

13º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2022

MECANISMO AUTOMATIZADO EM TEMPO REAL PARA DETECÇÃO DE PESSOAS SEM MÁSCARA UTILIZANDO APRENDIZADO DE MÁQUINA

OTÁVIO TORAÇA DIAS¹ e RODRIGO PALUCCI PANTONI²

¹ Graduando em Engenharia Elétrica, Bolsista PIBIFSP, IFSP, Câmpus Sertãozinho, otaviotoracadas@gmail.com

² Docente e Orientador do Projeto PIBIFSP, IFSP, Câmpus Sertãozinho, rpantoni@ifsp.edu.br

Área de conhecimento (Tabela CNPq): 1.03.03.05-7 - Processamento Gráfico (Graphics)

RESUMO: O reconhecimento facial é uma tecnologia moderna e muito aplicada graças ao avanço do *machine learning*, possibilitando novas implementações em detecção e identificação. Este é um projeto de iniciação científica referente ao reconhecimento do uso de máscara em pessoas em tempo real, onde um conjunto de bibliotecas treinadas na linguagem Python, são capazes de analisar e reconhecer as informações quadro a quadro e gerar respostas. O desenvolvimento foi impulsionado devido a pandemia do COVID-19, buscando novas formas de garantir a segurança das pessoas e se baseou em aprendizagem, pesquisa e testes, pois foram analisados alguns métodos de detecção facial e de utilização da máscara num ambiente de sala de aula. Os resultados obtidos foram condizentes com a expectativa inicial do projeto, tendo respostas consistentes e precisas.

PALAVRAS-CHAVE: Python; COVID-19; Inteligência Artificial.

Real-time Automated Mechanism for Person Detection No Mask Using Machine Learning

ABSTRACT: Facial recognition is a modern technology that is widely applied today thanks to the advancements of machine learning, enabling new implementations in security and identification. This project aims to aid on the recognition of the use of facial masks in real time, using a set of libraries trained in the Python programming language, it is possible to analyze and recognize images frame by frame and generate a prognostic. The development was driven due to the COVID-19 pandemic, seeking new ways to guarantee people's safety as masks were mandatory in many countries. The work mainly consisted of learning the libraries, doing research about methods and testing them, as algorithms of facial detection and mask utilization were analyzed on classroom environment. The results obtained were accordant with the initial expectation of the project, with consistent and accurate answers.

KEYWORDS: Python; COVID-19; Artificial Intelligence.

INTRODUÇÃO

Em meio a pandemia de COVID-19, a utilização de máscaras respiratórias vem sendo uma importante prática, sobretudo em locais fechados, como supermercados, lojas, escolas etc.

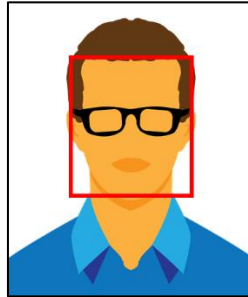
Assim, nos últimos anos, vários especialistas e empresas que dominam as técnicas de reconhecimento facial buscaram implementar algoritmos inteligentes com a finalidade de reconhecer se as pessoas estão ou não utilizando máscaras em vídeos e fotos, o que gerou novas implementações utilizando bibliotecas de reconhecimento facial. O objetivo geral deste trabalho é desenvolver e avaliar um sistema capaz de fazer a detecção de pessoas sem máscaras numa sala de aula.

MATERIAL E MÉTODOS

O processo de detecção de pessoas usando máscara é realizado em duas etapas: detecção de face (etapa 1) e a detecção se a pessoa está usando máscara propriamente dita (etapa 2).

Na execução do algoritmo a detecção de face retorna as coordenadas de um retângulo numa imagem (quadro do vídeo). A figura 1 mostra o “recorte” da figura que as coordenadas geram. Assim, quando uma face é detectada, o algoritmo traça um retângulo no quadro a fim de verificar quando a face foi detectada visualmente. Na sequência, a imagem é recortada apenas na face e depois normalizada, para posteriormente ser testada para verificar se a pessoa está utilizando máscara ou não.

FIGURA 1. Detecção de face e as coordenadas retornadas



Fonte: Adaptado de Pixabay (2022)

A detecção de face é considerada como tarefa extremamente difícil. Pode ser influenciada pela variação de pose, local, iluminação, variação de expressão, pelos faciais, presença de óculos, diferenças no ganho da câmera, iluminação condição e resolução da imagem. Para resolver esse problema, diferentes métodos têm sido propostos, como o Classificador *Cascade* (VIOLA AND JONES, 2001), MTCNN (AL-IMRAN, 2019), Dlib HOG (ADOUANI, 2019) e Dlib CNN (S ET AL, 2016) (JADHAV ET AL., 2021).

Os algoritmos avaliados neste trabalho foram desenvolvidos em Python, juntamente com o IDE PyCharm 2021.2.3 (Community Edition) (2022), que é um ambiente profissional gratuito para docentes e alunos. A escolha da linguagem Python se deu pela disponibilidade de várias bibliotecas para aprendizado de máquina e ciência de dados que, além de serem gratuitas, podem ser incorporadas facilmente em produtos finais, como dispositivos para capturar imagens/vídeos para tomada de alguma ação.

Assim, para a detecção facial avaliaram-se duas bibliotecas diferentes: Dlib e OpenCV. A primeira biblioteca possui duas formas de detecção facial, uma utilizando SVM (*Support Vector Machine*), e outra mais elaborada, que utiliza Redes Neurais Convolucionais (CNN) (ROSEBROCK, 2021). Dessa forma, a variação que trabalha com SVM utiliza o método citado Dlib HOG (ADOUANI, 2019), enquanto a segunda utiliza o método chamado Dlib CNN (S ET AL, 2016).

A biblioteca OpenCV, por outro lado, disponibiliza algumas maneiras de implementar a detecção facial, sendo cada uma delas um arquivo contendo métodos de classificação para encontrar as coordenadas (saída da resposta da biblioteca) da face no *frame*, foram testados dois desses métodos a fim de escolher os melhores resultados. O primeiro, um classificador *Haar Cascade*, possui muitos trabalhos relacionados e para a sua utilização é necessário apenas um arquivo XML contendo configurações para a biblioteca utilizar (BRAIN MENTORS PVT. LTD, 2021), o segundo método utiliza ferramentas de *Deep Learning*, batizado como DNN (*Deep Neural Network*), e para seu funcionamento correto com a biblioteca é preciso de um modelo e de uma configuração dos parâmetros desse modelo (MEGANHA, 2020). Para Agarwal (2020), em sua publicação no site *Towards Data Science*, o *Haar Cascade* apresenta os piores resultados tendo muitos falsos positivos, sendo que o método DNN oferece resultados bem melhores, levando vantagem por funcionar bem em movimentos rápidos e identificar faces laterais.

Assim, neste trabalho optou-se por avaliar a implementação da DNN na biblioteca OpenCV, por apresentar mais eficiência segundo a comunidade acadêmica (JADHAV ET AL., 2021) (ROSEBROCK, 2021), em comparação com o classificador *Haar Cascade* (também implementado na OpenCV).

A detecção do uso de máscaras em imagens, por outro lado, foi possível por conta das várias técnicas de reconhecimento facial existentes, mas que foram aplicadas em imagens com pessoas usando ou não máscara. Assim, o treinamento é usualmente supervisionado, com base nas imagens e a classificação de estar usando máscara ou não. Pode-se citar a utilização de Redes Neurais

Convolucionais (BAPIRAJU, AND KAUSHIK, 2021), SVM (BRAIN MENTORS PVT. LTD, 2021), assim como o classificador *Cascade* puramente, com base se a boca da pessoa foi detectada na imagem entre outras regras que produzem sistemas menos eficientes (PATIL, 2020).

Uma Rede Neural Convolucional possui um algoritmo de aprendizado profundo que pode captar uma imagem de entrada, atribuir importância (pesos e vieses que podem ser aprendidos) a vários atributos da imagem e ser capaz de diferenciar um do outro (DATA SCIENCE ACADEMY, 2022). Por outro lado, o SVM trabalha com análise de regressão (estatística), de modo a traçar uma linha de separação entre dados de duas classes.

A avaliação da face se usa máscara (etapa 2), foi realizada comparando as mesmas imagens de testes para uma CNN e uma SVM. Além disso, foram utilizados o mesmo banco de imagens para treinamento supervisionado. Mais detalhes sobre esse ensaio estão na próxima seção.

RESULTADOS E DISCUSSÃO

Em relação à detecção facial (etapa 1), verificou-se que um ponto importante a ser avaliado é a distância em que o método consegue detectar uma face numa sala de aula. Os testes foram feitos com dois métodos de detecção e em um único local com boa iluminação, para conseguir os melhores resultados. Para encontrar a distância máxima, o próprio pesquisador aluno se afastou até o ponto que não havia mais detecção e capturou a foto (figuras 2 e 3). A tabela 1 apresenta a distância máxima em que o método computacional pôde detectar a face, comparando o *Haar Cascade* e o DNN.

TABELA 1. Distância da câmera em que o método para de detectar a face

| | <i>HAAR CASCADE</i> | DNN |
|------------------------------|---------------------|-----|
| Distância Max. Medida | 1m | 4m |

FIGURA 2. Foto com a distância máxima que o método *Haar Cascade* pode detectar a face (1 metro)



FIGURA 3. Foto com a distância máxima que o método DNN pode detectar a face (4 metros)



O método *Haar Cascade* é bem simples de implementar a detecção facial, mas teve resultados menos satisfatórios em termos da distância e em relação a resposta menos regular alterando-se a pose, como por exemplo, de lado, ou a pessoa, o que Jadhav et al. (2021) relatam como “muitos falsos positivos, não trabalha com faces de lado e faces com diferentes oclusões”. O método DNN, por sua vez, demanda um pouco mais de complexidade para implementação em relação ao *Haar Cascade*, mas alcança maior distância e respostas mais regulares independente da pose e da pessoa.

Em relação à detecção se a pessoa está usando máscara (etapa 2), foram analisados e comparados o método SVM e Redes Neurais Convolucionais, que foram implementados em Python utilizando as bibliotecas scikit-learn (SVM) e TensorFlow-Keras (CNN). Neste experimento a detecção de face contou com o método DNN (uma vez que se saiu melhor nos testes de distância) e o mesmo banco de imagens (fotos) para o treinamento supervisionado, totalizando 170Mb, com 2.165 fotos de pessoas com máscaras e 1.930 pessoas sem máscara. Os testes foram realizados utilizando 20 fotos do aluno pesquisador, que não foram inseridas nos processos de treinamento. Assim, para cada foto testada, foi emitido o resultado se estava correto ou não.

A entradas utilizadas são matrizes tridimensionais com altura e profundidade, determinada pela quantidade de canais de cores. Em geral as imagens utilizam três canais, RGB, com os valores de cada pixel.

O método SVM trabalha com a separação linear de dados, o que segundo Ferreira (2021), gera uma quantidade maior de falsos positivos. Os testes apresentaram 87,5% de resultados falsos positivos quando a face detectada está sem máscara. Para esse método houve acerto para fotos com pessoas com máscara em 80% dos casos.

O segundo modelo, que utiliza Redes Neurais Convolucionais, apresenta resultados ótimos para análise de dados mais complexos, totalizando 100% de acerto tanto utilizando ou não máscaras nas fotos de teste.

A Tabela 2 resume os resultados obtidos na comparação dos dois métodos. As figuras 4 e 5 apresentam as telas de detecção correta tanto para sem máscara como com máscara.

TABELA 2. Taxa de acertos média em testes realizados

| | COM MÁSCARA | SEM MÁSCARA |
|-----|-------------|-------------|
| SVM | 80,0% | 12,5% |
| CNN | 100,0% | 100,0% |

FIGURA 4. Detecção de face e se usa máscara

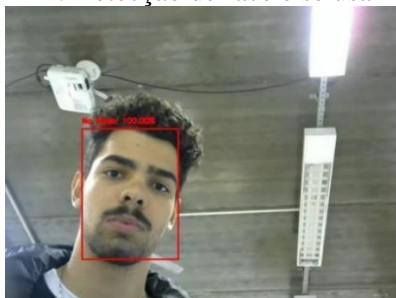
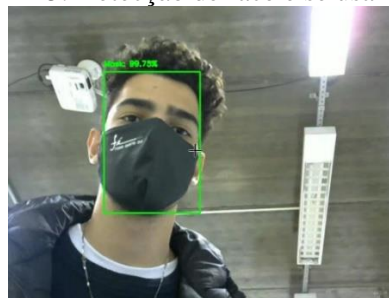


FIGURA 5. Detecção de face e se usa máscara



CONCLUSÕES

Por fim alguns dos problemas que influenciam no resultado da resposta são por questões de distância, iluminação e pose de lado. A solução desses problemas depende de uma adequação do local para uma boa iluminação e que as pessoas possam passar próximas a câmera e de preferência maximizando a leitura frontal da face.

Ao observar os resultados do método SVM tem-se uma tendência para que a resposta seja com máscara, inviabilizando julgar corretamente a análise desse método e indica que é preciso um trabalho de pesquisa para tentar corrigir esses problemas.

Por fim, para a aplicação desejada, que seria detectar se pessoas estão sem máscara numa sala de aula, a melhor opção foi a união do método DNN para detecção de face e Redes Neurais Convolucionais para detecção de máscara.

AGRADECIMENTOS

Os autores agradecem ao Câmpus Sertãozinho do IFSP que está financiando a bolsa de iniciação científica da modalidade PIBIFSP do referido aluno participante do projeto. Também agradecem ao Instituto Federal de Educação, Ciência, Tecnologia de São Paulo que está financiando os recursos deste projeto por meio do Edital Nº 329/2021 - "Edital de Fomento para Criação de Centros de Pesquisa e Inovação no IFSP".

REFERÊNCIAS

ADOUANI, Amal; HENIA, Wiem Mimoun Ben; LACHIRI, Zied. Comparison of Haar-like, HOG and LBP approaches for face detection in video sequences. In: 16TH INTERNATIONAL MULTI-CONFERENCE ON SYSTEMS, SIGNALS & DEVICES (SSD), Istanbul, 2019. **Proceedings [...]**. New York: IEEE, 2019, pages 266–271.

AL-IMRAN; SHAMS, Baniamin; NASIM, Faysal Islam. **Analysis on Face Recognition based on five different viewpoints of face images using MTCNN and FaceNet**, 2019. Disponível em: <http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/12814/15201007%2c%2015301030%2c%2015201051_CSE.pdf?sequence=1&isAllowed=y>. Acesso em 09 set. 2022.

AGARWAL, Vardan. Face Detection Models: Which to Use and Why?, 2020. Disponível em: <<https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c>>. Acesso em: 25 de maio de 2022.

BAPIRAJU, Yalla Lakshmi Sai; KAUSHIK, Praveen. Face Mask Detection in Images using Haar Cascade Classifier. **International Research Journal of Modernization in Engineering Technology and Science**, v.3 n. 06, 2021. Disponível em: <https://www.irjmets.com/uploadedfiles/paper/volume3/issue_6_june_2021/13489/1628083526.pdf>. Acesso em: 09 set. 2022.

BRAIN MENTORS PVT. LTD. Face Mask Detection Using OpenCV in Python, 2021. Disponível em: <<https://brain-mentors.com/face-mask-detection-using-opencv-in-python/>>. Acesso em: abril de 2022.

DATA SCIENCE ACADEMY. Deep Learning Book, 2022. Disponível em: <<https://www.deeplearningbook.com.br/>>. Acesso em: 10 jan. 2022.

FERREIRA, Michael Henrique Souza. Reconhecimento Facial para Detecção de Emoções Utilizando Redes Neurais Convolucionais com TensorFlow, 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação) – Pontifícia Universidade Católica de Goiás, Goiânia, 2021. Disponível em: <<https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3704>>. Acesso em: 07 set. 2022.

IDE PyCharm 2021.2.3 (Community Edition). Disponível em: <<https://www.jetbrains.com/pt-br/pycharm/>>. Acesso em: 08 set. 2022.

JADHAV, Anushka; LONE, Surabhi; MATEY, Sagarika; MADAMWAR, Tanvi; JAKHETE, Sumitra. Survey on Face Detection Algorithms. **International Journal of Innovative Science and Research Technology**, Jaipur, V. 6, n. 2, p. 291-297, 2021. Disponível em: <<https://ijisrt.com/assets/upload/files/IJISRT21FEB272.pdf>>. Acesso em 09 set. 2022.

MEGANHA, Felipe. Rodando uma simples DNN no seu PC com OpenCV, 2020. Disponível em: <<https://felipemeganha.medium.com/rodando-uma-simples-dnn-no-seu-pc-com-opencv-9b2671514b12>>. Acesso em: setembro de 2022

PATIL, Chinmay. Haar Cascades On Face Mask Detection. **International Journal of Scientific Research & Engineering Trends**, v. 7 n. 2, 2020. Disponível em: <https://ijsret.com/wp-content/uploads/2021/03/IJSRET_V7_issue2_208.pdf>. Acesso em: 08 set. 2022.

PIXABAY. Disponível em: <<https://pixabay.com/pt/vectors/avatar-garoto-desenho-animado-2027366/>>. Acesso em: 09 set. 2022.

ROSEBROCK, Adrian. Face detection with dlib (HOG and CNN), 2021. Disponível em: <<https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>> Acesso em: 12 abr. 2022.

S, Sharma; SHANMUGASUNDARAM, Karthikeyan; RAMASAMY, Sathees Kumar. FAREC — CNN based efficient face recognition technique using Dlib. IN: 2016 INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION CONTROL AND COMPUTING TECHNOLOGIES (ICACCCT), 2016, Ramanathapuram. **Proceedings [...]**. New York: IEEE, 2016, pages 511–518.

VIOLA, Paul; JONES, Michael. Rapid object detection using a boosted cascade of simple features. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2001, Kauai. **Proceedings [...]**. New York: IEEE, 2001, pages 511–518.