos da imagem de entrada (PARMAR; MEHTA, 2013, p. 85).

2.1.2.1 Métodos baseados em características

Neste método, as primeiras características a serem extraídas são os olhos, nariz e boca, assim como as suas localizações e estatísticas locais (geométricas e/ou aparências) que são alimentadas em um classificador estrutural (PARMAR; MEHTA, 2013). Os métodos de extração de recursos é um dos grandes desafios, pois o recurso de “restauração” é quando o sistema tenta recuperar recursos que são invisíveis devido às variações, como por exemplo, quando compara a posição da cabeça com uma imagem de perfil com uma frontal (ZHAO et al., 2003). Segundo Parmar e Mehta (2013), existem três métodos de extração, que são eles: baseados em bordas, linhas e curvas; baseados em modelos de recursos e baseados em correspondência estrutural que levam em consideração restrições geométricas nos recursos.

2.1.3.1 Métodos híbridos

Em sistemas de reconhecimento facial híbridos são utilizados extração de recursos e métodos holísticos, geralmente imagens 3D (PARMAR; MEHTA, 2013). A imagem do rosto é captada em 3D, permitindo o sistema registrar curvas de focos, como formas da testa ou queixo (PARMAR; MEHTA, 2013). Até um rosto em perfil poderia servir, pois o sistema utiliza o eixo da medida e profundidade, dando

informações suficientes para a construção de um rosto cheio (PARMAR; MEHTA, 2013).

Ainda segundo os mesmos autores, o sistema 3D geralmente apresenta uma sequência (PARMAR; MEHTA, 2013):

- a) detecção: é a captura (ou digitação) de um rosto em fotografia ou a fotografia do rosto em tempo real;
- b) posição: determinação do tamanho, localização e ângulo da cabeça;
- c) medição: atribuir medidas das curvas do rosto, criando um modelo com foco específico no interior e exterior do olho, e no ângulo do nariz;
- d) representação: conversão do modelo em código de uma representação numérica do rosto.
- e) correspondência: comparação dos dados recebidos com as imagens de rostos já existentes no banco de dados. As imagens 3D quando comparadas não precisam ter alterações. Porém, fotos inseridas em 2D, e nesse caso, a imagem 3D precisa de alterações, resultando em um dos maiores desafios que existe no campo atualmente.

Existem vários programas de reconhecimento faciais disponíveis, um deles é o *OpenFace*.

2.2 OPENFACE

Segundo os criadores, o interesse em criar a biblioteca originou-se pelo ambiente *mobile*, onde o reconhecimento facial está presente em tempo real e constantemente se adaptando a mudanças. Com isso, o principal foco a ser considerado é um sistema que fornece uma alta precisão com um baixo tempo de treinamento e predição (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

O número de dispositivo como *notebooks* e *smartphones* vem aumentando constantemente e normalmente estes são equipados com ao menos uma câmera. Torna-se cada vez mais raro ver algum destes dispositivos sem uma câmera inclusa (DELBIAGGIO, 2017, tradução nossa). O *OpenFace* é uma biblioteca de reconhecimento facial que utiliza duas linguagens de programação: *Python* e *Lua*.

Python é utilizada junto a biblioteca do *scikit-learn* para realizar tarefas relacionadas ao aprendizado de máquina, e Lua junto à biblioteca *Torch* que se encarrega de extrair informações das faces em aplicações de reconhecimento facial com *Deep Neural Network* (DNN), baseado no artigo *FaceNet* (SCHROFF; KALENICHENKO; PHILBIN, 2015).

A rede neural modificada foi treinada com uma base com 500 mil imagens que são a combinação das duas maiores bases de imagens de reconhecimento facial para pesquisa que se encontra na atualidade (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

A implementação realizada em *Python* é a responsável por efetuar a detecção, alinhamento e a classificação que usam bibliotecas. Já a implementação realizada em Lua é encarregada pela criação do descritor e a conversão da imagem para o descritor compreendido pela *Convolutional Neural Network* (CNN – em português Rede Neural Convolucional) (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa). CNN possui a capacidade de ser controlada variando sua profundidade e amplitude, além de fazer pressuposições fortes e principalmente corretas sobre a natureza das imagens (ou seja, estacionariedade das estatísticas e localidade das dependências de pixel). Assim, as CNNs têm muito menos conexões e parâmetros e, portanto, são mais fáceis de treinar, enquanto seu desempenho teoricamente melhor provavelmente será apenas um pouco pior (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

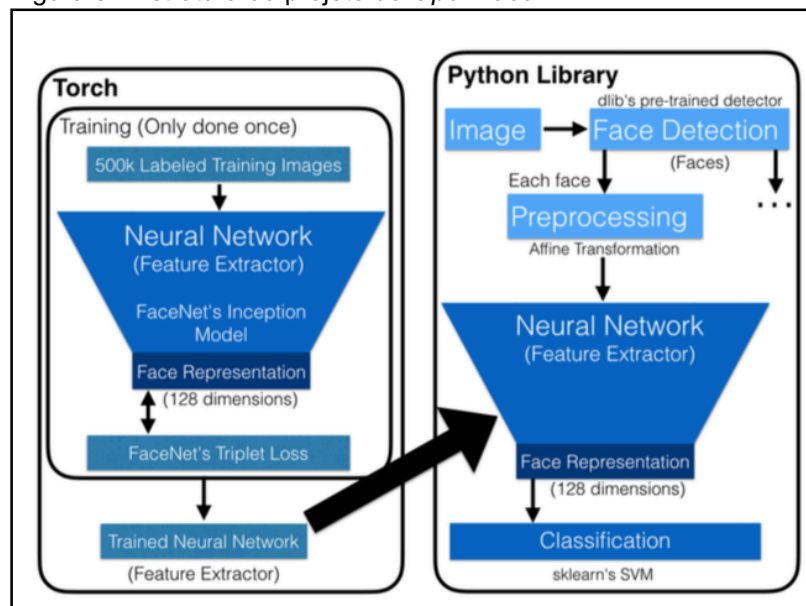
Já o componente de rede neural mapeia uma imagem pré-processada (alinhada) para uma representação de baixa dimensão. O *OpenFace* usa uma versão modificada da rede “nn4” do *FaceNet*. O nn4 é baseado na arquitetura *GoogLeNet* e a variante modificada (nn4.small2) reduz o número de parâmetros para um conjunto de dados menor. O *OpenFace* usa a perda do triplete do *FaceNet*, para que a rede forneça uma incorporação na hiperesfera da unidade, e a distância Euclidiana representa a similaridade (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

Na figura 3 a estrutura é dividida em duas partes, a esquerda há uma rede neural profunda usada para extrair os recursos (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa). Esta parte precisa ser executada

apenas uma vez, e como já foi realizada esta fase, o resultado pode ser usado diretamente na segunda parte da estrutura (DELBIAGGIO, 2017, tradução nossa).

A segunda parte da estrutura requer várias imagens por indivíduo, os rostos são detectados e extraídos das imagens com o detector pré-treinado do “dlib” que utiliza um *Histogram of Oriented Gradients* (HOG). Então, os rostos passam por uma fase de pré-processamento para finalmente serem usados na rede neural convolucional. A CNN utiliza os recursos extraídos na rede neural profunda da parte esquerda da estrutura como filtros (figura 3), e durante a fase de treinamento a CNN é adaptada de acordo com as diferentes classes. Já para reconhecer a pessoa de um rosto desconhecido, uma imagem é passada através da parte direita da estrutura, e a classe que representa a pessoa será dada como saída (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa; DELBIAGGIO, 2017, tradução nossa).

Figura 3 - Estrutura do projeto do OpenFace



Fonte: Amos; Ludwiczuk; Satyanarayanan (2016)

Ainda, segundo Amos, Ludwiczuk e Satyanarayanan (2016), para as implementações de treinamento e dedução da rede neural usa-se algumas bibliotecas, como o *Torch*, *Numpy*, *matplotlib*, *scikit-learn*, *OpenCV* e *Luajit* (figura 4).

Figura 4 - Bibliotecas utilizadas no OpenFace



Fonte: Do autor

Torch é uma biblioteca de aprendizado de máquina de código aberto, um *framework* de computação científica e uma linguagem de *script* baseada na linguagem de programação Lua. Tem objetivo de ter máxima versatilidade e velocidade na concepção de algoritmos científicos, tornando o processo simples. *Torch* vem com um grande ecossistema de pacotes voltados para a comunidade em aprendizado de máquina, visão computacional, processamento de sinais, processamento paralelo, imagem, vídeo, áudio e rede, entre outros (TORCH; 2018, COLLOBERT; KAVUKCUOGLU; FARABET, 2011, tradução nossa).

A biblioteca *Numpy* é um pacote fundamental na linguagem *Python*, permitindo trabalhar com tipos de *array* e matrizes de N dimensões com mais praticidade nas suas respectivas operações (NUMPY, s. d., tradução nossa; SARAIVA, 2014). Já o *Matplotlib* é uma “biblioteca popular para produção de gráficos de alta qualidade” (SARAIVA, 2014, p. 3), e pode ser utilizado em linguagem *Python* (MATPLOTLIB, s. d., tradução nossa).

OpenCV, é uma biblioteca para área da visão computacional, totalmente *OpenSource* e multiplataforma, possui módulos de processamento de imagens e vídeo, estrutura de dados, álgebra linear e conta com uma interface gráfica. Esta biblioteca foi desenvolvida nas linguagens de programação C/C++. Também, dá

suporte a programadores que utilizem Java, *Python* e *Visual Basic* e desejam incorporar a biblioteca a seus aplicativos (BRADSKI; KAEHLER, 2000).

A biblioteca LuaJIT é um compilador em tempo real para a linguagem de programação Lua (LUAJIT, s. d., tradução nossa). Ainda, o *OpenFace* utiliza duas linguagens de programação *Python* e *Lua* (figura 5).

Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível, com semântica dinâmica. A sintaxe simples e fácil de aprender do *Python* enfatiza a legibilidade e, portanto, reduz o custo de manutenção do programa. Além de suportar módulos e pacotes, que incentiva modularidade do programa e reutilização de código (VAN ROSSUM; DRAKE, 1995, tradução nossa). A linguagem de programação *Python* encontra-se em constante expansão e estabelece sendo como uma das linguagens mais populares para área da computação científica. Graças a sua natureza interativa de alto nível e seu amadurecido ecossistema de bibliotecas científicas, é uma escolha atraente para o desenvolvimento de algoritmos e análises exploratória de dados (MILMANN; AVAIZIS, 2011 apud PEDREGOSA et al., 2011).

Figura 5 - Linguagem de programação Python e Lua, respectivamente



Fonte: Do autor

Lua é uma linguagem de programação poderosa, eficiente e leve, projetada para estender aplicações (LUA, s. d.).

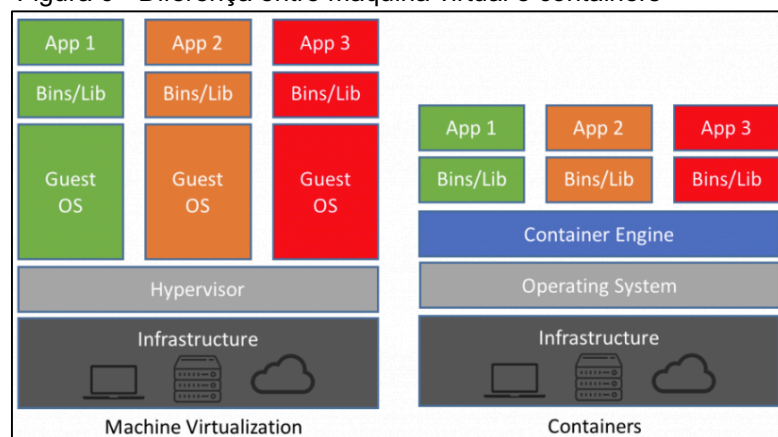
Diante de todas as bibliotecas necessárias, foi disponibilizado pelos criadores uma imagem em *Docker* com todas estas bibliotecas instaladas, fazendo que o uso do *OpenFace* seja mais simples, sem que seja necessário a instalação manualmente de todas as bibliotecas no sistema operacional.

2.2.1 DOCKER

É uma plataforma para desenvolvimento, envio e execução de vários aplicativos de maneira ágil. Ela permite que as aplicações sejam executadas separadamente da infraestrutura do servidor e sim trata as como um aplicativo gerenciável. Ainda, auxilia a enviar o código fonte mais rápido, realizar teste de maneira ágil e implantar rapidamente, diminuindo assim o ciclo de escrever e executar o código. Também possibilita o empacotamento de uma aplicação ou até um ambiente inteiro dentro de um ambiente chamado container, e este container se torna portátil para qualquer outro servidor que contenha o *Docker* instalado (DOCKER, 2018).

No ambiente com máquinas virtuais, o virtualizador é colocado acima do sistema operacional do *Host*, este é o que coordena os sistemas operacionais hospedeiros, e que são colocados acima dele. Diante de tal cenário todas estas complexidades podem ser evitadas, utilizando-se do *Docker*, os contêineres são disponibilizados diretamente no sistema operacional do *Host* (PREETH et al., 2015, tradução nossa). A figura 6 mostra a diferença de arquitetura entre os contêineres e a máquina virtual.

Figura 6 - Diferença entre máquina virtual e containers



Fonte: Brey (2018)

A virtualização de um container pode substituir uma máquina virtual (tabela 1), e sendo mais rápido iniciar e fechar um container do que uma máquina virtual (PREETH et al., 2015, tradução nossa).

Tabela 1 - Diferenças entre máquina virtual e *Docker*

Máquina Virtual	Docker
As máquinas virtuais são executadas em um hardware virtual e o sistema hospedeiro será carregado na memória virtual	Os convidados compartilham o mesmo sistema operacional, que é o sistema operacional host, no qual é carregado na memória física.
A comunicação entre os sistemas hospedeiro é por meio de dispositivos de rede, pode ser software.	A comunicação entre os containers é por meio de comando via terminal, <i>sockets</i> , entre outros.
A segurança depende do virtualizador.	Falta medidas de segurança.
Mais sobrecarga devido à sua complexidade.	Menos sobrecarga, pois é recipientes leves.
O compartilhamento de bibliotecas e arquivos não é possível.	Compartilhamento de arquivo possível (Ex: usando o comando SCP no LINUX).
Leva tempo na inicialização.	Inicialização mais rápida
Usa mais a memória, pois tem que armazenar o sistema operacional completo para cada convidado.	Menos uso de memória, pois compartilha o sistema operacional host.

Fonte: (Preeth et al., 2015, tradução nossa)

2.3 INTELIGÊNCIA ARTIFICIAL

O termo vem do inglês “*Artificial Intelligence*” (A. I. – I. A. em português) e foi utilizado pela primeira vez em 1956 por McCarthy (OSÓRIO; BITTENCOURT, 2000). Tem como fundamento “copiar” a inteligência humana, “e colocá-la a nosso serviço através de implementações que automatizem este tipo de processo/comportamento”, resultando em um “cérebro eletrônico” (OSÓRIO; BITTENCOURT, 2000).

A IA vem se destacando com o objetivo prático de simular a inteligência (NAKABAYASHI, 2009). Visto que, a palavra inteligência vem do latim *inter* (entre) e *legere* (escolher), inteligência seria as possibilidades de escolhas (NAKABAYASHI, 2009). Já a palavra artificial vem do latim *artificiale*, significando algo não natural, sem

a possibilidade de escolha, ou seja, algo produzido pelo homem (NAKABAYASHI, 2009).

“As ferramentas de IA, em sua evolução, começaram adquirindo conhecimentos que eram explicitados pelos especialistas de uma certa área (o que equivale a programar um sistema para resolver um problema)” (OSÓRIO; BITTENCOURT, 2000, p. 5).

A partir disso, estas ferramentas foram sendo dotadas de mecanismos de aquisição automática, como por exemplo o Prolog e sistemas especialistas (2ª geração), onde passaram a possuir mecanismos de máquinas (ou *machine learning*) (OSÓRIO; BITTENCOURT, 2000).

Estas ferramentas de aprendizado possuem limitações como:

(1) Usualmente assumem que os conhecimentos a serem adquiridos/manipulados e as informações disponíveis são corretos e devem estar completos em relação ao problema (o que dificilmente ocorre); (2) são orientadas para manipular informações simbólicas (informações quantitativas), onde os valores numéricos e contínuos (informações qualitativas) representam um problema difícil de ser tratado. Em relação a este último item, existem tentativas de integrar ao processamento simbólico com dados probabilísticos (regras bayesianas) e com uso de fatores de incerteza (regras fuzzy), como forma de expandir as potencialidades deste tipo de sistemas (OSÓRIO; BITTENCOURT, 2000).

2.3.1 Aprendizado de máquina

Aprendizado de Máquina (AM) é uma área da Inteligência Artificial que busca o desenvolvimento de técnicas computacionais com relação ao aprendizado, possibilitando a elaboração de sistemas computacionais aptos a adquirir conhecimento automaticamente por meio da experiência (MONARD; BARANAUSKAS, 2003).

Segundo Mitchell (1997), são programas de computador que melhoram seu desempenho em determinada tarefa por meio da experiência, o mesmo explica que um programa computacional aprende a partir da experiência E em relação a uma classe de tarefas T , com medida de desempenho P , e se seu desempenho nas tarefas T , medida por P melhora com a experiência E .

As técnicas de AM utilizam um princípio, no qual obtém-se conclusões a partir de um conjunto de exemplos dado como exemplo. Estas técnicas podem ser

divididas em duas categorias: aprendizado supervisionado e aprendizado não supervisionado.

Caso o processo de aprendizado seja informado um conjunto de exemplos e cada exemplo sendo formado por um conjunto de atributos de entrada e um conjunto de atributos de saída, este tipo de aprendizado é classificado como um aprendizado supervisionado. Por outro lado, se no processo de aprendizado, para cada exemplo houver apenas atributos de entrada, este é classificado como aprendizado não supervisionado. Este aprendizado tem por objetivo encontrar um conjunto de dados padrões ou sentidos que facilitem o entendimento destes dados (LORENA; CARVALHO, 2007; DE SOUTO et al., 2003).

A seguir, são apresentadas algumas definições de termos amplamente usados na literatura de AM.

- a) **exemplo (caso, registro, dado)**: são vetores de atributos. Um exemplo descreve objeto de interesse, como por exemplo as características da face de uma pessoa (SOUTO et al., 2003);
- b) **característica (atributo)**: descreve alguma característica de um exemplo, normalmente definidos em nominal e contínuo. Nominal quando não existe ordem entre valores. Contínuo quando existe uma ordem linear entre os valores (MONARD; BARANAUSKAS, 2003);
- c) **classe**: no aprendizado supervisionado, todo exemplo possui pelo menos um atributo especial denominado rótulo ou classe, que descreve o fenômeno de interesse (MONARD; BARANAUSKAS, 2003);
- d) **conjunto de exemplos (conjunto de dados)**: é composto por exemplos contendo valores de características (atributos), e no caso de aprendizado supervisionado cada exemplo também é associada uma classe. Geralmente, um conjunto de exemplos é dividido em dois subconjuntos, o conjunto de treinamento que é usado para o aprendizado do conceito e o conjunto de teste usado para medir o grau de efetividade do conceito aprendido (DE SOUTO et al., 2003);
- e) **erro e precisão (acurácia)**: uma medida de desempenho geralmente usada é a taxa de erro de um classificador h , também conhecida como taxa de classificação incorreta e apresentada por $err(h)$, taxa de erro

(*error rate*) da hipótese (h). Esta taxa de erro é obtida utilizando a fórmula a seguir:

$$err(h) = \left(\frac{1}{n}\right) \sum_{k=0}^n \|y_i \neq h(x_i)\|$$

Que a qual compara a classe verdadeira de cada exemplo com o rótulo atribuído pelo classificador. O complemento da taxa de erro, é a acurácia apresentada por $acc(h)$, precisão (*accuracy*) da hipótese h e pode ser obtida com a seguinte fórmula:

$$acc(h) = 1 - err(h)$$

O operador $\|E\|$ retorna 1 (um) se a expressão E for verdadeira e 0 (zero) casos contrários, e n é o número de exemplos. A acurácia é, em geral, estimada utilizando um conjunto independente de teste, que não foi usado em nenhum momento durante o processo de aprendizado (SOUTO et al., 2003; MONARD; BARANAUSKAS, 2003). Desta forma conclui-se que;

$$0 \leq acc(h) \leq 1 \text{ (Ou, equivalentemente, } 0\% \leq acc(h) \leq 100\%)$$

$$0 \leq err(h) \leq 1 \text{ (Ou, equivalentemente, } 0\% \leq err(h) \leq 100\%)$$

Há ainda outros meios de estimar a acurácia por meio de técnicas mais complexas, como o *cross-validation* (MITCHELL, 1997);

- f) **ruído**: o uso de dados imperfeitos é habitualmente comum. Estes podem ser derivados do próprio processo que gerou os dados, do processo de aquisição de dados, do processo de transformação ou mesmo de classes rotuladas incorretamente. Ou seja, nestes casos, diz-se que existe ruído nos dados (MONARD; BARANAUSKAS, 2003);
- g) **overfitting (super ajustamento)**: ocorre quando o modelo se especializa nos dados utilizados no seu treinamento, apresentando uma taxa de acurácia baixa para novos dados (MONARD; BARANAUSKAS, 2003).

2.3.1.1 Scikit-learn

Segundo Pedregosa et al. (2011, tradução nossa), o projeto *scikit-learn* fornece uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação *Python*. A pretensão do projeto é fornecer ferramentas de aprendizado de máquina eficientes e bem estabelecidas dentro de um ambiente de programação que seja acessível tanto para especialistas quanto para não especialistas em aprendizado de máquina.

Desde o seu lançamento em 2007, o *scikit-learn* tornou-se uma das mais populares bibliotecas de AM de código aberto para *Python*. O *scikit-learn* fornece algoritmos para tarefas de aprendizado de máquina, incluindo classificação, regressão, redução de dimensionalidade e *clustering*. A mesma também fornece módulos para extração de recursos, processamento de dados, e avaliação de modelos. Concebido como uma extensão da biblioteca *SciPy*, o *scikit-learn* foi construído sobre as populares bibliotecas do *Python*, *NumPy* e *matplotlib*. O *NumPy* estende o *Python* para oferecer suporte eficiente operações com grandes vetores e matrizes. O *matplotlib* fornece ferramentas de visualização, e *SciPy* fornece módulos de computação científica (HACKELING, 2017).

O *scikit-learn* é popular para pesquisa acadêmica porque possui uma API bem documentada, fácil de usar e versátil. Os desenvolvedores podem usar o *scikit-learn* para experimentar algoritmos diferentes, alterando apenas algumas linhas do código. O mesmo envolve algumas implementações populares de algoritmos de aprendizado de máquina, como o LIBSVM e o LIBLINEAR (HACKELING, 2017).

Com isso, o classificador é gerado a partir dos exemplos que a rede neural do *OpenFace* extrai de cada imagem fornecida assim utilizando as técnicas de AM que empregam um princípio de inferência denominado indução.

Para classificação o *OpenFace* utiliza um classificador SVM linear com um peso de regularização de 1, onde o mesmo executa consistentemente o mesmo ou até melhor que outros pesos de regularização e *kernels* (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

2.3.1.2 Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte constituem uma técnica de aprendizado que vem recebendo crescente atenção da comunidade de Aprendizado de Máquina (MITCHEL, 1997).

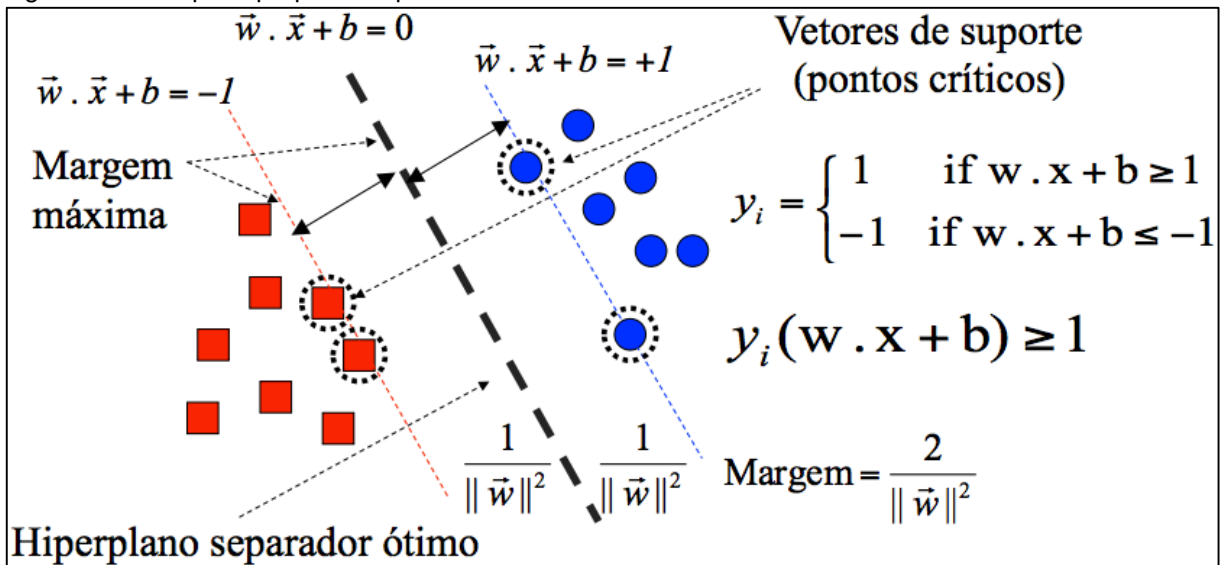
Ainda, são um conjunto de métodos de aprendizado supervisionados usados para classificação e regressão e detecção de exceções (HACKELING, 2017; PEDREGOSA et al., 2011, tradução nossa).

O SVM constrói um hiperplano ou um conjunto de hiperplanos em um espaço de alta ou infinita dimensão, que pode ser usado para classificação, regressão. Com isso, uma boa separação é alcançada pelo hiperplano que possui a maior distância até os pontos de dados de treinamento mais próximos de qualquer classe, pois em geral quanto maior a margem menor o erro de generalização do classificador (CAMPBELL, 2001, PEDREGOSA, 2011).

Segundo Campbell (2001), os classificadores que separam os dados por meio de um hiperplano são denominados classificadores lineares, e a equação que um hiperplano é apresentada por $f(x) = w \cdot x + b = 0$ em que $w \cdot x$ é o produto escalar entre os vetores w e x , $w \in X$ é o vetor normal ao hiperplano descrito e $\frac{b}{\|w\|}$ corresponde à distância do hiperplano em relação à origem, com $b \in R$ (CAMPBELL, 2001).

A margem é a distância perpendicular entre o hiperplano de separação e um hiperplano através dos pontos mais próximos, estes são os vetores de suporte. A região entre os hiperplanos de cada lado é chamada de faixa de margem e são exemplos de vetores de suporte de sinais opostos, como ilustrado na figura 7 (CAMPBELL, 2001).

Figura 7 - Exemplo hiperplano separador ótimo



Fonte: Carvalho (2015)

A figura 7 demonstra um exemplo com duas classes. Mas, conforme Campbell (2001) muitos conjuntos de dados da vida real envolvem classificação multiclasse e vários esquemas foram propostos para lidar com isso.

2.3.1.3 Validação cruzada

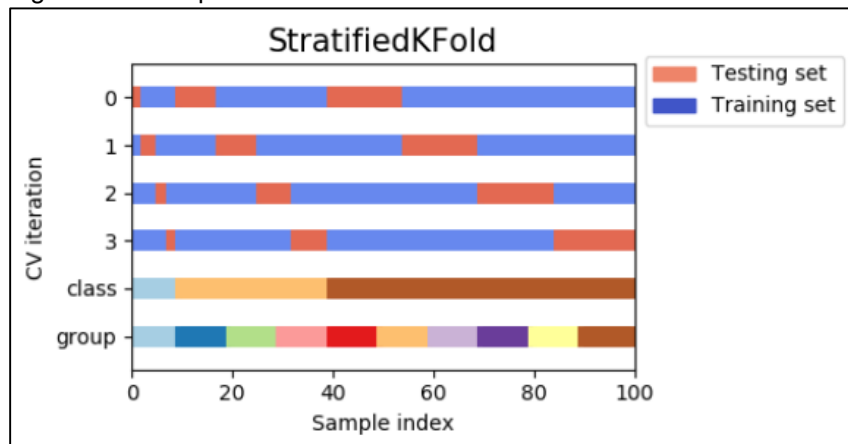
Métricas de avaliação são fundamentais para verificar se o modelo está com bons resultados e como podemos melhorar o mesmo. AM é uma ferramenta bem desenvolvida, mas não existe um único algoritmo que apresente o melhor desempenho para todos os problemas. Dessa forma, é importante compreender o poder e a limitação dos diferentes algoritmos utilizando alguma metodologia de avaliação que permita comparar algoritmos, como a validação cruzada.

A validação cruzada consiste em dividir todo o conjunto de dados em n partes, essas partes se chamam *folds* ou dobras bem como demonstrado na figura 10. Dessas partes uma será separada para teste e as outras restantes serão usadas para treinar o modelo. Isto é feito repetidamente até que o modelo seja treinado e testado com todas as partes. Assim, idealmente minimiza-se as chances de algum dado importante para a classificação ser deixado de fora durante o treinamento.

A biblioteca do *scikit-learn* fornece o método *cross_val_score* que foi usado para avaliar a pontuação do classificador em cada *fold*. Este método por padrão utiliza

outro objeto chamado de *StratifiedKFold*, que tem por objetivo retornar dobras estratificadas, ou seja, estas dobras são feitas preservando a porcentagem de amostras para cada classe como é representado na figura 8 (PEDREGOSA et al., 2011).

Figura 8 - Exemplo StratifiedKFold



Fonte: Pedregosa et al. (2011)

Outra estratégia adotada é a *OneVsRest*, consiste em ajustar um classificador por classe. Para cada classificador, a classe é ajustada contra todas as outras classes. Além de sua eficiência computacional, uma vantagem dessa abordagem é sua interpretabilidade. Como cada classe é representada apenas por um e um classificador, é possível obter conhecimento sobre a classe inspecionando seu classificador correspondente. Essa é a estratégia mais comumente usada para classificação multiclases e é uma opção padrão justa (PEDREGOSA et al., 2011).

2.3.1.4 Matriz de confusão

Hackeling (2017), cita que esta métrica de desempenho de classificação de multiclases, assim como na classificação binária, as matrizes de confusão são úteis para visualizar os tipos de erros feitos pelo classificador. A precisão, o recall e a pontuação F1 podem ser calculados para cada uma das classes, e a precisão de todas as previsões também pode ser calculada.

Uma forma de avaliar a qualidade de classificação é através da matriz de confusão (PRINA; TRENTIN, 2015).

Matos et al. (2009, p. 9), explicam que:

Precisão: taxa com que todos os exemplos classificados como positivos são realmente positivos. Nenhum exemplo negativo é incluído.

$$\text{Precisão} = \frac{VP}{VP+FP}$$

Revocação: Taxa com que classifica como positivos todos os exemplos que são positivos. Nenhum exemplo positivo é deixado de fora. Apresenta uma indicação do quanto do total de informação relevante foi recuperada.

$$\text{Revocação} = \frac{VP}{VP+FN}$$

Ainda, a matriz de confusão (ou matriz de erros) “é uma matriz quadrada que expressa a relação do número de unidades de amostra atribuída a uma categoria particular de duas classificações” (PASSO, 2013, p.22).

A matriz de confusão, é uma matriz de valores reais e valores preditos pelo classificador. Por exemplo, se uma matriz foi treinada para distinguir entre gatos, cães e coelhos, essa matriz irá assumir os possíveis resultado que o algoritmo oferecer (REVISTABW, 2018).

Ao considerarmos uma amostra de 27 animais (8 gatos, 6 cães e 13 coelhos), quando aplicado no sistema oferece o resultado demonstrado na figura 9 (REVISTABW, 2018).

Figura 9 - Demonstração de Matriz de confusão

		Prevista classe		
		Gato	Cão	Coelho
Classe real	Gato	5	3	0
	Cão	2	3	1
	Coelho	0	2	11

Fonte: Revistabw (2018)

Assim o classificador resultou em (REVISTABW, 2018):

- de 8 gatos reais, o sistema previu que 5 eram gatos e 3 eram cães;
- de 6 cães, o sistema previu que 2 eram gatos, 1 era coelho e 3 eram cães;
- dos 13 coelhos, o sistema previu que 2 eram cães e 11 eram coelhos.

Demonstrado esse exemplo, observa-se as maiores situações que o sistema possui. No caso, o sistema consegue detectar mais precisamente coelhos (84,62%), do que cães (50%) (REVISTABW, 2018).

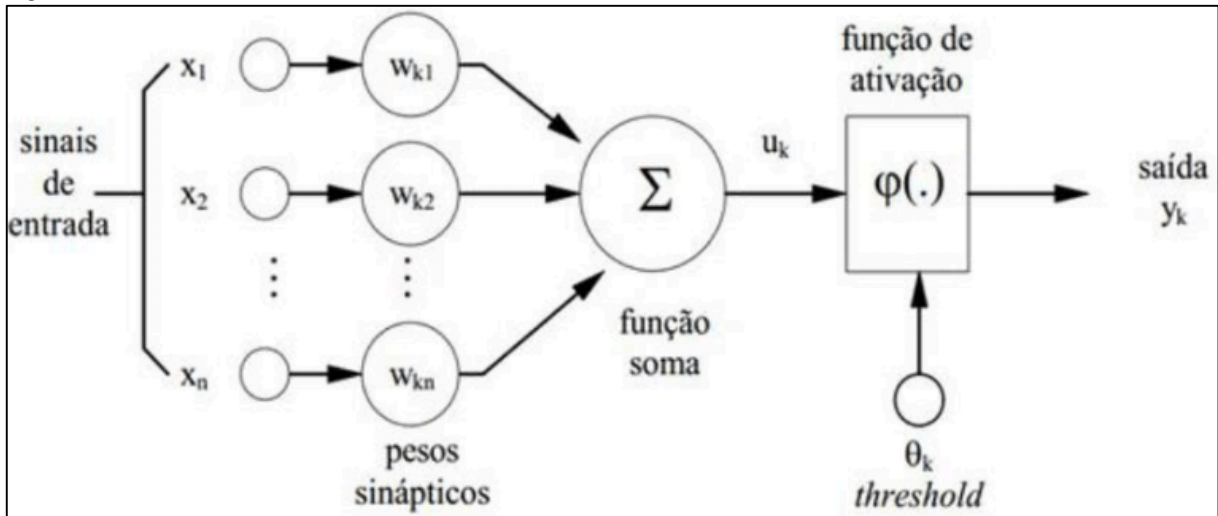
2.3.2 Redes Neurais Artificiais

As RNA inspiram-se em modelos biológicos, contudo, Haykin (2001) destaca que as redes desenvolvidas em computadores são bastante primitivas se comparadas ao cérebro humano e que os níveis estruturais de organização do sistema nervoso central não são replicáveis computacionalmente. Ainda, define que a rede neural artificial é uma máquina projetada para modelar a maneira como o cérebro humano realiza uma tarefa particular. Diversas características importantes encontradas nas redes biológicas podem ser encontradas nas redes artificiais, como o reconhecimento de padrões, processamento paralelo e distribuído e a tolerância a falhas.

Redes neurais artificiais são modelos matemáticos baseadas nas redes neurais biológicas do cérebro humano, isto é, são construídas a partir de neurônios interconectados (BARRETO, 2002). Da mesma maneira que um neurônio biológico, a estrutura do neurônio artificial é relativamente simples e responde a estímulos de outros neurônios conectados a ele (MOREIRA, 2002).

A figura 10 exemplifica um neurônio artificial, nas RNAs os receptores representam os neurônios da camada de entrada. As redes neurais propriamente ditas constituem as camadas ocultas e os atuadores correspondem os neurônios da camada de saída. Nessa representação, observa-se que cada terminal de entrada do neurônio está recebendo um valor e um peso associado a ele. As ponderações dos valores recebidos com os pesos de cada terminal são somadas e sobre o valor total recebido pelo neurônio é aplicada uma função de ativação, que restringe a amplitude de saída de um neurônio.

Figura 10 - Modelo de um neurônio artificial



Fonte: Haykin (2001)

Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. Segundo Haykin (2001, p.36-37), pode-se identificar três elementos básicos do modelo neuronal:

- a) um conjunto de *sinapses* (ou elos de conexão), cada uma caracterizada por um *peso* ou *força* própria. Especificamente, um sinal x_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico w_{kj} . É importante notar a maneira como são escritos os índices do peso sináptico w_{kj} . O primeiro índice se refere ao neurônio e o segundo se refere ao terminal de entrada da sinapse à qual o peso se refere. Ao contrário de uma sinapse do cérebro, o peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos bem como positivos;
- b) um *somador* para somar os sinais de entrada, ponderados pelas respectivas sinapses do neurônio; as operações constituem um *combinador linear*;
- c) uma *função de ativação* para restringir a amplitude da saída de um neurônio. A função de ativação é também referida como *função restritiva* já que limita o intervalo permissível de amplitude do sinal de saída a um valor finito. Regularmente, o intervalo normalizado da amplitude da saída de um neurônio é escrito como o intervalo unitário fechado $[0, 1]$ ou alternativamente $[-1, 1]$.

Com isso, compreende-se que as RNA podem ser utilizadas em diversos setores e para fins diversos. Ainda que essa área possa ser bem complexa, há um aumento nas pesquisas à medida que se tem mais conhecimento sobre.

3 TRABALHOS CORRELATOS

Vários trabalhos são desenvolvidos acerca do reconhecimento facial, devido a sua grande importância atualmente, como visto no referencial teórico, a área de reconhecimento facial está em grande ascendência.

Neste capítulo, são apresentados alguns trabalhos que se dedicaram a estudar as questões fundamentais na área de reconhecimento facial.

3.1 UMA COMPARAÇÃO DOS ALGORITMOS DO RECONHECIMENTO FACIAL

Neste estudo, o autor Delbiaggio (2017, tradução nossa), tem por objetivo de projeto da pesquisa realizar testes em diferentes algoritmos de reconhecimento facial e posteriormente efetuar testes de precisão e então compará-los. Os algoritmos que o autor utiliza são o *Eigenfaces*, *Fisher-faces*, Histograma de Padrão Binário Local e o algoritmo de rede neural convolucional profunda, o *OpenFace*.

Segundo Delbiaggio (2017, tradução nossa), realizou o experimento em três etapas diferentes. A primeira etapa consistiu em cinco pessoas, onde todas as fotos foram tiradas no mesmo ambiente e com a mesma luz. Já a segunda etapa contém dez pessoas e a terceira e última etapa tem quinze pessoas. Cada etapa é dividida em três testes. A diferença entre esses testes é o número de imagens por pessoa no conjunto de treinamento. Esta diferença nos testes se dá por terem dez pessoas no primeiro teste, vinte no segundo e quarenta no terceiro e último teste.

Por fim, o autor conclui que o uso da rede neural convolucional é mais eficiente para a realização de um reconhecimento facial do que uma abordagem estatística ou busca de padrões. Outra abordagem do autor é que, aumentando a quantidade de dados no conjunto de dados para extrair as características também é uma maneira de melhorar potencialmente os resultados (DELBIAGGIO, 2017, tradução nossa).

3.2 SISTEMA DE RECONHECIMENTO FACIAL

Segundo Braga (2013) seu objetivo foi o desenvolvimento de um sistema de reconhecimento facial automatizado. Este sistema se fez utilizando o algoritmo de Viola-Jones e uma análise discriminante.

O autor optou por usar o MATLAB como software para o desenvolvimento de reconhecimento facial. Segundo Braga (2013), o software escolhido sucedeu-se, pois, o mesmo oferece uma vasta biblioteca de funções matemáticas e algoritmos numéricos, além de um *toolbox* próprio para processar imagens.

De acordo com Braga (2013) a maioria das imagens testadas obtiveram um resultado satisfatório, entretanto algumas imagens apresentaram falsos positivos e falsos negativos.

Braga aborda ainda, que a implementação de sistema de reconhecimento facial exige algoritmos poderosos e que necessitam ser qualificados para lidar com as diversas variações de pose da face em uma foto, além de ruídos na imagem de entrada e outros fatores, como por exemplo a iluminação do ambiente.

Apesar das dificuldades apresentadas por Braga, o sistema obteve uma baixa taxa de erros e na maioria das condições testadas apresentou resultados satisfatórios, mesmo com a região dos olhos e da boca cobertas (BRAGA, 2013).

3.3 DETECÇÃO E RECONHECIMENTO FACIAL POR MEIO DE APRENDIZADO DE MÁQUINA

Neste trabalho, realizado na Universidade de Brasília, Maia (2016) propôs a identificação facial utilizando aprendizado de máquina, e realizando técnicas de extração de características e RNA *multilayer perceptron* (MLP).

Maia (2016) utilizou o MATLAB para implementar sua proposta, e para a coleta de imagens e criação do banco de dados utilizou o *Google Picasa*. Maia (2016) separou o banco de imagens em duas partes de tamanhos iguais, 906 faces para compor o conjunto de treino e 906 faces para compor o conjunto de teste, totalizando 1812 faces.

Conforme explica Maia (2016) através de seus resultados, pode-se observar que o desempenho da rede neural artificial em termos da capacidade de generalização, está diretamente ligada a fatores como qualidade da base de treino, distribuição das classes representatividade dos padrões, entre outros.

Já em relação ao processo de aprendizado da rede neural, Maia (2016) observou que não há como se obter uma solução única determinística e fechada, pois a função de erro é calculada a cada tempo e os pesos sinápticos ajustados na prática a cada tempo de treino. Portanto, para uma mesma estrutura de rede com a mesma base de treino, os resultados não serão idênticos (MAIA, 2016).

Maia (2016), também adverte que é importante citar que os valores dos pesos sinápticos são iniciados aleatoriamente a cada vez que a rede é criada. Logo, cada treino é único quanto ao número de épocas e erro quadrático médio obtido.

3.4 COMPARAÇÃO DE REDES NEURAI DE RECONHECIMENTO FACIAL

Neste trabalho, Uibo (2016, tradução nossa) realiza uma comparação entre redes neurais de reconhecimento facial.

Segundo Uibo (2016, tradução nossa), as redes neurais apresentaram bons resultados contendo imagens de celebridades de alta qualidade. Porém, o autor questiona se essas redes seriam capazes de executar também em um conjunto de dados diferente de imagens de arquivo de baixa qualidade.

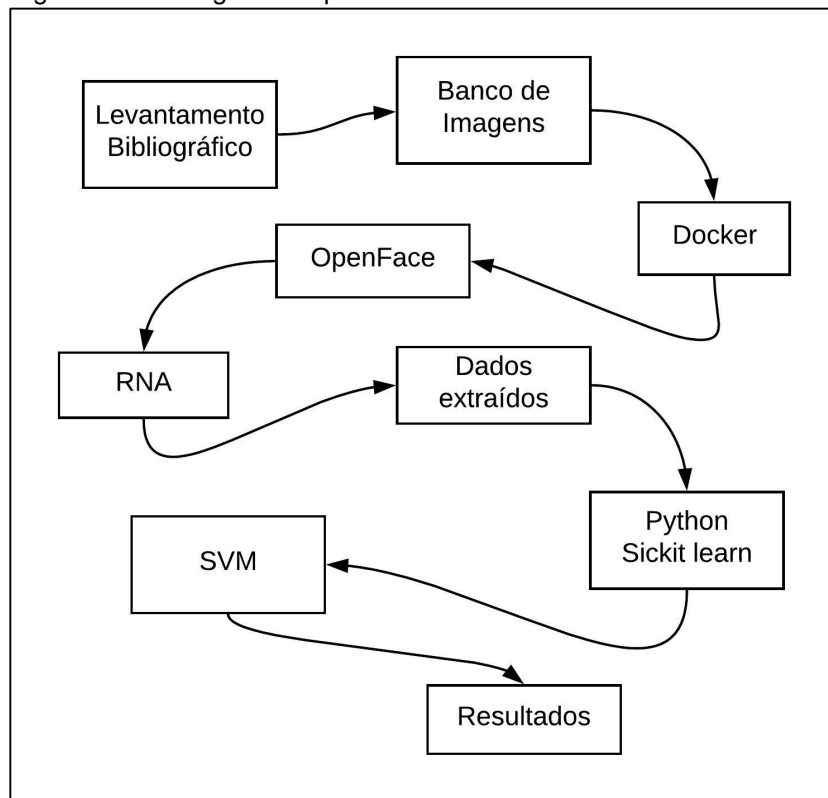
Os testes realizados por Uibo incluíram a extração de características para cada três das redes e para os conjuntos de dados LFW e FOTIS, produzindo o conjunto de dados de referência para o banco de dados FOTIS e analisando a precisão da classificação de cada conjunto de dados/rede.

Os resultados apresentados por Uibo (2016, tradução nossa) demonstraram que a rede VGG foi a melhor, seguido pela CASSIA e por último o *OpenFace*. Uilbo ainda destaca que discrepância mostrada entre a VGG e *OpenFace* se deve ao fato de conter algum erro da rede ou na configuração de *software*. Pois não há artigo publicado pelo *OpenFace*, e as informações disponibilizadas são apenas scripts de amostras e artigos separados publicados na página da *Web OpenFace*, o que aumenta a chance de erro humano.

4 ACURÁCIA DO SVM NO RECONHECIMENTO DE FACES EM BASES DE IMAGENS

A figura 11 exemplifica as etapas realizadas nesta pesquisa. Inicialmente, fez-se necessário realizar um levantamento bibliográfico acerca da temática do reconhecimento facial e inteligência artificial.

Figura 11 - Fluxograma experimental



Fonte: Do autor

Após, foi realizada uma pesquisa sobre os bancos de imagens existentes na internet. Dentre os diversos bancos de imagens disponíveis para teste, selecionou-se três bancos:

- a) *Labeled Faces in The Wild* (LFW)¹: possui cerca de 13 mil imagens diversas, com pessoas em diferentes poses, expressões e iluminação,

¹ <http://vis-www.cs.umass.edu/lfw/>

além do fundo diferente de cada imagem, ou seja, em ambientes não controlado;

- b) MS-CELEB-1M¹: contém 10 milhões de imagens para cerca de 100 mil celebridades, possuindo tamanho superior a 100Gb, devido a isto optou-se por trabalhar com um subconjunto de exemplos disponível para *download*. No qual, deste subconjunto foi escolhido cinco celebridades (Jennifer Lawrence, Justin Bieber, Ming-Na Wen, Luiz Fabiano e Sasha Pieterse), totalizando 566 imagens para o experimento. Assim como a LFW, o MS-CELEB-1M apresenta imagens em ambientes não controlados;
- c) FEI DATABASE²: é um banco de imagens brasileiro que possui 2800 imagens, contendo 200 pessoas no qual cada pessoa tem no total 14 imagens cada. Diferentemente das outras bases, esta apresenta um ambiente controlado, com imagens em um fundo branco e todas as fotos são obtidas da mesma forma para todas as pessoas.

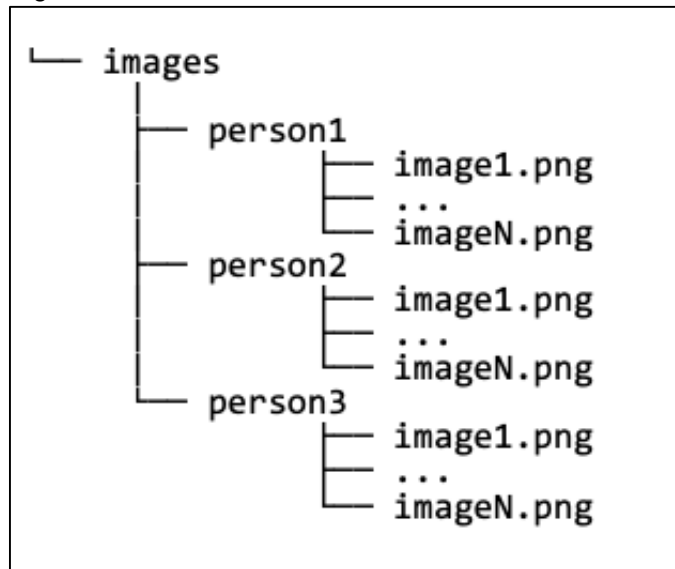
Primeiro, reproduziu-se o experimento de teste de acurácia da LFW, para validar os resultados apresentados no artigo “*OpenFace: A general purpose face recognition library with mobile applications*” (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa). Os resultados da LFW utilizando o *OpenFace* são obtidos calculando a distância euclidiana ao quadrado nos pares e de rotulagem sob um limiar como sendo a mesma pessoa acima do limiar como pessoas diferentes e abaixo como pessoas iguais. O melhor limite nos *folds* de treinamento é usado como limite na dobra restante (AMOS, LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

Posteriormente, empregou-se o uso das duas bases restantes (MS-CELEB-1M e FEI DATABASE) que possuem diferentes tamanhos, para avaliar o classificador SVM em diferentes cenários e tamanhos de exemplos. Para a utilização destas bases é necessária uma estrutura de diretórios, onde as imagens de cada indivíduo estejam separadas em diretórios diferentes (figura 12).

¹ <https://www.msceleb.org/>

² <https://fei.edu.br/~cet/facedatabase.html>

Figura 12 - Estrutura de diretórios em subdiretórios diferentes



Fonte: Do autor

Diferentemente da LFW e MS-CELEB-1M, a FEI DATABASE não possui uma estrutura de pastas para armazenamento de imagens para cada pessoa. Para isso, foi elaborado um simples algoritmo (figura 13), para que a separação não fosse realizada de forma manual. Então, para cada indivíduo foi criado uma pasta, resultando na separação das imagens.

Figura 13 - Algoritmo para criar subpastas na FEI DATABASE

```

1  import os
2  import shutil
3  import sys
4
5  directoryName = sys.argv[1]
6  directoryPath = os.getenv('PWD')
7  directoryPath = os.path.join(directoryPath, directoryName)
8  os.chdir(directoryPath)
9
10 print os.getcwd()
11
12 list_peoples = []
13
14 for file_found in os.listdir(os.getcwd()):
15     list_peoples.append(file_found)
16     list_peoples.sort(key=lambda f: int(filter(str.isdigit, f) or -1))
17
18 print('All files load to memory')
19
20 n = 1
21 count = 1
22 name = "Pessoa_%i" % n
23 for people in list_peoples:
24     print(name, people)
25     # if n == 42:
26     #     break
27     # n += 1
28     if not os.path.exists(name):
29         os.mkdir(name)
30     shutil.move(people, name)
31     if len(os.listdir(name)) == 14:
32         n += 1
33         name = "Pessoa_%i" % n
34         count += 1
35
36 print(str.format('total %i' % count))

```

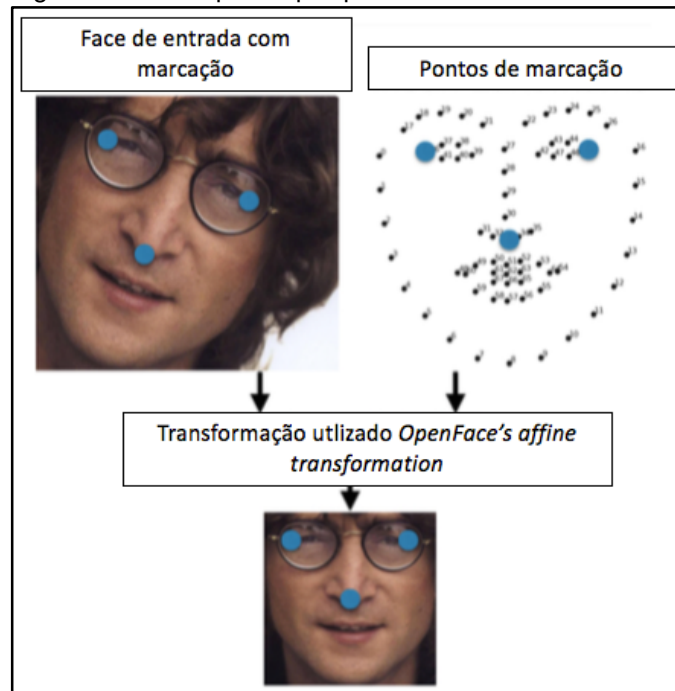
Fonte: Do autor

Segundo Amos, Ludwiczuk e Satyanarayanan (2016), a maneira mais fácil de executar o *OpenFace* é utilizando a imagem do ambiente para Docker (disponibilizado pelos criadores), onde contém todas as bibliotecas e *frameworks* para o funcionamento. Foi escolhido utilizar a imagem do container, a fim de evitar problemas de compatibilidade com o sistema operacional, pois a instalação manual de todas as bibliotecas e *frameworks*, além de algumas como *OpenCV*, necessita realizar a compilação, podendo causar interrupções e demora no experimento.

Dessa forma, baixou-se a imagem do *OpenFace* e posteriormente instalado no Docker. Com o container em execução e a imagem do *OpenFace* em funcionamento, inicia-se o procedimento de pré-processamento das imagens, no qual foi utilizado o detector de pontos de referência dlib (King, 2009) para centralizar e

rotacionar a face na imagem (com base no nariz e olhos) (figura 14). A partir deste momento, as imagens foram cortadas com base nos pontos de referência externo e redimensionadas em 96x96 para aparecerem em locais semelhantes na entrada da rede neural.

Figura 14 - Exemplo de pré-processamento



Fonte: Amos; Ludwiczuk; Satyanarayanan (2016, tradução nossa)

A rede neural escolhida foi “nn4.small2.v1”, onde usa o alinhamento que consiste em três pontos principais: a parte externa dos olhos e o nariz, totalizando 68 pontos de referência em uma face. Ainda, “nn4.small2.v1” é pré-treinada e foi escolhida com base no desempenho apresentado no desenvolvimento no artigo do *OpenFace*.

A tabela 2 demonstra os resultados de desempenho das RNAs do *OpenFace*, utilizando um CPU de 8 núcleos de 3.70 GHz e uma GPU Tesla K40 (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

Tabela 2 - Performance da rede neural

Modelo	Tempo de Execução (CPU)	Tempo de Execução (GPU)
nn4.v1	75.67 ms ± 19.97 ms	21.96 ms ± 6.71 ms
nn4.v2	82.74 ms ± 19.96 ms	20.82 ms ± 6.03 ms
nn4.small1.v1	69.58 ms ± 16.17 ms	15.90 ms ± 5.18 ms
nn4.small2.v1	58.9 ms ± 15.36 ms	13.72 ms ± 4.64 ms

Fonte: Amos; Ludwiczuk; Satyanarayanan (2016)

Neste trabalho, o *Docker* foi necessário para executar a RNA do *OpenFace* à qual é responsável por extrair as características das faces, já as demais etapas do AM foram executadas diretamente no sistema operacional, sendo necessária a instalação de algumas bibliotecas citadas anteriormente, como *scikit-learn*, *numpy*, *matplotlib*.

Finalizado o pré-processamento das imagens, as mesmas foram utilizadas na RNA para extração das representações faciais. Para sua aplicação, executou-se a linha de comando da figura 15, na qual a rede neural está implementada na linguagem Lua, para este comando são necessários dois parâmetros, “-outDir” este diretório em que são gerados os arquivos “.csv” e “-data”, no qual encontra-se as imagens pré-processadas, que foi utilizada pela RNA.

Figura 15 - Comando usado para gerar as representações

```
./batch-represent/main.lua -outDir "feature dir" -data "aligned IMG dir"
```

Fonte: Do autor

Para criação e avaliação do SVM foram usados métodos disponíveis na biblioteca *scikit-learn*. Onde, por meio do `cross_val_score` juntamente com o *K-fold*, dividiu-se o conjunto de dados em um subconjunto de treinamento e teste utilizando $k=5$. Ainda, devido ao fato de existir mais de uma classe usou-se o padrão *StratifiedKFold*, onde os *folds* são feitas preservando a porcentagem de amostras para cada classe.

A figura 16 mostra o código fonte implementado, onde utiliza os arquivos “.csv” gerado pela RNA e que consiste nas representações faciais e suas respectivas

classes. Após a criação do SVM com *kernel* linear é utilizado o método `cross_val_score`, passando por parâmetro, classificador criado, dados de exemplos, classes e a quantidade de *folds*.

Figura 16 - Demonstração do código fonte

```

35 def crossVal(args):
36
37     fname = "{}labels.csv".format(args.workDir)
38     labels = pd.read_csv(fname, header=None).as_matrix()[:, 1]
39     labels = [os.path.split(os.path.dirname(label))[1] for label in labels]
40     le = LabelEncoder()
41     le.fit(labels)
42     labelsNum = le.transform(labels)
43     nClasses = len(le.classes_)
44
45     fname = "{}reps.csv".format(args.workDir)
46     embeddings = pd.read_csv(fname, header=None).as_matrix()
47
48     X, y = embeddings, labelsNum
49
50     ln_clf = SVC(C=1, kernel='linear', probability=True)
51     clf = OneVsRestClassifier(ln_clf)
52
53     print("\nObtaining results cross-validation")
54     t_start = time.clock()
55     scores = cross_val_score(clf, X, y, cv=5, n_jobs=2)
56     t_end = time.clock()
57     print("Acuracy: %0.4f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
58     print('Cross validation time: {:.4f}s.'.format(t_end-t_start))
59     print("Scores:{}".format(scores))
60     print("=====")

```

Fonte: Do autor

Para que seja executado este método, a linha de comando demonstrada na figura 17 precisou ser executada. Neste comando consta a versão do *Python*, o nome do arquivo e o nome do método desenvolvido. Este último, é necessário informar como parâmetro o caminho onde se encontra os arquivos “.csv”.

Figura 17 - Comando para executar o algoritmo

```
[Lucass-MacBook-Pro:evaluation lucasorso$ python3 ROC_CROSS.py crossVal $(pwd)
```

Fonte: Do autor

Assim, ao executar o algoritmo desenvolvido é retornado um vetor de pontuações do classificador para cada execução da validação cruzada.

5 RESULTADOS E DISCUSSÃO

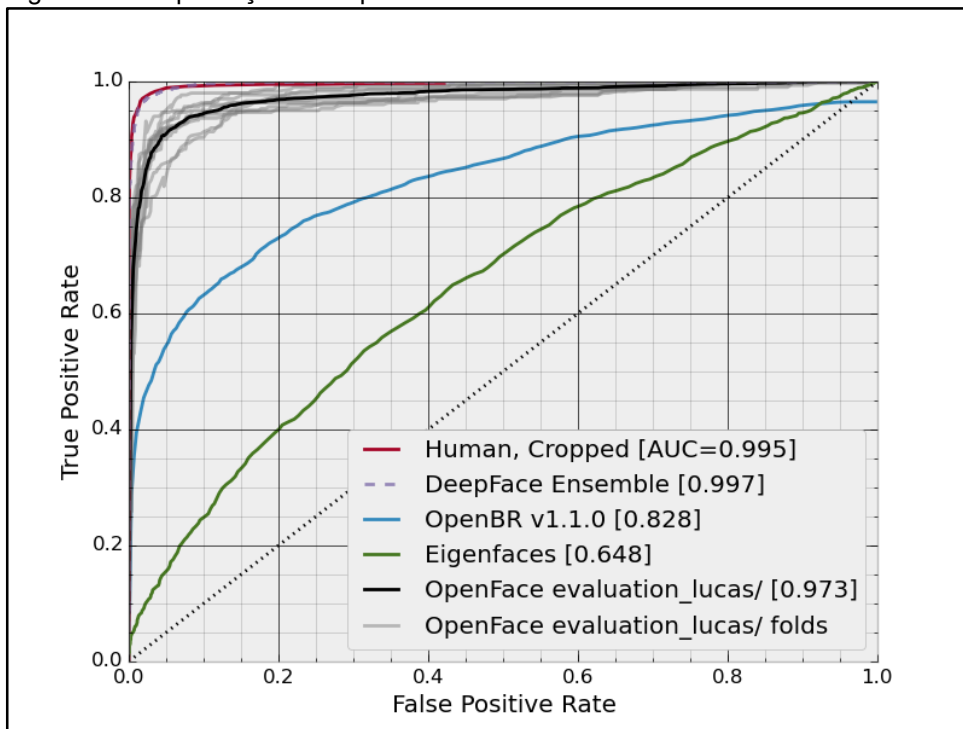
De acordo com o referencial teórico elaborado na pesquisa, o reconhecimento facial vem apresentando avanço nas últimas décadas. Para entendimento dos resultados apresentados, foi necessário aplicar os conceitos apreendidos durante esta pesquisa.

Neste capítulo apresenta-se os resultados obtidos bem como a discussão dos mesmos.

5.1 REPRODUÇÃO DO EXPERIMENTO DE ACURÁCIA NA BASE LFW

Nesse experimento, os resultados (figura 18) coincidiram com os mesmos resultados apresentado por Amos, Ludwiczuk e Satyanarayanan (2016), onde demonstram uma alta precisão na identificação de faces, quando utilizando a base de dados LFW.

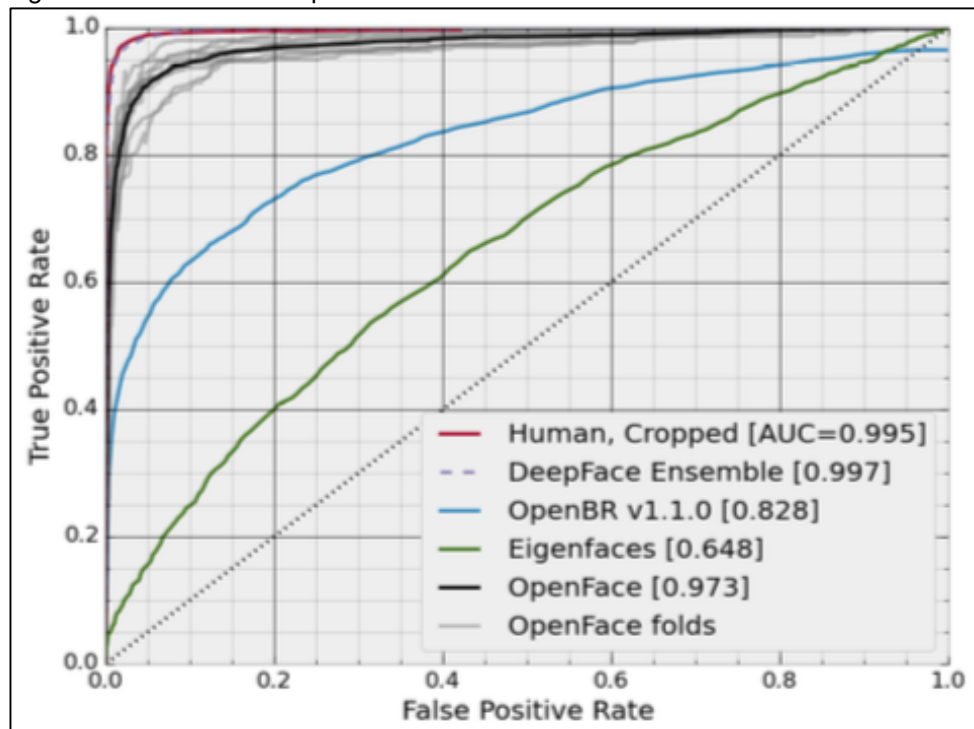
Figura 18 - Reprodução do experimento



Fonte: Do autor

A figura 19 representa o gráfico obtido do trabalho de Amos, Ludwiczuk e Satyanarayanan (2016). Esses resultados quando comparados aos nossos, é possível afirmar que a construção do experimento é replicada, obtendo-se os mesmos resultados e acurácia.

Figura 19 - Acurácia do OpenFaces no teste LFW



Fonte: Amos, Ludwiczuk e Satyanarayanan (2016)

Com esta reprodução dos resultados pode se verificar como o *OpenFace* utiliza as técnicas de inteligência artificial para o processo de reconhecimento facial, com isso ficou esclarecido a forma de como utiliza-lo em outras bases de imagens.

A biblioteca do *OpenFace* contém o algoritmo também em *Python*, responsável por realizar o experimento na base da LFW. Porém, este algoritmo é somente para a base LFW, já que ela contém suas regras para a realização do experimento. Sendo assim, não foi possível testar e plotar a curva ROC utilizando o mesmo algoritmo em outros bancos de imagens como foi presumido no início desta pesquisa.

5.2 VALIDAÇÃO CRUZADA COM DUAS BASES DE IMAGENS

Antes de iniciar o treinamento do classificador se fez necessário realizar o pré-processamento das imagens. Este pré-processamento consiste em identificar uma face em uma imagem, bem como alinhá-la conforme descrito no item 4.5 da metodologia. A partir disso, obteve-se os dados listados abaixo (Tabela 3) para cada uma das bases.

Tabela 3 - Pós processamento dos dados

	MS-CELEB-1M	FEI DATABASE
Total	566	2800
Alinhadas	549	2627
Diferença	17	173

Fonte: Do autor

Conforme demonstrado em nossos resultados, o algoritmo *dlib OpenFace* não conseguiu identificar algumas imagens. Para o subconjunto da base de dados MS-CELEB-1M não foram identificadas 17 imagens, e para a base de dados FEI DATABASE não foram identificadas 173 imagens. Segundo Braga (2013), as diferentes variações de pose, ruídos, iluminação de ambiente das imagens pode dificultar a identificação facial.

Após o pré-processamento, a RNA é utilizada nas imagens alinhadas, onde foram obtidos dois arquivos no formato “.csv”. O arquivo “**reps.csv**” são as representações das faces, definem-se como um vetor bidimensional, em que x é a quantidade de imagens alinhadas e y é a quantidade de características que a RNA extraiu de cada face. Para o *OpenFace* a saída da RNA é um vetor com 128 posições para cada imagem, e como foi aplicado várias imagens, o resultado é uma matriz, conforme demonstrado na tabela 4. Já o “**labels.csv**” também é um vetor bidimensional, no qual encontra-se o caminho completo de cada imagem utilizada, bem com seu índice.

Tabela 4 - Matriz de representações

	MS-CELEB-1M	FEI DATABASE
Matriz	$\begin{bmatrix} 1 & \dots & 128 \\ \vdots & \ddots & \vdots \\ 549 & \dots & 128 \end{bmatrix}$	$\begin{bmatrix} 1 & \dots & 128 \\ \vdots & \ddots & \vdots \\ 2627 & \dots & 128 \end{bmatrix}$

Fonte: Do autor

A tabela 5 demonstra os resultados da validação cruzada nas bases MS-CELEB-1M e FEI DATABASE, respectivamente, que foram obtidos com o método `cross_val_score`.

Tabela 5 - Resultados da validação cruzada

	MS CELEB-1M	FEI DATABASE
<i>Fold1</i>	1.0	0.9
<i>Fold2</i>	0.97	0.91
<i>Fold3</i>	0.99	0.88
<i>Fold4</i>	0.99	0.89
<i>Fold5</i>	1.0	0.88
Acurácia	0.99%	0.89%

Fonte: Do autor

A MS-CELEB-M1 contém mais exemplos por cada classe e contém menos classes. Os resultados demonstraram uma maior acurácia com 0.99%, ao contrário dos resultados da FEI DATABASE, que mesmo contendo mais classes do que a MS-CELEB-M1 e com menos exemplos em cada classe, obteve-se uma acurácia de 0.89%. Com este resultado compreende-se que, conforme maior o número de exemplos utilizados para o treinamento de um classificador o mesmo predirá com uma maior acurácia.

Maia (2016), cita em seu trabalho o desempenho da RNA em termos da capacidade de generalização, onde está diretamente ligada a fatores como qualidade da base de treino, distribuição das classes e representatividade dos padrões.

Presume-se ainda, que a RNA utilizada pelo *OpenFace* conseguiu gerar com qualidade as representações de cada classe, gerando assim uma boa base de treino para ser utilizado no AM e conseqüentemente, conseguindo uma boa pontuação

no modelo de classificador. Da mesma forma, Delbiaggio (2017) demonstrou que teve êxito em relação aos outros algoritmos de reconhecimento facial, mesmo utilizando poucos exemplos para treinamento. Segundo ainda o mesmo autor, ainda que alcance resultados melhores que os demais testados em seu trabalho, o *OpenFace* encontrou alguns erros, mas que estes erros foram minimizados com dados de treinamento maiores. Sendo assim, fica evidenciado que conforme mais exemplos utilizados no AM para criação de classificadores, melhor será a pontuação deste.

No estudo de Meloni (2009), foi utilizado o SVM para classificar as imagens de sensoriamento remoto por meio da validação cruzada, onde foi utilizado o método *Leave-One-Out*, diferentemente do método deste estudo (*OneVsRest*), o mesmo obteve bons resultados, mesmo que as imagens sejam usadas para diferentes aplicações o comportamento do classificador é excelente.

Já no trabalho de Horta et al. (2011), utilizou-se várias técnicas para classificação de empresas, como Regressão Logística (RL), SVM, MLP e Árvore de Decisão (AD), os autores destacam que os classificadores com melhores resultados foram AD e o SVM. Neste caso, o AD obteve um desempenho melhor que o SVM. Contudo, Santos (2002), em sua dissertação sobre o uso de SVM para aprendizagem e reconhecimento de objetos baseado na aparência, apresentou que o tipo de *Kernel* geralmente não interfere na precisão final do classificador. No qual, os autores Amos, Ludwiczuk e Satyanarayanan (2016) citam em seu trabalho sobre a escolha do *Kernel* Linear para ser utilizado como padrão no *OpenFace*.

Ainda, Santos (2002) realizou outro experimento comparando MLP e SVM, e considerando apenas 10 classes, os resultados da SVM apresentou uma pequena superioridade em relação a MLP, que é uma técnica totalmente viável para problemas de classificação de padrões, em especial para problemas de reconhecimento baseado na aparência, o que corrobora ainda mais com nossos resultados.

Meloni (2009) em seu estudo complementa que a adição de características favorece na evolução dos resultados, criando classificadores com um desempenho maior, e ressalta que as escolhas de suas características colaboraram na classificação das imagens de satélite.

Nossos resultados demonstraram que os dados de exemplos da MS-CELEB-1M obtiveram-se uma acurácia de 99%, já para a FEI DATABASE obteve uma

acurácia de 89%. Para ambas as bases de dados o classificador foi avaliado com a estratégia da validação cruzada, ficando claro que a RNA que o *OpenFace* utiliza para a extração de características de uma face são bons dados de exemplos para a geração de classificadores.

5.3 TESTES ADICIONAS

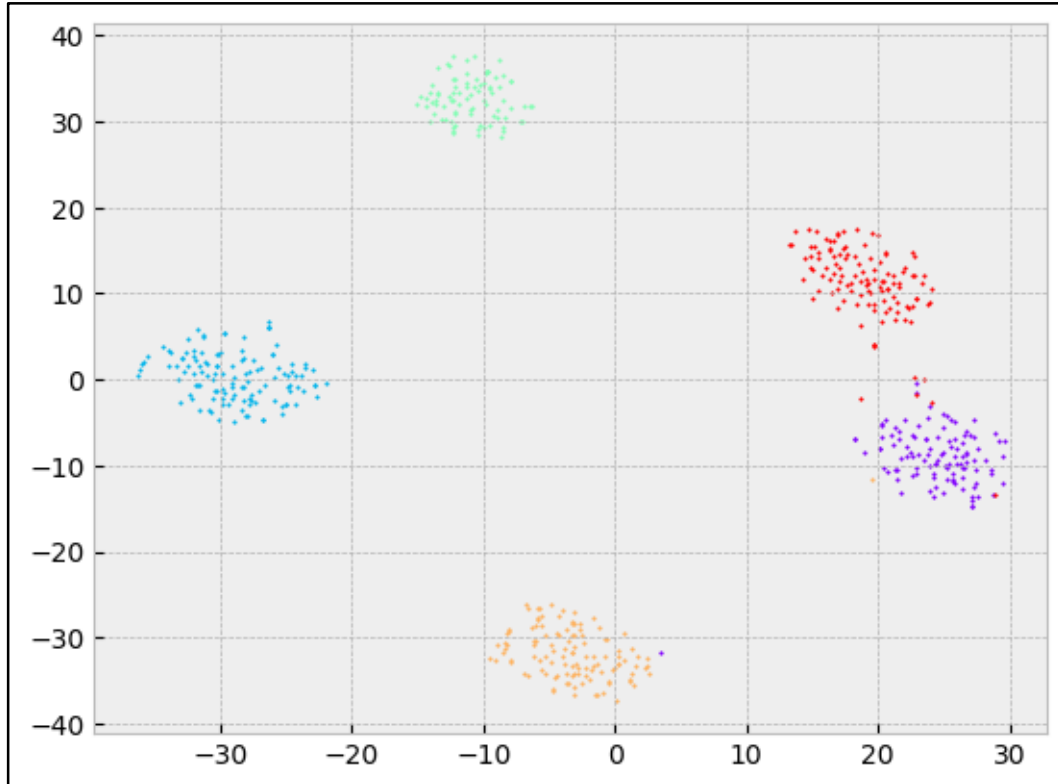
Com o intuito de complementar esta pesquisa, foi gerado ainda dois gráficos correspondentes aos dados das bases MS-CELEB-1M (figura 18) e FEI DATABASE (figura 19), no qual utilizou-se a técnica t-SNE para demonstrar a matriz de 128 dimensões dos dados extraídos pela RNA.

A t-SNE é uma técnica para redução de dimensionalidade que é particularmente adequada para a visualização de conjuntos de dados de alta dimensão. A técnica pode ser implementada via aproximações do Barnes-Hut, permitindo que ela seja aplicada em grandes conjuntos de dados do mundo real (t-SNE, MAATEN; HILTON, 2008). Converte semelhanças entre pontos de dados para probabilidades conjuntas e tenta minimizar a divergência de Kullback-Leibler¹ entre as probabilidades (PEDREGOSA et al., 2011, tradução nossa).

¹ É uma medida de como uma distribuição de probabilidades diverge de outra distribuição de probabilidades.

Os resultados demonstraram na base MS-CELEB-1M (figura 20) que a RNA conseguiu segregar as classes distantes entre si, isso se dá pelo fato de haver menos classes e mais quantidade de exemplos do que a base FEI DATABASE.

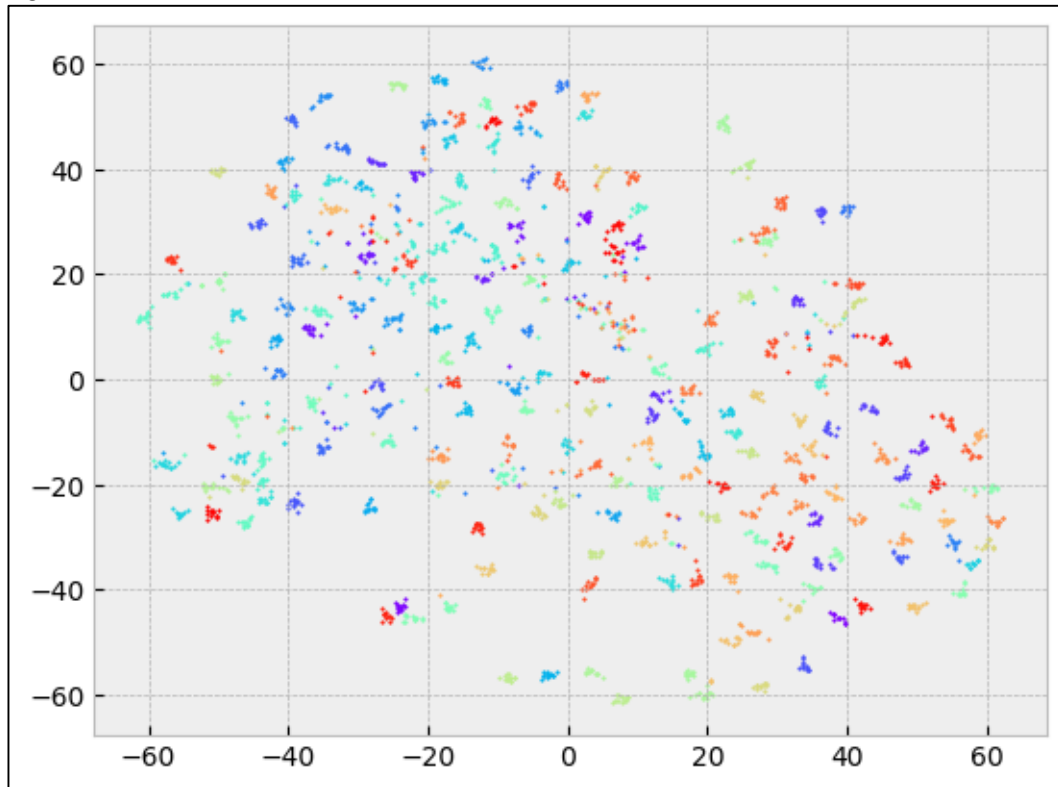
Figura 20 - Resultado t-SNE MS-CELEB-1M



Fonte: Do autor

Já a base FEI DATABASE que contém mais classes e menos exemplos, apresentou uma maior aproximação entre as classes no seu resultado (figura 21). Mesmo assim, é possível distinguir a segregação entre as classes.

Figura 21 - Resultado t-SNE FEI DATABASE

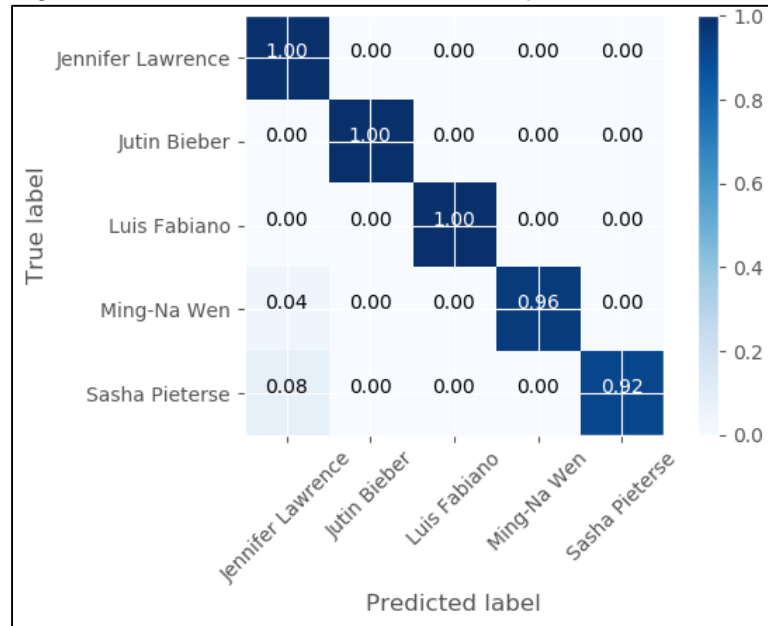


Fonte: Do autor

Para finalizar, foi realizado o teste de matriz de confusão, que demonstra como o classificador criado conseguiu distinguir cada classe. A figura 22, é uma amostragem do que foi obtido a partir de um dos *folds*, da base MS-CELEB-1M, utilizando método `confusion_matrix` no qual está disponível na biblioteca *scikit-learn*. É possível verificar qual classe verdadeira foi predita erroneamente. Onde, a classe real Sasha Pieterse, nosso classificador classificou com uma pontuação de 0.8 como sendo Jennifer Lawrence.

Para a FEI DATABASE a matriz também foi gerada, porém como ela é obtida por meio de classes reais e preditas, e também por conter 200 classes ficou inviável a sua visualização.

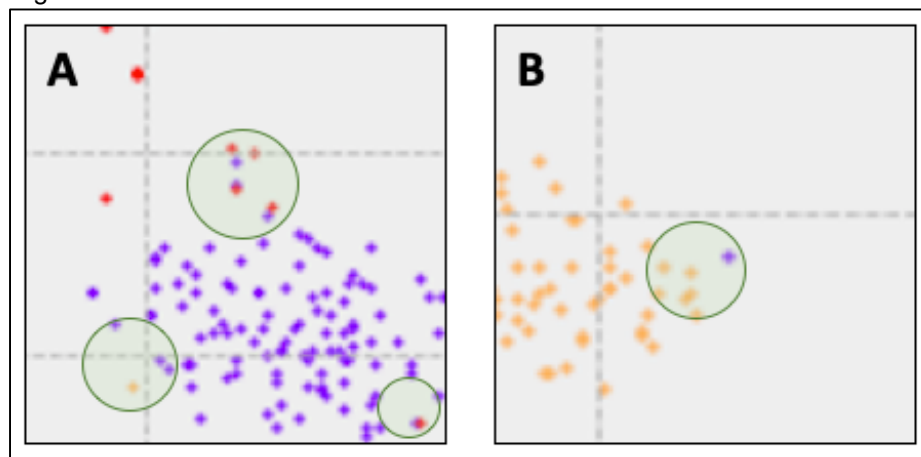
Figura 22 - Amostra da matriz de confusão para cada classe



Fonte: Do autor

A diferença que o classificador encontrou na figura 22, deu-se pelo fato da RNA extrair características semelhantes para classes distintas (figura 23 A e B).

Figura 23 - Características semelhantes entre as classes



Fonte: Do autor

Visto tudo isso, pode-se dizer que os resultados apresentados demonstraram uma acurácia significativa no reconhecimento facial o que corrobora que a biblioteca *OpenFace* é uma opção relevante de estudo para área de reconhecimento facial.

6 CONCLUSÃO

A replicação do teste por meio da LFW na biblioteca *OpenFace* e a utilização de dois bancos de imagens (MS-CELEBE-1M e FEIDATABASE) nos possibilitou compreender melhor como a I.A. é utilizada na área de reconhecimento facial e como ocorre a extração das características faciais, bem como o A.M. para o treinamento e classificação de novos exemplos. Assim, todos os experimentos realizados nesta pesquisa contribuíram para uma compreensão prática da técnica, de reconhecimento facial bem como seu uso.

Ainda a validação cruzada demonstrou uma melhor acurácia na MS-CELEB-1M (0.99%) do que na FEI DATABASE (0.89%), indicando que a utilização de mais exemplos por classe pode interferir no classificador.

De modo geral, a utilização da rede neural pré-treinada do *OpenFace*, para a extração de características, juntamente com um classificador SVM obteve resultados considerados satisfatórios. De modo que, evidentemente mais testes sejam necessários, diante dos resultados apresentados pode-se considerar o *OpenFace* como uma boa opção para o uso do reconhecimento facial.

Como sugestão de continuação para este trabalho de pesquisa propõe-se:

- a) criar uma base de dados com celebridades brasileiras, com o intuito de fomentar o estudo do reconhecimento facial;
- b) realizar um protótipo de sistema de segurança com reconhecimento facial que utiliza biblioteca do *OpenFace*;
- c) comparar o *scikit-learn* e *libsvm* na classificação de faces;
- d) comparar os dados de exemplos com outra biblioteca de A.M., como por exemplo o WEKA.

REFERÊNCIAS

- AGARWAL, M.; AGRAWAL, H.; JAIN, N.; KUMAR, M. Face Recognition Using Principle Component Analysis, Eigenface and Neural Network. **International Conference On Signal Acquisition And Processing**. p. 310-314, 2010.
- AMOS, B.; LUDWICZUK, B.; SATYANARAYANAN, M. **OpenFace: A general-purpose face recognition library with mobile applications**. 2016. 20 f. TCC (Graduação) - School Of Computer Science, Carnegie Mellon University, Pittsburgh, Pa 15213, Eua, 2016.
- AMOS, B.; LUDWICZUK, B.; SATYANARAYANAN, M. **OpenFace Documentation**. 2017. 17 f. CMU School Of Computer Science, Carnegie Mellon University, Pittsburgh, Pa 15213, Eua, 2017.
- BARRETO, J. M. **Introdução às redes neurais artificiais**. In: V Escola Regional de Informática. Sociedade Brasileira de Computação, Regional Sul, Santa Maria, Florianópolis, Maringá, p. 5-10, 2002.
- BARRETO, R. M. **Aprendizagem de Métrica baseada na Distância Euclidiana aplicada ao Reconhecimento de Faces**. 2013, 68f. Dissertação (Mestrado) - Universidade Federal de Pernambuco, Recife – PE, 2013.
- BIANCHI, M. F. de. **Extração de características de imagens de faces humanas através de wavelets, PCA e IMPCA**. 2006. 137f. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos – SP, 2006.
- BOMBARDELLI, F. G. **Estudo sobre Reconhecimento Facial**. 2015. 5 f. Monografia (Especialização) - Curso de Ciência da Computação, Universidade Federal do Paraná, Paraná, 2015.
- BRADSKI, G; KAEHLER, A. OpenCV. **Dr. Dobb's journal of software tools**, v. 3, 2000.
- BRAGA, L. F. Z. **Sistemas de reconhecimento facial**. 2013. 84 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica Com Ênfase em Eletrônica), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013.
- BREY, P. **Containers vs. Virtual Machines (VMs): What's the Difference?** 2018. Disponível em: <<https://blog.netapp.com/blogs/containers-vs-vms/>>. Acesso em: 10 nov. 2018.
- CAMPBELL, C. An introduction to kernel methods. **Studies in Fuzziness and Soft Computing**, v. 66, p. 155-192, 2001. Disponível em: <http://calla.rnet.missouri.edu/cheng_courses/com/Quadratic_programming_svmintro.pdf>. Acesso em: 10 nov. 2018.

CARVALHOF de. **Aprendizado de Máquina**. Minicurso. In. SYMPOSIUM ON KNOWLEDGE DISCOVERY, MINING AND LEARNING, 3., 2015, Pretrópolis, Rio de Janeiro: 2015. Disponível em: <http://www2.ic.uff.br/~kdmile/MachineLearning_Andre.pdf>. Acesso em: 10 nov

COLLOBERT, R; KAVUKCUOGLU, K; FARABET, C.. Torch7: A matlab-like environment for machine learning. In: **BigLearn, NIPS workshop**. 2011.

CHELLAPPA, R.; WILSON, C. I.; SIROHEY, S. Human and machine recognition of faces: a survey. **Proceedings Of The IEEE**. v. 83, n. 5, p. 705-741, 1995.

DELBIAGGIO, N. **A comparison of facial recognition's algorithms**. 2017. 41f. Trabalho de Conclusão de Curso (Graduação em Tecnologia da Informação Empresarial) - Universidade de ciências aplicadas, Finlândia, 2017.

DE SOUTO, M. C. P.; LORENA, A. C.; DELBEM, A. C. B.; CARVALHO, A. C. P. L. F. DE. Técnicas de aprendizado de máquina para problemas de biologia molecular. **Sociedade Brasileira de Computação**, 2003.

DOCKER (Webpage). **Docker overview**. Disponível em: <<https://docs.docker.com/engine/docker-overview/>>. Acesso em: 21 maio. 2018.

GOLDSTEIN, A. J.; HARMON, L. D.; LESK, A. B. Identification of human faces. **Proceedings Of The IEEE**. v. 59, n. 5, p. 748-760, 1971.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. New Jersey: USA, 2006. p.103.

GUO, Y.; ZHANG, L.; HU, Y.; HE, X.; GAO, J. . **European Conference on Computer Vision**. v. 9907, n. 3, p. 87-102, 2016.

HACKELING, G. **Mastering Machine Learning with scikit-learn**. Packt Publishing Ltd, 2017. Não Tá no texto

HAYKIN, S. **Redes Neurais – Princípios e Práticas**. 2a Edição. Tradução: Paulo Martins Engel. Porto Alegre: Bookman. 2001.

HORTA, R. A. M.; BORGES, C. C. H.; CARVALHO, F. A. A. DE.; ALVES, F. J. S.; Previsão de Insolvência: Uma Estratégia para Balanceamento da Base de Dados Utilizando Variáveis Contábeis de Empresas Brasileiras. **Sociedade, Contabilidade e Gestão**. v. 6, n. 2, 2011.

HUANG, G. B.; RAMESH, M.; BERG, T.; LEARNED-MILLER, E. **Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments**. University of Massachusetts, Amherst, Technical Report 07-49, October, 2007. Disponível em: <<http://vis-www.cs.umass.edu/lfw/index.html>>. Acesso: 26 nov. 2017.

KING, D. E. Dlib-ml: A Machine Learning Toolkit. **Journal of Machine Learning Research**. v. 10, p. 1755–1758, 2009.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **ImageNet Classification with Deep Convolutional Neural Networks**. P. 1106–1114. 2012.

LORENA, A. C.; CARVALHO, A. C. P. L. F de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

LOPES, R. G. C.; MARIO, M. C.; SILVA-FILHO, J. I. Estudo para medição de peças através de imagens utilizando Lógica Paraconsistente Anotada. **Science and Technology**. v. 1, n. 1, p. 28-32, 2012.

LOPES, E. C. **Detecção de Faces e Características Faciais**. Relatório Técnico, Rio Grande do Sul. 2016. Disponível em: <<http://www.pucrs.br/facin-prov/wp-content/uploads/sites/19/2016/03/tr045.pdf>>. Acesso em: 28 Nov. 2017.

LUAJIT (WebPage). **LuajIT**. S. d. Disponível em: <<http://luajit.org/luajit.html>>. Acesso em: 12 de Nov. 2018.

MAATEN, L. V. D.; HINTON, G. Visualizing High-Dimensional Data Using t-SNE. **Journal of Machine Learning Research**. v. 9, p. 2579-2605, 2008.

MAIA, H. L. F. **Detecção e reconhecimento facial por meio de aprendizado de máquina**. 2016. 50f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Redes de Comunicação) - Universidade de Brasília, Brasília, 2016.

MARQUES-FILHO, O.; VIEIRA-NETO, H. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. p. 307.

MELONI, R. B. da S. **Classificação de Imagens de Sensoriamento Remoto usando SVM**. 2009. 64 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

MENEZES, G. G. S. **Reconhecimento de faces usando análise de componentes principais e morfologia matemática**. 2009. 43f. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) - Universidade de Pernambuco. 2009

MITCHELL, T. M. **Machine learning**. McGraw-Hill Science. 1997. 414p.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas Inteligentes-Fundamentos e Aplicações**. v. 1, n. 1, p. 32, 2003.

MONTIBELER, D. C. G. C.; FERNANDES, J. C. L. Controle de acesso através dos sistemas biométricos como ferramenta de combate a fraudes. **Periódico Eletrônico da FATEC**. v.1, n. 6, p. 37-48, 2012.

MOREIRA, F. C. **Reconhecimento e Classificação de Padrões de Imagens de Núcleos Linfócitos do Sangue Periférico Humano com a Utilização de Redes Neurais Artificiais**. 2002. 69f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis – SC, 2002.

NAKABAYASHI, L. A. **A contribuição da Inteligência Artificial (IA) na Filosofia da Mente**. 2009. 109f. Dissertação (Mestrado) - Pontifícia Universidade Católica de São Paulo, São Paulo - SP, 2009.

NSTC – National Science and Technology. **Face Recognition**. 2006. Disponível em: <<https://www.hSDL.org/?view&did=463904>>. Acesso em: 20 de jun. 2017.

NUNES, F. T.; SILVA, J. B.; NICOLETE, P. C.; PEREIRA, J.; CRISTIANO, M. A. S. **Um Estudo sobre Técnicas de Biometria Baseadas em Padrões Faciais e sua Utilização na Segurança Pública**. p. 113 -129. In: SPANHOL, F. J.; LUNARDI, G. M.; SOUZA, M. V. de. *Tecnologias da Informação e Comunicação na Segurança Pública e Direitos Humanos*. São Paulo: Blucher, 2016.

NUMPY (WebPage). **NumPy**. S. d. Disponível em: <<http://www.numpy.org/>>. Acesso em: 12 de Nov. 2018.

OLIVEIRA, D. R. **Reconhecimento de faces usando redes neurais e biometria**. 2006. 164f. Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2003.

OMAIA, D. **Um sistema para detecção e reconhecimento de face em vídeo utilizando a transformado cosseno discreta**. 2009. 93f. Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, João Pessoa, 2009.

OSÓRIO, F. BITTENCOURT, J. R **Sistemas Inteligentes baseados em Redes Neurais Artificiais aplicados ao Processamento de Imagens**. In: WORKSHOP DE INTELIGÊNCIA ARTIFICIAL, I, 2000, UNISC – Universidade de Santa Cruz do Sul, 2000, p.30.

PARMAR, D. N.; MEHTA, B. B. Face Recognition Methods & Applications. **International Journal Of Computer Technology & Applications**. v. 4, n. 1, p. 84-86, 2013.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in python. **The Journal of Machine Learning Research**. v. 12, p. 2825–2830, 2011.

PRAZERES, S. G. **Reconhecimento de Faces Utilizando Redes Neurais Autoassociativas**. 2010. 39f. Trabalho de Conclusão de Curso (Graduação em

Engenharia da Computação) - Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, 2010.

PREETH, E. N.; MULERICKAL, F. J. P.; PAUL, B.; SASTRI, Y. **Evaluation of Docker containers based on hardware utilization**. In: International Conference On Control Communication & Computing India (iccc), 2015, India, p.697-700.

REVISTABW – RBW TECNOLOGIA. **Aprendizado de máquina: matriz de confusão**. 2018. Disponível em: <<http://www.revistabw.com.br/revistabw/matriz-de-confusao/>>. Acesso em: 12 de Nov. 2018

RODRIGUES, B. U. **Reconhecimento do tipo de cachaça utilizando visão computacional e reconhecimento de padrões**. 2015. 119 f. Dissertação (Mestrado em Ciência da Computação), Universidade Federal de Goiás, Goiânia, 2015.

SANTANA, L. M. Q.; GOMES, F. R.; SANTOS, T. S. R. Uma análise do processo reconhecimento facial. **Ciências exatas e tecnológicas**. v. 2, n.2, p. 49-58, 2014.

SANTOS, E.M. DOS. **Teoria e Aplicação de Support Vector Machines à Aprendizagem e Reconhecimento de Objetos Baseado na Aparência**. 2002. 111 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Departamento de Informática, Universidade Federal da Paraíba, Campina Grande, 2002. Disponível em: <http://docs.computacao.ufcg.edu.br/posgraduacao/dissertacoes/2002/Dissertacao_EulandaMirandadosSantos.pdf>. Acesso em: 11 nov. 2018.

SARAIVA, F. de O. **Cantor: Ambiente Integrado de Desenvolvimento Voltado à Computação Científica Utilizando Python**. 17p. 2014. Disponível em: <<http://wsl.softwarelivre.org/2014/0004/cantor-ambiente-integrado-de-desenvolvimento-voltado-a-computacao-cientifica-utilizando-python-saraiva-wsl-2014-.pdf>>. Acesso em: 12 de Nov. 2018.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. **FaceNet: A unified embedding for face recognition and clustering**. In: IEEE Conference On Computer Vision And Pattern Recognition. p.815-823, 2015.

SILVA, A. L.; CINTRA, M. E. **Reconhecimento de padrões faciais: Um estudo**. In: Encontro Nacional de Inteligência Artificial e Computacional, 2015, Proceedings ENIAC, p. 224-231, 2015.

SILVA NETO, J. G. **Método de reconhecimento de face baseado em estatísticas de ordem superior**. 2016. 72f. Dissertação (Mestrado em Engenharia de Sistemas e Automação) - Universidade Federal de Lavras, Lavras – MG, 2016.

SIROVICH, L.; KIRBY, M. Low-dimensional procedure for the characterization of human faces. **Journal of the Optical Society America A**. v. 4, p. 519-524, 1987.

SORTE, L. X. B. **Reconhecimento de Faces: Aplicação de algoritmos de Redes Neurais**. In: SEMINÁRIO INTERNO DE COGNIÇÃO ARTIFICIAL, III, 2011, 3., 2011, Campinas. Campinas: Unicamp, 2011. p.5.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern recognition**. San Diego: Elsevier (usa), 2003. p.689.

THOMAZ, C. E. **FEI Face Database**. 2006. Disponível em: <<https://fei.edu.br/~cet/facedatabase.html>>. Acesso em: 14 de Abr. 2018.

TORCH, **Why Torch?** 2018. Disponível em: <<http://torch.ch/>>. Acesso em: 12 nov. 2018.

TRAMONTIN, E. D. **Análise e aplicação de reconhecimento facial em sistema embarcado**. 2016. 75f. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) - Universidade Federal de Santa Catarina, Araranguá, 2016.

TURK, M.; PENTLAND, A. Eigenfaces for Face Detection/Recognition. **Journal of Cognitive Neuroscience**. v. 3, n. 1, p. 71-86, 1991.

UIBO, Z. **Comparison of Face Recognition Neural Networks**. 2016. 23 f. Curso de Ciência da Computação, University Of Tartu Faculty Of Science And Technology, Tartu, 2016. Disponível em: <https://comserv.cs.ut.ee/home/files/Uibo_informaatika_2016.pdf?study=ATILoputoo&reference=81695FBBBEA2972D7AB1BFB1F3A6A0DBD3475C0F>. Acesso em: 22 abr. 2018.

VALLEJO, J. A. S.; NEIRA, J. C. S. **Diseño e implementación de un sistema de seguridad basado en reconocimiento de rostros**. 2005. 162f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Eletrônica e Controle) - Escuela politécnica nacional, Quito, 2005.

VAN ROSSUM, G.; DRAKE JR, F. L. **Python reference manual**. Amsterdam: Centrum voor Wiskunde en Informatica, 1995.

VINODHINI, M; AISHWARYA, C.; SAFRINA, K. J.; RAMYA, K. A comparative analysis of linear discriminant, principal component and evolutionary pursuit for recognition of human faces. **International Journal of Modern Trends in Engineering and Research (IJMTER)**. v. 3, 2016.

ZHAO, W.; CHELLAPPA, R.; PHILLIPS, P. J.; ROSENFELD, A. Face Recognition: A Literature Survey. **ACM Computing Surveys**. v. 35, n. 4, p. 399–458, 2003.

YANG, M. H; KRIEGMAN, D. J.; AHUJA, N. Detecting Faces in Images: A Survey. **IEEE Transactions on pattern analysis and machine intelligence**. v. 24, n. 1, p. 34-58, 2002.

APÉNDICE (S)

APÊNDICE A – ARTIGO

Estudo de caso de um sistema de reconhecimento facial utilizando o *OpenFace* para identificação de faces em bancos de imagens

Lucas Ricardo Orso¹, Paulo João Martins²

¹Acadêmico do curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC – Brasil

²Departamento de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC - Brasil

lucasorso@gmail.com, pjm@unesc.net

Abstract. *Facial recognition has emerged as an effective area in pattern recognition research. In the area of security systems, the identification of criminals using facial recognition is an important use, being an alternative for security, since it is not invasive and provides the biometric information presenting the unique characteristics of each individual. Therefore, we are looking for a better knowledge about facial recognition techniques.*

Resumo. *O reconhecimento facial vem emergindo como uma área efetiva em pesquisas no reconhecimento de padrões. Na área de sistemas de segurança, a identificação de criminosos usando reconhecimento facial caracteriza-se como uma importante utilização, sendo uma alternativa para a segurança, pois esta é pouco invasiva e provê as informações biométricas apresentando as características únicas de cada indivíduo. Por isso, busca-se um maior conhecimento sobre técnicas de reconhecimento facial.*

1. Introdução

As inúmeras aplicações práticas do reconhecimento facial e seus desafios o tornaram uma área de pesquisa popular em visão computacional. Isto é evidenciado por muitos algoritmos de reconhecimento de face desenvolvidos ao longo das últimas três décadas e, nos últimos anos, o aparecimento de um número considerável de sistemas de reconhecimento de face comerciais (SILVA NETO, 2016). Foram realizados estudos mais detalhados ao passar dos últimos anos, devido à necessidade de encontrar meios de reconhecer indivíduos através de suas características (VALLEJO; NEIRA, 2005). Neste sentido, o reconhecimento facial se dá pela possibilidade de identificar pessoas por meio de suas características (TURK; PENTALAND, 1991, tradução nossa).

Para se reconhecer uma face, deve-se primeiro detectá-la, portanto, a primeira etapa de um sistema de reconhecimento facial é a detecção de uma face em uma imagem de entrada. Por isso, a detecção de imagens está diretamente relacionada ao reconhecimento facial, e em seguida é realizada a comparação com os modelos conhecidos pelo sistema (SUNG et al., 1994 apud OLIVEIRA, 2006). A detecção da face é um fator importante, pois é a primeira etapa necessária para qualquer sistema de reconhecimento facial. A maioria das técnicas utilizadas admite uma detecção ideal, conseqüentemente, a capacidade do sistema depende da precisão da etapa de detecção (OMAIA, 2009). Existem vários programas de reconhecimento facial

disponíveis, como por exemplo o *OpenFace*. Ainda, por se tratar de uma biblioteca disponível para todos os interessados em estudá-la, pode-se estudar a sua implementação e verificar o seu funcionamento.

Este estudo tem como objetivo avaliar o classificador SVM utilizado pelo *OpenFace* na classificação das imagens através do método *Cross-Validation*.

2. Metodologia

Foram utilizados três bancos de imagens disponíveis na internet e de acesso gratuito: *Labeled Faces in the Wild* (LFW), MS-CELEB-1M e FEI DATABASE.

O LFW possui cerca de 13 mil imagens diversas, com pessoas em diferentes poses, expressões e iluminação, além do fundo diferente de cada imagem, ou seja, em ambientes não controlado. O MS-CELEB-1M contém 10 milhões de imagens para cerca de 100 mil celebridades, possuindo tamanho superior a 100Gb, devido a isto optou-se por trabalhar com um subconjunto de exemplos disponível para *download*. No qual, deste subconjunto foi escolhido cinco celebridades (Jennifer Lawrence, Justin Bieber, Ming-Na Wen, Luiz Fabiano e Sasha Pieterse), totalizando 566 imagens para o experimento. Assim como a LFW, o MS-CELEB-1M apresenta imagens em ambientes não controlados. E por último o FEI DATABASE, que é um banco de imagens brasileiro que possui 2800 imagens, contendo 200 pessoas no qual cada pessoa tem no total 14 imagens cada. Diferentemente das outras bases, este apresenta um ambiente controlado, com imagens em um fundo branco e todas as fotos são obtidas da mesma forma para todas as pessoas.

Primeiro, reproduziu-se o experimento de teste de acurácia da LFW, para validar os resultados apresentados no artigo “*OpenFace: A general purpose face recognition library with mobile applications*” (AMOS; LUDWICZUK; SATYANARAYANAN, 2016).

Posteriormente, empregou-se o uso das duas bases restantes (MS-CELEB-1M e FEI DATABASE) que possuem diferentes tamanhos, para avaliar o classificador SVM em diferentes cenários e tamanhos de exemplos. Para a utilização destas bases é necessária uma estrutura de diretórios, onde as imagens de cada indivíduo estejam separadas em diretórios diferentes. Diferentemente da LFW e MS-CELEB-1M, a FEI DATABASE não possui uma estrutura de pastas para armazenamento de imagens para cada pessoa. Para isso, foi elaborado um simples algoritmo (figura 1), para que a separação não fosse realizada de forma manual. Então, para cada indivíduo foi criado uma pasta, resultando na separação das imagens.

```

1  import os
2  import shutil
3  import sys
4
5  directoryName = sys.argv[1]
6  directoryPath = os.getenv('PWD')
7  directoryPath = os.path.join(directoryPath, directoryName)
8  os.chdir(directoryPath)
9
10 print os.getcwd()
11
12 list_peoples = []
13
14 for file_found in os.listdir(os.getcwd()):
15     list_peoples.append(file_found)
16     list_peoples.sort(key=lambda f: int(filter(str.isdigit, f) or -1))
17
18 print('All files load to memory')
19
20 n = 1
21 count = 1
22 name = "Pessoa_%i" % n
23 for people in list_peoples:
24     print(name, people)
25     # if n == 42:
26     #     break
27     # n += 1
28     if not os.path.exists(name):
29         os.mkdir(name)
30     shutil.move(people, name)
31     if len(os.listdir(name)) == 14:
32         n += 1
33         name = "Pessoa_%i" % n
34         count += 1
35
36 print(str.format('total %i' % count))

```

Figura 1. Algoritmo para criar subpastas na FEI DATABASE.

Segundo Amos, Ludwiczuk e Satyanarayanan (2016), a maneira mais fácil de executar o *OpenFace* é utilizando a imagem do ambiente para Docker (disponibilizado pelos criadores), onde contém todas as bibliotecas e *frameworks* para o funcionamento correto do *OpenFace*. Foi escolhido utilizar a imagem do Docker, a fim de evitar problemas de compatibilidade com o sistema operacional, pois a instalação manual de todas as bibliotecas e *frameworks*, além de algumas como *OpenCV*, necessita realizar a compilação, podendo causar interrupções e demora no experimento.

Dessa forma, foi então efetuado o *download* da imagem do *OpenFace* e posteriormente instalado o Docker. Com o Docker em execução e a imagem do container do *OpenFace* em funcionamento, inicia-se o procedimento de pré-processamento das imagens, no qual foi utilizado o detector de pontos de referência dlib (King, 2009) para centralizar e rotacionar a face na imagem (com base no nariz e olhos) (figura 2). A partir deste momento, as imagens foram cortadas com base nos pontos de referência externo e redimensionadas em 96x96 para aparecerem em locais semelhantes na entrada da rede neural.

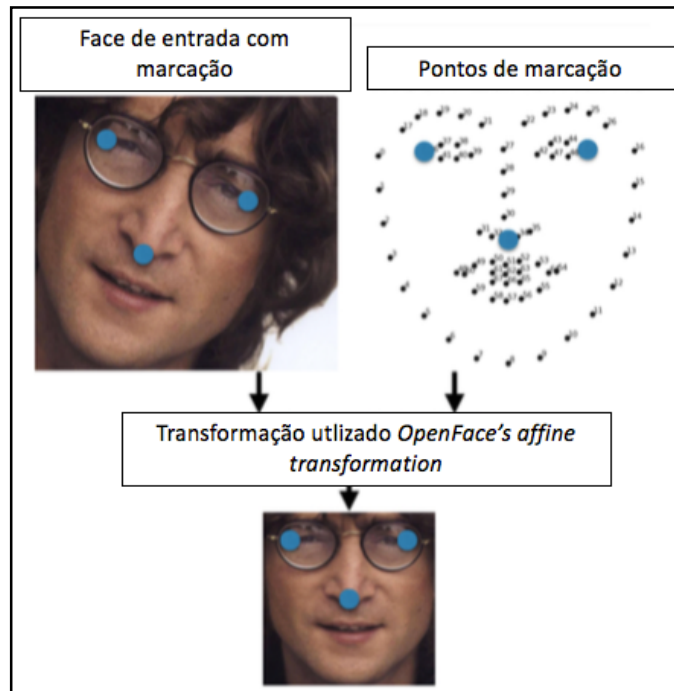


Figura 2. Exemplo de pré-processamento.

A rede neural escolhida foi “nn4.small2.v1”, onde usa o alinhamento que consiste em três pontos principais: a parte externa dos olhos e o nariz, totalizando 68 pontos de referência em uma face. Ainda, “nn4.small2.v1” é pré-treinada e foi escolhida com base no desempenho apresentado no desenvolvimento no artigo do *OpenFace*.

A tabela 1 demonstra os resultados de desempenho das RNAs do *OpenFace*, utilizando um CPU de 8 núcleos de 3.70 GHz e uma GPU Tesla K40 (AMOS; LUDWICZUK; SATYANARAYANAN, 2016, tradução nossa).

Tabela 1. Performance da rede neural.

Modelo	Tempo de Execução (CPU)	Tempo de Execução (GPU)
nn4.v1	75.67 ms ± 19.97 ms	21.96 ms ± 6.71 ms
nn4.v2	82.74 ms ± 19.96 ms	20.82 ms ± 6.03 ms
nn4.small1.v1	69.58 ms ± 16.17 ms	15.90 ms ± 5.18 ms
nn4.small2.v1	58.9 ms ± 15.36 ms	13.72 ms ± 4.64 ms

Finalizado o pré-processamento das imagens, as mesmas foram utilizadas na RNA para extração e representações faciais. Ao final desta etapa, os dados extraídos foram armazenados em arquivos “.csv”.

Neste trabalho, o *Docker* foi necessário para executar a RNA do *OpenFace* à qual é responsável por extrair as características das faces, já as demais etapas do AM foram executadas diretamente no sistema operacional, sendo necessária a instalação de algumas bibliotecas como *scikit-learn*, *numpy*, *matplotlib*.

Para criação e avaliação do SVM foram usados métodos disponíveis na biblioteca *scikit-learn*. Onde, por meio do `cross_val_score` juntamente com o *K-fold*, dividiu-se o conjunto de dados em um subconjunto de treinamento e teste, utilizando $k=5$. Ainda, devido ao fato de existir mais de uma classe usou-se o padrão *StratifiedKFold*, onde os *folds* foram feitas preservando a porcentagem de amostras para cada classe.

A figura 3 mostra o código fonte implementado, onde utiliza os arquivos “.csv” gerado pela RNA e que consiste nas representações faciais e suas respectivas classes. Após a criação do SVM com *kernel* é utilizado o método `cross_val_score`, passando por parâmetro, classificador criado, dados de exemplos, classes e a quantidade de *folds*.

```

35 def crossVal(args):
36
37     fname = "{}/labels.csv".format(args.workDir)
38     labels = pd.read_csv(fname, header=None).as_matrix()[:, 1]
39     labels = [os.path.split(os.path.dirname(label))[1] for label in labels]
40     le = LabelEncoder()
41     le.fit(labels)
42     labelsNum = le.transform(labels)
43     nClasses = len(le.classes_)
44
45     fname = "{}/reps.csv".format(args.workDir)
46     embeddings = pd.read_csv(fname, header=None).as_matrix()
47
48     X, y = embeddings, labelsNum
49
50     ln_clf = SVC(C=1, kernel='linear', probability=True)
51     clf = OneVsRestClassifier(ln_clf)
52
53     print("\nObtaining results cross-validation")
54     t_start = time.clock()
55     scores = cross_val_score(clf, X, y, cv=5, n_jobs=2)
56     t_end = time.clock()
57     print("Acuracy: %0.4f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
58     print('Cross validation time: {:0.4f}s.'.format(t_end-t_start))
59     print("Scores:{}".format(scores))
60     print("=====")

```

Figura 3. Demonstração do código fonte.

Para que seja executado este método, a linha de comando demonstrada na figura 4 precisou ser executada. Neste comando consta a versão do *Python*, o nome do arquivo e o nome do método desenvolvido. Este último, é necessário informar como parâmetro o caminho onde se encontra os arquivos “.csv”.

```
[Lucass-MacBook-Pro:evaluation lucasoroso$ python3 ROC_CROSS.py crossVal $(pwd)
```

Figura 4. Comando para executar o algoritmo.

Assim, ao executar o algoritmo desenvolvido é retornado um vetor de pontuações do classificador para cada execução da validação cruzada.

3. Resultados e Discussão

3.1 Reprodução do experimento de acurácia na base LFW

Nesse experimento, os resultados (figura 5) coincidiram com os mesmos resultados apresentado por Amos, Ludwiczuk e Satyanarayanan (2016), onde demonstram uma alta precisão na identificação de faces, quando utilizando a base de dados LFW.

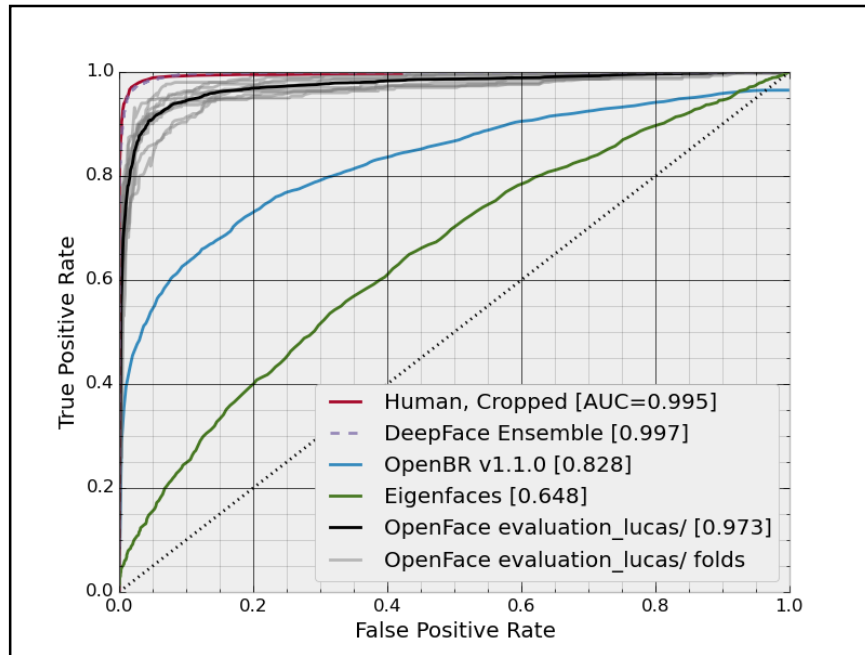


Figura 5. Reprodução do experimento.

Com esses resultados é possível afirmar que a construção do experimento é replicada, obtendo-se os mesmos resultados e acurácia dos resultados dos autores Amos, Ludwiczuk e Satyanarayanan (2016).

A biblioteca do *OpenFace* contém o algoritmo também em *Python*, responsável por realizar o experimento na base da LFW, porém este algoritmo é somente para a base LFW, já que ela contém suas regras para a realização do experimento. Sendo assim, não foi possível testar e plotar a curva ROC utilizando o mesmo algoritmo em outros bancos de imagens como foi presumido no início desta pesquisa.

3.2 Validação cruzada com duas bases de imagens

Antes de iniciar o treinamento do classificador se fez necessário realizar o pré-processamento das imagens. Este pré-processamento consiste em identificar uma face em uma imagem, bem como alinhá-la e a partir disso, obteve-se os dados para cada uma das bases. (Tabela 2).

Tabela 2. Pós processamento dos dados.

	MS-CELEB-1M	FEI DATABASE
Total	566	2800
Alinhadas	549	2627
Diferença	17	173

Conforme demonstrado em nossos resultados, o algoritmo dlib *OpenFace* não conseguiu identificar algumas imagens. Para o subconjunto da base de dados MS-CELEB-1M não foram identificadas 17 imagens, e para a base de dados FEI DATABASE não foram identificadas 173 imagens. Segundo Braga (2013), as diferentes variações de pose, ruídos, iluminação de ambiente das imagens pode dificultar a identificação facial. Após o pré-processamento, a RNA é utilizada nas imagens alinhadas, onde foram obtidos dois arquivos no formato “.csv”. O arquivo “**reps.csv**” são as representações das faces, definem-se como um vetor bidimensional, em que x é a quantidade de imagens alinhadas e y é a quantidade de características que a RNA extraiu de cada face. Para o *OpenFace* a saída da RNA é um vetor com 128 posições de cada imagem, e como foi aplicado várias imagens, o resultado é uma matriz, conforme demonstrado na tabela 3. Já o “**labels.csv**” também é um vetor bidimensional, no qual encontra-se o caminho completo de cada imagem utilizada, bem com seu índice.

Tabela 3 - Matriz de representações

	MS-CELEB-1M	FEI DATABASE
Matriz	$\begin{bmatrix} 1 & \dots & 128 \\ \vdots & \ddots & \vdots \\ 549 & \dots & 128 \end{bmatrix}$	$\begin{bmatrix} 1 & \dots & 128 \\ \vdots & \ddots & \vdots \\ 2627 & \dots & 128 \end{bmatrix}$

A tabela 4 demonstra os resultados da validação cruzada nas bases MS-CELEB-1M e FEI DATABASE, respectivamente.

Tabela 4 - Resultados da validação cruzada.

	MS CELEB-1M	FEI DATABASE
<i>Fold1</i>	1.0	0.9
<i>Fold2</i>	0.97	0.91
<i>Fold3</i>	0.99	0.88
<i>Fold4</i>	0.99	0.89
<i>Fold5</i>	1.0	0.88
Acurácia	0.99%	0.89%

A MS-CELEB-M1 contém mais exemplos por cada classe e contém menos classes. Os resultados demonstraram uma maior acurácia com 0.99%, ao contrário dos resultados da FEI DATABASE, que mesmo contendo mais classes do que a MS-CELEB-M1 e com menos exemplos em cada classe, obteve-se uma acurácia de 0.89%.

Com este resultado compreende-se que, conforme maior o número de exemplos utilizados para o treinamento de um classificador o mesmo predirá com uma maior acurácia.

Maia (2016), cita em seu trabalho o desempenho da RNA em termos da capacidade de generalização, onde está diretamente ligada a fatores como qualidade da base de treino, distribuição das classes e representatividade dos padrões.

Presume-se ainda, que a RNA utilizada pelo *OpenFace* conseguiu gerar com qualidade as representações de cada classe, gerando assim uma boa base de treino para ser utilizado no AM e conseqüentemente, conseguindo uma boa pontuação no modelo de classificador. Da mesma forma, Delbiaggio (2017) demonstrou que teve êxito em relação aos outros algoritmos de reconhecimento facial, mesmo utilizando poucos exemplos para treinamento. Segundo ainda o mesmo autor, ainda que alcance resultados melhores que os demais testados em seu trabalho, o *OpenFace* encontrou alguns erros, mas que estes erros foram minimizados com dados de treinamento maiores. Sendo assim, fica evidenciado que conforme mais exemplos utilizados no AM para criação de classificadores, melhor será a pontuação deste.

No estudo de Meloni (2009), foi utilizado o SVM para classificar as imagens de sensoriamento remoto por meio da validação cruzada, onde foi utilizado o método Leave-One-Out, diferentemente do método deste estudo (OneVsRest), o mesmo obteve bons resultados, mesmo que as imagens sejam usadas para diferentes aplicações o comportamento do classificador é excelente.

Já no trabalho de Horta et al. (2011), utilizou-se várias técnicas para classificação de empresas, como Regressão Logística (RL), SVM, MLP e Árvore de Decisão (AD), os autores destacam que os classificadores com melhores resultados foram AD e o SVM. Neste caso, o AD obteve um desempenho melhor que o SVM. Contudo, Santos (2002), em sua dissertação sobre o uso de SVM para aprendizagem e reconhecimento de objetos baseado na aparência, apresentou que o tipo de Kernel geralmente não interfere na precisão final do classificador. No qual, os autores Amos, Ludwiczuk e Satyanarayanan (2016), citam em seu trabalho sobre a escolha do Kernel Linear para ser utilizado como padrão no *OpenFace*.

Ainda, Santos (2002) realizou outro experimento comparando MLP e SVM, e considerando apenas 10 classes, os resultados da SVM apresentou uma pequena superioridade em relação a MLP, que é uma técnica totalmente viável para problemas de classificação de padrões, em especial para problemas de reconhecimento baseado na aparência, o que corrobora ainda mais com nossos resultados.

Meloni (2009) em seu estudo complementa que a adição de características favorece na evolução dos resultados, criando classificadores com um desempenho maior, e ressalta que as escolhas de suas características colaboraram na classificação das imagens de satélite.

Nossos resultados demonstraram que os dados de exemplos da MS-CELEB-1M obtiveram-se uma acurácia de 99%, já para a FEI DATABASE obteve uma acurácia de 89%. Para ambas as bases de dados o classificador foi avaliado com a estratégia da validação cruzada, ficando claro que a RNA que o *OpenFace* utiliza para a extração de características de uma face são bons dados de exemplos para a geração de classificadores.

Visto tudo isso, pode-se dizer que os resultados apresentados demonstraram uma acurácia significativa no reconhecimento facial o que corrobora que a biblioteca *OpenFace* é uma opção relevante de estudo para área de reconhecimento facial.

4. Conclusão

A replicação do teste por meio da LFW na biblioteca *OpenFace* e a utilização de dois bancos de imagens (MS-CELEBE-1M e FEIDATABASE) nos possibilitou compreender

melhor como a I.A. é utilizada na área de reconhecimento facial e como ocorre a extração das características faciais, bem como o AM. para o treinamento e classificação de novos exemplos. Assim, todos os experimentos realizados nesta pesquisa contribuíram para uma compreensão prática da técnica, de reconhecimento facial bem como seu uso.

Ainda a validação cruzada demonstrou uma melhor acurácia na MS-CELEB-1M (0.99%) do que na FEI DATABASE (0.89%), indicando que a utilização de mais exemplos por classe pode interferir no classificador.

De modo geral, a utilização da rede neural pré-treinada do OpenFace, para a extração de características, juntamente com um classificador SVM linear obteve, resultados de acurácia do classificador, considerados satisfatórios. Ainda que, evidentemente mais testes sejam necessários, diante dos resultados apresentados pode-se considerar o *OpenFace* como uma boa opção para o uso do reconhecimento facial.

Referências

- Amos, B.; Ludwiczuk, B.; Satyanarayanan, M. OpenFace: A general-purpose face recognition library with mobile applications. 2016. 20 f. TCC (Graduação) - School Of Computer Science, Carnegie Mellon University, Pittsburgh, Pa 15213, Eua, 2016.
- Braga, L. F. Z. Sistemas de reconhecimento facial. 2013. 84 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica Com Ênfase em Eletrônica), Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2013.
- Delbiaggio, N. A comparison of facial recognition's algorithms. 2017. 41f. Trabalho de Conclusão de Curso (Graduação em Tecnologia da Informação Empresarial) - Universidade de ciências aplicadas, Finlândia, 2017.
- Horta, R. A. M.; Borges, C. C. H.; Carvalho, F. A. A. DE.; Alves, F. J. S.; Previsão de Insolvência: Uma Estratégia para Balanceamento da Base de Dados. Utilizando Variáveis Contábeis de Empresas Brasileiras. Sociedade, Contabilidade e Gestão. v. 6, n. 2, 2011.
- King, D. E. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research. v. 10, p. 1755–1758, 2009.
- Maia, H. L. F. Detecção e reconhecimento facial por meio de aprendizado de máquina. 2016. 50f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Redes de Comunicação) - Universidade de Brasília, Brasília, 2016.
- Meloni, R. B. da S. Classificação de Imagens de Sensoriamento Remoto usando SVM. 2009. 64 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.
- Oliveira, D. R. Reconhecimento de faces usando redes neurais e biometria. 2006. 164f. Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2003.
- Omaia, D. Um sistema para detecção e reconhecimento de face em vídeo utilizando a transformado cosseno discreta. 2009. 93f. Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, João Pessoa, 2009.
- Santos, E.M. DOS. Teoria e Aplicação de Support Vector Machines à Aprendizagem e Reconhecimento de Objetos Baseado na Aparência. 2002. 111 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Departamento de Informática, Universidade Federal da

Paraíba, Campina Grande, 2002. Disponível em: <http://docs.computacao.ufcg.edu.br/posgraduacao/dissertacoes/2002/Dissertacao_EulandaMirandadosSantos.pdf>. Acesso em: 11 nov. 2018.

Silva Neto, J. G. Método de reconhecimento de face baseado em estatísticas de ordem superior. 2016. 72f. Dissertação (Mestrado em Engenharia de Sistemas e Automação) - Universidade Federal de Lavras, Lavras – MG, 2016.

Vallejo, J. A. S.; Neira, J. C. S. Diseño e implementación de un sistema de seguridad basado en reconocimiento de rostros. 2005. 162f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Eletrônica e Controle) - Escuela politécnica nacional, Quito, 2005.