

Identificação automática do uso de máscaras de proteção facial: um estudo comparativo

Automatic identification of the use of face protection masks: a comparative study

Identificación automática del uso de mascarillas de protección facial: un estudio comparativo

José Voltan¹, Ronaldo Goldschmidt¹, Jefferson Oliva², Julio Duarte¹,
Dalcimar Casanova², Marcelo Teixeira².

1 Programa de Pós-graduação em Sistemas e Computação, Instituto Militar de Engenharia - IME, Rio de Janeiro (RJ), Brasil.

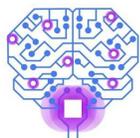
2 Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná – UTFPR, Pato Branco (PR), Brasil.

Autor correspondente: José Voltan
E-mail: voltan.jose@ime.eb.br

Resumo

Objetivo: O uso de máscaras de proteção é uma importante medida para diminuir a transmissão da COVID-19 e de outras doenças. O presente trabalho teve como objetivo realizar um estudo comparativo entre diferentes modelos de aprendizado profundo aplicados na identificação da máscara de proteção (pessoa sem máscara, com máscara ou com máscara incorretamente posicionada). **Métodos:** foram implementados e avaliados os modelos de Redes Neurais Convolucionais *MobileNetV3*, *Xception*, *VGG19*. Foi utilizada a técnica de transferência de aprendizado na implementação e ajuste desses modelos. **Resultados:** Os modelos avaliados apresentaram acurácias que variaram entre 42% e 86%, esta última obtida pelo modelo *Xception*, superando os resultados reportados na literatura correlata. **Conclusão:** Os resultados apontam para o potencial promissor do modelo *Xception* que, ao viabilizar o monitoramento automático, permite a orientação de pessoas quanto ao correto uso de máscaras de proteção, dessa forma, contribuindo para diminuir o espalhamento de doenças através das vias aéreas.

Descritores: Máscaras; Aprendizado de Máquina; Aprendizado Profundo



Abstract

Objectives: The use of protective masks is an important measure to reduce the transmission of COVID-19 and other diseases. The present work aims to carry out a comparative study between different deep learning models applied to the identification of face masks (person without mask, with mask or with incorrectly positioned mask).

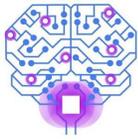
Methods: Convolutional Neural Network models MobileNetV3, Xception, VGG19 were implemented and evaluated. The transfer of learning technique was used in the implementation and adjustment of these models. **Results:** The evaluated models presented accuracies that varied between 42% and 86%, the latter obtained by the Xception model, surpassing the results reported in the related literature. **Conclusion:** The results point to the promising potential of the Xception model, which, by enabling automatic monitoring, allows people to be guided on the correct use of protective masks, thus helping to reduce the spread of diseases through the airways.

Keywords: Mask; Machine Learning; Deep Learning

Resumen

Objetivos: El uso de mascarillas protectoras es una medida importante para reducir la transmisión del COVID-19 y otras enfermedades. El presente trabajo tiene como objetivo realizar un estudio comparativo entre diferentes modelos de aprendizaje profundo aplicado a la identificación de la mascarilla protectora (persona sin mascarilla, con mascarilla o con mascarilla mal colocada). **Métodos:** Se implementaron y evaluaron los modelos de redes neuronales convolucionales MobileNetV3, Xception, VGG19. En la implementación y ajuste de estos modelos se utilizó la técnica de transferencia de aprendizaje. **Resultados:** Los modelos evaluados presentaron precisiones entre 42% y 86%, esta última obtenida por el modelo Xception, superando los resultados reportados en la literatura relacionada. **Conclusión:** Los resultados apuntan al potencial promisorio del modelo Xception que, al posibilitar el monitoreo automático, permite orientar las personas sobre el uso correcto de las mascarillas protectoras, contribuyendo así a reducir la propagación de enfermedades a través de las vías respiratorias.

Descriptor: Máscaras; Aprendizaje Automático; Aprendizaje Profundo



Introdução

A pandemia ocasionada pelo vírus SARS-CoV-2 começou em dezembro de 2019 na China, e rapidamente se alastrou pelo mundo. O vírus SARS-CoV-2 pertence a uma família de vírus denominada Coronavírus e ao infectar humanos pode desenvolver a doença COVID-19 ⁽¹⁾. No Brasil, o Ministério da Saúde declarou emergência de saúde pública de importância nacional em fevereiro de 2020 ⁽¹⁾.

Buscando diminuir a transmissão do vírus SARS-CoV-2 entre pessoas, a Organização Mundial da Saúde (OMS) e a Organização Pan-Americana da Saúde (OPAS) recomendaram o uso de máscara de proteção ⁽²⁾. A máscara de proteção é uma medida não-farmacológica, de baixo custo, que pode diminuir, ou até mesmo evitar a disseminação de gotículas infecciosas entre as pessoas e o ambiente ⁽²⁾.

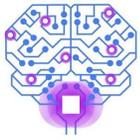
Além de proteger contra a disseminação da COVID-19, o uso da máscara de proteção individual também pode evitar o contágio de outras doenças que utilizam como forma de propagação gotículas respiratórias inaladas através das vias aéreas, como é o caso da varíola dos macacos (*Monkeypox*) ⁽³⁾.

Para ser eficiente, a máscara deve ser posicionada corretamente no rosto, cobrindo a boca e o nariz e ser amarrada de forma a ficar justa ⁽²⁾. Nesse contexto, existem diversos modelos de máscaras de proteção, conforme ilustrado na figura 1.



Figura 1 – Modelos de máscara de proteção

Mesmo sendo uma medida simples, barata e eficaz, muitas pessoas não utilizam a máscara corretamente ou simplesmente esquecem de utilizá-la. Dessa forma é interessante que o controle do uso da máscara seja automatizado para que se possa orientar as pessoas, principalmente ao entrarem em ambientes fechados. Na literatura,

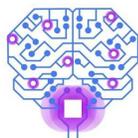


existem algumas soluções baseadas em Aprendizado de Máquina voltadas para o controle de acesso a ambientes e outras para o monitoramento de ambientes. Algumas soluções separaram o problema em duas classes, identificando se o indivíduo está com máscara ou sem máscara, nesse conjunto se inserem os trabalhos de Junior et al. ⁽⁴⁾, Costa et al. ⁽⁵⁾ e Das et al. ⁽⁶⁾. Já outras soluções, utilizaram três classes, identificando pessoas com a máscara corretamente posicionada, incorretamente posicionada ou sem máscara. Essa abordagem é mais condizente com o que se pode observar no dia a dia, em que algumas pessoas estando com a máscara na face, a posicionam de forma incorreta. Alguns dos autores que fizeram a abordagem a duas classes, sugeriram a análise a 3 classes ⁽⁶⁾.

Dentre os trabalhos que adotaram a abordagem em 3 classes, Quiñonez-Cuenca et al. ⁽⁷⁾ desenvolveram um sistema de Inteligência Artificial das Coisas (AIoT). Os autores analisaram o desempenho do algoritmo YOLOv3, identificado por eles como sendo o estado da arte para detecção de objetos reais voltados para AIoT. Quiñonez-Cuenca et al. utilizaram dois *datasets*, o Kaggle Medical Mask Dataset ⁽⁸⁾ e o MAFA ⁽⁹⁾. Para o treinamento, 716 imagens foram utilizadas, sendo 3.070 faces com máscara correta, 113 com máscara incorretas e 675 sem máscara (uma distribuição não balanceada). A validação contou com 179 imagens, sendo 162 faces com máscara correta, 10 com máscara incorreta e 42 sem máscara. Quanto ao desempenho do algoritmo YOLOv3, os autores obtiveram uma precisão para a classe máscara correta de 85,97%, sem máscara de 68,72% e máscara incorreta de 73,15%.

Sobre o *dataset* Kaggle Medical Mask Dataset ⁽⁸⁾, ele apresenta imagens reais de pessoas com diferentes tipos de máscaras. Entretanto também há imagens geradas através de *Data Augmentation*, a partir da rotação ou espelhamento de outras imagens que o compõem, o que prejudica a sua avaliação. Já o *dataset* MAFA ⁽⁹⁾ é composto por muitas imagens de profissionais de saúde em ambiente hospitalar, havendo muitas imagens posadas e similares entre si.

Outro trabalho que também utilizou 3 classes foi o de Siradjuddin et al. ⁽¹⁰⁾. Foram comparados o desempenho de dois modelos, o Faster R-CNN e o Fast R-CNN. Para isso, os autores utilizaram a junção de dois *datasets* diferentes, o MAFA ⁽¹¹⁾ e o Annotated Facial Landmarks in the Wild (AFLW dataset). Sendo que este último não se encontra



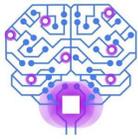
mais acessível através da URL citada por Siradjuddin et al. Algumas das imagens possuíam mais do que uma face. Do primeiro *dataset*, Siradjuddin et al. utilizaram 25.876 imagens e do segundo, 14.587. Assim, para o treinamento utilizaram 17.019 faces sem máscara, 3.842 com máscara incorreta e 25.610 com máscara correta. Por fim, o teste utilizou 1.000 faces de cada tipo. Como resultado, o Faster R-CNN obteve uma precisão média de 73% e o Fast R-CNN 39%. A classe com máscara incorreta apresentou o pior desempenho nos dois modelos, 57% para o primeiro e 17% para o segundo.

O quadro 1 consolida a revisão da literatura, apresentando os autores, quantidade de classes utilizadas, os modelos e os *datasets* utilizados. Foi possível identificar na literatura o incentivo na procura por modelos com melhor acurácia.

Quadro 1 – Resumo dos principais trabalhos da literatura correlata

Ref.	Título	Classes	Modelos	<i>Dataset</i> utilizado
(4)	Aplicação de visão computacional para o monitoramento do uso de máscaras de proteção	2	MTCNN/ Haar-cascade	(12)
(5)	On the Performance Analysis of a TensorFlow based Neural Network for Face Mask Detection	2	MobileNetV2	(13)
(6)	Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV	2	CNN próprio	(14) e (15)
(7)	Evaluación de AIoT en modelos computacionales en la nube y en el borde aplicado a la detección de mascarillas	3	YOLOv3	(8) e (9)
(10)	Faster Region-based Convolutional Neural Network for Mask Face Detection	3	Faster/ Fast R-CNN	(11) e -

O quadro 2 consolida os *datasets* utilizados na literatura. Além das referências, são apresentados os locais onde eles se encontram disponíveis, a natureza das imagens para o caso da máscara, a perspectiva da imagem, a quantidade de pessoas que aparecem nas imagens, a quantidade de classes apresentadas e por fim se existe uma licença de uso. Quanto à licença de uso, Maranhão⁽¹⁵⁾, apesar de ter informado uma licença no site Kaggle, não explica a origem das imagens e o site disponibilizado como referência, <https://makeml.app/datasets/mask>, não existe. Da análise das imagens que compõem os *datasets* também foram identificadas, em alguns deles, imagens muito similares ou produto de técnicas de *Data Augmentation*, tais como rotação, recorte e espelhamento. Assim, é possível notar a lacuna existente em relação a *datasets* frontais, voltados para 3 classes e que tenham uma licença de uso bem definida.

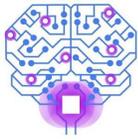
**Quadro 2** – Datasets utilizados na literatura

Ref.	Local	Natureza	Perspectiva	Pessoas	Classes	Licença
(8)	Kaggle	Real	Variado	1 ou +	3	Não
(9)	Google Drive	Real	Variado	1 ou +	3	Não
(11)	Kaggle	Real	Variado	1 ou +	3	Não
(12)	data-flair.training	Sintético	Frontal	1	2	Não
(13)	Github	Real	Variado	1 ou +	2	Não
(14)	Github	Sintético	Frontal	1	2	Não
(15)	Kaggle	Real	Variado	1 ou +	3	Sim*

Diante da análise da literatura correlata em que é considerado o problema de detecção de três classes, percebe-se que os trabalhos são limitados com relação à quantidade de modelos de detecção considerados nos estudos. No máximo, cada trabalho compara apenas dois modelos de aprendizado de máquina. Desta forma, percebe-se como uma lacuna de pesquisa a ausência de um estudo comparativo envolvendo mais modelos de aprendizado aplicados ao problema de detecção de três classes. Estudos nessa direção são importantes pois podem identificar os modelos de aprendizado de máquina com maior potencial para a construção de soluções computacionais que, ao viabilizar o monitoramento automático e a orientação de pessoas quanto ao uso (correto) de máscaras de proteção, contribuam para mitigar o espalhamento de doenças que se utilizam das vias aéreas superiores.

Diante da lacuna de pesquisa mencionada, este trabalho realizou um estudo voltado à comparação de desempenho de quatro arquiteturas de Redes Neurais Convolucionais (do inglês *Convolutional Neural Network* – CNN) através da abordagem em 3 classes, uma vez que no dia a dia é comum verificar pessoas com a máscara incorretamente posicionada, o que diminui a proteção oferecida pela máscara. Uma abordagem que considera apenas as duas classes com máscara e sem máscara, teria dificuldade em lidar com imagens de pessoas com a máscara incorretamente posicionada. Também foi adotada a abordagem da análise de controle de acesso a ambientes, que pressupõe a detecção em face frontal.

As quatro arquiteturas de CNN analisadas foram uma mais simples com quatro variações, e outras que são consideradas como o estado da arte na detecção automática de objetos: MobileNetV3⁽¹⁶⁾, Xception⁽¹⁷⁾ e VGG19⁽¹⁸⁾. A MobileNetV3-Large é a próxima



geração das MobileNets, estando ajustada para CPUs de celulares ⁽¹⁶⁾. A Xception implementa o módulo *inception* e obtém melhores resultados que a Inception V3 para o *dataset* ImageNet ^(17, 19). Por fim, a VGG19 é uma CNN com 19 camadas profundas ⁽¹⁸⁾.

Outra contribuição deste trabalho foi a construção de um *dataset* de imagens a partir de outros dois existentes, o que foi feito utilizando imagens sintéticas para as classes com máscara correta e com máscara incorreta. Quanto à avaliação dos modelos, foi utilizado outro *dataset*, com imagens reais.

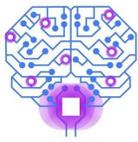
Métodos

Base de dados

Para o desenvolvimento deste trabalho, foram considerados os *datasets* Flickr-Faces-HQ3 (FFHQ) ⁽²⁰⁾ e MaskedFace-Net ⁽²¹⁾. O *dataset* Flickr-Faces-HQ3 (FFHQ) ⁽²⁰⁾ possui 70.000 imagens de faces humanas de diferentes etnias, idades, perspectiva e iluminação do rosto. Quanto à licença de uso, todas as imagens permitem o uso gratuito, redistribuição e adaptação para finalidades não comerciais ⁽²⁰⁾. Assim, o *dataset* Flickr-Faces-HQ3 proveu as imagens da classe “sem máscara”.

As imagens das classes “com máscara correta” e “com máscara incorreta”, foram obtidas no *dataset* MaskedFace-Net ⁽²¹⁾. Nele, as imagens do *dataset* FFHQ ⁽²⁰⁾ receberam de forma artificial uma máscara descartável azul, através da aplicação de um modelo deformável da máscara na face, o que traz mais realismo à imagem gerada. O *dataset* MaskedFace-Net possui 137.016 imagens. Quanto à classe “com máscara incorreta”, foram utilizados três tipos de erro: máscara cobrindo apenas o nariz e boca; apenas a boca e o queixo ou a máscara sob a boca. Quanto à licença de uso, ele está disponível para finalidades não comerciais (Creative Commons BY-NC-SA 4.0) ^(21, 22).

Dessa forma, a partir da junção dos dois *datasets* apresentados ^(20, 21) foi montado o conjunto de imagens utilizado no treinamento dos modelos propostos neste trabalho. A figura 2 ilustra essa junção. Para o teste e comparação dos modelos, foi utilizado outro *dataset*, com imagens reais, com diferentes modelos de máscaras. A utilização desse outro conjunto de imagens permitiu avaliar a capacidade de generalização dos modelos treinados com um *dataset* sintético diante de imagens reais. Para o *dataset* real, foram



escolhidas 50 imagens de pessoas sem máscara, a partir do *dataset* Flickr-Faces-HQ3 (FFHQ) ⁽²⁰⁾, outras 50 imagens de pessoas com máscara correta e 50 com máscara incorreta a partir dos *datasets* Kaggle Medical Mask Dataset ⁽⁸⁾ e MAFA ⁽⁹⁾. Para as imagens das classes com máscara correta e incorreta, foi realizado um recorte na imagem, deixando uma única face por imagem. Na avaliação experimental, não houve imagens repetidas ou similares entre os conjuntos de treinamento e teste.



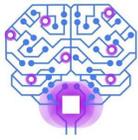
Figura 2 – Montagem do *dataset* de treinamento

Seleção de imagens

O método proposto foi aplicado em 4 etapas: seleção das imagens, construção dos modelos, treinamento e avaliação. Na primeira etapa, foi realizada a seleção das imagens a serem utilizadas no treinamento. As imagens foram selecionadas de forma pseudo-aleatória. Assim, a composição do subconjunto de imagens utilizado no treinamento foi de 8.000 imagens para a classe “sem máscara”, 7.624 imagens para a classe “com máscara incorreta” e 7.683 imagens para a classe “com máscara correta”.

Construção de modelos

Na etapa seguinte, foi realizada a construção dos modelos. Primeiro da arquitetura mais simples com algumas variações, e em seguida das arquiteturas MobileNetV3, Xception e VGG19. Para essa construção foram utilizadas as ferramentas TensorFlow e a API Keras, por meio da linguagem de programação Python. TensorFlow é uma interface



voltada para algoritmos de aprendizado de máquina ⁽²³⁾. Keras é uma API de alto nível que utiliza a plataforma TensorFlow 2 e que é compatível com Python 3.6–3.9. Ela foi escrita em Python e é voltada para aprendizado profundo ⁽²⁴⁾.

A primeira arquitetura, ilustrada pela figura 3, consistiu em duas camadas convolucionais e duas camadas de *pooling*. A primeira camada convolucional contou com 200 filtros e a função de ativação ReLU. Ela foi seguida por uma camada de *pooling* do tipo MaxPooling2D. A combinação de camadas convolucionais e camadas de *pooling* é uma recomendação comum na literatura ⁽²⁵⁾. Em seguida, foi utilizada outra camada convolucional, com 100 filtros, seguida por outra de MaxPooling2D. Essa segunda camada convolucional faz uma combinação das características extraídas pela primeira camada convolucional. Em seguida, foi utilizada uma camada de achatamento.

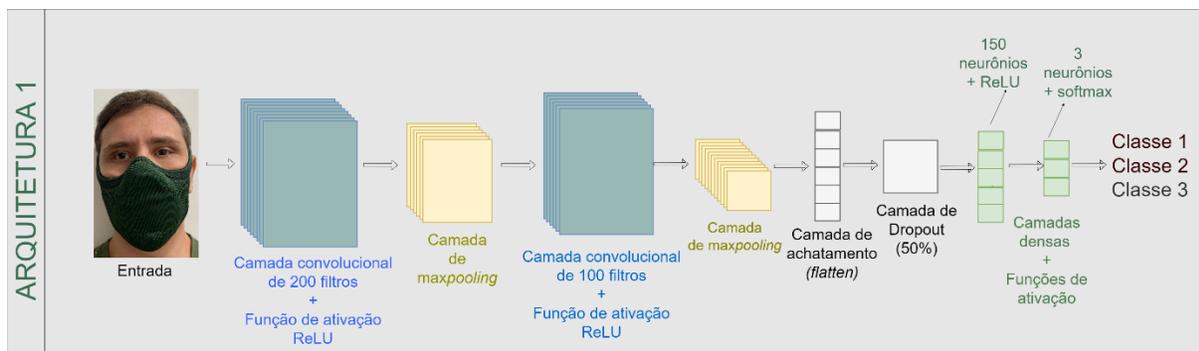
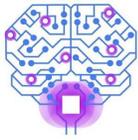


Figura 3 – Estrutura de camadas que compõe a Arquitetura 1

Visando prevenir o sobre-ajuste, foi adicionada também uma camada de *Dropout*, com taxa de 0,5, a qual só é utilizada durante o treinamento. Tal taxa significa que cada neurônio possui 50% de chance de ser momentaneamente desligado ⁽²⁵⁾, forçando a rede a se adaptar. Finalmente duas camadas densas, a primeira com 150 neurônios e função de ativação ReLU, e a última com apenas 3 neurônios (quantidade de classes do problema) e função de ativação *Softmax*.

Quatro variações dessa arquitetura foram treinadas e testadas. A primeira (A) utilizou os três canais (RGB), onde os pixels não foram normalizados. A segunda (B) converteu as imagens de 3 canais (RGB) para uma escala cinza. A terceira (C) realizou a normalização dos pixels (0 - 255 para 0 - 1) por intermédio de uma camada de



normalização no início da arquitetura. Por fim, a quarta variação (D) realizou a conversão para escala cinza e normalização dos pixels. Para o treinamento dos modelos, foram utilizadas 50 épocas e uma taxa de aprendizado de 0,0001.

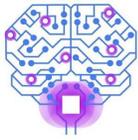
As arquiteturas pré-treinadas MobileNetV3, Xception e VGG19 foram modificadas de forma a contar ainda com as seguintes camadas: uma camada de entrada das imagens com dimensões (150, 150, 3), seguida por uma camada para normalizar os pixels para a faixa entre [-1, 1]. Ao final do núcleo pré-treinado, foram adicionadas uma camada de achatamento do tipo *Flatten* e uma camada de *Dropout* com taxa de 0,2, visando evitar um *overfitting*. Também se adicionou uma camada densa com 150 neurônios e a função de ativação ReLU. Por fim, uma camada de saída, do tipo densa com 3 neurônios (quantidade de classes do problema) e a função de ativação softmax.

Para o treinamento e ajuste dos pesos, duas abordagens foram utilizadas. Na primeira abordagem (I), o núcleo foi congelado, isto é, não teve seus pesos e parâmetros ajustados durante o treinamento. O treinamento com o *dataset* sintético foi realizado em 50 épocas, com uma taxa de aprendizado de 0,0001. Já na segunda abordagem (II), o núcleo também foi congelado para um treinamento inicial. Foi realizado o treinamento utilizando o *dataset* sintético, com uma taxa de aprendizado de 0,001 e 50 épocas. Depois disso, o núcleo foi descongelado, e todo modelo foi treinado com uma taxa de aprendizado de 0,00001 e 20 épocas.

Resultados e Discussão

Após a fase de treinamento, os modelos foram avaliados com o *dataset* não sintético (150 imagens). A tabela 1 apresenta os resultados obtidos com a primeira arquitetura, em que a (A) representa o modelo com 3 canais e não normalizado, (B) o modelo em escala cinza e não normalizado, (C) com 3 canais e normalizado, e por fim (D) em escala cinza e normalizado.

Os resultados mostram que tanto o modelo (A) como o (D) obtiveram uma acurácia muito próxima. Todos os modelos conseguiram uma boa revocação para a classe “sem máscara”, conseguindo identificar corretamente todas as pessoas que estavam de fato sem máscara. O problema é que algumas pessoas com máscara (correta ou incorreta) também foram identificadas como estando sem máscara, por isso a precisão ficou em



torno de 50% para essa classe nas quatro abordagens. Esses resultados foram inferiores aos obtidos por ⁽⁷⁾ e ⁽¹⁰⁾ com os modelos YOLOv3 e Faster R-CNN, respectivamente.

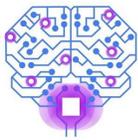
Tabela 1 – Resultados arquitetura 1

	A	B	C	D
Acurácia total	59,3%	58,0%	54,0%	60,0%
Precisão “sem máscara”	46,7%	50,5%	43,5%	51,0%
Precisão “máscara incorreta”	100,0%	70,6%	90,9%	76,9%
Precisão “máscara correta”	86,7%	76,5%	87,5%	76,9%
Revocação “sem máscara”	100,0%	100,0%	100,0%	100,0%
Revocação “máscara incorreta”	26,0%	48,0%	20,0%	40,0%
Revocação “máscara correta”	52,0%	26,0%	42,0%	40,0%
F1 “sem máscara”	63,7%	67,1%	60,6%	67,6%
F1 “máscara incorreta”	41,3%	57,1%	32,8%	52,6%
F1 “máscara correta”	65,0%	38,8%	56,8%	52,6%

A estratégia de utilizar a cor da imagem em escalas de cinza com a finalidade de o modelo generalizar o tipo de máscara, não obteve resultados satisfatórios. Isso pode ter ocorrido por haver mais detalhes além da cor que diferenciam os tipos de máscara.

Passando à análise dos modelos pré-treinados com o *dataset* Imagenet ⁽¹⁹⁾, a tabela 2 apresenta o desempenho de cada um dos modelos, tanto na abordagem I e II.

Para todas as três arquiteturas pré-treinadas, a abordagem II mostrou melhores resultados (acurácia). A arquitetura Xception utilizando a abordagem II obteve a melhor acurácia (86,0%), tendo inclusive superado os modelos apresentados na literatura ⁽⁷⁾ e ⁽¹⁰⁾. O Xception, abordagem II, obteve uma precisão igual ou superior a 80,0% nas três classes. Um ponto negativo, foi a revocação para a classe “máscara incorreta” em 68%, o que demonstra os casos de elementos dessa classe, classificados erroneamente na classe “sem máscara” ou “máscara correta”. O modelo MobileNetV3Large apresentou valores de acurácia próximos nas duas abordagens, seu menor desempenho pode ser explicado devido ao seu uso ser voltado para dispositivos móveis, com menor poder computacional. Já o modelo VGG19 apresentou a maior variação de acurácia entre as duas abordagens, com a acurácia quase duplicando.

**Tabela 2** – Resultados usando transferência de aprendizado

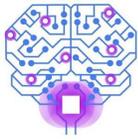
	MobileNetV3Large		Xception		VGG19	
	I	II	I	II	I	II
Acurácia total	59,3%	66,0%	67,3%	86,0%	42,0%	78,7%
Precisão “sem máscara”	48,1%	52,6%	56,8%	90,9%	36,5%	73,5%
Precisão “máscara incorreta”	100,0%	90,5%	85,7%	87,2%	100,0%	92,3%
Precisão “máscara correta”	82,1%	88,2%	81,3%	80,4%	100,0%	78,6%
Revocação “sem máscara”	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Revocação “máscara incorreta”	14,0%	38,0%	24,0%	68,0%	10,0%	48,0%
Revocação “máscara correta”	64,0%	60,0%	78,0%	90,0%	16,0%	88,0%
F1 “sem máscara”	64,9%	69,0%	72,5%	95,2%	53,5%	84,7%
F1 “máscara incorreta”	24,6%	53,5%	37,5%	76,4%	18,2%	63,2%
F1 “máscara correta”	71,9%	71,4%	79,6%	84,9%	27,6%	83,0%

Mesmo promissores, os resultados mostraram limitações na capacidade de generalização do modelo de classificação, uma vez que este foi treinado em um *dataset* que possui um único modelo de máscara e é sintético.

Conclusão

Este trabalho teve por objetivo a comparação de modelos de CNN para o problema da identificação da máscara de proteção em pessoas. Nesse problema, foram consideradas três classes: “sem máscara”, “com máscara incorreta” e “com máscara correta”. Também, neste trabalho foi construído um *dataset* de faces frontais, com imagens das três classes a partir de outros existentes.

Os modelos foram treinados a partir do *dataset* sintético construído, e depois avaliados com um *dataset* real. O modelo Xception, utilizando a abordagem de treinamento II, obteve uma acurácia de 86,0%, o que supera os resultados existentes na literatura analisada. Dessa forma, as duas principais contribuições deste trabalho foram o estudo comparativo e o *dataset* voltado para as 3 classes, com 203.708 imagens. Toda

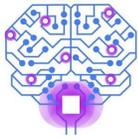


a implementação desenvolvida neste trabalho pode ser obtida através do link https://github.com/jlvoltan/cbis_covid.

Como oportunidade de trabalhos futuros, sugere-se a construção de um *dataset* com imagens reais das três classes estudadas em nossa proposta e a criação de imagens artificiais envolvendo outros tipos e modelos de máscaras.

Referências

1. Freitas CM, Barcellos C, Villela DAM, organizadores. Covid-19 no Brasil: cenários epidemiológicos e vigilância em saúde [Internet]. Série Informação para ação na Covid-19. Fiocruz; 2021. doi: <https://doi.org/10.7476/9786557081211>
2. Organização Pan-Americana da Saúde. Orientação sobre o uso de máscaras no contexto da COVID-19: Orientação provisória, 6 de abril de 2020.
3. Agência Nacional de Vigilância Sanitária (Brasil). Nota técnica GVIMS/GGTES/ANVISA nº 03/2022: orientações para prevenção e controle da Monkeypox nos serviços de saúde. Brasília: ANVISA; 2022.
4. Pereira Junior A, Teixeira F, Homem T. Aplicação de visão computacional para o monitoramento do uso de máscaras de proteção. CONICT. 2020.
5. Costa VL, Teixeira EH, Mafra SB, Rodrigues JJPC. On the Performance Analysis of a TensorFlow based Neural Network for Face Mask Detection. 2021 IWCMC. 2021.
6. Das A, Ansari MW, and Basak R, Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. 2020 IEEE 17th INDICON. 2020, 1-5, doi: 10.1109/INDICON49873.2020.9342585.
7. Quiñonez-Cuenca F, Maza-Merchán C, Cuenca-Maldonado N, Quiñones-Cuenca M, Torres R, Sandoval F, et al. Evaluación de AIoT en modelos computacionales en la nube y en el borde aplicado a la detección de mascarillas. Ingenius. 2022 Jun; (27).
8. Vich I. Medical masks dataset images tfrecords [Internet]. Kaggle. 2020. [cited 2022 Jun 06]. Available from: <https://bit.ly/3er0tb8>
9. Ge S, Li J, Ye Q, Luo Z. MAFA [Internet]. 2018. [cited 2022 Jun 06]. Available from: <https://bit.ly/3FBC52o>
10. Siradjuddin IA, Agustien, Muntasa A. Faster Region-based Convolutional Neural Network for Mask Face Detection. 5th ICICoS. 2021. 282-286, doi: 10.1109/ICICoS53627.2021.9651744.
11. Mangalampalli R. MAFA data [Internet]. Kaggle. 2020. [cited 2022 Jun 10]. Available from: <https://www.kaggle.com/datasets/rahulmangalampalli/mafa-data>



12. DataFlair. Download Face Mask Data – DataFlair [Internet]. [cited 2022 Jun 08]. Available from: <https://data-flair.training/blogs/download-face-mask-data/>
13. Wang Z, Wang G, Huang B, Xiong Z, Hong Q, Wu H, et al. Masked face recognition dataset and application. arXiv; 2020. doi <https://doi.org/10.48550/arxiv.2003.09093>
14. Bhandary P. Prajnash [Internet]. Github. 2020. [cited 2022 Jun 12]. Available from: <https://github.com/prajnasb/observations/tree/master/experiments/data>
15. Maranhão A. Face Mask Detection [Internet]. Kaggle. 2020. [cited 2022 Jun 12]. Available from: <https://www.kaggle.com/andrewmvd/face-mask-detection>
16. Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, et al. Searching for MobileNetV3 [Internet]. arXiv; 2019. doi: 10.48550/ARXIV.1905.02244 [cited 2022 Jun 25]. Available from: <https://arxiv.org/abs/1905.02244>
17. Chollet F. Xception: Deep learning with depthwise separable convolutions [Internet]. arXiv; 2017. doi: 10.48550/ARXIV.1610.02357 [cited 2022 Jun 27]. Available from: <https://arxiv.org/abs/1610.02357>
18. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv; 2014. doi: 10.48550/ARXIV.1409.1556
19. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural network. Advances in neural information processing systems 2012; 25.
20. NVIDIA. FFHQ dataset [Internet]. 2019. [cited 2022 Jun 25]. Available from: <https://github.com/NVLabs/ffhq-dataset>
21. Cabani A, Hammoudi K, Benhabiles H, Melkemi M. MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19. Smart Health. 2021; 19. [cited 2022 Jun 25]. Available from: <https://www.sciencedirect.com/science/article/pii/S2352648320300362>
22. Hammoudi K, Cabani A, Benhabiles H, Melkemi M. Validating the correct wearing of protection mask by taking a selfie: design of a mobile application "CheckYourMask" to limit the spread of COVID-19. CMES-Computer Modeling in Engineering & Sciences. 2020. 124 (3):1049-59.
23. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado G, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv; 2016. doi 10.48550/ARXIV.1603.04467.
24. Chollet F, et al. Keras [Internet]. GitHub; 2015. [cited 2022 Jun 29]. Available from: <https://github.com/fchollet/keras>.
25. Géron A. Mãos à Obra Aprendizado de Máquina com Scikit-Learn e TensorFlow: Conceitos, Ferramentas e Técnicas Para a Construção de Sistemas Inteligentes. Rio de Janeiro: Alta Books; 2019.