

Universidade Federal do ABC



Processamento de Vídeo

Detecção do uso de Máscara Facial

Gabriel Nigro - 11201810443

Jessi Leandro - 11201810509

Joannis Basile - 11201810529

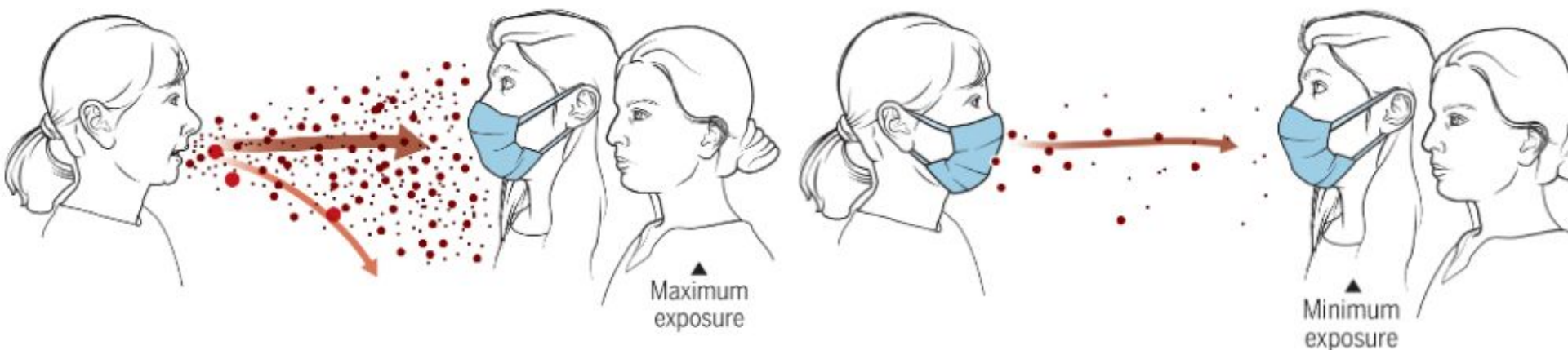
Contextualização

Em março de 2020 fomos acometidos pela pandemia originada pelo Sars-CoV-2, o Covid-19. Devido a isso, a maneira com que estávamos acostumados a viver mudou de maneira drástica, agora com isolamentos e uso de materiais para medida preventiva contra o vírus, sendo o uso da máscara facial que cubra as vias aéreas o mais marcante.



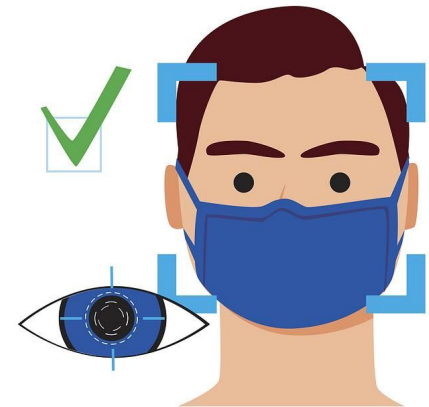
Contextualização

Mesmo com toda essa crise instaurada, as pessoas ainda recusam-se a usar a máscara, colocando assim a vida de muitos terceiros em risco, visto que a máscara é o método mais efetivo para impedir a propagação do vírus do infectado para os saudáveis.



Objetivo

- Criar uma aplicação web que, a partir de algoritmos de inteligência artificial, identifique o uso de máscara facial, operando no controle de acesso em estabelecimentos e locais de uso comum.
- Além disso, temos como objetivo aprender e praticar técnicas de Processamento de Vídeo, de ML e AI



Cenário

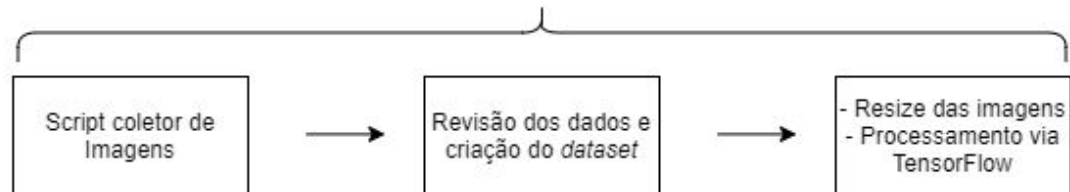
Local: Realizar a detecção de máscara a partir de uma webcam e do navegador.

Ideal: Aplicação ser utilizada em locais de grande movimentação, como em catracas de transporte público, totens de entradas de shoppings e comércios.

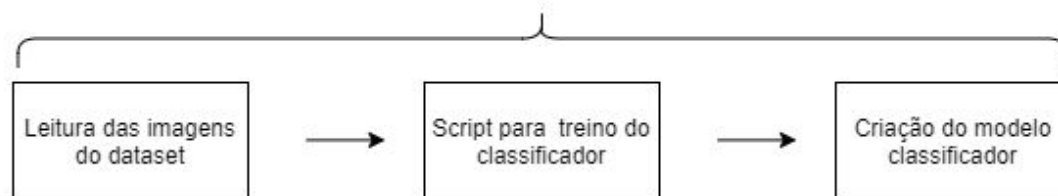


Fluxo de trabalho

Pré-processamento de dados



Treinamento do classificador

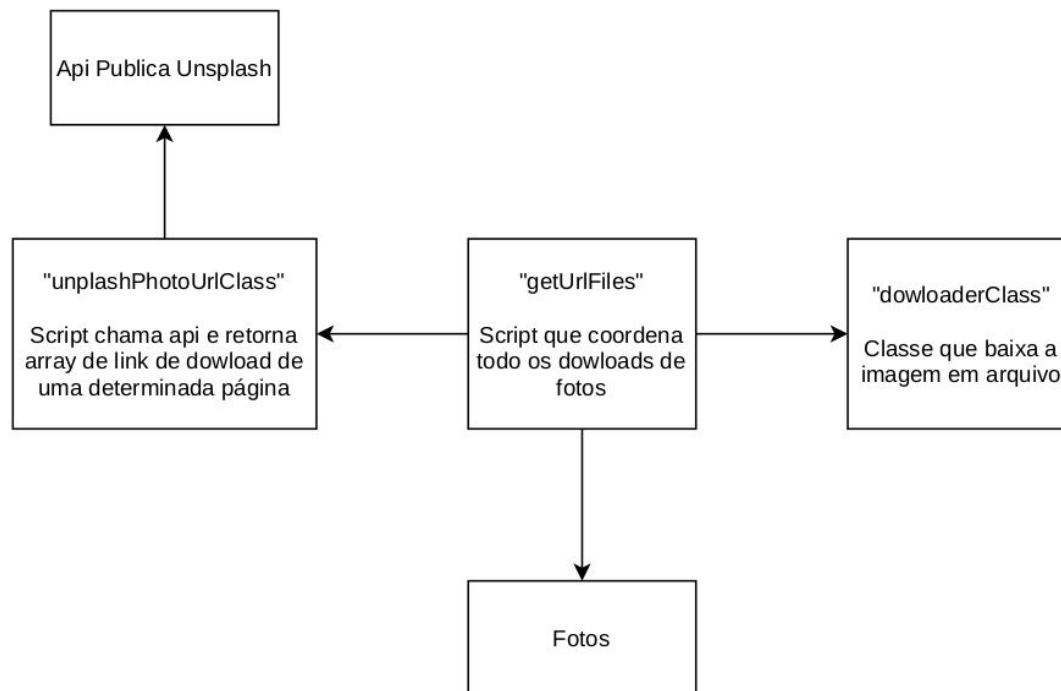


Testes e Análise de Resultados



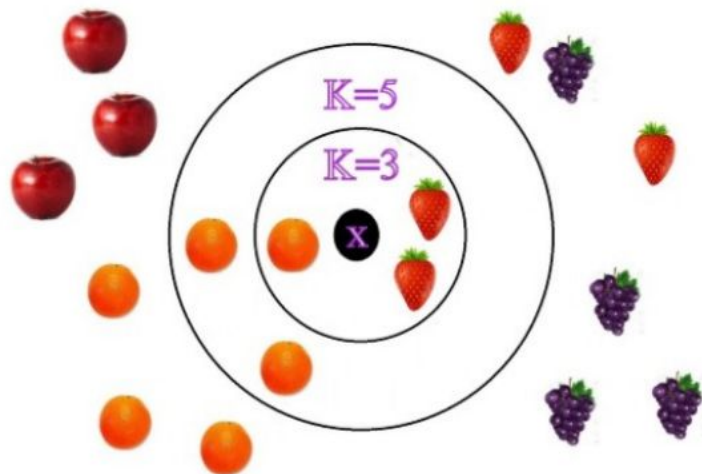
Desenvolvimento

Corpus: Para confecção da aplicação é necessário criar o corpus que será analisado pelos algoritmos de inteligência artificial.



Desenvolvimento

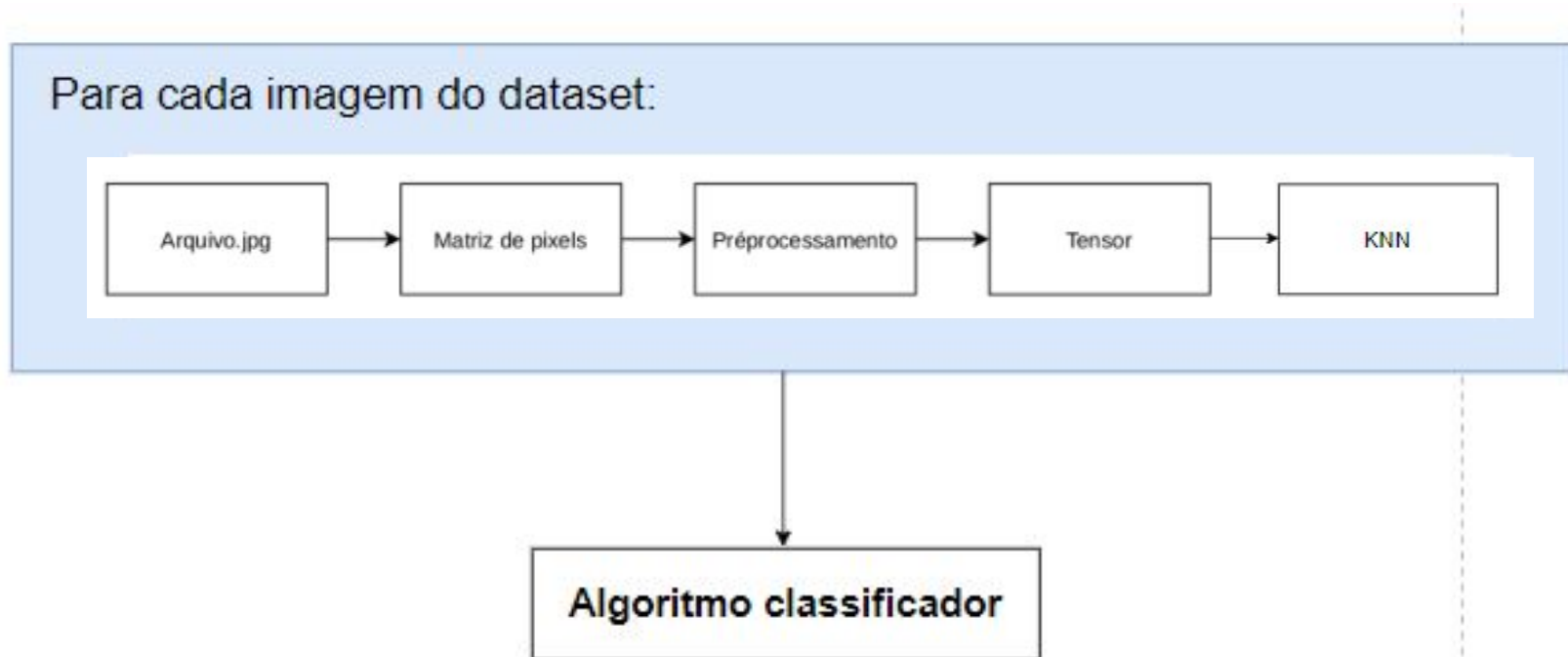
Metodologia: O algoritmo de inteligência artificial utiliza a técnica *KNN* - *K Nearest Neighbor*, o qual a partir das distâncias euclidianas e a posição dos pontos, adquiridos de cada foto, é realizada a predição do novo ponto, via agrupamento dos pontos vizinhos mais próximos.



Desenvolvimento

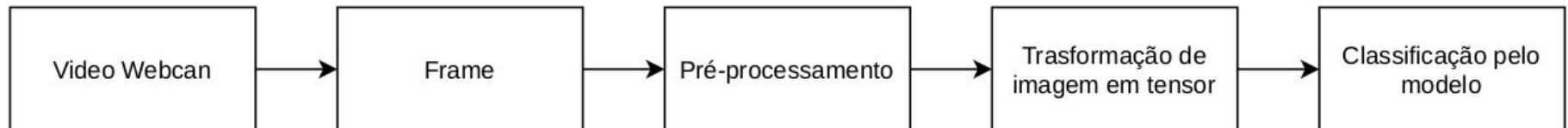
- **Canvas & Tensor-flow:** presença de funções nativas para interação com o navegador, para linearidade de análise entre front/back-end
- **MobileNet - Keras:** capaz de ler/analisar os dados e transformá-los em um tensor.
- **KNN:** imputação dos tensores, especificando a presença ou não de máscara em cada tensor, criando um modelo de aprendizagem. Esse processo foi realizado para todo o dataset (1400 imagens)
- **Algoritmo resultante:** Extraído um JSON de todo o processo, resultando no algoritmo a ser utilizado pelo classificador

Desenvolvimento



Funcionamento do sistema

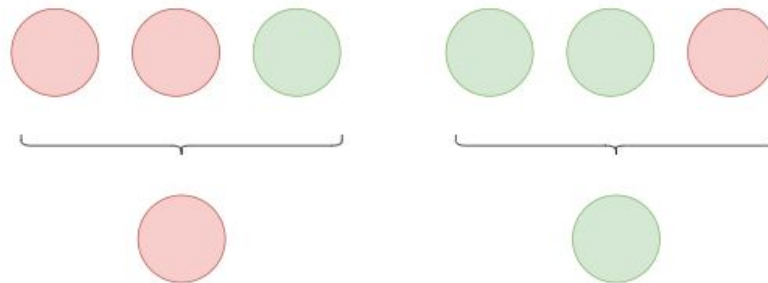
Visão geral: O projeto consiste numa aplicação web que herda o modelo classificador, pré-processando e analisando os frames obtidos via captura de vídeo e, assim, classificando-os para determinação do controle de acesso



Funcionamento do sistema

Por se tratar de um vídeo, nem todo frame é capturado com qualidade, pois a pessoa pode estar se movimentando, resultando em frames desfocados, embaçados ou tremido.

Estratégia: São capturados 3 frames a cada segundo (otimização do processamento), os quais são transformados em tensores e analisados um a um pelo classificador. Assim, é feita uma estratégia de “melhor de três”, na qual a maioria dos resultados indicará o controle de acesso positivo ou negativo.



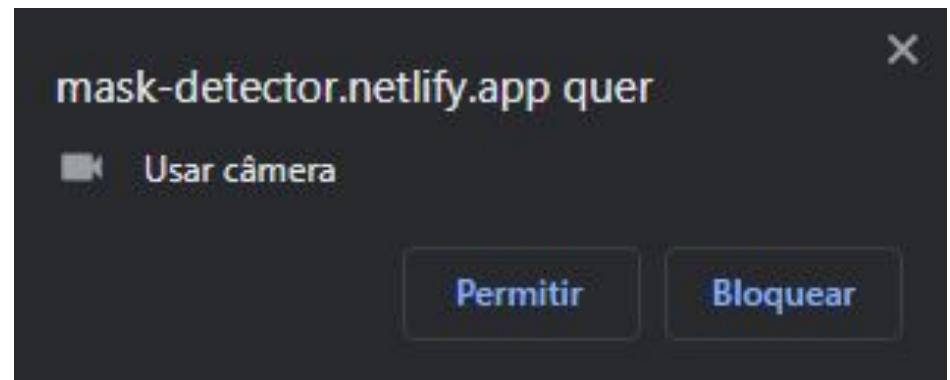
Funcionamento do sistema

Aplicação Web:

- Utiliza o framework Angular para gerar todo o bundle da aplicação.
- hospedada no [Netlify](https://mask-detector.netlify.app/), logo para utilizá-la basta acessar a seguinte url: <https://mask-detector.netlify.app/>

Utilização:

Basta clicar em
“**Permitir**” no pop-up
aberto pelo navegador



Funcionamento do sistema

User Interface (UI):

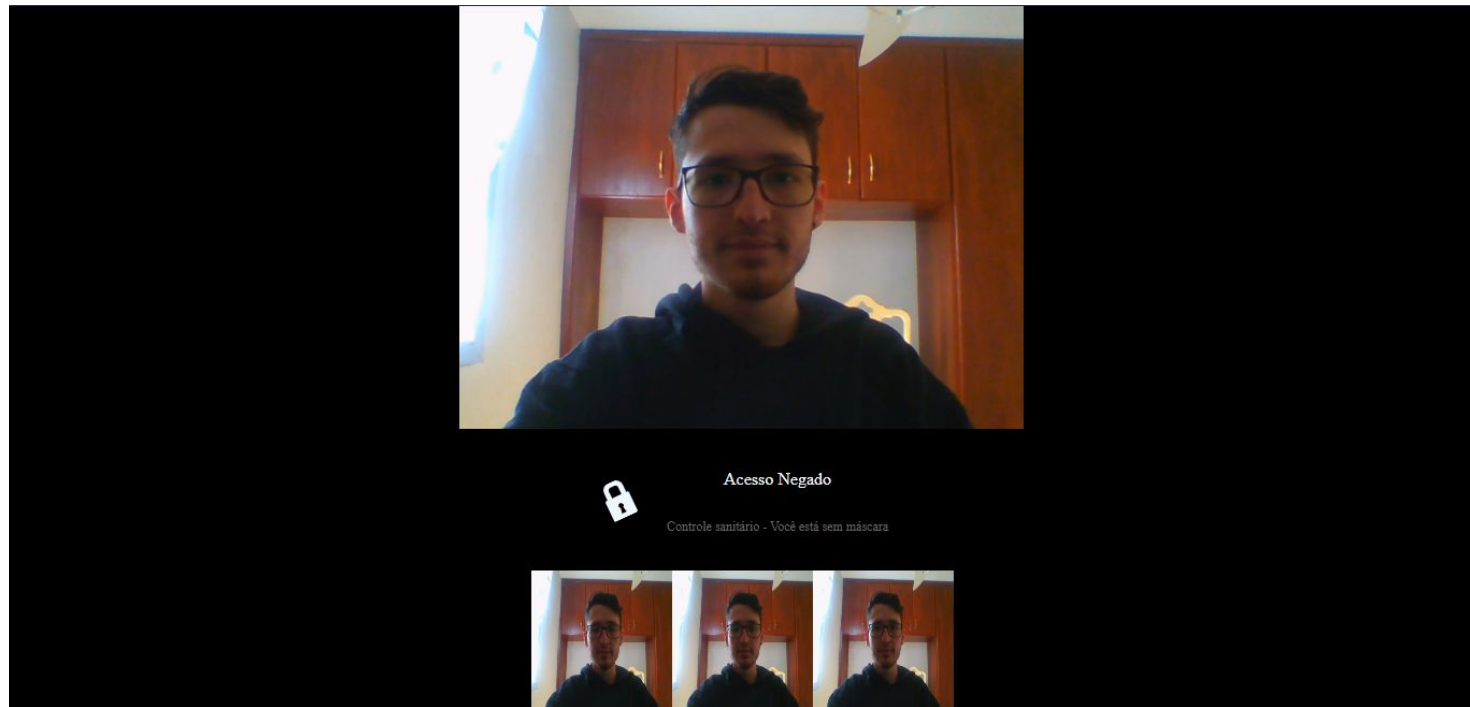
- Coleta do vídeo
- Processamento dos dados recebidos
- Execução do controle de acesso



Funcionamento do sistema

User Interface(UI):

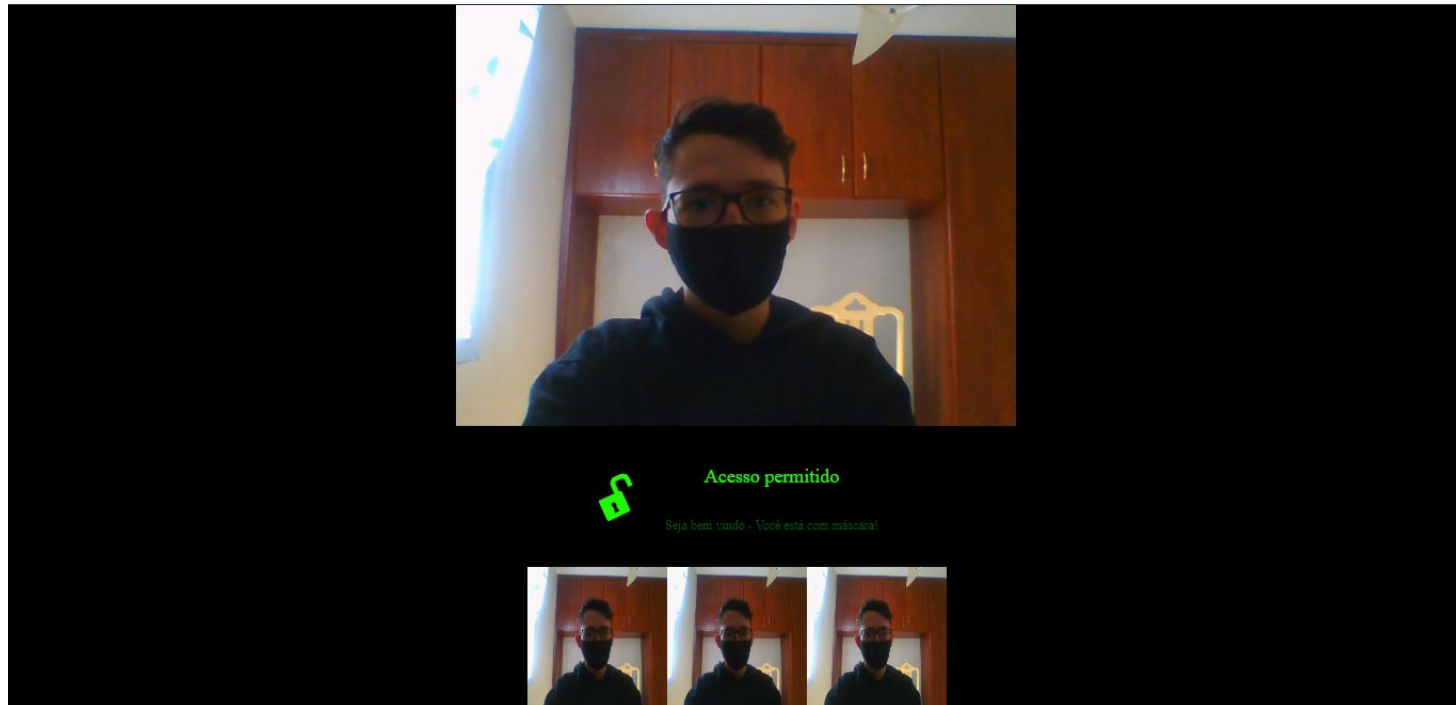
- Caso negativo (sem máscara)



Funcionamento do sistema

User Interface(UI):

- Caso positivo (com máscara)



Resultados

A aplicação comporta-se de maneira satisfatória, sendo possível identificar a presença, ou não, da máscara facial.

O algoritmo classificador obteve melhor performance para imagens, pois obter frames de um vídeo é suscetível a complicações, como dito anteriormente.

Entretanto, da maneira como está hoje, não seria possível utilizá-lo como controle de acesso 100% confiável, pois há diferentes cenários que podem gerar falhas, os quais serão abordados na análise.



Análises

É visto que, diante de cenários adversos, como falta de luminosidade, presença de mais de uma pessoa, falta ou excesso de proximidade da câmera, a aplicação não é capaz de identificá-los e tratá-los.

Isso ocorre pois o dataset utilizado para treino possui fotos de boa qualidade, boa luminosidade, até mesmo de certa forma artísticas. Logo, o classificador irá buscar por informações correlatas aos cenários em que foi confeccionado.

Para resolver esse problema seria necessário um dataset maior, o que acarretaria numa maior necessidade de processamento, o que é uma grande complicação para o computador comum que possuímos.



Análises

Há também cenários de utilização incorreta da máscara, no qual a aplicação não possui inteligência para lidar com. Da mesma forma, seria necessário incluir imagens do uso incorreto de máscara no dataset, e classificá-los como “no-mask”.



Conclusão

Com isso concluimos que, para a confecção de um classificador com altas taxas de acurácia e precisão, as quais são mandatórias para obtermos um alto nível de confiança e o utilizarmos como controle de acesso, é necessário a construção de um dataset contendo situações adversas, para “blindá-lo” contra falhas.

A aplicação estruturada serve como uma POC (*proof of concept*) de que é possível identificar a presença de máscara através de um streaming de vídeo, trazendo a oportunidade de ser utilizada para o bem de todos e, assim, diminuir a transmissão do vírus.



Lista de arquivos

Aplicação Web:

MASKDETECTOR

- angular.json
- karma.conf.js
- package.json
- README.md
- tsconfig.app.json
- tsconfig.json
- tsconfig.spec.json

src

- index.html
- main.ts
- polyfills.ts
- styles.css

app

- app.component.css
- app.component.html
- app.component.spec.ts
- app.component.ts
- app.module.ts
- modelo-1.json

assets

- cadeado-aberto.svg
- cadeado-fechado.svg

environments

- environment.prod.ts
- environment.ts

A lista de arquivos foi obtida a partir da execução do comando “*tree*” presente no Windows/Linux.

Repositório do GitHub contendo os arquivos fontes: [maskDetector](#)



Lista de arquivos

Classificador & Extrator: Para confecção do extrator e do classificador foram criados scripts unitários, responsável pela importação dos módulos, leitura do dataset, pré-processamento e retorno do modelo classificador.

```
trainingMaskClassifier.js X
14 // Usando binários de c no kernel -- only linux
15 const tf = require('@tensorflow/tfjs-node');
16
17 // mobilenet transforma imagem em vetor
18 const mobilenetModule = require('@tensorflow-models/mobilenet');
19 const knnClassifier = require('@tensorflow-models/knn-classifier');
20
21 const fs = require('fs')
22
23 // lista todos os arquivos do diretório
24
25 // readsync ele lista todos os nomes dos arquivos no diretório
26 const filesMask = fs.readdirSync('with_mask');
27 const filesFace = fs.readdirSync('without_mask');
28
```

Parte do código do arquivo classificador

Repositório do GitHub
contendo os arquivos
fontes: [treinador](#)



Referências bibliográficas

GETTY IMAGES. Unsplash: Beautiful Free Images & Pictures, 2021. Documentation. Disponível em: <https://unsplash.com/documentation>. Acesso em: 06 de jul. de 2021

USO DE MÁSCARA REDUZ CHANCE DE INFECÇÃO POR CORONAVÍRUS EM 87%, DIZ ESTUDO DE UNIVERSIDADES DO RS. G1 - Globo, 2021. Disponível em: <https://g1.globo.com/rs/rio-grande-do-sul/noticia/2021/03/05/uso-de-mascara-reduz-chance-de-infeccao-por-coronavirus-em-87percent-diz-estudo-de-universidades-do-rs.ghtml>. Acesso em: 06 de jul. de 2021

ROSEBROK, Adrian. COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning, 2020. Disponível em: <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/>. Acesso em: 06 de jul. de 2021

BHANDARY, Prajna. Observations, 2020. Disponível em: <https://github.com/prajnasb/observations>. Acesso em: 06 de jul. de 2021

TensorFlow. Construa uma rede neural para executar a classificação (Coding TensorFlow em português), 2019.(07m30s). Disponível em: <https://www.youtube.com/watch?v=qaIX5ZLiR1g>. Acesso em: 06 de jul. de 2021

YASTREMSKY, David. Creating a Powerful COVID-19 Mask Detection Tool with PyTorch, 2021. Disponível em: <https://towardsdatascience.com/creating-a-powerful-covid-19-mask-detection-tool-with-pytorch-d961b31dcd45>. Acesso em: 06 de jul. de 2021

Filipe Luz, "Algoritmo Knn para classificação". Disponível em: <https://inferir.com.br/artigos/algoritmo-knn-para-classificacao/>. Acesso em: 05/08/2021