FUNDAÇÃO GETÚLIO VARGAS – FGV MANAGEMENT

MBA Executivo em Economia e Gestão – Business Analytics e Big Data

Disciplina: Desafios e Requisitos dos Projetos Analíticos (1S/2020)

PROJETO ANALÍTICO EM GRUPO

DANIELE F. LYRA

EDUARDA ROUSSEAU

GABRIELA LEA ALVES

DEEP LEARNING PARA O RECONHECIMENTO DO USO DE MÁSCARA DE PROTEÇÃO FACIAL

SÃO PAULO

2020

RESUMO

Redes neurais são utilizadas para identificação e reconhecimento de imagem desde sua popularização no mercado, criando espaço para o desenvolvimento de novas arquiteturas como convolutivas e profundas. Em vista da atual pandemia e as mudanças em ambientes sociais, este projeto propõe o uso de uma rede neural convolucional profunda, VGG19, para a detecção do uso de máscaras faciais protetoras para evitar a propagação da COVID-19. Para a criação do conjunto de imagens foi utilizado a técnica de web scraping em Python e, testes foram feitos para determinar todos os cenários que precisamos incluir no conjunto para que o modelo tivesse as classes relevantes para a identificação, classes classificadas como Mask e No Mask. Como complemento, para viabilizar o teste do modelo em cenários reais, utilizamos features das bibliotecas OpenCV e MTCNN junto ao modelo desenvolvido, o que permitiu a aplicação do modelo à imagens em condições de luminosidade e contraste reais. Tivemos resultados satisfatórios para o conjunto treinado como apresentamos na seção respectiva, mas existe a necessidade de melhorar a constância do modelo e a análise de ambientes com pouca luminosidade.

Palavras-chave: Aprendizagem Profunda, Rede Neural Convolucional, OpenCV, MTCNN e COVID 19.

ABSTRACT

Neural networks have been the breakthrough for image recognition and identification since its gained relevance on the market, creating the scenario for the development for new architectures: deep and convolutional networks. Considering the current global pandemic and social environment, this project uses a convolutional deep neural network, VGG19, to detect the use of facial protective masks to avoid COVID-19 infection. The image dataset was created through a python developed web scraper and tests were conducted to make sure all relevant scenarios were included to identify our classes classified as Mask and No Mask. In addition, to test the model on real scenarios, features of OpenCV e MTCNN libraries were included to the model, which made possible to test it on real conditions of light and contrast. Satisfactory results were obtained to our dataset as presented on given section, even though further development is important regarding prediction's constancy in the model and in the analysis of low-light environments.

Keywords: Deep Learning, Convolutional Neural Network, OpenCV, MTCNN e COVID 19.

LISTA DE FIGURAS

FIGURA 1 - Exemplos de expressões faciais e suas características. 7
FIGURA 2- Estrutura básica de uma CNN tradicional. 8
FIGURA 3- Exemplo do uso do OpenCV. 19
FIGURA 4-Exemplo de detecção de face com o método MTCNN. 20
FIGURA 5-Apresentação dos resultados do modelo. 24
FIGURA 6- Reconhecimento facial de um grupo de pessoas. 25
FIGURA 7- Reconhecimento de diversidade de gênero. 26
FIGURA 8-Reconhecimento com a mão na face. 26
FIGURA 9- Reconhecimento com objetos na face. 27
FIGURA 10-Reconhecimento de face em ambiente de pouca iluminação. 28
FIGURA 11- Representação gráfica da Matriz Confusão. 30
FIGURA 12- Avaliação do treinamento do modelo 31

LISTA DE ABREVIAÇÕES

AI – Artificial Intelligence

AMIB- Associação de Medicina Intensiva Brasileira

ANN – Artificial Neural Network

API - Application Programming Interface

BDA – Big Data Analytics

CAFFE - Convolutional Architecture for Fast Feature Embedding

CNN- Convolutional Neural Network

COVID 19 - Coronavírus SARS-CoV-2

DL – Depp Learning

GPU - Unidade de Processamento Gráfico

ILSVRC – ImageNet Large Scale Visual Recognition Challenge

MLP - Multi Layer Perceptron

MTCNN – Multitask Cascaded Convolutional Networks

OMS – Organização Mundial da Saúde

OPENCV - Open Source Computer Vision Library

RGB - Red, Green, Blue

RNA – Rede Neural Artificial

RNC - Rede Neural Convolutiva

SIS – Sistemas de Informação em Saúde

UFBA - Universidade Federal da Bahia

URL - Uniform Resource Locator

UTI - Unidade de terapia intensiva

SUMÁRIO

1. INTRODUCÃO 1

2. REVISÃO DA LITERATURA 5

2.1 Inteligência artificial aliada a prevenção da Covid-19 5

2.2 Aprendizado de máquina aplicados em reconhecimento facial..... 6

3. MÉTODO 11

3.1 Seleção e apresentação das variáveis do estudo 11

3.2 Rede Neural com Teachable Machine..... 12

3.3 Ajustes ao conjunto de dados e validação no modelo Teachable Machine..... 13

3.4 Rede Neural Convolucional pré-treinada VGG19..... 13

3.5 Implementando o modelo no reconhecimento facial por vídeo 18

3.6 Elaborando o modelo na VGG 19 com nosso conjunto de dados e aplicá-lo a OpenCV e MTCNN...... 20

4. AMOSTRA E RESULTADOS 22

4.1 Apresentação descritiva da amostra do estudo..... 22

4.2 Apresentação dos Resultados..... 23

5. ANÁLISES E DISCUSSÃO 29

5.1 Métricas de desempenho do modelo..... 29

5.2 Avaliação do treinamento do modelo..... 31

6. LIMITAÇÕES DA PESQUISA / ESTUDOS FUTUROS 32

7. CONCLUSÃO 33

REFERÊNCIAS..... 35

APÊNDICES / ANEXOS..... 39

1. INTRODUCÃO

A Inteligência Artificial (AI) tem crescido exponencialmente e com o aprendizado de máquina tem impactado o modo de vida das pessoas facilitando a rotina social, melhorando o desempenho de negócios e possibilitando inovações na saúde, indústria e comércio.

O termo aprendizado de máquina se refere ao conjunto de técnicas para construir sistemas capazes de adquirir conhecimento de forma automática (MONARD e BARANAUSKAS, 2003). E esse aprendizado pode ser categorizado em aprendizado supervisionado ou aprendizado não supervisionado da máquina (HAYKIN, 2008).

O Machine Learning, de forma simplificada, é o uso de algoritmos para coletar dados, aprender com eles, e então fazer uma determinação ou predição sobre alguma coisa, por exemplo, para as empresas as vendas do próximo ano. Através dos anos, incluiu árvores de aprendizado reforçado, redes Bayesianas, rede neurais artificiais e entre outros.

As redes neurais artificiais foram inspiradas pelo sistema nervoso central do cérebro humano, que são capazes de forma não linear e paralelo de organizar seus constituintes estruturais, conhecidos como neurônios, de modo a realizar reconhecimento de padrões, percepção e o controle motor.

O Deep Learning, aprendizado em profundidade, é uma das técnicas para implementar o machine learning, e seus modelos envolvem conhecimentos de matemática, estatística, programação, visão computacional, pré-processamento de imagens e entre outras áreas. As técnicas de deep learning têm possibilitado soluções para muitos problemas de reconhecimento de imagens, reconhecimento de fala e processamento de linguagem natural.

O reconhecimento facial é uma das tarefas que utiliza as redes neurais profundas e as técnicas de deep learning em suas soluções. O reconhecimento facial é um sistema biométrico de funcionamento pela retina, que assim como os olhos são únicas ao rosto. Os sistemas de reconhecimento precisam trabalhar com uma alta taxa de precisão, algo que ainda não foi alcançado pela ciência (ÇARIKÇI; ÖZEN, 2011).

As redes convolucionais são, atualmente, as arquiteturas de rede neurais que alcançam a melhor taxa de acertos na classificação de imagens, e os métodos de aprendizado de máquina supervisionados são os mais utilizados para casos de classificação. Combina três arquiteturas, pesos compartilhados ou replicados, receptores locais e, às vezes, subamostras

espaciais ou temporais. Tais elementos, permitem que sejam identificados pontas, curvas e limites, aprendizado este que é combinado ao longo da rede (LECUN & BENGIO, 1995).

Temática do Trabalho

Rede Neural Deep Learning Aplicada ao Reconhecimento Facial

Pergunta/Questão de Pesquisa

Como usar uma rede neural para o reconhecimento do uso de máscara de proteção facial durante a pandemia?

Justificativa do Fenômeno

O Deep Learning é uma classe de algoritmos de aprendizado de máquina baseados principalmente nos princípios de redes neurais artificiais. Tornou-se uma solução ideal para problemas que antes era solucionados apenas pelo cérebro humano, seu potencial de aplicação é imenso tanto na academia quanto na indústria. Uma das possibilidades do uso de rede neural de aprendizado profundo é no reconhecimento preciso da fala e processamento de linguagem natural, auxiliando tecnologia para carros, casas e eletrodomésticos.

Cada vez mais, modelos são integrados com serviços essenciais e políticas públicas, um exemplo recorrente é estimar a densidade populacional de uma região com base na contagem de pessoas em multidões. A aplicação abrange diversas áreas como saúde, tráfego, segurança e urbanismo (SINDAGI & PATEL, 2017).

As redes neurais profundas, como as Redes Neurais Convolucionais (CNN), foram aplicadas em problemas de reconhecimento e obtiveram sucesso (CIRESAN, MEIER, MASCI & SCHMIDHUBER, 2012). A estrutura da CNN foi introduzida por Lecun para reconhecimento de face e dígito (LECUN & BENGIO, LECUN et al., 1998, LECUN et al., 2004), que demonstraram melhores resultados de reconhecimento do que outras metodologias, por exemplo, abordagens bayesianas gaussianas ou K classificadores de vizinhos mais próximos.

Modelos de redes neurais convolutivas adquirem aprendizado pelos dados que processam e sabemos que quanto maior o volume de dados, melhor o aprendizado resultante. Visando tal melhoria, são usadas técnicas de data augmentation que provaram ser eficientes quando a tarefa que será aprendida na rede é complexa ou a identificação a ser feita é de um objeto pequeno (ZOPH et al, 2019). Data augmentation consiste em aleatoriamente extrair

perspectivas diferentes das imagens do dataset, como por exemplo, uma rotação em 90 ou 180 graus, com intuito de fornecer maiores visões sobre o cenário para o modelo (CUBUCK et al. 2019).

Simonyan e Zisserman (2015) provaram que redes convolucionais muito profundas são eficazes para a classificação de imagens em larga escala. À medida que a profundidade aumenta, o desempenho da rede aprimora tarefas complexas de reconhecimento.

Justificativa do Contexto

Diante do atual cenário mundial de pandemia da COVID-19, torna-se recomendável, e até mesmo obrigatório em regiões do país, o uso de máscaras faciais protetoras para evitar o contágio da doença.

Mesmo com a recomendação da Organização Mundial da Saúde (OMS) para o uso de máscaras apenas para os casos em que já se constata a infecção pelo vírus, analisando que não existe no momento a comprovação da efetividade do uso indiscriminado, a "ausência de evidência de efetividade não equivale a evidência de inefetividade" desta prática, como pontua Garcia (2020).

Por se tratar de um vírus novo, ainda não foram realizadas pesquisas sobre o impacto de máscaras no contágio da COVID-19, porém estudos anteriores já consideraram o uso de máscaras como prevenção à outras doenças infecciosas e demonstraram que este comportamento, quando incentivado, previne e auxilia na redução da contaminação no período logo após a infecção pelo vírus, quando o indivíduo ainda demonstra poucos ou nenhum sintoma e tem alto potencial de transmissão. Ainda que não seja uma pesquisa inicial, o resultado do uso de máscaras, mesmo que não cirúrgicas, quando combinado com políticas públicas preventivas, poderiam ter forte impacto na taxa de transmissão observada, sendo este um resultado estatisticamente plausível, ainda que incerto (HOWARD, et al., 2020).

Objetivo Geral

O objetivo principal deste trabalho é utilizar um modelo de rede neural convolutiva para a identificação de pessoas com e sem a máscara facial protetora, através do reconhecimento de imagens.

Para atingir o objetivo proposto, utilizamos o Teachable Machine para treinar nossa rede neural de modo não supervisionado. Uma das técnicas de machine learning usada nessa plataforma para geração dos modelos é a rede neural convolucional e o transfer learning.

2. REVISÃO DA LITERATURA

O presente trabalho se propõe a contribuir no combate à disseminação da Covid-19 ao fazer uso de rede neurais convolucionais e deep learning para um sistema de monitoramento do uso de máscaras protetoras faciais pela população. Teremos três pilares a considerar, o uso de inteligência artificial no combate a Covid-19, o reconhecimento facial através do aprendizado de máquina e o impacto do Deep Learning nas redes neurais convolucionais.

2.1 Inteligência artificial aliada a prevenção da Covid-19

O artigo científico publicado em abril deste ano na revista Cadernos de Prospecção mantido pela Universidade Federal da Bahia (UFBA), com o tema sobre Epidemiologia e Prevenção da Covid-19 Aliada à Inteligência Artificial. Discute sobre os métodos preventivos e o uso da Inteligência Artificial (IA) para monitoramento e predição da incidência da doença auxiliando na contenção de disseminação da Covid-19.

Segundo os autores, é urgente a preparação de estratégias para o controle e prevenção da doença, dado que o Brasil carece de uma grande quantidade de leitos de UTI (Unidade de terapia intensiva) para enfrentar o surto da Covi-19 como mencionado pela Associação de Medicina Intensiva Brasileira (AMIB, 2020) e a medicina não está preparada em questões de vacinas e medicação.

Propõem estabelecer estratégias de fortalecimento de medidas protetivas e aconselham aliar o uso dos métodos de IA que poderá auxiliar no decréscimo do número de casos. Mencionam que, um dos métodos para auxílio preventivo seria estimar o número de infectados, as características dos doentes e o avanço do número de infectados pelo tempo é uma forma de obter informações para o controle do surto, entretanto, o uso de IA como sistema de monitoramento, controle e prevenção epidemiológica de doenças é um ramo atualmente pouco explorado (DE FREITAS, et al., 2020).

O estudo sugere a implementação de Sistemas de Informação em Saúde (SIS), com isso seria possível um controle de epidemias com o uso da Inteligência Artificial para prever cenários. Com base nas estatísticas dos dados registrados pelos órgão de saúde pelo mundo

e divulgados pela OMS, implementando o recurso de IA poderia melhorar a compreensão desses números, atualizando em tempo real os dados e as informações com base nas medidas restritivas implementadas e no monitoramento de aglomerações em locais específicos (DE FREITAS, et al., 2020).

Freitas, et al., sugerem que o uso de bases existentes, incluindo as bases geradas pela Covid-19 podem facilitar o aprendizado de máquinas na construção de sistema de IA, em que a integração dos dados epidemiológicos com outros dados sobre transporte e comportamento humano pode gerar sistemas de monitoramento e de análise preditivas mais eficientes e auxiliando na redução da mobilidade de pessoas infectadas e no isolamento e distanciamento social.

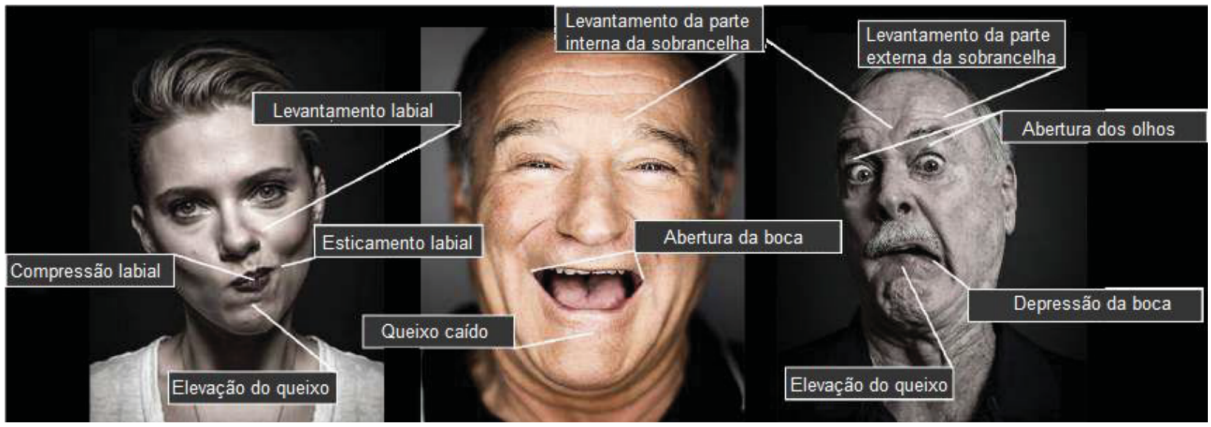
2.2 Aprendizado de máquina aplicados em reconhecimento facial

As expressões faciais é uma das formas de compreender o comportamento humano e a linguagem corporal associada, desempenham um papel significativo na interação interpessoal.

Segundo Andrade (2019), o reconhecimento automático de expressões faciais ainda é um desafio para os computadores, dado que é complicado identificar e separar as características relevantes de cada expressão. A capacidade visual humana é um sistema complexo que fornece respostas rápidas e detalhadas sobre o ambiente, e atrelados a capacidade humana de analisar e compreender informações sensoriais propiciam a tomada de decisão em um curto espaço de tempo.

De acordo com Zhang, et al. (2016), conforme citado por Andrade (2019), as expressões faciais são a forma de comunicação não verbal mais eficiente para a transmissão de emoções humanas, fornecem informações importantes que são codificadas sob diversas características faciais complexas sobre o estado mental, emocional e físico dos indivíduos envolvidos, como pode ser visto na FIGURA 1.1.

FIGURA 1 - Exemplos de expressões faciais e suas características.



Fonte: Adaptado de (ANDRADE, 2019, apud KIMMEL,2017)

O desenvolvimento de um sistema que permita aos computadores processarem informações visuais da mesma forma ou "similar" ao sistema visual humano, ou seja, compreendam conceitos de alto nível a partir de imagens ou vídeos digitais ainda é um desafio do ponto de vista computacional. Devido ao fato que para uma máquina o reconhecimento de expressões faciais simples e distintas é complexo e ainda ser capaz de lidar com imagens capturadas de ambientes com variações de iluminação, ruído e ângulo da face, sendo esta uma tarefa trivial ao ser humano por causa das diversas analogias e conexões criadas pelo cérebro (ANDRADE, 2019).

Um sistema deve ter métodos para aquisição, processamento e análise de imagens digitais para fazer a extração de dados e interpretá-los. A maioria dos sistemas atuais que faz o reconhecimento de expressões faciais apoiam-se em base de imagens estáticas ou de vídeos em duas dimensões (ANDRADE, 2019 apud LYONS, 1999; KANADE et al., 2000; HAPPY et al., 2017; THE FIELD PROJECT, 2018).

Um dos campos de estudo da visão computacional é o reconhecimento de expressões faciais que tem crescido muito nos últimos dez anos, pois apresentam maior força expressiva e facilidade de identificação, produzindo maior área de aplicação (ANDRADE, 2019 apud MAO, 2015).

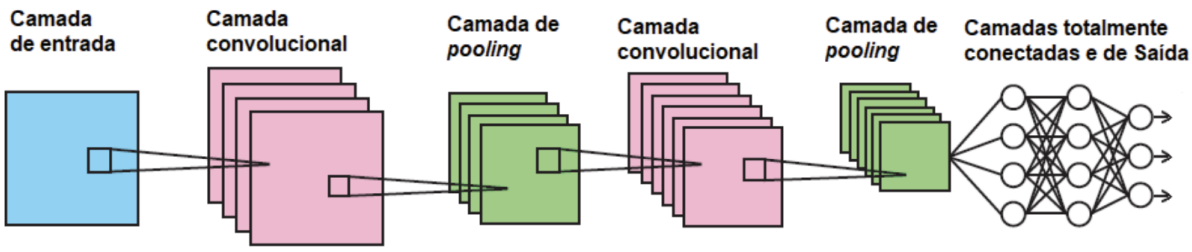
A primeira aplicação com sucesso de uma Redes Neurais Convolucionais (do inglês, Convolutional Neural Networks, CNNs) foi desenvolvida por Yann Lecun em 1998, com sete camada entre convoluções e fully connected. Desde então as técnicas de de aprendizado profundo se popularizou e ficou cada vez mais profundas e complexas, dentre os trabalhos

desenvolvidos Andrade (2019) cita o trabalho de Mayya et al. (2016) em que apresentaram um modelo de classificação por meio de uma rede neural convolucional profunda, sendo responsável pela extração e o processamento de características, usando a arquitetura Caffe (Convolutional Architecture for Fast Feature Embedding) o que permitiu alcançar um tempo significantemente menor para processamento dos dados comparado a outras técnicas mais antigas de aprendizado de máquina.

As CNNs são tipos especializados de Rede Neural Artificial (RNAs) que modelam sistemas por meio de circuitos (conexões) que podem simular o sistema nervoso humano, envolvendo a capacidade de aprender e agir diante de situações apresentadas, e adquirem conhecimento por meio da experiência e da observação. Seus princípios fundamentais foram retirados da neurociência, as CNNs foram desenvolvidas inicialmente para processamento de imagens, porém também foi implementado com sucesso em aplicações de processamento de vídeo e linguagem natural (ANDRADE, 2019).

A estrutura básica de uma CNNs é composta por três tipos diferentes de camadas, convolucionais, agrupamento (do inglês, *pooling*) e totalmente conectada e cada tipo de camada tem regras diferentes para a propagação de sinal para frente ou para trás (ANDRADE, 2019 apud ZAPLETAL, 2017). Andrade (2019) aponta para o fato de não haver regras definidas sobre como a estrutura de camadas individuais devem ser organizadas, porém tipicamente está estruturada em duas partes: extração de recurso que combina as camadas convolucionais e gera o vetor das características e a camada de classificação que é constituída por camadas conectadas, como pode ser visto na FIGURA 1.2.

FIGURA 2- Estrutura básica de uma CNN tradicional.



Fonte: Adaptado de (ANDRADE, 2019 apud STENROOS, 2017).

A primeira camada da rede é uma camada de entrada que é uma imagem, a operação de convolução é executada sobre a entrada com um filtro, chamado *kernel*, que por padrão possui um formato de quadrado e sua largura pode variar de 3 a N pixels. Em seguida, sobre cada elemento do vetor de entrada é criado pela convolução do *kernel* o mapa das características que será usado na próxima parte. Os valores da matriz do *kernel* viram parâmetros e são ajustados durante o treinamento da rede. A operação de multiplicação de uma camada de rede neural tradicional é substituída pela operação convolucional (ANDRADE, 2019 apud ZAPLETAL, 2017).

Cada uma das camadas convolucionais representam um neurônio da camada e cada um deles está ligado a uma região específica da imagem, sendo possível que uma parte de uma região pode ser entrada de dois ou mais neurônios. Geralmente, os neurônios da camada convolucional compartilham os parâmetros e são conectados apenas a uma região local de entrada (ANDRADE, 2019 apud LOPES, 2016).

2.3 Deep learning em Redes Neurais Convolucionais

Uma pesquisa publicada no Artificial Intelligence Review elenca a relevância das arquiteturas atuais de Deep Learning nos resultados exemplares obtidos atualmente com os modelos mais recentes.

Acredita-se que a capacidade de aprendizado de uma rede neural profunda esteja baseada nos múltiplos estágios de extração das representações de dados. A pesquisa aponta que inovações na arquitetura e diferentes combinações de funções de ativação e perda representam os principais avanços explorados (KHAN, Et al., 2020).

Parte do grande uso de modelos de redes neurais profundas vem da atenção que ganharam de empresas de renome no mercado. Empresas como Google, Facebook e Microsoft trouxeram a pesquisa para grupos internos e tem trabalhos publicados relacionados a novas arquiteturas e orientações de uso. Abordagens indicam que o ganho que redes neurais convolucionais apresentam ao aprender problemas complexos se dá ao combinar lineares e não lineares arquiteturas, com isso agregando a perspectiva de ver a representação em diversos níveis de abstração (KHAN, Et al., 2020).

Khan, et al.(2020) ressaltam que isso demanda processamento e memória o que torna as soluções pouco práticas para o cotidiano, uma vez que aplicações práticas mais corriqueiras como veículos autônomos ou robótica tem recursos limitados.

Iniciativas atuais propõem adequar as arquiteturas para ambientes com recursos restritos, como GoogleNet, ShuffleNet e ANTNet. O uso de blocos genéricos que podem ser incluídos em qualquer momento da aprendizagem de uma rede neural para direcionar o foco, também está presente nos estudos mais recentes, tornando o modelo mais assertivo e de mais fácil compreensão, agregando não apenas performance, mas ganhando ainda mais abrangência (KHAN, Et al., 2020).

A arquitetura VGG é uma dessas novas abordagens, trazendo camadas em módulos com diferentes níveis de profundidade. Nas versões com 19 e 16 camadas obteve-se resultados comparáveis aos ganhadores da competição ILSVRC nos anos de 2011 e 2012. (SIMONYAN & ZISSERMAN, 2015). A relevância desta arquitetura, que ficou em segundo lugar na competição de 2014, deve-se a simplicidade mantida com a profundidade que atinge (KHAN, Et al., 2020).

3. MÉTODO

O presente trabalho fez uso de imagens públicas para desenvolver um modelo em Rede Neural Convolutiva Profunda que fosse capaz de reconhecer o uso da máscara facial. Essas imagens foram recolhidas através da técnica de web scraping em Python e posteriormente tratadas manualmente pela equipe. Uma vez que tínhamos a base tratada, um modelo de Rede Neural foi ajustado a partir do Teachable Machine, foi feito o treinamento não supervisionado e ele exigiu inclusão de novas imagens em nossa base, uma vez que não englobava alguns cenários possíveis. Uma vez concluído a validação das imagens no modelo fornecido pelo Teachable Machine, aplicamos o mesmo conjunto ao modelo convolutivo VGG, pré treinado da API Keras com 19 camadas, buscando resultados ainda melhores. A seguir iremos nos aprofundar nas etapas do processo.

3.1 Seleção e apresentação das variáveis do estudo

A primeira fase teve como objetivo constituir uma base de dados para a alimentação da rede neural; pensando no modelo que planejamos desenvolver, tínhamos dois objetivos encontrar imagens de face de pessoas com e sem máscara protetora facial. Foram utilizadas imagens públicas disponibilizadas na web, encontradas através dos sites de busca Google e Bing.

Para a construção do conjunto de imagens no volume que precisávamos para treinar e validar o modelo, tomamos a iniciativa de automatizar o processo. Foi desenvolvido, para isso, um script para fazer o web scraping em Python que coletou as imagens desejadas dos sites de busca indicados anteriormente.

Para as imagens correspondentes a face de pessoas sem máscara facial protetora, utilizamos bases de artistas e figuras públicas. Já para o conjunto de imagens de pessoas com máscara, como não encontramos uma base de dados pública disponível, foi necessário coletar as imagens na web.

Após a coleta feita através da web scraping mencionado anteriormente, foi necessário fazer um tratamento das imagens descartando as que não possuíam características de máscara protetora facial, como recomendado pelos órgãos de saúde, e imagens que usavam animações ou manequins simulando uma face humana.

Ao finalizarmos o processo de coleta, checamos a diversidade de raça, gênero e idade em toda a base, buscando garantir a aderência e aplicação do modelo em diversas amostras.

3.2 Rede Neural com Teachable Machine

Próxima etapa consistia em elaborar um modelo a partir de nosso conjunto de dados para classificar as imagens. Usamos o **Teachable Machine,** ferramenta gratuita do Google baseada na web o que torna a criação de modelos de aprendizado de máquina rápida, fácil e acessível a todos. Como o processamento da rede neural era rápido, o treinamento pôde ser feito de forma fluida e os parâmetros ajustados diversas vezes até o que apresentaremos à frente.

Essa ferramenta usa o Tensorflow.js, uma biblioteca de aprendizado de máquina em Javascript, para treinar e executar os modelos que o usuário cria no seu navegador da web. A biblioteca foi criada sobre o Tensorflow.js e está disponível no repositório de código aberto do Teachable Machine Communities no GitHub.

Os modelos gerados pelo Teachable Machine usam uma técnica chamada aprendizado de transferência (transfer learning). Existe uma rede neural pré-treinada e, quando o usuário cria suas próprias classes, pode imaginar que suas classes estão se tornando a última camada ou etapa da rede neural. Especificamente, os modelos de imagem e poses estão aprendendo com os modelos MobileNet pré-treinados.

Nesta fase, inicialmente utilizamos 3.000 imagens coletadas usando a web scraping, criando assim apenas 02 classes: pessoas com máscara e pessoas sem máscara.

Foram feitos diversos testes visuais, com uso da webcam, para validar o modelo treinado; foi possível validar o modelo do uso de máscara ou não dos integrantes deste trabalho e fazendo alguns pequenos testes, tais como: colocar a mão na face, usar a máscara de forma incorreta, usar objetos diversos na frente da face, a ausência de pessoa frente a webcam, entre outros.

Após os testes das classes iniciais, verificamos inconsistências, como fato de que na ausência de uma pessoa o modelo entendia que o plano de fundo era uma pessoa com máscara, o que claramente não seria correto.

3.3 Ajustes ao conjunto de dados e validação no modelo Teachable Machine

Tínhamos duas situações que precisavam ser corrigidas, a classificação como pessoas usando máscara protetora de pessoas com mão na face e imagens de fundo. A forma mais assertiva de promover tal aprendizado é colocando estes cenários na amostra de treino da Rede Neural e foi o que buscamos nos próximos passos.

Fizemos uma nova coleta de imagens usando a web scraping, para as imagens de pessoas com a mão na face e para imagens de fundo (background) e passamos a ter um total de 5.570 imagens. Detalhando as imagens consideradas como plano de fundo, foram consideradas os mais diversos locais possíveis de circulação de pessoas, todas as imagens possuem apenas o ambiente e seus objetos.

De volta ao Teachable Machine, foi necessário criarmos mais uma classe para o plano de fundo (background) e treinar novamente para que o modelo aprendesse a presença de uma pessoa frente a webcam e só depois validar o uso ou não da máscara.

Portanto, ao final, foram criadas 03 classes para o treino do modelo na ferramenta, são elas:

* Classe 1: NoMask - Pessoas Sem máscara
* Classe 2: Mask - Pessoas Com máscara
* Classe 3: Background - Imagens de plano de fundo

Definição dos Hiperparâmetros:

* Epochs: 3
* Batch Size: 32
* Learning Rate: 0,001

3.4 Rede Neural Convolucional pré-treinada VGG19

A Rede Neural Convolucional pré-treinada VGG19 está disponível no Keras, uma API com estrutura de aprendizado profundo escrita em Python que roda sobre a plataforma de aprendizado de máquina TensorFlow e possui uma lista de modelos pré-treinados no conjunto de dados ImageNet.

Este conjunto de dados, com cerca de 14 milhões de imagens, é fornecido apenas miniaturas e URLs de imagens, de maneira semelhante ao que os mecanismos de pesquisa de imagens na web.

A portabilidade dos modelos é uma característica exclusiva dos modelos de deep learning. O VGG19 carrega pesos pré-treinados no ImageNet, sendo construído com o formato de dados 'channels_first' (canais, altura, largura) ou com o formato de dados 'channels_last' (altura, largura, canais) e tendo tamanho da entrada padrão de imagem de 224x224.

Funciona da seguinte forma: durante o treinamento, a entrada para o ConvNets é uma imagem RGB de tamanho fixo de 224 × 224. O único pré-processamento que feito é subtrair o valor RGB médio, calculado no conjunto de treinamento, de cada pixel. A imagem é passada através de uma pilha de camadas convolucionais (conv.), onde são usados filtros com um campo muito pequeno receptivo: 3 × 3 (que é o menor tamanho para capturar a noção de esquerda / direita, cima / baixo, centro).

Em uma das configurações, é utilizado filtros de convolução 1 × 1, que podem ser vistos como uma transformação linear dos canais de entrada (seguida pela não linearidade). O passo da convolução é fixo em 1 pixel; o preenchimento espacial de conv. entrada de camada é tal que a resolução espacial é preservada após a convolução, ou seja, o preenchimento é de 1 pixel para 3 × 3 camadas convolucionais (SIMONYAN & ZISSERMAN, 2015).

O agrupamento espacial é realizado por cinco camadas de pool máximo, que seguem algumas das conv. camadas (nem todas as camadas convolucionais são seguidas por pool máximo). O pool máximo é realizado em uma janela de 2 × 2 pixels, com passo 2. Uma pilha de camadas convolucionais (que tem uma profundidade diferente em diferentes arquiteturas) é seguida por três camadas totalmente conectadas (FC): as duas primeiras têm 4096 canais cada, a terceira executa 1.000 maneira a classificação ILSVRC e, portanto, contém 1000 canais (um para cada classe). A camada final é a camada soft-max. A configuração das camadas totalmente conectadas é a mesma em todas as redes (SIMONYAN & ZISSERMAN, 2015).

Os hiperparâmetros do modelo podem ser personalizados pelo usuário, como a quantidade de épocas do treino, o tamanho dos batches, a taxa de aprendizado, tamanho da

imagem e o random state. Algumas features também podem ser aplicadas ao modelo, para otimizá-lo. Os hiperparâmetros que utilizados posteriormente no modelo, foram:

- *Época (Epochs)* - uma época significa que cada amostra do conjunto de dados de treinamento foi alimentada pelo modelo de treinamento pelo menos uma vez. Se as épocas estiverem definidas como 50, por exemplo, significa que o modelo treinado funcionará 50 vezes todo o conjunto de dados de treinamento.

- *Taxa de Aprendizado (Learning Rate)* - recomenda-se cautela na definição desse valor, pois tem grande influência durante o processo de treinamento da rede neural. Geralmente, o valor varia de 0.1 a 1.0, uma taxa de aprendizado muito alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado, por outro lado uma taxa muito baixa torna o aprendizado da rede muito lento.

- *Tamanho do Lote (Batch Size)* - é a quantidade de dados que se deseja processar em cada iteração de treinamento, normalmente se escolhe o valor do bach como múltiplo de 2. Supondo que o tamanho escolhido seja 16 e que a base de dados é formada por 80 imagens, isso significa que os dados serão divididos em 04 lotes (80/16). Depois que todos os 4 lotes tiverem sido alimentados através do modelo, exatamente uma época será concluída.

- *Tamanho da Imagem (Input_Shape)* - Deve ter exatamente 3 canais de entrada e largura e altura e não devem ser menores que 32. O tamanho da entrada padrão para o modelo da VGG19 é de 224x224 (altura x largura).

- *Estado aleatório (Random_State)* - é usado para manter a reprodutibilidade do modelo da mesma forma toda vez que é executado. Caso não seja atribuído um valor, toda vez que fizer a divisão dos dados para teste e treino o conjunto de dados serão diferentes o que atrapalha na depuração caso haja algum problema.

O **Transfer Learning** nos permite utilizar o conhecimento de tarefas aprendidas anteriormente e aplicá-las a tarefas relacionadas mais recentes. O conhecimento de uma tarefa existente atua como uma entrada adicional ao aprender uma nova tarefa de destino.

O aprendizado de transfer learning é uma técnica pela qual o modelo de rede neural é treinado primeiro em um problema semelhante ao que está sendo resolvido. Uma ou mais

camadas do modelo treinado são usadas em um novo modelo treinado no problema de interesse. Um dos benefícios dessa técnica é de diminuir o tempo de treinamento para um modelo de rede neural e resulta em menor erro de generalização (BROWNLEE, 2020).

Data Augmentation, como o conjunto de dados que geramos é pequeno o nosso principal problema seria o overfitting e para evitarmos isso usamos a técnica de aumento de dados. Com isso, o modelo será exposto a diferentes tipos de informações permitindo uma melhor generalização.

O aumento de dados é uma técnica para criar artificialmente novos dados de treinamento a partir dos dados de treinamento existentes. Isso é feito aplicando técnicas específicas de domínio a exemplos dos dados de treinamento que criam novos e diferentes exemplos de treinamento. As transformações incluem uma variedade de operações no campo de manipulação de imagens, como trocas, inversões, zooms entre outros. A intenção é expandir o conjunto de dados de treinamento com exemplos novos e plausíveis, dessa forma as variações das imagens do conjunto de treinamento que provavelmente serão vistas pelo modelo (BROWNLEE, 2019).

Um método de **callback** é qualquer código executável que é passado como parâmetro para outro código, esse outro código deve retornar (executar) o argumento em um determinado momento para completar alguma ação ou tipo de rotina (MOZILLA, 2019).

No Keras existe exemplos de framework para callbacks usado dentro de modelos. Callbacks de praxe são executados ao final de cada época, dessa forma evita que a rede neural caia em mínimos locais, uma vez que a otimização dos pesos da rede neural não se trata de um problema de otimização convexa. Também, pode ser executado em outros estágios do treinamento como no início de uma época, antes ou depois de um único lote, etc. Dessa forma, podemos usar o callback para salvar periodicamente o modelo em disco, para monitorar as métricas após cada lote de treinamento, entre outras funcionalidades (KERAS, 2020).

Os modelos de Callback que usamos para auxiliar o treinamento do modelo, são:

- *ModelCheckpoint*: Salva o modelo para cada época, com isso podemos treinar o modelo e não se preocupar com possíveis problemas como o travamento da máquina.

- *ReduceLROnPlateau*: Reduz a taxa de aprendizado quando uma métrica parar de melhorar.

Segundo Perervenko (2017), as redes neurais são muito sensíveis ao método de inicialização de neurônios em camadas ocultas, dessa forma o objetivo do treinamento da rede é destacar estruturas ocultas, tais como representação, imagens e relacionamentos nos dados. A inicialização de neurônios com pesos de uma rede perceptron multicamadas - MLP (do inglês, Multi Layer Perceptron), obtidos durante o pré-treinamento, permite diminuir o número de dados rotulados e as épocas usando o método do ajuste fino.

O método do **Ajuste Fino** consiste no aperfeiçoamento de parâmetros no qual o treinamento da rede é realizado a partir de uma rede neural pré-treinada. As abordagens mais comuns para o aperfeiçoamento da rede durante o ajuste fino do treinamento são: o treinamento ocorre normal, sendo os valores dos parâmetros atualizados conforme for necessário, o outro caso é atualizar somente a última camada da rede e os demais parâmetros permanecem inalterados durante o treinamento (DE ANDRADE, 2018).

Como as camadas iniciais da rede aprendem características muito gerais e à medida que descemos (deep) a rede, as camadas tendem aprender padrões mais específicos para a tarefa em que está sendo treinada.

Por isso, utilizamos o método do ajuste fino no re-treino de parte da rede pré-treinada VGG19. Dessa forma, consistiu em manter intactas as camadas iniciais (ou congelá-las) e treinar novamente as camadas inferiores para que ela possa entender nossos padrões. A técnica introduz um pequeno ajuste nas representações mais abstratas do modelo que está sendo reutilizado, de modo a torná-las mais relevantes para o problema atual. O ajuste fino ajuda em duas limitações:

- Quando a quantidade de dados necessários para o treinamento não é grande, porém podemos dar maior capacidade de entendimento para a rede pré-treinada retreinando parte de sua rede.

- Quando a parte que está sendo re-treinada não é treinada do zero. Como os parâmetros precisam ser atualizados são menores, a quantidade de tempo de processamento necessária também será menor.

As etapas envolvidas no processo de ajuste fino são as seguintes:

1) Criar o modelo, acrescentando o modelo Sequencial com camadas Densas denominado classificador na saída de uma base convolutiva pré-treinada.

2) Congelar a base convolutiva pré-treinada.

3) Treinar o classificador.

4) Descongelar uma ou mais camadas da base convolutiva.

5) Treinar o modelo.

3.5 Implementando o modelo no reconhecimento facial por vídeo

Com o modelo criado, propusemos uma validação por meio de sistema para o reconhecimento facial por vídeo usando uma webcam. As bibliotecas OpenCV e MTCNN foram as ferramentas escolhidas que melhor atendiam aos requisitos necessários do projeto.

O **OpenCV** (Open Source Computer Vision Library) é um software para captura e manipulação de imagens e vídeos. A biblioteca possui mais de 2500 algoritmos otimizados, que incluem um conjunto abrangente de algoritmos clássicos e avançados de visão computacional e aprendizado de máquina. Esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmeras, rastrear objetos em movimento, etc (OPENCV.org, 2020).

Utilizamos o OpenCV para a criação de uma interface de monitoramento das imagens capturadas e processadas com a webcam que são transformadas em frames (imagens em sequência). Nas imagens manipuladas o software desenha um retângulo na posição do rosto assumindo as cores verde para o retorno da classe Mask e vermelha para a classe NoMask, além de exibir a contagem de pessoas no vídeo de acordo com a classe que o modelo retorna, como podemos observar na foto abaixo.

FIGURA 3- Exemplo do uso do OpenCV.

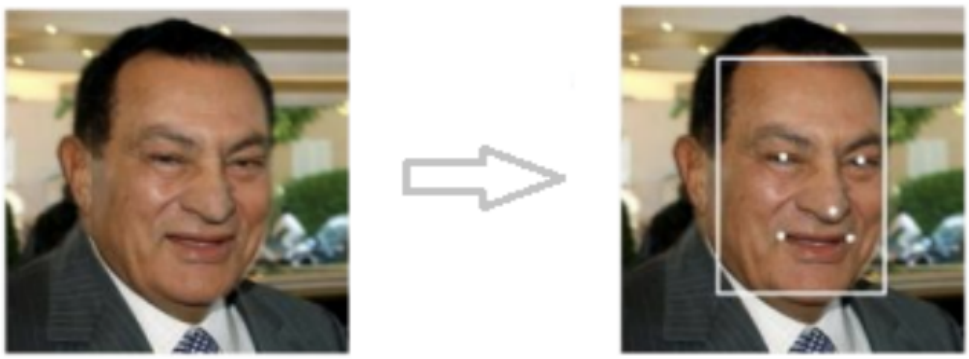
Fonte: Elaborado pelos autores (Crédito: Getty Images/iStockphoto).

O conjunto das imagens capturadas pelo OpenCV foi submetido a biblioteca que implementa o método **MTCNN** (Multitask Cascaded Convolutional Networks), para a detecção de face no qual para cada imagem detectada é feito um enquadramento da face e o posicionamento dos olhos.

Esse método foi desenvolvido com uma abordagem utilizando redes de aprendizado profundo, e superou resultados quando comparado a outros métodos mais tradicionais. Possibilita a segmentação do rosto realizando o enquadramento da face, e retorna as informações de posicionamento de olhos, nariz e boca na imagem. A aplicação do método facilita também a detecção das regiões oculares da esquerda e da direita baseadas na antropometria da face humana, isto é, com o posicionamento dos olhos, nariz e enquadramento da face, traçar segmentos para cada um dos olhos que enquadrem na região (ZHANG,Et al., 2016).

Na Figura 3, temos a demonstração visual do uso do método MTCNN na detecção de facial, e de pontos chaves do rosto (olhos, nariz e boca).

FIGURA 4-Exemplo de detecção de face com o método MTCNN.



Fonte: Adaptado de Zhang et al. (2016).

3.6 Elaborando o modelo na VGG 19 com nosso conjunto de dados e aplicá-lo a OpenCV e MTCNN.

Uma vez que possuíamos indicativos de que o conjunto de dados obtidos era relevante e suficiente para treinarmos uma rede neural, buscamos nosso objetivo primário de trabalhar com uma rede neural profunda com as imagens capturadas.

Fizemos uso do modelo pré-treinado VGG 19 disponível no Keras, uma vez que é uma das arquiteturas mais usadas e relevantes para o mercado. Aproveitando o aprendizado adquirido através do conjunto ImageNet e, considerando que a profundidade da rede tem impacto significativo na identificação de imagens, foi escolhida a versão com 19 camadas.

Com o modelo escolhido, a definição dos hiperparâmetros foi feita em sequência, tendo como resultado os seguintes valores:

* Epochs: 30 # Número de época a serem treinadas
* Batch Size: 32 # Processa 32 imagens a cada lote
* Learning Rate: 1e-5 # Taxa de aprendizado
* Input Shape: 1e-5 (160, 160, 3) # Tamanho da imagem
* Random State: 1e-5 (160, 160, 3) 42 # Semente usada pelo gerador de números aleatórios

Em seguida definimos as classes para o bom funcionamento do modelo. Utilizamos as técnicas de Callback: ModelCheckpoint e ReduceLROnPlateau, o primeiro para garantir que teremos os dados salvos em cada época de treino, o segundo para guiar a taxa de aprendizado do modelo conforme resultados do treino.

Em seguida, dividimos o dataset em treino e teste, cada qual correspondente ao momento respectivo do desenvolvimento e fizemos o processo de Data Augmentation para que nossa amostra fosse ampliada com diferentes espectros das imagens.

Ao aplicarmos tais configurações ao modelo, o passo seguinte consistiu em aplicar a técnica levantadas de ajuste fino e retreinar o modelo da VGG 19, buscando com que a rede aprenda de forma mais assertiva as características do nosso conjunto de dados.

O modelo é compilado a partir do aprendizado ganho com o ajuste fino executado e submetemos nossa amostra de teste para que possamos analisar e medir seus resultados. Por último inserimos uma nova imagem, diferente das imagens do nosso dataset, para que o modelo faça a predição se a pessoa da foto está com máscara ou sem máscara facial.

Como complemento e tendo em vista futuras aplicações práticas, fizemos outra validação com imagens extraídas diretamente da câmera do dispositivo, para isso usamos em conjunto das bibliotecas OpenCV e MTCNN. A primeira captura as imagens na webcam, enquanto a segunda identifica nas imagens capturadas a presença de pessoas no enquadramento e, caso exista, a imagem passa pelo processamento no modelo criado. O retorno da classificação da imagem como Mask ou No Mask acontece por meio da biblioteca OpenCV que exibe o resultado de forma visual para o usuário.

4. AMOSTRA E RESULTADOS

4.1 Apresentação descritiva da amostra do estudo

O dataset utilizado para o treino no Teachable Machine é composto em sua totalidade por 5.570 imagens dentro de três categorias, sendo elas: pessoas com máscara, pessoas sem máscara e plano de fundo. Temos 1.667 correspondentes à classe de pessoas com máscara, 1.799 de pessoas sem máscara e 2.104 imagens de plano de fundo.

Não conseguimos obter um volume de imagens iguais para todas as classes, devido a dificuldade de extrair imagens distintas dos sites de busca, uma vez que, muitos sites usam a mesma imagem em publicações distintas.

A classe de **pessoas com máscara** chamamos de **"Mask"**, foi composta por imagens de perfil de pessoas com o rosto coberto pela máscara protetora e foram coletadas apenas imagens de máscara que atendam aos critérios da recomendação feita pela OMS.

Foi necessária uma curadoria na coleta das imagens a fim de se obter uma diversidade das imagens, respeitando as diferenças de raça/cor, etnias, gêneros, faixas etárias, características físicas (por exemplo, barba e bigode) e acessórios (por exemplo, óculos, bonés, etc).

Para melhor assertividade do modelo, não foram inseridas imagens de pessoas usando máscara de forma incorreta (por exemplo, o uso da máscara pela metade com o nariz descoberto), assim como os tipos de máscara não aprovadas pelas organizações de saúde (por exemplo, máscara de mergulho snorkel).

A classe de **pessoas sem máscara** chamamos de **"NoMask"**, foi composta por imagens de perfil de pessoas, assim como a classe anterior também tivemos o cuidado de coletar as imagens respeitando as diferenças de raça/cor, etnia, gêneros, faixas etárias, características físicas (por exemplo, barba e bigode) e acessórios (por exemplo, óculos, bonés, etc).

Para melhor aprendizado da máquina sobre o uso ou não de máscara, inserimos nessa classe imagens de perfil de pessoas com uma mão ou as duas mãos na boca cobrindo a face, assim como outros tipos de posições e gestos com os braços e mãos na frente do rosto.

A classe de **plano de fundo** chamamos de **"Background"**, composta por imagens de diversos locais passíveis de circulação de pessoas, porém sem a presença delas. As imagens coletadas foram: interior de escritórios, interior de casas, quartos, cozinhas, ruas e praças de cidades, escolas, praias, favelas, parques públicos, parques infantis (playground), aeroportos, estações de metrô, igrejas, shoppings, mercados e cidades em geral. Consideramos como quadro de imagens válidos aqueles que seria possível ter uma pessoa presente, por exemplo, a imagem de um parque com a perspectiva bidimensional em que poderia existir uma pessoa.

Imagens com perspectiva tridimensional foram evitadas por mudar as dimensões dos objetos em cena, um exemplo desse tipo de imagem seria a foto panorâmica do topo de um prédio. Fotos em perspectiva é uma técnica usada por fotógrafos para projetar a fotografia na mesma dimensão que os olhos vêem, que é o plano tridimensional no qual cada ângulo nos dá uma visão diferente.

Na construção do modelo utilizamos apenas as imagens das classes Mask e NoMask, porque utilizamos os softwares MTCNN para o reconhecimento de face e o OpenCV com o uso da webcam para a captura, manipulação e monitoramento das imagens processadas. Neste cenário, quando não há pessoas no enquadramento dos frames capturados, ele não é reconhecido pelo MTCNN nem apresentado ao modelo para predição, tornando desnecessário o aprendizado da classe background e a identificação desta classe.

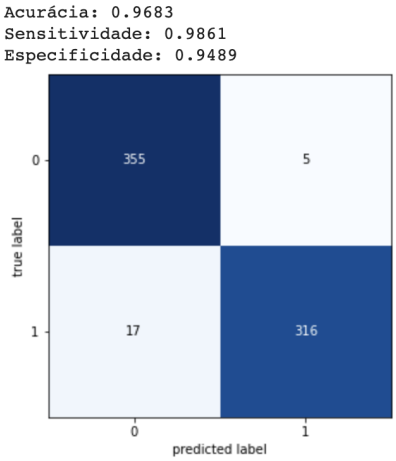
4.2 Apresentação dos Resultados

Os resultados obtidos foram, de certa forma, satisfatórios uma vez que inserida uma nova imagem diferente das usadas no dataset o modelo reconheceu corretamente se a pessoa da imagem estava usando máscara ou não.

Obtivemos um índice de cerca de 96% de acurácia, de acordo com os últimos ajustes efetuados no modelo. A sensibilidade do mesmo, foi de 98% e a especificidade do modelo ficou em 94%.

Observando a amostra de teste apresentada ao modelo, tivemos acerto em 671 imagens dos 693 totais.

FIGURA 5-Apresentação dos resultados do modelo.



Fonte: Elaborado pelos autores.

A distribuição desse acerto foi de 316 delas de uso da máscara e 355 de não uso. Nas 22 imagens que o modelo classificou de forma equivocada, 05 delas apresentavam a máscara, mas o modelo não conseguiu identificar e em 17 delas não tínhamos o uso da máscara e a classificação estava apontando com o uso de máscara. Na figura abaixo, podemos ver todos os resultados de forma simplificada.

O modelo foi testado nas seguintes condições:

- Ambiente com iluminação da luz do dia.
- Em períodos noturnos com ambiente de baixa iluminação.
- Identificar o número de faces existente na imagem capturada pela webcam.
- Reconhecer o uso de máscara e sem máscara das faces capturas.
- Reconhecer uma pessoa com a mão no rosto como classe sem máscara.
- Reconhecer que um objeto qualquer frente a face não representa o uso de máscara.
- Reconhecer faces distintas a fim de garantir a diversidade na predição.

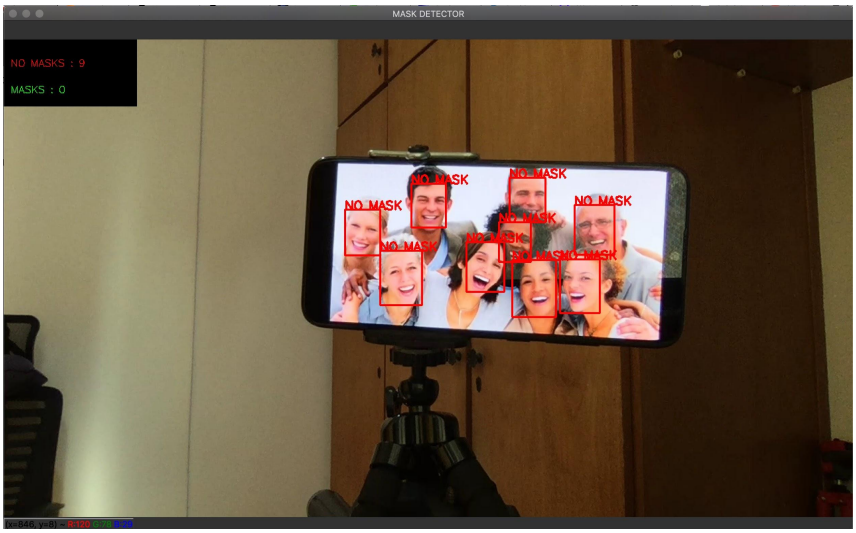
Resultados Positivos

Grupo de Pessoas

Reconheceu corretamente o número de faces existentes na imagem da webcam, para esse teste devido a pandemia e a orientação das autoridades sanitárias de distanciamento social não foi possível fazer com um grupo de pessoas, por isso realizamos o teste usando uma imagem disponível na internet.

Na imagem como pode ser visto abaixo, o grupo é composto por 09 pessoas e o modelo previu corretamente em torno de 99% para a classe sem máscara para todos os integrantes da imagem.

FIGURA 6- Reconhecimento facial de um grupo de pessoas.

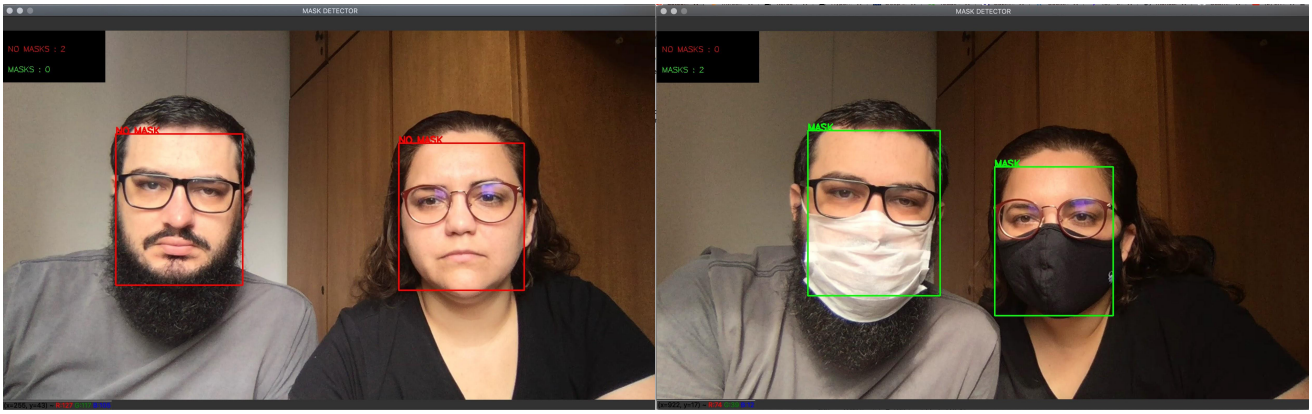


Fonte: Elaborado pelos autores (Crédito: Getty Images/iStockphoto).

Diversidade

O modelo reconhece faces distintas e não faz distinção de gênero ou cor/raça, como pode ser visto na imagem da figura 7. Na imagem abaixo, podemos constatar que o modelo não fez a distinção de gênero e previu corretamente cerca de 85% para a classe mask e 73% para classe mask para ambos os gêneros.

FIGURA 7- Reconhecimento de diversidade de gênero.

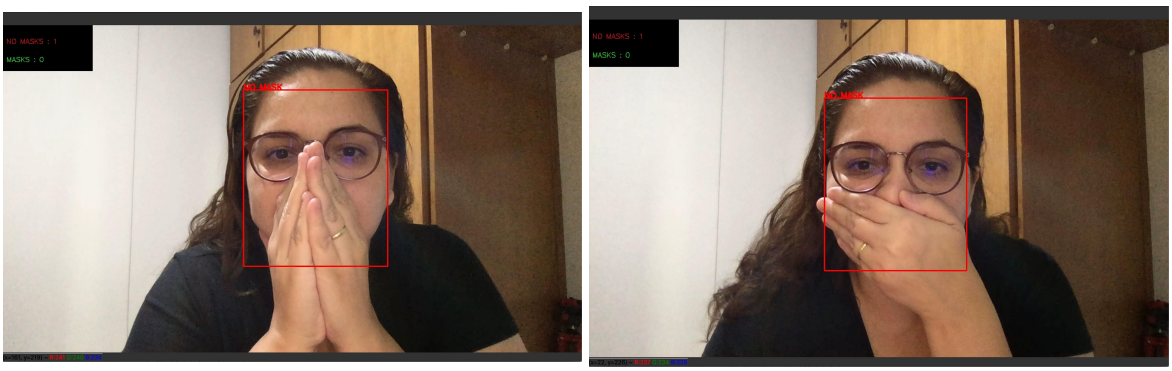


Fonte: Elaborado pelos autores.

Mão na face

Um hábito comum das pessoas, é o de colocar a mão no rosto tampando assim a face, por isso treinamos o modelo com esse tipo de imagem para que aprendesse a reconhecer essa característica e classificar corretamente na classe de no mask. O resultado obtido foi satisfatório o modelo previu corretamente, testamos as algumas posições para a mão no rosto como pode ser visto na figura 8 abaixo.

FIGURA 8-Reconhecimento com a mão na face.

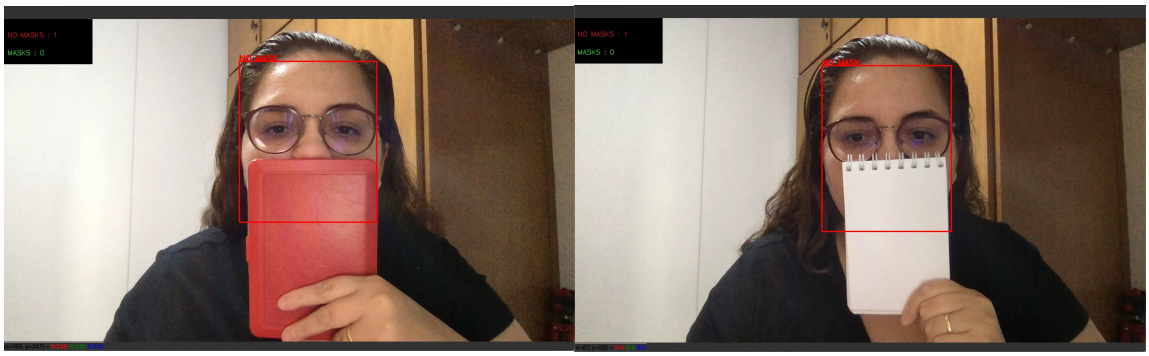


Fonte: Elaborado pelos autores.

Objetos na face

Fizemos teste com objetos frente da face cobrindo o rosto e simulando o que poderia ser uma máscara, e o modelo classificou corretamente como No Mask. Como de fato, podemos verificar nas imagens abaixo da figura 9 que os objetos não se trata de uma máscara de proteção facial.

FIGURA 9- Reconhecimento com objetos



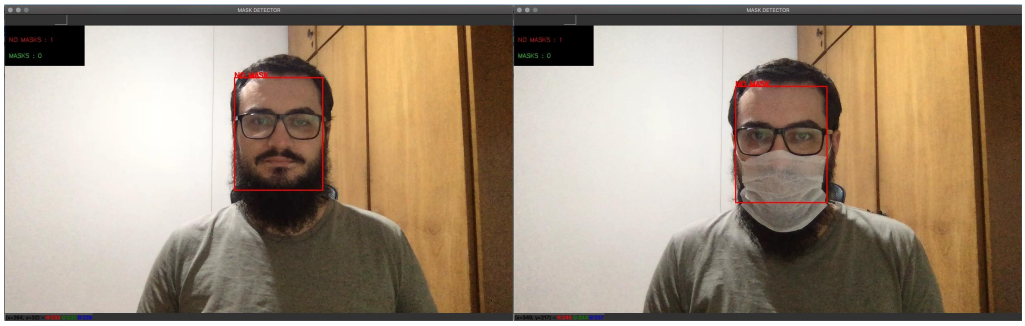
Fonte: Elaborado pelos autores.

Resultados Negativos

Em ambientes com pouca iluminação o modelo previu de maneira incorreta as classes. Na imagem da esquerda a pessoa estava sem máscara e o modelo previu com 91% a classe de No Mask e 8% para a classe de Mask.

Entretanto, para a imagem da direita a pessoa estava com máscara e o modelo previu de forma errada 86% para No Mask e 13% para Mask. Como podemos observar nas imagens abaixo, o modelo errou ao prever que a pessoa estava sem máscara quando na verdade ela estava com.

FIGURA 10-Reconhecimento de face em ambiente de pouca



Fonte: Elaborado pelos autores.

5. ANÁLISES E DISCUSSÃO

A acurácia obtida de 96% é um resultado satisfatório, quer dizer que a proporção de casos que foram corretamente previstos, sejam eles verdadeiro positivo ou verdadeiro negativo. A sensibilidade de 98%, nos indica uma performance da capacidade de identificar o uso da máscara.

No entanto, a especificidade do modelo ficou em 94%, menor que a sensibilidade. Acreditamos que isso, se deva ao fato de termos incluído também no dataset pessoas com a mão na face, o que torna mais complexa a distinção.

Na Matriz Confusão, vimos que de uma amostra de 693 imagens usadas como teste, o modelo classificou corretamente as classes 45% de acerto para uso de máscara e 51% de não uso da máscara. O modelo classificou de forma equivocada em 22 imagens o que representa 3% de erro.

Nas configurações e aprendizado atual do modelo, ao aplicá-lo ao teste usando a webcam, obtivemos um resultado satisfatório na maioria das vezes em que o ambiente dispunha de iluminação boa. Entretanto, em ambientes com baixa iluminação, o modelo não apresentou o comportamento esperado para o reconhecimento da face com ou sem máscara.

5.1 Métricas de desempenho do modelo

Existem diversas métricas desenvolvidas para se avaliar o quão bom é um modelo, utilizamos as métricas de desempenho da predição mais comuns e usadas no mercado para avaliar o funcionamento do nosso modelo. São elas:

- **Acurácia (Accuracy):** é a proporção de predições corretas do modelo, sem considerar o que é verdadeiro positivo ou que é verdadeiro negativo e sim o acerto total.
- **Sensibilidade (Sensibility):** é a proporção de casos verdadeiros positivos, ou seja, avalia a capacidade do modelo classificar um indivíduo corretamente.
- **Especificidade (Specificity):** é a proporção de verdadeiros negativos, ou seja, avalia a capacidade do modelo predizer um indivíduo como não evento que foram identificados corretamente.

- **Matriz confusão (Confusion Matrix):** É uma tabela que mostra as frequências de classificação para cada classe do modelo, verdadeiro positivo, verdadeiro falso, falso positivo e falso negativo.

- **Verdadeiro positivo (true positive - TP):** é a proporção de verdadeiros positivos em relação a todas as predições positivas,ou seja, o modelo previu corretamente que o indivíduo estava sem máscara.

- **Verdadeiro negativo (true negative - TN):** é a proporção de verdadeiros negativos em relação a todas predições negativas, ou seja, o modelo previu corretamente o indivíduo com máscara.

- **Falso positivo (false positive - FP):** também chamado de erro Tipo I, é a proporção de falso positivo em relação a todas as predições, ou seja, é quando o indivíduo estava com máscara e o modelo previu incorretamente que não estava.

- **Falso negativo (true negative - TN):** também chamado de erro Tipo II, é a proporção de falso negativo em relação a todas as predições, ou seja, é quando o indivíduo estava sem máscara e o modelo previu incorretamente que estava usando.

FIGURA 11- Representação gráfica da Matriz Confusão.

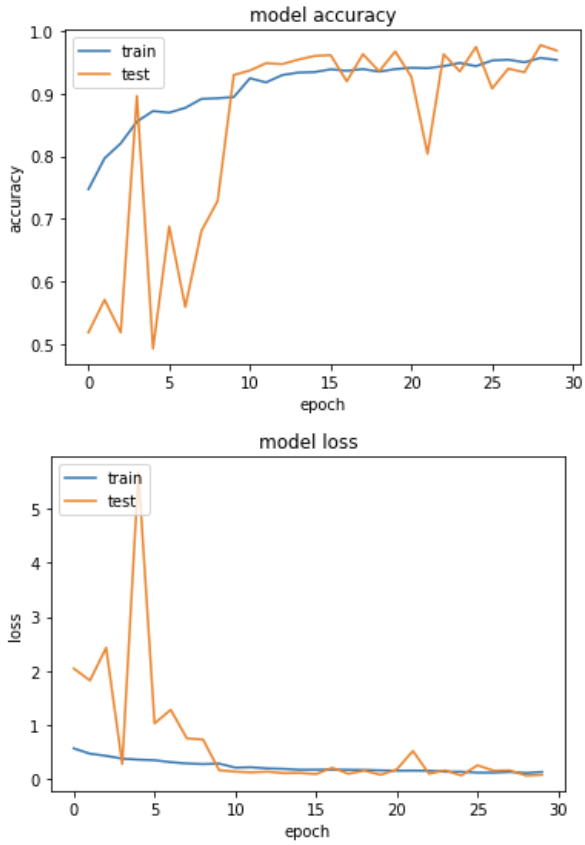
		Classe Verdadeira (confirmado por análise)	
		positivos	**negativos**
Classe Prevista (predito pelo teste)	**positivos**	**VP** Verdadeiro Positivo	**FP** Falso Positivo (erro tipo I)
	negativos	**FN** Falso Negativo (erro tipo II)	**VN** Verdadeiro Negativo

Fonte: Elaborado pelos autores.

5.2 Avaliação do treinamento do modelo

Como forma de comparar a acurácia do modelo a amostra e o erro dele sobre tais observações, temos a Figura 12. Nela conseguimos perceber a oscilação da acurácia e erro ao longo das épocas de treino e compará-las aos resultados na amostra teste.

FIGURA 12- Avaliação do treinamento do modelo



Fonte: Elaborado pelos autores.

Logo ao início das épocas, a variabilidade dos resultados de erro e acurácia são predominantes, demonstrando o impacto dos métodos de callback utilizados como ReduceLROnPlateau, que alteraria a taxa de aprendizado usada na próxima época.

Conseguimos identificar que a partir da 10ª época de treino a variação deixa de acontecer na mesma magnitude que anteriormente, demonstrando o aprendizado do modelo adequado a ambas amostras, treino e teste.

O gráfico indica um ponto de atenção, próximo a 15ª época a amostra de teste volta a apresentar variabilidade na acurácia, ainda que isso não seja observado em mesma amplitude no erro. Possivelmente este seja um indicativo de que o modelo possa ter sofrido overfit, ainda que exclusivamente não confirme isto.

6. LIMITAÇÕES DA PESQUISA / ESTUDOS FUTUROS

Considerando o tempo hábil que tivemos para o desenvolvimento e a limitação do conjunto de imagem, uma vez que o uso de máscaras é algo recentemente generalizado para a população ocidental, os resultados obtidos na identificação do uso de máscaras faciais usando a rede convolucional VGG 19 obteve um resultado muito bom, superando as expectativas para o trabalho.

Além de um conjunto de dados mais diverso para treino e teste do modelo incluindo cenários noturnos do uso de máscaras, por exemplo, melhorias já vislumbradas são relacionadas a parâmetros, tratamento prévio das imagens, uso de redes com arquiteturas diferentes, mas também relevantes no mercado e novos métodos e classes.

Considerando parâmetros do modelo, a busca por parâmetros melhores pode ser feita visando aprimorar métricas obtidas, utilizar a biblioteca Keras Tuner ajudaria a escolher o conjunto ideal de hiperparâmetros para o modelo.

Visando ter um conjunto com maior gama de cenários atendidos, tratar previamente algumas das imagens coletadas retirando o brilho ou alterando saturação, assim escurecendo-as, para inserir no aprendizado do modelo e que pode então reconhecer melhor o uso da máscara em ambientes com baixa iluminação. Assim como, fazer incrementos de imagens enriquecendo a base de dados com mais imagens e melhorando o aprendizado da máquina mitigando assim o risco de overfitting de quando a base de dados é pequena.

Ajustes no Callback já utilizado, incluindo outros métodos e classes como EarlyStopping, que pare de treinar quando uma métrica monitorada parar de melhorar minimizando a perda, ou LearningRateScheduler que funciona como um agendador da taxa de aprendizado que para cada época obtém o valor da taxa atualizado.

Por fim, a busca por outros modelos de rede pré-treinadas, como a ResNet que é uma rede mais profunda que a VGG19 e que usa o pool médio global. Outra possibilidade é a Inception-ResNet-v2 essa rede possui 164 camadas de profundidade e por isso aprendeu representações ricas de recursos para uma ampla variedade de imagens.

7. CONCLUSÃO

O trabalho reuniu informações sobre o uso de rede neural convolucional para o reconhecimento facial de pessoas com máscaras e sem máscaras. Com as imagens coletadas na web e o treinamento da rede realizado no Teachable Machine foi possível observar que, apesar das dificuldades de se obter imagens de qualidade e suficientes para a construção de um sistema de reconhecimento facial confiável, conseguimos alcançarmos bons resultados em nosso modelo.

O trabalho se propôs a utilizar técnicas de rede neural convolutiva (RNC) para uma realidade nova da sociedade devido a pandemia do Covid-19, o uso de máscaras faciais protetoras. Reunimos informações sobre o uso de RNC para o reconhecimento facial que nos dessem a oportunidade de realizar a distinção entre pessoas com máscaras e pessoas sem máscaras.

Não dispúnhamos de um dataset pronto com essas imagens que fosse de domínio público, foi necessário construir uma alternativa que possibilitasse a coleta das imagens da web, utilizamos para isso a técnica de web scraping.

Para imagens de pessoas sem máscaras, utilizamos um dataset público de imagens de artistas, políticos e figuras públicas. Também, incluímos algumas imagens para enriquecer o dataset.

As imagens de pessoas com máscaras encontramos dificuldades por não ter um dataset público disponível. O único que encontramos foi um dataset composto por foto de pessoas asiáticas com máscara, porém estava indisponível seu acesso. Com isso, tivemos que coletar as imagens na web para compor nosso dataset.

Em ambos os casos, procuramos ter a maior diversidade possível de etnias, raças e idades em nossos datasets, levando em conta a diversidade do país e procurando tornar o modelo aplicável em situações reais. Garantimos isso ao checar manualmente e visualmente as imagens coletadas antes de iniciarmos o treino da rede.

A dificuldade encontrada para obtermos imagens de qualidade suficientes para o aprendizado da RNC, foi superada e conseguimos realizar o treinamento da rede usando o Teachable Machine. Através desta abordagem, usando a GPU para processamento do código

e utilizando a rede convolutiva pré-treinada VGG19, conseguimos alcançar resultados satisfatórios em nosso modelo.

REFERÊNCIAS

ANDRADE, Gabriel Herman Bernardim. Paradigmas de Aprendizado de Máquina Aplicados em Reconhecimento Facial. Dissertação de mestrado (Programa de Pós-Graduação em Engenharia Elétrica) - Universidade Federal do Paraná, Curitiba, 2019.

BROWNLEE, Jason. How to Improve Performance With Transfer Learning for Deep Learning Neural Networks, 2020. Machine Learning Mastery. Disponível em: https://machinelearningmastery.com/how-to-improve-performance-with-transfer-learning-for-deep-learning-neural-networks/. Acesso em: 26 mai.2020.

BROWNLEE, Jason. How to Configure Image Data Augmentation in Keras, 2019.Machine Learning Mastery. Disponível em: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/. Acesso em: 26 mai.2020.

ÇARIKÇI, Müge; ÖZEN, Figen. A Face Recognition System Based on Eigenfaces Method. Procedia Technology. p.118-123, 2012. doi: 10.1016/j.protcy.2012.02.023. Disponível em: https://www.researchgate.net/publication/257743676_A_Face_Recognition_System_Based_on_Eigenfaces_Method. Acesso em: 20 mai.2020.

CHAVES, Rodrigo Reuse. Redes Neurais Deep Learning Aplicados ao Reconhecimento Facial. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) - Universidade de Caxias do Sul, Rio Grande do Sul, 2018.

CIRESAN, Dan. Et al.. Multi- column depp neural network for traffic sign classification. Neural Networks. v.32, p.333-338, 2012. doi 10.1016/j.neunet.2012.02.023. Disponível em:https://www.sciencedirect.com/science/article/abs/pii/S0893608012000524.Acesso em: 20 mai.2020.

CUBUK, Ekin D. El al.. AutoAugment: Learning Augmentation Strategies from Data. The Computer Vision Foundation, p. 113-123, 2019. doi: 10.1109/CVPR.2019.00020. Disponível em:http://openaccess.thecvf.com/content_CVPR_2019/papers/Cubuk_AutoAugment_Learning_Augmentation_Strategies_From_Data_CVPR_2019_paper.pdf. Acesso em: 20 mai.2020.

DE ANDRADE, Kovaleski, Patrícia.Implementação de redes neurais profundas para reconhecimento de ações em vídeo. Trabalho de Conclusão de Curso (Curso de Engenharia de Computação e Informação) - Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2018.

DE FREITAS, Robson A. B. El al.. Prospecção Científica sobre Epidemiologia: Prevenção da Covid-19 Aliada à Inteligência Artificial. Cadernos de Prospecção, Salvador, v.13, n.2, p.543-558, 2020. doi: 10.9771/cp.v13i2.COVID-19.36190. Disponível em: https://portalseer.ufba.br/index.php/nit/article/view/36190/21006. Acesso em: 26 mai.2020.

DEEP LEARNING BOOK, Reconhecimento de Imagens com Redes Neurais Convolucionais em Python- Parte 1. Disponível em: http://deeplearningbook.com.br/reconhecimento-de-imagens-com-redes-neurais-convolucionais-em-python-parte-1/. Acesso em: 26 mai.2020.

GARCIA, Leila Posenato. Use of facemasks to limit COVID-19 transmission. Epidemiol. Serv. Saúde, Brasília , v.29, n.2, e2020023, 2020 . doi: 10.5123/s1679-49742020000200021. Disponível em: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S2237-96222020000200902. Acesso em: 20 mai.2020.

GOOGLE, Teachable Machine. Disponível em: https://teachablemachine.withgoogle.com/. Acesso em: 10 maio.2020.

HALL, Patrick; WEN, Phan; WHITSON, Katie. (2016). The Evolution of Analytics: Opportunities and Challenges for Machine Learning in Business. 1ª. Ed (First Edition). United States of America: O'Reilly Media, 2016.

HAYKIN, Simon. Redes Neurais, 2ª. Ed. Porto Alegre: Bookman, 2008.

HOWARD, Jeremy. El al.. Face Masks Against COVID-19: An Evidence Review. Preprints, v.2, 2020040203 , 2020, doi: 10.20944/preprints202004.0203.v2. Disponível em: https://www.preprints.org/manuscript/202004.0203/v2. Acesso em: 20 mai.2020.

KERAS, Callbacks API, 2020. Disponível em: https://keras.io/api/callbacks/. Acesso em: 26 mai.2020.

KHAN, Asifullah. El al.. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. Artificial Intelligence Review, v.7, 2020. doi: 10.1007/s10462-020-09825-6. Disponível em: https://arxiv.org/abs/1901.06032. Acesso em: 26 mai.2020.

LECUN, Yann; BENGIO, Yoshua. Convolutional Networks for Images, Speech, and Time-Series. The Handbook of Brain Theory and Neural Networks, MIT Press, 1995. Disponível em: http://yann.lecun.com/exdb/publis/index.html. Acesso em: 26 mai.2020.

MACEDO, Sandeco. Detectando Faces em Vídeo com Deep Learning. Github, abril.2020. Disponível em: https://github.com/sandeco/CanalSandeco/blob/master/Deep%20Learning%20s%C3%A9rie/%2316%20-%20Detectando%20Faces%20em%20v%C3%ADdeo%20com%20Deep%20Learning/create%20model.ipynb. Acesso em: 10 abr.2020.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre Aprendizado de Máquina. Sistemas Inteligentes Fundamentos e Aplicações. 1ª. Ed, p.39-56. Barueri, São Paulo: USP, 2003. Disponível em: http://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf. Acesso em: 20 mai.2020.

MOZILLA, Developer Network. Função Callback, 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Glossario/Callback_function. Acesso em: 26 mai.2020.

OPENCV, Equipe. Sobre o OpenCV (Open Source Computer Vision Library), 2020. Disponível em: https://opencv.org/about/. Acesso em: 20 mai.2020.

PERERVENKO, Vladimir. Redes Neurais Profundas (Parte IV). Criação, Treinamento e Teste de um Modelo de Rede Neural. 2017. Disponível em: https://www.mql5.com/pt/articles/3473. Acesso em: 20 mai.2020.

SIMONYAN, karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. Cornell University. v.6, 2015. Disponível em: https://arxiv.org/abs/1409.1556. Acesso em: 26 mai.2020.

SINDAGI, Vishwanath; PATEL, Vishal. A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation. Pattern Recognition Letters, v.107, p.3-16, 2018. doi: 10.1016/j.patrec.2017.07.007. Disponível em:

https://www.sciencedirect.com/science/article/abs/pii/S0167865517302398. Acesso em: 26 mai.2020.

VIZONI, Marcelo Vilela. Reconhecimento da Região Ocular para a Identificação Biométrica de Pessoas Utilizando Aprendizado em Profundidade. Dissertação (Mestrado) - Universidade Estadual Paulista "Júlio de Mesquita Filho". Faculdade de Ciências, Bauru, 2019.

ZHA, Shengxin. El all.. Exploiting Image-trained CNN Architectures for Unconstrained Video Classification. Cornell University, v.3, 2015. doi:10.5244/C.29.60. Disponível em: https://arxiv.org/abs/1503.04144. Acesso em: 26 mai.2020.

ZHANG, K. et al. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, IEEE, v. 23, n. 10, p. 1499–1503, 2016, doi:10.1109 / LSP.2016.2603342.

ZOPH, Barret. El all.. Learning Data Augmentation Strategies for Object Detection. Cornell University, v.1, 2019. Disponível em: https://arxiv.org/abs/1906.11172. Acesso em:Acesso em: 26 mai.2020.

APÊNDICES / ANEXOS

Esse apêndice é para descrever os scripts disponíveis através do link abaixo e como deve ser utilizado https://github.com/Dlyra/mask_detector.

Antes de executar os scripts é necessário instalar as dependências do projeto para isso utilize o arquivo **requirements.txt**. No terminal utilize o comando *pip install -r requirements.txt,* isso deverá ser feito dentro do diretório do projeto. Para mais detalhes leia o arquivo **README.md**.

O script **download_imagens.py** é para aplicar o web scraping e fazer o download das imagens através do site de busca https://images.google.com.

Para gerar o modelo execute o script **detector_mask.ipynb**, sendo recomendado o uso da GPU, do contrário o processamento da rede neural será lento. É necessário informar o diretório onde está localizado os arquivos das imagens obtidas no passo anterior, ou pode também utilizar o dataset disponibilizado no repositório. A imagem nomeada de *img_1.jpg* pode ser utilizada para validar o modelo criado.

Para a execução do modelo usando a webcam utilize o script **webcam_mask_detector.py**. Nessa etapa é necessário que tenha uma webcam instalada e configurada no computador.