Método para Reidentificação Facial

Yves Augusto Lima Romero [*] **Ajalmar Rêgo da Rocha Neto** [**]

[*] *Departamento de Telemática, Instituto Federal do do Ceará, CE, (e-mail: yvesromero1998@gmail.com).*
[**] *Departamento de Telemática, Instituto Federal do Ceará, CE, (e-mail: ajalmar@gmail.com)*

Abstract: Recenlty, many face reid strategies have been developped. These strategies create vectorial spaces capable of representing data on reduced dimensions. Such representations are produced by deep learning models that learn to maximize the intra-class similarity and minimize the inter-class similarity. The method described in this article proposes a new face reid strategy based on the Facenet model. It consists of training support vector machines on the euclidean distances calculated between the embeddings of the Facenet model, notably reducing the false positive rate.

Resumo: Ultimamente se tem desenvolvido diversas estratégias para identificar se dois rostos foram capturados do mesmo indivíduo. Para tanto, é preciso criar espaços vetoriais capazes de representar os dados em dimensões reduzidas. Essas representações são geradas por modelos de aprendizado profundo, que aprendem, durante o treinamento, a maximizar a similaridade entre amostras de uma mesma classe, e diminui-la entre amostras de classes distintas. O método descrito neste trabalho utiliza o modelo treinado no artigo da Facenet para propor uma nova estratégia de reidentificação facial, que consiste em treinar máquinas de vetores suporte a partir de distâncias euclidanas calculadas entre os *embeddings* extraídos pelo modelo da Facenet, reduzindo significativamente a taxa de falsos positivos.

Keywords: deep learning; convolutional neural networks; person reid; one-shot learning; triplet loss; machine learning; biometric systems.

Palavras-chaves: aprendizado profundo; redes neurais convolucionais; reidentificação de indivíduos; aprendizado *one-shot*; *triplet loss*; aprendizado de máquina; sistemas biométricos.

1. INTRODUÇÃO

Nos últimos anos se tem visto uma profusão de novas tecnologias voltadas para sistemas biométricos. O uso desses sistemas tornou-se bastante comum, transformou-se num elemento do cotidiano. O uso da biometria para identificar seres humanos de maneira automática se popularizou sobretudo nos anos 90, época em que havia grandes discussões sobre a confiabilidade dos sistemas biométricos, Cole (1999), Cohen (1994). Pode-se capturar a biometria de um indivíduo de diversas maneiras, por exemplo, extraindo informações da face, das impressões digitais, ou mesmo da voz. Em virtude da pandemia do COVID-19, V.M. Corman (2020), Zumla (2020), os sistemas de biometria facial passaram a ter mais preferência, pelo fato de a captura das imagens não exigir contato físico com um dispositivo, mas poder ser efetuada a uma certa distância. Em todos os tipos de biometria, faz-se uma comparação entre informações, para verificar se foram extraídas do mesmo indivíduo. No caso do reconhecimento facial, as informações extraídas de capturas do rosto de um mesmo indivíduo terão um elevado grau de similaridade entre si. E seu grau de similaridade com as de outros indivíduos será menor. Idealmente, essas características extraídas devem refletir o padrão dos dados de modo a minimizar a semelhança entre as amostras de classes(indivíduos) diferentes, e maximizar essa semelhança para amostras pertecentes à mesma classe. A depender da estratégia escolhida para obter tais informações, ou seja, para extrair as características da imagem, essa divisão dos dados pode ser alcançada com maior ou menor sucesso.

No reconhecimento facial, existem basicamente duas abordagens principais, a *one versus one*, ou reidentificação, e a *one versus all*, ou classificação. No primeiro caso, busca-se identificar se duas imagens pertencem ao mesmo indivíduo, computando suas características, e efetuando uma comparação por meio de critérios de similaridade. Já o *one versus all* é utilizado quando se deseja pesquisar, numa série de indivíduos cadastrados, qual deles mais se assemelha com uma determinada imagem. O *one versus all* é um problema de classificação, que consiste em apontar a classe cujos padrões mais se aproximam do padrão recebido. Geralmente, quando se utiliza esta abordagem, implementa-se, logo a seguir, uma etapa de reidentificação, para validar os resultados da classificação. Pois a classificação não garante que o indivíduo apontado é o correto, mas apenas que, dentre os indivíduos cadastrados, é o que obteve maior pontuação de similaridade. Disso decorre a necessidade de um estágio de reidentificação de face, que é acoplado ao final do processo, conferindo maior segurança ao sistema.

No intuito de produzir técnicas capazes de extrair informações relevantes de imagens, desenvolveram-se diversos métodos, como o *Padrão Binário Local*, Ojala et al. (2002),

o *Histograma de Gradientes Orientados*, Dalal and Triggs (2005), e o Eigenfaces, que utiliza autovetores da matriz de covariância dos dados, seguindo a mesma linha do algoritmo PCA, Turk and Pentland (1991). As informações obtidas por essas técnicas podem ser utilizadas para comparar faces humanas, Ahonen et al. (2006). O traço mais marcante dos extratores de características é abstrair informações dos dados, gerando um espaço vetorial de dimensões reduzidas onde a comparação é efetuada de maneira mais eficaz, tanto em tempo de execução, como em assertividade.

Entretanto, com o advento dos algoritmos de aprendizado profundo, como as redes neurais convolucionais, Lecun (1998), houve uma mudança de foco. Pesquisadores começaram a formular metodologias para resolver o problema da reidentificação facial por meio de redes neurais. A grande vantagem das redes neurais, além de sua grande capacidade de representar os dados, é que essas estruturas oferecem uma solução *end-to-end*, sem que seja necessário quebrar o problema em etapas de extração e comparação. A rede neural já realiza estes dois processos de maneira otimizada, mediante a correção automática dos parâmetros de suas matrizes de peso. As redes convolucionais têm um papel muito importante na extração de características de imagens. Mediante aplicação de filtros ao longo dos pixels da imagem, as *CNN's* geram mapas de características, que representam padrões em diversos graus de abstração.

Utilizando essas características, é possível treinar classificadores, de diversas modalidades, a fim de categorizar os dados. Tendo isso vista, elaborou-se o método de redes neurais siamesas, Koch et al. (2015), cujo treinamento é aplicado a pares de imagens — positivos e negativos —, resultando numa previsão binária, onde o valor 1 representa que as duas imagens do par fornecido pertencem ao mesmo indivíduo, e o valor 0 representa o contrário.

Mais adiante, a ideia das redes siamesas tornou-se mais sofisticada com a publicação da Facenet, Schroff et al. (2015), uma metodologia que já não utiliza mais um par, mas uma tripla de imagens, duas das quais foram capturadas do mesmo indivíduo(a imagem âncora e a imagem positiva), sendo a outra um padrão negativo, retirado de outro indivíduo. Por meio do treinamento, a rede neural aprende a produzir um espaço vetorial que maximiza a distância euclidiana entre padrões de indivíduos diferentes(ou seja, minimiza a similaridade) e minimiza a distância euclidiana entre padrões pertencentes a um mesmo indivíduo(ou seja, maximiza a similaridade). A função de custo que a rede neural procura otimizar é a *triplet loss*, que consiste na diferença entre a distância do âncora para a imagem positiva e a distância do âncora para a imagem negativa. Durante o treinamento dessa rede, as triplas são formadas de acordo com um critério de similaridade. Privilegia-se as triplas mais difíceis, nas quais a distância entre o âncora e o negativo é menor ou muito próxima da distância entre o âncora e o positivo. Esse processo se chama *Hard Triplet Mining*, que consiste em pesquisar triplas na base de dados que satisfaçam tal critério, Hermans et al. (2017). Essa inovação abriu espaço para uma série de esforços no intuito de produzir modelos neurais capazes de efetuar a reidentificação em quaisquer imagens, por exemplo, o modelo da Openface, Amos et al. (2016b). Na Figura 1, deseja-se aferir se a imagem âncora pertence

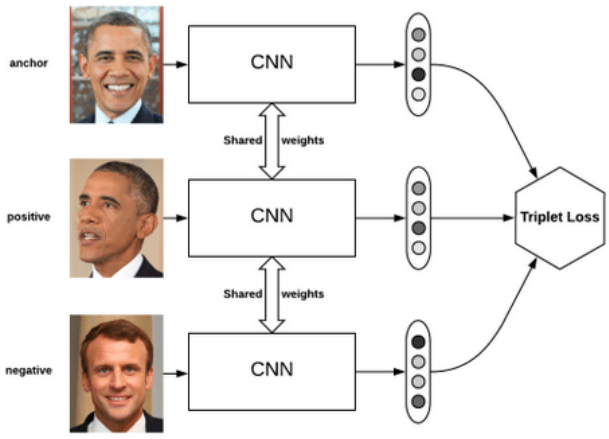


Figura 1. *Triplet loss*: Imagens âncora, positiva e negativa. Fonte da imagem: Olivier Moindrot (2018)

Figura 2. *Triplet loss*: após o aprendizado, a representação dos dados maximiza a distância entre o âncora e o negativo, e a minimiza entre o âncora e o positivo. Imagem retirada do artigo da Facenet

ao mesmo indivíduo retratado na imagem positiva, ou seja, Barack Obama. Para isso, é preciso introduzir as imagens à camada de entrada de uma rede convolucional, a fim de produzir os *embeddings* dessas imagens, ou seja, vetores de características, os quais serão comparados entre si.

Na Figura 2, tem-se uma ilustração do que acontece no processo de aprendizado da rede neural treinada a partir de triplas de imagens. Vê-se que, no início do treinamento, a distância euclidiana entre as amostras âncora e negativa é pequena, e grande entre as amostras âncora e positiva. Ao final da etapa, essa relação se inverte, e a amostra âncora fica mais distante da negativa, e mais próxima da positiva, no espaço de características da rede.

1.1 Trabalhos relacionados

Recentemente, diversos esforços foram feitos na direção de criar métodos para reconhecimento facial, com foco na *reidentificação*. Os estudos mais conhecidos se concentram na geração de *embeddings* por meio de algoritmos de aprendizado profundo, como a Openface Amos et al. (2016a), a VGG Face Parkhi et al. (2015), o modelo da Dlib King (2009), e a ArcFace Deng et al. (2019). Outros métodos, como em Wu (2015), não fazem *embedding* das características da imagens, o que dificulta sua aplicação em tarefas de classificação e clusterização. Atualmente, a técnica que representa o estado da arte é a *Facenet*.

1.2 Proposta

Neste artigo se fará uma proposição de método para reidentificação facial, combinando as técnicas de aprendizado profundo com classificadores de margem mínina. Para este

fim, a rede neural treinada no artigo da *Facenet* será reutilizada para extração de características e, a seguir, será colocada mais uma etapa, proposta por este trabalho na seção de metodologia, que consiste em utilizar os vetores, ou seja, os *embeddings* extraídos com a rede neural da *Facenet*, e calcular distâncias euclidianas entre essas vetores para, em seguida, treinar máquinas de vetores suporte com os valores de distância obtidos. Foi gerada uma base de dados por meio da qual se comparou os métodos, para avaliar o desempenho do método aqui proposto. Este documento está organizando do seguinte modo: na segunda seção está descrita a metodologia usada, na terceira seção constam os resultados e tabelas, e na quarta, por fim, está a conclusão.

2. METODOLOGIA

2.1 Base de dados

Para desenvolver o método proposto neste artigo foi criada uma base de dados de rostos humanos, com um total de 3021 imagens, pertencentes a 258 indivíduos famosos, obtidas no Google Images mediante *web scraping*, detecção facial e posterior limpeza dos dados(para remover imagens repetidas e inconsistentes), alinhamento de face e obtenção da região de interesse.

Pesquisando os nomes dos famosos no Google Images por meio de um código python, foram armazenadas as imagens de cada um dos indivíduos em pastas separadas, cada uma com o nome do indivíduo. A seguir, aplicou-se a detecção facial a cada uma das imagens, extraindo o recorte dos rostos, e descartando as demais regiões das imagens. O modelo utilizado para a detecção facial foi uma rede neural treinada na estratégia de *Single Shot Detection*, Liu et al. (2016), por meio do framework *Caffe*. Logo a seguir, aplicou-se a detecção de *landmarks*, para obter a localização de partes do rosto, como olhos, nariz, sobrancelha e boca, a fim de aplicar o alinhamento às faces encontradas e, depois, efetuar novos recortes, preservando somente a região de interesse do rosto, excluindo cabelos, orelhas, etc. Segue uma ilustração na Figura 3. O modelo utilizado para a detecção de *landmarks* foi encontrado nos arquivos da biblioteca *dlib*, King (2009). Esta base de dados foi disponibilizada na *web* [1].

2.2 Arquitetura da rede

Primeiramente, utilizou-se uma rede neural profunda, construída de acordo com a arquitetura Inception-ResNet-V1, Szegedy et al. (2017), treinada na base de dados LFW, Huang et al. (2008). Esta arquitetura tem por finalidade combinar as vantagens das arquiteturas Inception com a redução de complexidade propiciada pelas conexões residuais da arquitetura ResNet, He et al. (2016). Como ilustrado na Figura 4, as primeiras conexões aplicam operações de convolução com filtros unitários(1x1), Szegedy et al. (2015). Isso é feito com a finalidade de reduzir o número de filtros das camadas seguintes, sem que seja necessário criar camadas adicionais para transformar a dimensão dos dados. Efetua-se, simplesmente, a projeção dos filtros da camada anterior nas dimensões da seguinte, resultando

[1] https://www.kaggle.com/datasets/yveslr/open-famous-people-faces

(a) Glenn Close (b) Abigail Breslin

(c) Cuba Gooding Jr. (d) Gerard Butler

Figura 3. Imagens recortadas com a exclusão de partes irrelevantes do rosto. Centraliza-se os elementos significativos para efetuar o reconhecimento facial.

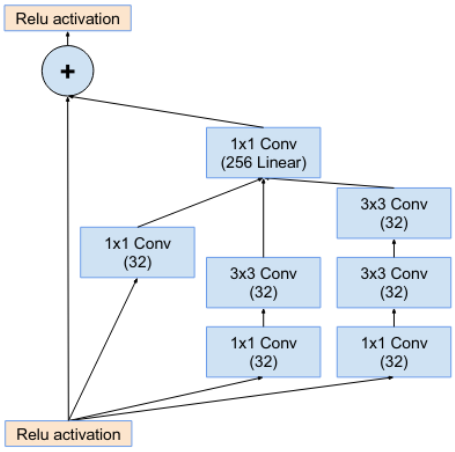


Figura 4. Desenho esquemático de um dos módulos da arquitetura Inception-ResNet-v1, intitulado Inception-ResNet-A. Fonte: Szegedy et al. (2017)

numa camada que possui o número de filtros desejado, sem que seja necessário realizar operações computacionalmente custosas. Ainda na Figura 4, observa-se uma conexão residual, que consiste na soma, ao final do bloco, entre os dados de entrada e o resultado das convoluções. Essa conexão visa evitar que, em redes muito profundas, a intensidade do sinal da primeira camada se perca ao longo das diversas transformações dos dados. Conservando-se a integridade do sinal, é possível ampliar o número de camadas sem que o modelo sofra penalizações em razão da complexidade das operações.

2.3 Extração de Características

A rede utilizada neste artigo se encontra disponível em formato h5 no github do criador da Facenet, David Sandberg, e seus parâmetros foram otimizados tendo em vista a função *triplet loss*, conforme prescreve a metodologia do artigo da Facenet.

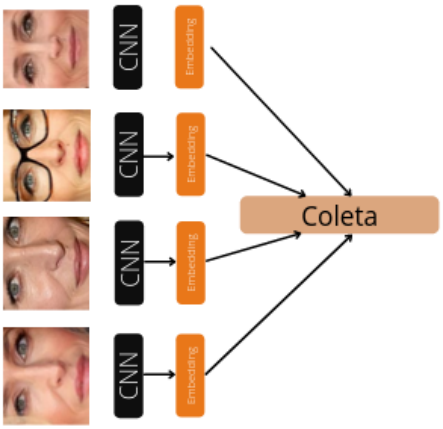


Figura 5. *Embeddings*: extração de características das imagens de Gillian Anderson.

Assim, utilizou-se essa rede convolucional para capturar *embeddings* das imagens. Os *embeddings* são vetores de características recolhidos na última de camada de neurônios, em resposta a uma imagem introduzida na camada de entrada da rede. Foram coletados os *embeddings* de cada indivíduo contido na base de dados, como ilustra a Figura 5 A camada de saída do modelo da Facenet possui 128 neurônios, ou seja, os vetores de característica(*embeddings*) possuem 128 elementos.

Antes de introduzir as imagens na rede neural, foi feita a normalização dos pixels mediante divisão pelo número inteiro 255, obrigando os elementos da matriz a assumirem valores entre 0 e 1. Pei and Lin (1995)

2.4 Método

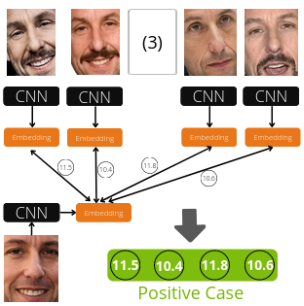
Na estratégia proposta pelo artigo da Facenet, para saber se uma determinada imagem pertence a um indivíduo, computa-se a distância euclidiana, ou quadrática, entre o vetor de características dessa imagem e o de alguma imagem pertencente a esse indivíduo; depois, calcula-se a mesma distância com relação ao vetor de características obtido a partir de uma imagem de um outro indivíduo qualquer. Faz-se a subtração dessas distâncias e, se o resultado for menor que um determinado limiar, considera-se que a imagem pertence ao indivíduo. Caso contrário, considera-se que não. Uma das vantagens dessa técnica, é que só se necessita de duas imagens para efetuar a previsão, um exemplo positivo, e um negativo, sem que seja necessário cadastrar várias imagens do mesmo usuário.

Entretanto, é possível utilizar mais de uma imagem da mesma pessoa, com o fito de aperfeiçoar a reidentificação facial, reduzindo a taxa de falsos negativos e falsos positivos, como será mostrado na seção de resultados. Quando se possui, por exemplo, quatro imagens de um indivíduo, a previsão do sistema tem uma base de apoio mais ampla, e os possíveis ruídos contidos numa determinada imagem são compensados quando se leva em consideração as outras. Ademais, o erro mais grave que pode ocorrer num sistema de reconhecimento facial é o falso positivo. E a probabilidade de um impostor passar despercebido pelo sistema é bem maior quando o *embedding* deste impostor é comparado somente a um dos *embeddings* de um determinado indivíduo, ao invés de ser comparado a vários.

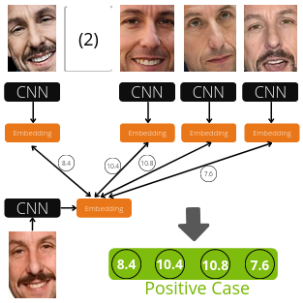
O método proposto por este artigo efetua a reidentificação levando em conta as distâncias euclidianas entre os diversos *embeddings* pertencentes a um mesmo indivíduo e o *embedding* da imagem que se deseja validar, para saber se pertence ou não ao indivíduo em questão. Neste intuito, para cada indivíduo da base de dados, foi treinada uma máquina de vetores suporte, totalizando 258 modelos. Antes do treinamento, a base de dados foi dividida em quatro partes, três das quais foram separadas para o treinamento desses modelos, e a outra parte para o estágio de teste. Ou seja, o indivíduo que tinha 8 imagens, ficou, na base de treinamento, somente com 6, e com 2 na de teste. E assim a proporção foi aplicada aos demais indivíduos da base, de acordo com seu número total de imagens.

Durante o treinamento, a máquina de vetores suporte é alimentada com vetores que contém os valores das diversas distâncias euclidianas, calculadas entre os *embeddings* da base de dados de treinamento. Assim, são formadas amostras positivss e amostras negativas, e os modelos são treinados para emitir uma resposta binária, onde o valor 1 aponta para o caso positivo, e 0 para o negativo.

Na Figura 6, há uma esquema da formação dos pares positivos. Cada vetor de características de um indivíduo é comparado com os vetores de características das outras imagens desse mesmo indivíduo, gerando as distâncias euclidianas circuladas na figura. Depois, essas distâncias são agrupadas num vetor. A totalidade dos vetores gerados nessa etapa representam as amostras positivas que serão colocadas na entrada da máquina de vetores suporte do indivíduo em questão. Isso é feito para todos os indivíduos da base.



(a) Formação de uma amostra positiva a partir da segunda imagem. Valores de distância obtidos: 8,4; 10,4; 10,8; 7,6



(b) Formação de uma amostra positiva a partir da terceira imagem. Valores de distância obtidos: 11,5; 10,4; 11,8; 10,6

Figura 6. Formação das amostras positivas a serem introduzidas na máquina de vetores suporte.

Na Figura 7, há um esquema da formação de pares negativos. Toma-se uma imagem de rosto que não pertence ao indivíduo cujo modelo será treinado, descarta-se aleatoriamente uma das imagens positivas, e faz-se o cálculo da distância euclidiana entre a imagem negativa e as imagens positivas restantes. Após isso, junta-se essas distâncias num vetor. Cabe notar que, como já era esperado, as distâncias do caso negativo assumem valores maiores, pois a similaridade entre os dados é menor.

Uma vez realizada a geração das amostras, treina-se a máquina de vetores suporte associada ao indivíduo em questão, utilizando a técnica de busca de parâmetros *Grid*

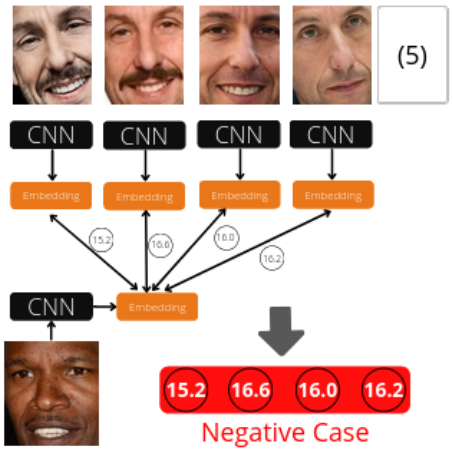


Figura 7. Formação de uma amostra negativa com exclusão aleatória de uma das cinco imagens positivas. Valores de distância obtidos: 15,2; 16,6; 16,0; 16,2

Search, separando uma parte da base de treinamento para a validação dos parâmetros, a fim de encontrar o melhor modelo. Na base de dados utilizada neste artigo, o *kernel* da máquina de vetores suporte que demonstrou melhores resultados nos dados de validação foi, em todos os indivíduos, o *kernel* linear. E o parâmetro C assumiu o valor 0,25. Os conjuntos de parâmetros que utilizaram o *kernel* RBF resultaram em baixas taxas de acerto, o que indica que, no espaço vetorial gerado pela *Facenet*, o problema da reidentificação é um problema linear, quando se utiliza a distância euclidiana.

2.5 Metodologia dos Testes

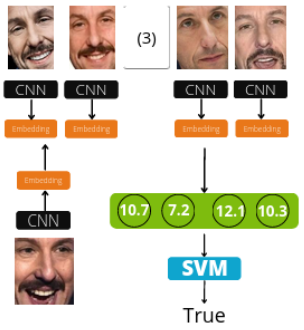
Cada modelo foi testado individualmente, para aferir a taxa de falsos negativos e falsos positivos, utilizando a base dados de teste, com um total de 852 imagens. Ao final, foi feita calculada a média dos falsos positivos e falsos negativos dos modelos.

Cada modelo foi testado da seguinte maneira. Para calcular a taxa de falsos negativos, tomou-se, dentre as 852 imagens, os *embeddings* referentes a imagens do indivíduo ao qual este modelo está associado. Para cada um desses *embeddings*, computou-se a distância para os *embeddings* de treinamento desse indivíduo, resultando no vetor de distâncias. E esse vetor foi passado na entrada da máquina de vetores suporte, para determinar a saída binária. Nos casos, em que a resposta foi o valor 0, considerou-se isso um falso negativo. E nos casos em que foi 1, considerou-se um verdadeiro positivo, ou seja, um previsão correta.

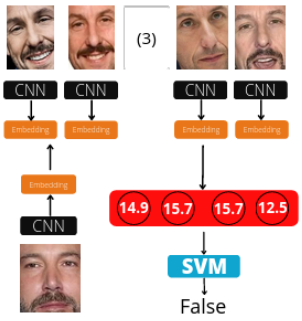
Para calcular a taxa de falsos positivos realizou-se um processo semelhante, porém tomando os *embeddings* das imagens associadas aos outros indivíduos da base de dados. Neste caso, os casos em que a máquina de vetores suporte entregaram o valor 1 na saída foram considerados falsos positivos. E nos casos em que o valor foi 0, considerou-se isso um verdadeiro negativo, ou seja, uma previsão correta.

Considere-se o seguinte exemplo numérico. Um indivíduo teve 4 imagens alocadas para teste. Isso significa que, das 852 imagens de teste da base de dados, 4 pertencem a este indivíduo, e 848 pertencem a outros. Se, dentre as 4 imagens, 3 forem consideradas positivas pelo modelo, e a outra for considerada negativa, a taxa de falsos negativos, para este indivíduo, é de 25%. Caso, dentre as 848 imagens pertinentes a outros indivíduos, 12 sejam consideradas positivas pelo modelo, e as demais sejam classificadas como negativas, a taxa de falss positivos, para este indivíduo, é 1,41%. Esse processo é aplicado a todos os indivíduos.

Na Figura 8, segue um esquema dos testes. Cada vez que uma imagem é testada no modelo, após a extração do vetor de características, as distâncias são computadas, havendo a exclusão aleatória de uma das imagens de treino, para que o vetor de distâncias tenha o mesmo tamanho das entradas nas quais a máquina de vetores suporte foi treinada.



(a) Testagem de uma amostra positiva. Valores de distância obtidos: 10,7; 7,2; 12,1; 10,3



(b) Testagem de uma amostra negativa. Valores de distância obtidos: 14,9; 15,7; 15,7; 12,5

Figura 8. Ilustração de testes em amostras positivas e negativas.

O exemplo da Figura 8 é bastante expressivo, pois se pode observar que, no caso positivo, uma das distâncias calculadas assumiu o valor 12,1, enquanto, no caso negativo, uma das distâncias resultou no valor 12,5, valores muito próximos, porque a imagem de Ben Affleck se assemelha a um dos padrões de treinamento do indivíduo Adam Sendler. Porém, esta semelhança é compensada pelos outros valores do vetor de distâncias. Isso ilustra a vantagem de se empregar uma metodologia que leva em conta mais de uma imagem do indivíduo para fins de inferência. Caso se utilizasse somente o valor de 12,5 para julgar se o indivíduo da foto é o mesmo do modelo, provavelmente se configuraria um caso de falso positivo. Da mesma maneira, o valor 12,1 poderia configurar um falso negativo, caso não se levasse em conta os outros valores distância obtidos.

2.6 Métodos comparados

O método intitulado *Facenet*, na Tabela 1, segue a inferência proposta no artigo da Facenet, utilizando a função *triplet loss*. A taxas foram calculadas por indivíduo, efetuando-se uma média, levando-se em conta as imagens de testes relativas ao indivíduo em questão, que formaram os casos positivos, e as imagens pertinentes a outros indivíduos da base, que formaram os casos negativos. Para tanto, selecionam-se aleatoriamente uma imagem positiva e uma negativa, calcula-se a distância euclidiana entre a âncora e a positiva, e entre a âncora e a negativa; em seguida, subtraem-se essas distâncias e, se o valor obtido for inferior a um certo limiar, considera-se que a imagem âncora é da mesma classe da imagem positiva. Este limiar é calculada por validação cruzada no conjunto de treinamento.

Já no que se refere ao outro método, intitulado *Embeddings* na SVM, trata-se de uma estratégia que consiste em colocar, na entrada das máquinas de vetores suporte, os próprios *embeddings*, positivos e negativos, no lugar do vetor com as distâncias. Para cada indivíduo, gera-se os *embeddings* por meio do modelo da *Facenet*, de imagens deste indivíduo, e de imagens que não são dele. A seguir, treina-se uma máquina de vetores suporte para separar as duas classes: positivo ou negativo. Os testes, nesse caso, foram feitos do mesmo modo como no método proposto por este trabalho, ou seja, foi computada a sensitividade e a especificidade para cada indivíduo, e a taxa resultante, exibida na Tabela 1, é uma média.

3. RESULTADOS E DISCUSSÃO

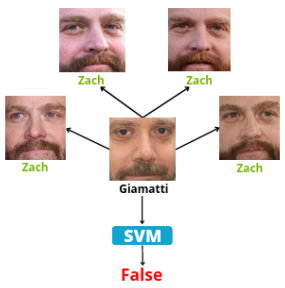
Método	Sensib.(%)	Especif.(%)	Inferência($10^{-5}s$)
Vetor Distâncias	95,07 ± 0,97	99,94 ± 0,004	18.79
Embeddings na SVM	95,71 ± 0,92	99,86 ± 0,092	5.93
Facenet	91,24 ± 1,31	97,81 ± 0,954	3.17

Tabela 1. Resultados

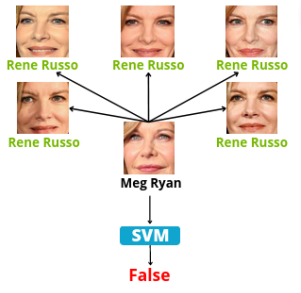
Observa-se na Tabela 1 que o uso das máquinas de vetores suporte fez uma grande diferença em relação à simples aplicação da estratégia da Facenet. Nos dois primeiros métodos dessa tabela, a taxa de falsos positivos foi inferior a 1%. No entanto, o método proposto, muito embora tenha apresentado uma taxa de falsos negativos mais elevada, resultou numa taxa de falsos positivos 2,3 vezes menor(de 0,14% para 0,06%).

Nas Figuras 9, 10 e 11 seguem exemplos dos resultados obtidos. Vê-se, na Figura 9, os atores Paul Giamatti e Zach Galifianakis, dotados de aparência similar. Ainda assim, Giamatti foi detectado como um verdadeiro negativo; do mesmo modo a atriz Meg Ryan, quando comparada à Rene Russo. Na Figura 10, pode-se ver casos de verdadeiro positivo.

Na Figura 11, tem-se casos de falso positivo e falso negativo, respectivamente nos itens (a) e (b). Percebeu-se que, em certos casos, embora utilizando diversas imagens do mesmo indivíduo, o método não foi capaz de contornar valores baixos de distância entre *embeddings* de indivíduos diferentes.

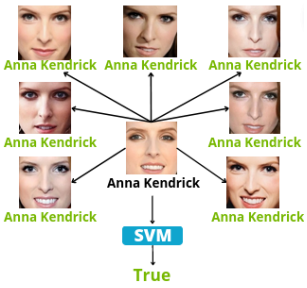


(a) Exemplo de amostra negativa classificada corretamente: Paul Giamatti *versus* Zach Galifianakis

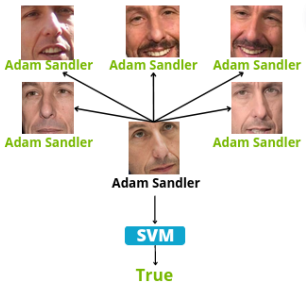


(b) Exemplo de amostra negativa classificada corretamente: Meg Ryan *versus* Rene Russo

Figura 9. Exemplos de verdadeiro negativo alcançados após o treinamento

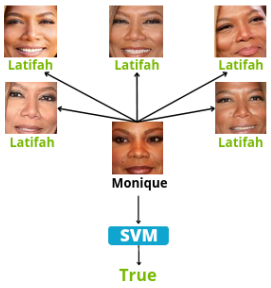


(a) Exemplo de amostra positiva classificada corretamente: Anna Kendrick *versus* Anna Kendrick

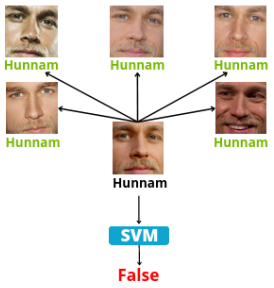


(b) Exemplo de amostra positiva classificada corretamente: Adam Sandler *versus* Adam Sandler

Figura 10. Exemplos de verdadeiro positivo alcançados após o treinamento



(a) Exemplo de amostra negativa classificada incorretamente: Monique *versus* Queen Latifah



(b) Exemplo de amostra positiva classificada incorretamente: Hunnam *versus* Hunnam

Figura 11. Exemplos de previsões incorretas, falso positivo e falso negativo, obtidas após treinamento

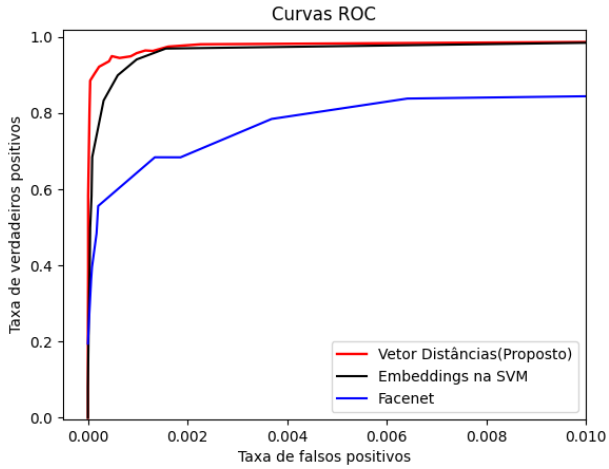


Figura 12. Curvas ROC. Note que a curva da *Facenet* ocupa uma área inferior à dos métodos que utlizam a SVM, e que o método proposto, assinalado com a linha vermelha, abrange uma área maior que os outros dois.

Ademais, na Figura 12 pode-se ver as curvas ROC, construídas com os resultados de cada algoritmo. Vê-se que o método proposto, embora apresente, na Tabela 1, uma taxa de falsos negativos mais alta que a do método *Em-*

beddings na SVM, abrange uma secção maior no gráfico da curva ROC em comparação a este último método.

4. CONCLUSÃO

Assim, a utilização de múltiplas imagens dos indivíduos é capaz de compensar o fato de que o mesmo indivíduo pode aparecer, em uma ou outra imagem, com uma aparência sensivelmente diferente das demais. Como se viu nas Figuras 6 e 8, há imagens em que o indivíduo da base possui um bigode no rosto, e outras em que o rosto está limpo.

Ademais, aplicar diretamente os vetores de características nas entradas das máquinas de vetores suporte(que é a estratégia do método *Embeddings* na SVM, da Tabela 1), embora seja uma solução mais efetiva que a inferência tradicional proposta pela Facenet, não alcança a curva ROC do método proposto neste artigo. A explicação para isso é que este método tira mais proveito da multiplicidade de imagens que aquele. No segundo método da Tabela 1, os casos de elevada similiridade entre imagens de indivíduos diferentes(casos de *outliers*) repercutem bastante na descoberta dos vetores suporte do SVM, e acabam interferindo negativamente na construção da margem de separação entre as classes; e do mesmo modo influenciam os casos de baixa similaridade entre imagens do mesmo indivíduo. Já no método proposto, a presença desses *outliers* nas amostras é disfarçada pela construção do vetor de distâncias, porque o vetor que é introduzido na entrada da SVM contém não somente a distância calculada a partir da amostra *outlier*, mas também as distâncias calculadas a partir de outras amostras. Desse modo, a presença do *outlier* provoca menos repercussão na descoberta do hiperplano de margem máxima, o que gera contribuições na especificidade do método, reduzindo a taxa de falsos positivos.

No entanto, o preço para alcançar este resultado é um aumento significativo no tempo de inferência, como se viu na Tabela 1, em razão do cálculo das distâncias euclidianas, que antecede a introdução da amostra na entrada da SVM; além disso, outra desvantagem do método proposto é a necessidade de se ter um número razoável de imagens por indivíduo. Experimentalmente, o número mínimo para obter resultados satisfatórios no treinamento dos modelos é de 4 imagens.

AGRADECIMENTOS

A pesquisa desenvolvida nesse trabalho foi suportado parcialmente pelo projeto Apple Developer Academy - IFCE, ao qual os autores oferecem seus votos de gratidão.

REFERÊNCIAS

Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12), 2037–2041.

Amos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016a). Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science.

Amos, B., Ludwiczuk, B., Satyanarayanan, M., et al. (2016b). Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6(2), 20.

Cohen, L. (1994). Whodunit?–violence and the myth of fingerprints: Comment on harding. *Configurations*, 2(2), 343–347.

Cole, S. (1999). What counts for identity? the historical origins of the methodology of latent fingerprint identification. *Science in Context*, 12(1), 139–172.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, 886–893. Ieee.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hermans, A., Beyer, L., and Leibe, B. (2017). In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Huang, G.B., Mattar, M., Berg, T., and Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*.

King, D.E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10, 1755–1758.

Koch, G., Zemel, R., Salakhutdinov, R., et al. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Lecun, Y. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), 971–987.

Olivier Moindrot (2018). Triplet loss in tensorflow. URL `https://github.com/omoindrot/tensorflow-triplet-loss`. [Online; accessed April 27, 2013].

Parkhi, O.M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition.

Pei, S.C. and Lin, C.N. (1995). Image normalization for pattern recognition. *Image and Vision computing*, 13(10), 711–723.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A.A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71–86.

V.M. Corman, O. Landt, M.K.e.a. (2020). Detection of 2019 novel coronavirus (2019-ncov) by real-time rt-pcr. *Euro Surveillance*.

Wu, X. (2015). Learning robust deep face representation. *arXiv preprint arXiv:1507.04844*.

Zumla, D.H..E.A..T.M..F.N..R.K..O.D..G.I..T.M..Z.M..C.D..A. (2020). The continuing 2019-ncov epidemic threat of novel coronaviruses to global health—the latest 2019 novel coronavirus outbreak in wuhan, china. *International Journal of Infectious Diseases*.