Proposta de um sistema de autenticação por reconhecimento facial em linguagem de programação Python e banco de dados MySQL

Proposal for a facial recognition authentication system in Python programming language MySQL database

Guilherme de Freitas Anacleto * Eduardo Alves Moraes †

junho de 2022

Resumo

O reconhecimento facial está em evolução a cada dia. Reconhecer um rosto ou identificar quem seja apenas pela face, é uma tarefa simples para qualquer indivíduo, que se adaptou ao longo de milhões de anos de existência. A visão é uma das características fundamentais para a existência da espécie humana. Com todos os avanços tecnológicos, conseguiu-se programar uma visão computacional que simula a visão humana. No entanto reproduzir tal característica tão complexa em um sistema computacional, não seria tão simples sem o avanço continuo e rápido da tecnologia. Por meio de um sistema criado a partir de uma linguagem de programação de alto nível chamada Python, as complexidades encontradas no desenvolvimento deste tipo de funcionalidade foram reduzidas em apenas algumas linhas de programação, Sendo possível identificar um individuo por meio de sua face utilizando uma simples *webcam*.

Palavras-chave: reconhecimento facial, Python, autenticação facial, visão computacional.

Abstract

Facial recognition has been evolving day after day. Recognizing a face or identifying who it is just by the face is a simple task for any individual, who has adapted over millions of years of existence. Vision is one of the fundamental characteristics for the existence of the human species. With all the technological advances, it has been possible to program a computer vision that simulates human vision. However, reproducing such

*"Aluno do curso de Tecnologia em Segurança da Informação - Faculdade de Tecnologia de Ourinhos - e-mail: guilherme.analeto@fatec.sp.gov.br"

†"Professor Orientador do curso de Tecnologia em Segurança da Informação - Faculdade de Tecnologia de Ourinhos - e-mail: eduardo.moraes@fatec.sp.gov.br"

a complex feature in a computer system would not be so simple without the continuous and rapid advancement of technology. Through a system created from a high-level programming language called Python, the complexities found in the development of this type of functionality have been reduced in just a few programming lines, making it possible to identify an individual through their face.

Keywords: facial recognition, Python, facial authentication, computer vision.

1 Introdução

Atualmente os mecanismos mais utilizados na verificação de identidade são *passwords* ou cartões magnéticos. No entanto, pretende-se que as tecnologias biométricas venham elevar o grau de certeza e substituir os mecanismos mais convencionais (**??**).

Com o propósito de contribuir para a ampliação do conhecimento da área, o presente artigo cientifico tem como objetivo analisar os resultados obtidos por meio da autenticação facial baseada na linguagem de programação Python, utilizando algumas de suas bibliotecas voltadas para visão computacional.

O campo da biometria ganhou a maior atenção e fez seu lugar como a opção mais confiável de reconhecimento durante o passado, devido à disponibilidade de tecnologia viável após extensa pesquisa neste campo e lacunas em outros sistemas de identificação (**??**).

Esse software desenvolvido em Python, tem como finalidade, analisar o rosto de um ser humano, sendo capaz de determinar, se esse usurário já esta na base de dados cadastro ou não, ao identificar o usuário, o nome do individuo será apresentado logo abaixo de seu rosto, caso não seja possível identificá-lo, o sistema exibirá que a entidade é desconhecida.

Utilizando a linguagem Python, se abriu um leque de possibilidades, com uma facilidade sem igual, com suas bibliotecas que trabalham diretamente com aplicações visuais e com um vasto suporte de detecção facial podendo posteriormente se transformar em uma autenticação.

Mesmo com toda essa facilidade e evolução ao longo dos anos, seria possível gerar uma autenticação facial baseada na linguagem de programação Python em poucas linhas, capaz de identificar e autenticar usuários, por meio de suas faces? Dependendo dos resultados encontrados será possível responder essa questão.

2 Revisão Bibliográfica

Este estudo traz como revisão bibliográfica temática de autenticação facial, utilizando linguagem de programação Python, a definição do tema, as questões de pesquisa levantadas, objetivos e sua implementação são meios de se obter resultados a respeito de sua acurácia. Para tal, realizou-se a utilização de diversos artigos científicos, com seus temas voltados a autenticação facial e seus métodos, reconhecimento facial e sua história, linguagem de programação Python.

A busca foi realizada de maneira remota, por meio de pesquisa de artigos científicos, monografias, sendo periódicos disponíveis no Google Acadêmico.

Sendo selecionado como analise, todos os artigos que mencionassem "Autenticação", "Reconhecimento Facial", "Visão Computacional". Priorizou-se por incluir as palavras chaves como "Autenticação Facial", já que é a definição mais clara do que será executado neste artigo. Com o material obtido, foi possível o desenvolvimento deste documento e suas demais citações, bem como seus devidos testes.

2.1 Reconhecimento facial

O reconhecimento facial é uma função vital para nós seres humanos, a partir do momento que nascemos e abrimos nossos olhos, já estamos utilizando essa função evolutiva primordial.

Pensando nessa linha de raciocínio, o quão difícil seria tentar reproduzir essa função em uma máquina que só trabalha com zeros e uns, segundo pesquisas de cientistas atuais, existem as chamadas células da face (*face patches*), localizadas no córtex temporal inferior.

"Os cientistas descobriram que, em vez de representar uma identidade específica, cada célula da face representa um eixo específico dentro de um espaço multidimensional (**??**)".

Um bom exemplo que segue o mesmo conceito são as cores vermelho, azul e verde que se combinam de maneiras distintas para criar todas as cores de um espectro, os eixos podem se combinar e criar infinidades de faces.

Com o avanço da tecnologia o reconhecimento facial foi se adaptando ao longo das ultimas décadas, sendo utilizado em diversos mecanismos, como ao sorrir para tirar uma foto, em enquadramento de rostos em câmera de celular ou até mesmo para a segurança governamental.

O reconhecimento facial computacional funciona seguindo os mesmos parâmetros de como nós humanos identificamos um rosto, assim como em nosso sistema nervoso existem as células da face, responsáveis pela identificação de rostos, nas máquinas existem algoritmos que simulam esse processo, pegando as características de um rosto, sendo eles divididos em dois tipos, a interna como nariz, olhos, boca e a externa como a forma do rosto, estilo do cabelo, contorno da cabeça e orelhas, a máquina pega todos esses parâmetros e transforma em um código, para que ela possa trabalhar com ela posteriormente.

Mas nem sempre o reconhecimento facial foi tão sofisticado, antes dele se tornar o que é hoje, existiu uma variedade de eventos durante a história que contribuirão para sua evolução.

??) desenvolveram um sistema que o permitia classificar fotos de rostos manualmente, usando um dispositivo que necessitava que fossem inseridas as coordenadas horizontais e verticais usando uma caneta que emitia pulsos eletromagnéticos, esse sistema pode ser utilizado para registrar manualmente as localizações de coordenadas das várias caracterizas da face, sendo olhos, linha do cabelo, e boca (**??**).

??) adicionaram maior precisão ao reconhecimento facial, utilizaram 21 marcadores subjetivos específicos, incluindo a cor do cabelo e a espessura dos lábios, para automatizarem o reconhecimento. O problema residia no fato das medições e localizações terem de ser calculadas manualmente também (**??**).

Em 1988, início dos anos 1990, foi desenvolvida a técnica de Eigenfaces, que é basicamente o conjunto de autovetores de uma matriz de covariância formada por imagens de faces. Esta técnica foi desenvolvida para representar padrões encontrados em imagens

de rostos utilizando o método de Análise de Componentes Principais (**??**).

??) expandiram a abordagem Eigenface, descobrindo como detectar rostos em imagens. Isso levou às primeiras ocorrências de reconhecimento automático de rosto. Sua abordagem foi limitada por fatores tecnológicos e ambientais, mas foi um avanço significativo na comprovação da viabilidade do reconhecimento facial automático (**??**).

Em 1998, para incentivar a indústria e a academia a avançar neste tópico, a Agência de Projetos de Pesquisa Avançada de Defesa (DARPA) desenvolveu o programa de tecnologia de reconhecimento facial (FERET), que forneceu ao mundo um banco de dados considerável e desafiador composto por 2.400 imagens para 850 pessoas (**??**).

No ano de 2006, o FRGC[1] avaliou os algoritmos de reconhecimento facial mais recentes disponíveis. Imagens de rosto de alta resolução, varreduras 3D de rosto e imagens de íris foram usadas nos testes. Os resultados indicaram que os novos algoritmos eram 10 vezes mais precisos do que os algoritmos de reconhecimento facial de 2002 e 100 vezes mais precisos do que os de 1995, mostrando os avanços da tecnologia de reconhecimento facial na última década (**??**).

Nos anos seguintes a tecnologia de reconhecimento facial, só aumentou, se expandindo nas redes sócias, sendo implementado pelo Facebook, segundo **??**), o recurso foi muito controverso no começo, más após alguns anos os usuários não se incomodaram.

A Apple lançou o iPhone X em 2017, anunciando o reconhecimento facial como um de seus novos recursos principais. O sistema de reconhecimento de rosto do telefone é usado para segurança do dispositivo. O novo modelo de iPhone esgotou quase instantaneamente, provando que os consumidores agora aceitam o reconhecimento facial como o novo padrão ouro para segurança (**??**).

Mais recentemente, utiliza-se técnicas de aprendizado de máquina (*machine learning*), para identificar diversos rostos, por meio de linguagem de programação.

2.2 Autenticação

O processo de autenticação fornece como finalidade identificar um indivíduo ou verificar se a identidade do mesmo é verdadeira ou falsa (**??**).

Existem três métodos de autenticação utilizados nos tempos atuais:

- Autenticação baseada em conhecimento: onde o usuário precisa ter uma informação prévia para que possa ser autenticado (Ex.: Senhas formadas por um conjunto de caracteres), sendo a menos segura.
- Autenticação baseada em posse: quando o usuário só pode ser autenticado, possuindo um dispositivo que possa autentica-lo (Ex.: *Tokens*, Cartões de acesso.), sendo mais segura que a anterior.
- Autenticação baseada em característica: é quando se utiliza uma característica única do usuário para identifica-lo (Ex: Biometria, como leitura da íris), sendo a mais segura entre as três.

[1] *Face Recognition Grand Challenge*

Autenticação refere-se a necessidade de confirmar ou negar uma determinada identidade de um indivíduo, enquanto identificação é estabelecer a identidade, desconhecida à partida, de um indivíduo.

A autenticação biométrica oferece muitas vantagens sobre os sistemas de autenticação convencionais que dependem de posses ou conhecimento especial (**??**).

Portanto, será utilizada nessa pesquisa o meio de autenticação biométrica, sendo focada especificamente na autenticação por meio da face.

A face será lida, por meio de uma imagem, carregada no sistema, é possível identificar apenas o rosto na imagem, e em seguida transformar essa face em um vetor, onde se obtém os dados do rosto do indivíduo em um vetor de 128 posições, sendo posteriormente utilizado para a comparação das faces já registradas no banco de dados.

3 Materiais e Métodos

No quesito modo de pesquisa, foi aderido obter resultados, informações e aprendizado existentes em artigos, vídeo aulas, e sites como GitHub[2], para fins de teste com o reconhecimento facial, utilizando informações relevantes a respeito da biblioteca *face recognition*, que se dedicavam explicitamente em apresentar sobre os seguintes temas, autenticação facial, linguagem de programação Python e suas bibliotecas.

GitHub é uma plataforma de código-fonte e hospedagem de arquivos que usa Git para controle de versão. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribua com projetos privados e *open source* de qualquer usuário.

Para efetuar a programação foi utilizado o Visual Studio Code[3]. É o editor de código-fonte da Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle de versão Git integrado, realce de sintaxe, conclusão de código inteligente, *snippets* e refatoração de código.

Ao longo da elaboração do artigo existiu alguns obstáculos para se obter o conteúdo necessário, sendo um deles, instalar todas as bibliotecas necessárias, levando a problemas na execução do código, sendo o principal detalhe, a versão do *Python*, onde a biblioteca do *face recognition* falhou na versão mais atual até o momento da publicação deste artigo, sendo ela *Python 3.9.7*.

Foi executado a aplicação com êxito, com a versão do *Python 3.8.3*, onde não existiu falhas significativas. Porém a biblioteca utilizada exige uma quantidade considerável de processamento, onde o video apresenta uma quantidade baixa de quadros por segundo.

O material foi utilizado para o desenvolvimento deste artigo científico. Portanto foi elaborado testes, para que seja provada sua eficácia a respeito de autenticação e sua funcionalidade.

4 Resultados Obtidos

Todos os testes foram realizados em um computador de mesa, com as seguintes configurações, processador Intel i5 750, 8 GB de memória RAM, placa de video GTX 750

[2] Link de acesso ao projeto GitHub: <https://github.com/>
[3] Link de acesso: <https://code.visualstudio.com/>

ti, sendo usado uma *webcam*, com a capacidade máxima de gravar vídeos em 480 pixeis.

Com a ferramenta Visual Studio Code, foi possível executar o código em Python, onde foi possível realizar todos os testes desejados.

Seguindo os processos que o programa executa, temos, carregar a foto, obter apenas o rosto da foto, transformar o rosto em um vetor, cadastrar esse vetor no banco, por fim comparar o rosto visualizado na *webcam* com os dados no banco e retornar se esse rosto existe na base dados ou não.

Ao realizar um teste utilizando a imagem apresentado na Figura 1, inicialmente o programa vai carregar a imagem, depois localizar apenas o rosto nessa imagem.

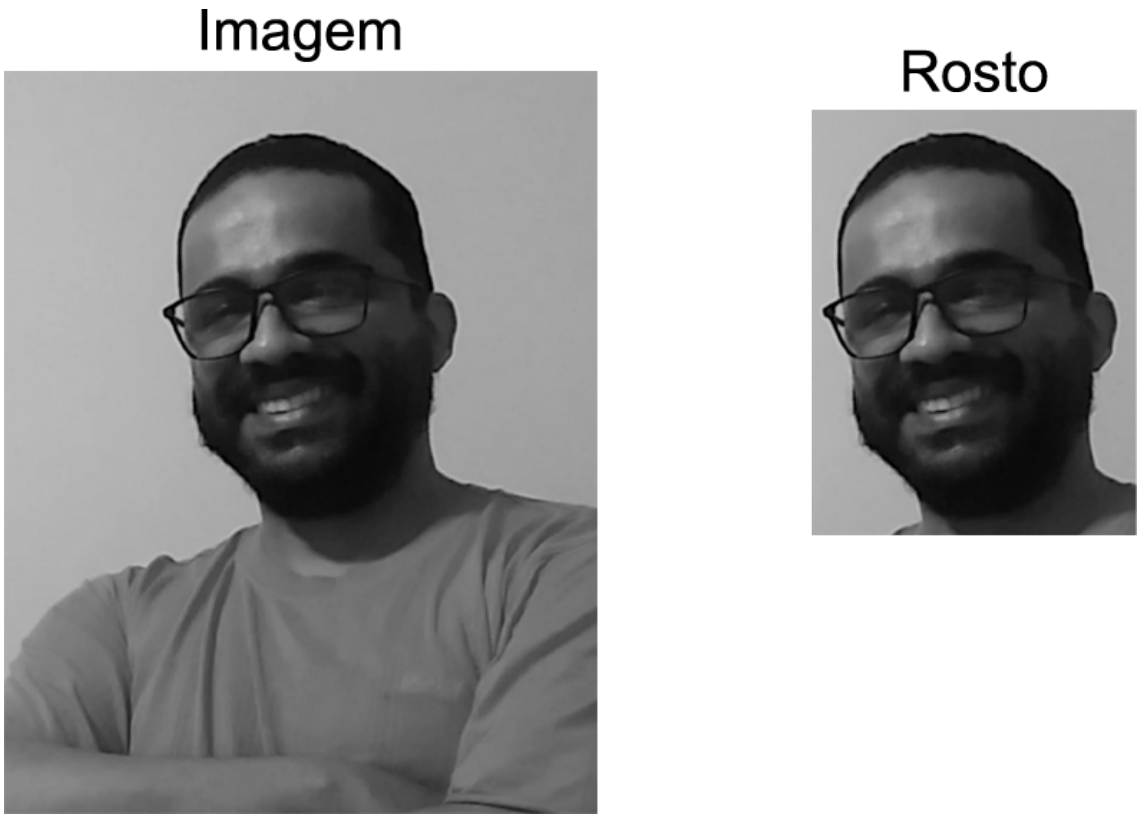

Figura 1 – Imagem de teste 1
Fonte: Do próprio autor.

A partir do rosto encontrado na Figura 1, é gerado um vetor estruturado e apresentado na Figura 2.

```
[-9.31140333e-02  1.36347309e-01  5.76222986e-02 -3.95874828e-02
  8.10826314e-04 -7.97739625e-03  3.56786996e-02 -4.16545803e-03
  1.73047379e-01 -4.78014462e-02  1.90434322e-01 -5.37134334e-02
 -1.87315032e-01 -9.71490815e-02 -1.75816361e-02  1.08819373e-01
 -1.83032200e-01 -1.10453986e-01 -2.01985501e-02 -5.74079342e-02
  1.65647827e-02 -5.67717571e-03  1.87942735e-03  5.61386943e-02
 -9.84536484e-02 -4.08220977e-01 -8.91670734e-02 -1.94804385e-01
  5.63235059e-02 -1.59697443e-01 -2.04914100e-02  5.33060580e-02
 -1.62762374e-01 -2.23314390e-02 -6.25901446e-02  3.67623344e-02
 -7.90311210e-03 -4.73111942e-02  1.38298973e-01  8.62961262e-02
 -6.87812269e-02  1.11290412e-02 -1.12692509e-02  3.72567505e-01
  2.08722264e-01  2.72146259e-02 -3.11672688e-04  3.97700891e-02
  6.31182492e-02 -2.21053973e-01  2.25325581e-02  1.05703309e-01
  8.50314423e-02  8.33195075e-02  5.31794652e-02 -1.14681661e-01
  2.23344024e-02 -2.32209004e-02 -2.22390860e-01  6.08249567e-02
  2.17610933e-02 -2.61916276e-02 -5.59190288e-02  1.18112676e-02
  2.97494650e-01  9.49992910e-02 -1.15631178e-01 -4.00296822e-02
  1.73038945e-01 -1.56199053e-01  3.83101888e-02 -3.96796837e-02
 -1.43434525e-01 -1.13570675e-01 -1.96788773e-01  1.53188497e-01
  3.51463079e-01  1.66165248e-01 -1.22764722e-01  4.38887775e-02
 -1.52755916e-01 -1.61146279e-04  1.97935523e-03  5.96992187e-02
 -1.46991387e-01  3.20443958e-02 -6.91718757e-02  7.82248229e-02
  8.86496753e-02  2.18483154e-02 -3.27690616e-02  1.97523877e-01
 -5.96015938e-02  3.41963954e-03  3.45214717e-02 -7.30164871e-02
 -8.55271593e-02 -1.67789571e-02 -1.11325935e-01 -2.49817930e-02
  4.13780138e-02 -1.57796279e-01 -1.34396367e-02  1.44075021e-01
 -2.15163201e-01  8.87601003e-02  2.36421395e-02 -4.41100076e-02
  2.99421623e-02  1.04323268e-01 -1.33155674e-01 -7.17625171e-02
  1.36491194e-01 -2.37803206e-01  1.76193103e-01  1.98602721e-01
  7.50521123e-02  1.18576325e-01  4.24303859e-02  5.71078658e-02
 -1.64657626e-02 -7.37689435e-04 -1.09145984e-01 -1.09757613e-02
  4.75674495e-02  2.66618654e-02 -9.81043559e-03  1.52333416e-02]
```

Figura 2 – Vetor gerado a partir de um rosto
Fonte: Do próprio autor.

O vetor informa ao programa que esse conjunto de dados representam uma face de um individuo, após essa etapa o programa já consegue identificar indivíduo por meio de seus vetores.

Na Figura 3 é exibido o resultado obtido após o teste de autenticação, o sistema identifica o rosto da pessoa comparando o vetor com os vetores que estão no banco de dados, caso o sistema consiga achar o vetor correspondente ao rosto do indivíduo, significa que ele cadastrado, dessa forma o sistema retorna o nome do usuário, caso o sistema não consiga identificar o indivíduo, ele exibe a mensagem "Não identificado".

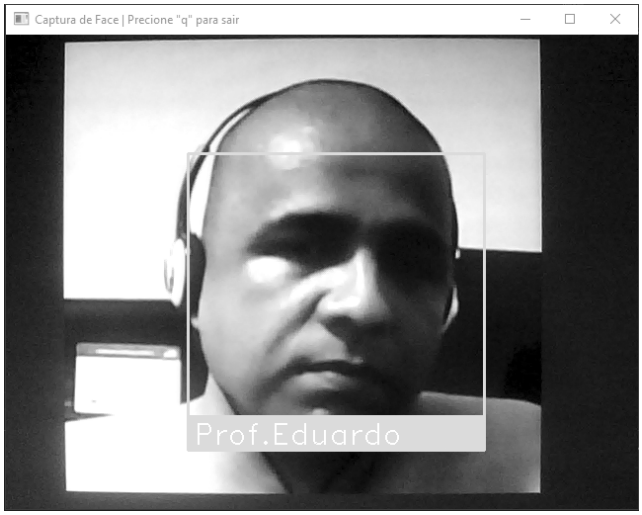

Figura 3 – Autenticação obtida por meio do vetor
Fonte: Do próprio autor.

Segundo a documentação da biblioteca *face recognition*, ela foi construída usando o reconhecimento de rosto de última geração da dlib, que foi construído com aprendizado profundo. O modelo tem precisão de 99,38 no referencial *Faces in the Wild* (**??**).

Os resultados obtidos foram satisfatórios, pois ao testar com uma parcela de pessoas, o programa não teve dificuldade em identifica-las, assim alcançando seu objetivo de identificar e autenticar faces humanas.

Ao longo do desenvolvimento desse projeto, uma das maiores dificuldades foi a instalação das bibliotecas que o *face recognition* dependia, além disso algumas versões mais recentes do Python não dão suporte a essa biblioteca.

5 Considerações Finais

Desde o surgimento da tecnologia, já se pensava em proteção de dados, um grande exemplo disso. são as cifras e senhas criadas por usuários, para proteger seus dados de outros usuários, seja em um aplicativo do banco ou em um *website* de faculdade.

Levando em conta a evolução da tecnologia, hoje em dia existem diversos meios de se autenticar, em diversas plataformas diferentes, sendo por meio de uma senha, biometria, *tokens*, entre outros.

Analisando os tipos de autenticação, todas possuem vantagens e desvantagens, senhas ou palavras-passe, geram ao usuário a dependência da memorização, forçando o usuário a lembrar dela, alguns podem optar por anotar, mas ao fazerem isso essa senha perde o seu propósito, já que outras pessoas podem obtê-las através de vários meios.

Pensando em *tokens* e em cartões de autenticação, são utilizados como objetos para se ter o acesso a algum conteúdo protegido, podem ser comparados a uma chave, que permite que o usuário possa abrir uma porta, onde terá o acesso autenticado, umas das maiores falhas desse tipo de autenticação, seria a dependência no objeto autenticador, em casos de perca ou de danos ao objeto, o usuário perderia o acesso.

Considerando a autenticação biométrica, não cria dependência alguma com a memória ou a necessidade de se ter um objeto autenticador, pois já seria autenticado pelas suas características. O usuário pode ser autenticado através de suas características biométricas, tais como as íris dos olhos, seus dedos, palmas das mãos, voz ou rosto.

Tudo isso só foi possível por conta da evolução exponencial da tecnologia, junto á capacidade do ser humano aprender e se adaptar, permitindo que a raça humana, criasse algoritmos aparentemente simples, porém ao mesmo tempo com um grau complexidade.

Esse artigo científico pode ser de grande utilidade em sistemas, que precisam de um autenticador biométrico baseado em faces humanas, podendo ser responsável pelo controle dos funcionários ou na identificação de uma pessoa em meio a um grupo, portanto disponibilizo esse trabalho de graduação para estudos futuro.

Agradecimentos

A toda a minha família, principalmente meus pais e irmãos.

Aos meus amigos, que estiveram sempre ao meu lado.

A minha namorada, pelo carinho, amor, apoio e incentivos.

E a todos que, direta ou indiretamente, fizeram parte de minha jornada.