

Implementação de Rede Neural Convolutacional para Predição de COVID-19 Através de Imagens de Raio X

Erik Gabriel Cruz Sena
Graduando em Engenharia Mecânica
Instituto de Ciência e Tecnologia
Universidade Federal
dos Vales do Jequitinhonha e Mucuri
Diamantina - Minas Gerais
e-mail: erik.gabriel@ufvjm.edu.br

Honovan Paz Rocha
Instituto de Engenharia, Ciência e Tecnologia
Universidade Federal
dos Vales do Jequitinhonha e Mucuri
Janaúba - Minas Gerais
e-mail: honovan.rocha@ufvjm.edu.br

Resumo—A COVID-19 é uma das grandes ameaças do século, desta forma, tornou-se uma necessidade prevenir a propagação do vírus para diminuir a carga nos leitos de UTI. Desta maneira, torna-se relevante o desenvolvimento de ferramentas para auxílio ao diagnóstico de casos positivos da doença. Neste contexto, este trabalho tem como objetivo o estudo, implementação e ajuste de uma Rede Neural Convolutacional para classificação de pacientes com a doença. O modelo implementado foi inicialmente aplicado ao banco de dados MNIST com o propósito de ajustar e validar a arquitetura da rede, em seguida, foi ajustado e aplicado a uma base de dados composta por imagens de radiografia da região torácica de pacientes saudáveis e com COVID-19. Os experimentos realizados mostraram que a abordagem proposta obteve desempenho superior a alguns métodos do estado da arte, classificando corretamente mais de 97% dos pacientes com COVID-19. Os resultados demonstram que a abordagem proposta gera boa separabilidade entre classes, podendo ser usado como ferramenta de auxílio ao diagnóstico.

Palavras-chave—Redes Neurais Convolutacionais, Reconhecimento de Padrões, COVID-19, Imagens de raio X.

I. INTRODUÇÃO

Segundo [1], as espécies de coronavírus são um grupo de vírus de RNA altamente diversos, envelopados, de sentido positivo e de fita simples. Eles causam várias doenças envolvendo os sistemas respiratório, entérico, hepático e neurológico. Considerando-se os sintomas físicos, envolvem frequentemente tosse, febre e dificuldades respiratórias [2].

Com a rápida escalada da doença a nível global, a COVID-19 logo se tornou uma pandemia. Alguns países têm adotado medidas, tais como isolamento de casos suspeitos, fechamento de escolas e universidades, distanciamento social e *lockdown* a fim de reduzir os impactos da pandemia, diminuindo o pico de incidência e o número de mortes [3].

Dessa forma, considerando a grande disponibilidade de dados relativos à pacientes suspeitos de COVID-19, vários trabalhos nas áreas de Ciência dos Dados e Inteligência Artificial começaram a ser desenvolvidos com o objetivo de auxiliar no entendimento e contenção desta doença. A disponibilidade de bases de dados contendo imagens de raio

X da região torácica de pacientes com suspeita de COVID-19, alavancou pesquisas na área de aprendizado de máquina aplicada, destacando-se algumas técnicas com casos recentes de sucesso no campo de reconhecimento/classificação de padrões em imagens. Dentre estas técnicas estão as conhecidas como Redes Neurais Profundas (em inglês, *Deep Neural Networks* - DNN), Transferência de Aprendizado (em inglês, *Transfer Learning*) e Redes Neurais Convolutacionais (em inglês, *Convolutional Neural Networks* - CNN).

As CNN's, com origens no final da década de 1970 através do trabalho pioneiro de Kunihiko Fukushima [4] e consolidadas pelo trabalho de Yann Lecun [5] em 1998, é um tipo de modelo de aprendizado profundo que se tornou dominante em tarefas de visão computacional. O que torna as CNN's eficientes são os filtros convolutacionais capazes de extrair características importantes para classificação, além de levarem em consideração a estrutura espacial das imagens [6].

Nesse contexto, este trabalho tem o objetivo o estudo, implementação e ajuste de uma CNN para classificação de pacientes com COVID-19. A arquitetura da CNN foi ajustada e validada utilizando-se uma base de dados de *benchmark*, em seguida o modelo foi aplicado a uma base de dados contendo imagens de raio X da região torácica de pacientes, com o objetivo de classificá-los indicando se existem ou não lesões pulmonares, e por conseguinte apontando se provavelmente estão infectados pela COVID-19.

Os resultados obtidos através dos experimentos realizados, demonstraram eficácia do modelo proposto de 97,58%, considerando-se a detecção de pacientes com a doença.

Este trabalho está organizado da seguinte maneira: A Seção II apresenta alguns trabalhos recentes relacionados à detecção de COVID-19 utilizando-se Redes Neurais. A Seção III, apresenta estudos relacionados à classificadores e aprendizado de máquinas. A Seção V, descreve a metodologia utilizada para ajuste e validação da arquitetura da CNN, bem como a adequação e aplicação do modelo final para a base de dados de raios X. Ademais, a Seção VI apresenta o experimento

proposto, como também os resultados e discussões relativos à aplicação da CNN à base de dados, objeto de estudo. Por fim, a Seção VII demonstra as conclusões que obtivemos com este trabalho, além de listarmos possíveis trabalhos futuros.

II. TRABALHOS RELACIONADOS

Considerando-se que a detecção de COVID-19 é um assunto extremamente recente, é possível perceber uma grande quantidade de trabalhos aplicados ao tema, em parte isso é devido à emergência para estudo desta doença que demonstra letalidade e alta taxa de espalhamento global. Desta forma, elencamos alguns dos trabalhos mais recentes relacionados à aplicação de técnicas de *machine learning* à detecção de COVID-19 com base em imagens de raio X.

No trabalho de [7] propõe usar *Transfer Learning* para detecção de casos positivos de COVID-19. O aprendizado por transferência é um método de aprendizado que reutiliza uma rede pré-treinada como ponto de partida. Dito isso, o modelo foi treinado com 224 imagens positivas da doença, junto com 714 imagens de pneumonia bacteriana e viral e 504 imagens em condições normais. As imagens, por sua vez, foram redimensionadas para um tamanho de 200 x 266 pixels e treinadas em diferentes modelos de Aprendizagem por Transferência. Conclui que o melhor modelo é usando MobileNet v2, obtendo uma sensibilidade de 98,66%.

Em sua pesquisa, [8] utilizou imagens de radiografias de tórax prontamente disponíveis com achados clínicos de casos positivos de COVID-19 (154 imagens). O modelo de rede convolucional selecionado foi a ResNet-101, entretanto, esse modelo já foi pré-treinado para reconhecer objetos de outras várias imagens. Dessa forma, para completar o conjunto de testes, foi simulado um cenário cuja taxa de prevalência, calculada como o número de casos de uma doença em população durante um período de tempo, é de 2,57%. Assim, extraiu-se 5.828 imagens não rotuladas do conjunto de imagens ChestX-ray14. Os autores concluíram que usando a CNN pré-treinada obteve uma sensibilidade de 77,3%.

Em [9] aborda um estudo utilizando 3 modelos de CNN, Inception V3, Xception e ResNeXt a fim de selecionar a melhor dentre elas. Para análise do desempenho, foram coletadas 6.432 amostras do repositório *Kaggle*, todas elas reduzidas para um tamanho de 128 x 128 pixels. Por fim, a conclusão deste trabalho é que dos 3 modelos, a Xception tem o melhor desempenho, podendo ser usada para classificar pacientes.

Para [10], foram usados cinco técnicas de aprimoramento de imagem para detecção de COVID-19 utilizando *dataset* com 3 classes diferentes. Essas técnicas como correção gamma e complemento, tem por objetivo melhorar a percepção da patologia. Foram usadas 6 redes convolucionais para o treinamento, testando para imagens segmentadas e não segmentadas do pulmão. Finalmente, conclui que o melhor modelo é aquele que emprega a correção gamma como aprimoramento juntamente com imagem não segmentada.

III. CLASSIFICADORES

Nesta seção, abordaremos os principais algoritmos para classificação implementados neste trabalho, incluindo-se algoritmos clássicos utilizados apenas para efeitos de comparação durante o processo de validação da CNN proposta.

A. Multilayer Perceptron

Segundo [11], as redes neurais artificiais (RNAs) são muito úteis quando utilizadas para a resolução de problemas, tais como, reconhecimento de padrões, classificação e previsão. Nesse contexto, a Perceptron de Múltiplas Camadas (em inglês, *Multilayer Perceptron* - MLP) destaca-se pela sua popularidade e aplicabilidade. Uma rede MLP é constituída por uma camada de entrada (*input layer*), uma ou mais camadas escondidas (*hidden layer*) e uma camada de saída (*output layer*).

O funcionamento de um neurônio é dado pelo ponderamento dos pesos sinápticos com um sinal recebido e o resultado dessa operação, é processado por uma função matemática chamada de função de ativação. Essas funções de ativação são essenciais, pois elas fazem transformações não-linear nos dados de entrada, tornando a rede neural capaz de aprender e executar tarefas mais complexas [12]. Dentre as funções de ativação que são usadas, destacam-se a Sigmóide, a Tangente Hiperbólica, a ReLU e a Softmax.

A função Sigmóide é amplamente utilizada, já que simula uma saída binária, mas com geração de uma não linearidade obtida através da geração de valores reais que tem limites definidos na faixa entre 0 e 1. A função sigmóide ou logística e a sua respectiva derivada estão representadas em (1).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (1)$$

Outra importante função de ativação é a Tangente Hiperbólica. Sendo definida a partir do seno e cosseno hiperbólicos, funciona como uma versão escalada da função logística, atuando no intervalo [1, -1]. A função e a sua respectiva derivada podem ser vistas em (2).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \tanh'(x) = \frac{1}{\cosh^2(x)} \quad (2)$$

A função Unidade Linear Retificada (em inglês, *Rectified Linear Unit* - ReLU) é uma função fácil de otimizar, pelo fato de ser similar a função linear. Além disso, ela tende a ter mais efetividade comparada com as funções sigmoidais, principalmente na questão de execução. Entretanto, a sua eficácia varia com o problema. A função ReLU e a sua derivada estão representadas em (3).

$$\begin{aligned} ReLU(x) &= \max(0, x) \\ ReLU'(x) &= \left\{ \begin{array}{l} 1, \text{ se } x \geq 0 \\ 0, \text{ caso contrário} \end{array} \right\} \end{aligned} \quad (3)$$

Por fim, a função Softmax força a saída de uma rede neural a representar a probabilidade dos dados serem de uma das classes definidas. Ela não costuma ser aplicada nas camadas escondidas da rede, mas sim na camada de saída, onde realmente há interesse em gerar as probabilidades para definir a classe. A função Softmax pode ser representada em (4).

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ para } j = 1, \dots, K. \quad (4)$$

B. Redes Neurais Convolucionais

No contexto de redes MLP, quando passamos a utilizar duas ou mais camadas ocultas para construção de sua arquitetura, temos o surgimento do termo aprendizado profundo (em inglês, *Deep Learning* - DP). Dentre os diversos tipos de DP, considerando-se os mais conhecidos e utilizados estão as chamadas Redes Neurais Convolucionais, que trouxeram avanços em processamento de imagens, voz e vídeo. Esses avanços são obtidos através de modelos computacionais compostos por várias camadas de processamento, e isso faz com que aprendam representações de dados com múltiplos níveis de abstrações [13].

De forma geral, pode-se tentar compreender como uma rede neural convolucional funciona, através da Fig. 1. Nela podemos observar uma imagem passando por um processo de convolução, *pooling* e *fully connected*.

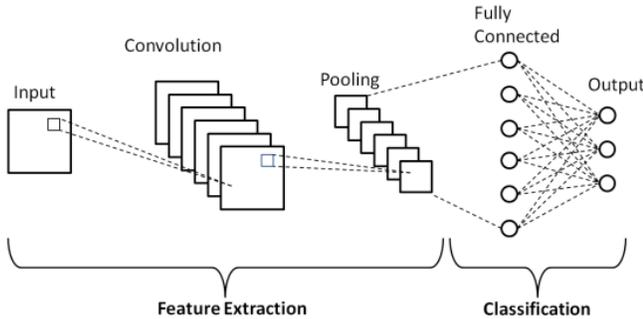


Figura 1. Exemplo de rede neural convolucional [14]

1) *Convolução*: A convolução é um processo em que utilizam-se filtros, também chamado de *kernels*, para captar características importantes para a rede. Esses filtros são formados por pesos inicializados aleatoriamente e atualizando-os através do processo de retropropagação do erro.

Além disso, segundo [15] os filtros são usados em locais diferentes da imagem. Isto acaba fazendo com que os padrões que ocorrem com frequência, situado em qualquer ponto da matriz, possam ser aprendidos. Essa característica, é denominada compartilhamento de parâmetros.

A operação de convolução, representada pelo símbolo $*$, é descrita a partir de (5)

$$g(i, j) = (x * k)[i, j] = \sum_m \sum_n x[m, n] k[i - m, j - n], \quad (5)$$

em que, $g(i, j)$ é a saída ativada para a próxima camada, x é a entrada da imagem e k é a matriz de filtros.

O passo seguinte após a convolução, é a aplicação de uma função de ativação para geração de uma não linearidade na saída desta camada. Ela pode ser usada para ajustar ou cortar o resultado, sem ela o modelo se torna apenas um modelo de regressão linear.

2) *Pooling*: A camada de *pooling* nada mais é que uma camada para simplificar a informação da camada anterior, a fim de reduzir a complexidade para outras camadas. O *maxpooling* é o método mais utilizado, ele divide a imagem em uma sub-região determinada, por exemplo 2x2, e retorna o valor máximo dentro dessa sub-região [12]. Esse tipo de método serve para diminuir a quantidade de pesos a serem aprendidos e também para evitar *overfitting*, quando o algoritmo se ajusta bem ao conjunto de dados, mas é ruim para generalização, e alto custo operacional.

3) *Fully Connected*: Nesta etapa, a camada totalmente conectada (em inglês, *fully connected*) é semelhante como os neurônios são organizados em uma rede *Multilayer Perceptron*. Dessa forma, cada nó em uma camada totalmente conectada está diretamente conectado a todos os nós na camada anterior e na próxima.

C. Support Vector Machines

Máquina de Suporte Vetorial (em inglês, *Support Vector Machines* - SVM) é um algoritmo cujo objetivo é classificar um determinado conjunto de pontos de dados que são mapeados para um espaço de características usando uma função denominada de *kernel* [16]. Nesse sentido, essas funções realizam transformações de dados complexas fazendo com que os atributos passem a ser linearmente separáveis.

De forma mais simplificada, o SVM encontra uma reta de separação, chamada de hiperplano, em que se busca maximizar a distância entre os atributos mais próximos de cada classe.

Outro importante parâmetro a definir é o parâmetro de custo C . Para [17], a margem é maior para $C = 0,01$ do que $C = 10.000$. Portanto, valores maiores de C focam em pontos, classificados corretamente, próximos ao limite de decisão, enquanto que valores menores envolvem dados mais distantes.

D. Random Forest

Criado em 1995 por Tim Kan Ho [18], as florestas aleatórias (em inglês, *Random Forest* - RF) são classificadores de conjuntos de árvores de decisão, com cada árvore cultivada usando algum tipo de randomização. As RFs têm a capacidade de processar grandes quantidades de dados com altas velocidades de treinamento, com base em uma árvore de decisão.

De acordo com [17], as *Random Forests* são uma forma de calcular a média de várias árvores de decisão, treinadas em diferentes partes do mesmo conjunto de treinamento, com o objetivo de reduzir a variância e o viés.

Uma vantagem desse algoritmo é a sua aplicação, podendo ser utilizado tanto para tarefas de classificação quanto para regressão.

E. Gradient Boosting

O *Gradient Boosting* (GBM) é um algoritmo de *Boosting* bastante popular devido a sua grande adaptabilidade e é usado principalmente em competições de *machine learning*. Essa técnica usada no GBM, constrói um conjunto de árvores, com cada árvore aprendendo e melhorando a anterior, diferentemente das *Random Forests* que constroem um conjunto de árvores independentes [19]. Além disso, o GBM treina iterativamente novos modelos com um enfoque nos erros dos modelos anteriores, tornando a predição mais resistente ao viés.

Segundo [20] uma das maiores motivações de usar o GBM é que ele permite otimizar uma função de custo especificada pelo usuário, em vez de uma função de perda que geralmente oferece menos controle e não corresponde a algumas aplicações do mundo real.

IV. MÉTRICAS DE AVALIAÇÃO

Dentre as principais formas de avaliação de modelos preditivos, temos a matriz de confusão, que permite avaliação consistente do modelo. A matriz de confusão é uma matriz que contém informações relacionadas à classificações reais e previstas. Utilizada em problemas de classificação binária, ela busca entender a relação entre acertos e erros que o modelo apresenta [21]. Uma matriz de confusão é representada pela Fig. 2.

		Real	
		Sim	Não
Predito	Sim	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Não	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Figura 2. Matriz de confusão para problema de classificação binária. Adaptado de [22]

em que, VP significa que a classe prevista e real originalmente fazem parte da classe positiva, FP significa que a classe predita retornou positiva, porém a classe original observada era negativa, VN expressa que tanto os valores preditos e reais fazem parte da classe negativa e FN representa que o valor predito resultou na classe negativa, mas a classe original era positiva.

Após encontrar esses valores e obter a matriz de confusão, é possível calcular métricas de avaliação para estabelecer o quão preciso é o modelo [22]. Existem várias métricas, e cada uma delas tem peculiaridades que se encaixa melhor em um determinado estudo.

A Acurácia, por exemplo, indica o desempenho em geral do modelo, ou seja, representa a taxa de acerto de todo

o classificador. A Precisão, indica quantas classes positivas o modelo acertou dentre todas as classificações de classe positiva. Em outras palavras, quanto maior a precisão, menor o erro de falsos positivos. Outra métrica é a Sensibilidade ou *Recall*, no qual representa a taxa de valores classificadas como positivo, comparada com quantos deveriam ser. Um *recall* alto indica que o classificador produziu poucos falso negativos. Por fim, o F1-Score tem como objetivo unir precisão e sensibilidade a fim de ter um único valor que determine a qualidade geral do modelo. Dessa forma, quando essa métrica tiver um valor alto, o modelo terá uma boa precisão quanto uma boa sensibilidade. O cálculo dessas métricas de avaliação podem ser encontrados na Tabela I.

Tabela I
MÉTRICAS DE AVALIAÇÃO

Acurácia	Precisão	Sensibilidade	F1-Score
$\frac{VP+VN}{VP+VN+FP+FN}$	$\frac{VP}{VP+FP}$	$\frac{VP}{VP+FN}$	$\frac{2 \cdot VP}{2 \cdot VP+FP+FN}$

A curva AUC-ROC como [23] descreve, é uma métrica de avaliação no qual é medida o desempenho de um modelo através de gráfico. ROC (em inglês, *Receiver Operating Characteristics*) é uma curva de probabilidade, que mede o número de vezes que o classificador errou a predição. Já o AUC (em inglês, *Area Under the Curve*), representa quanto o modelo é capaz de distinguir entre as classes.

Um bom algoritmo de classificação tem um AUC próximo de 1, o que simboliza que o algoritmo separa bem as classes. AUC próximo de 0, tem a pior medida de separabilidade. E quando o valor é 0,5 significa que o modelo não tem capacidade de separação.

V. METODOLOGIA

A fim de se realizar um estudo sobre o ajuste de parâmetros em Redes Neurais, foi utilizado o banco de dados da MNIST de dígitos manuscritos, que é comumente usado por pesquisadores que desejam obter técnicas de aprendizado em reconhecimento de padrões.

Os modelos de redes neurais e operações de convolução utilizados neste trabalho foram implementados em linguagem *Python* versão 3.7, com o auxílio apenas de bibliotecas básicas de operações matemáticas como a *Numpy*. Considerando-se os outros métodos implementados (SVM, RF e GBM) o pacote *scikit-learn* [24] na versão 0.24.1 foi utilizado como *framework* base. As configurações gerais da máquina utilizada para os experimentos são: CPU i5-9300H 2.4GHz, RAM 8GB e sistema operacional Windows 10 64-bit.

Dessa forma, para essa base de dados precisou-se primeiramente ajustar a estrutura da MLP. Sendo assim, o algoritmo implementado foi configurado da seguinte forma: gradiente descendente como método de otimização, um tamanho de lote de 50, escolhido empiricamente através de uma busca linear e um número máximo de 100 épocas. Além disso, foi definido uma taxa de aprendizagem adaptativa, cujo o objetivo é fazer grandes mudanças no início do treinamento e ir diminuindo

no decorrer do procedimento, para que pequenas mudanças fossem feitas a fim de otimizar o treinamento. O algoritmo foi executado 10 vezes para cada quantidade de neurônios na camada escondida a fim de colher métricas de estatística descritiva para a acurácia obtida, como o desvio padrão, a média e também o melhor e pior caso possível.

Variando a quantidade de neurônios na camada escondida, um resultado com bom *tradeoff* entre acurácia e tempo de treinamento foi obtido utilizando 300 neurônios. Com esse parâmetro encontrado, foram feitos vários outros testes variando as funções de ativação. Os resultados desse experimento podem ser vistos na Tabela II.

Tabela II
DADOS OBTIDOS UTILIZANDO 300 NEURÔNIOS JUNTO COM A TAXA DE APRENDIZAGEM ADAPTATIVA

Funções		Acurácia (%)	Melhor caso (%)	Pior caso (%)	Tempo de treinamento médio (s)
Camada escondida	Camada de saída				
ReLU	Softmax	97,78 ± 0,12	97,94	97,54	610,46
ReLU	Sigmóide	96,71 ± 0,14	97,00	96,53	567,89
ReLU	Tanh	96,18 ± 0,08	96,28	96,01	613,60
Sigmóide	Sigmóide	94,77 ± 0,17	95,15	94,50	692,39
Sigmóide	Tanh	95,63 ± 0,20	96,01	95,19	658,95
Tanh	Sigmóide	94,16 ± 0,16	94,49	93,96	641,77
Tanh	Tanh	95,18 ± 0,10	95,40	95,08	695,67

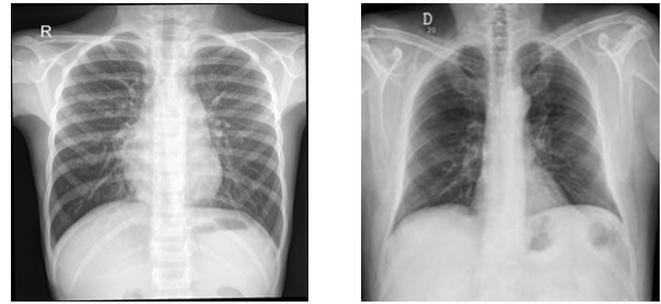
Pode-se perceber que a combinação das funções $ReLU(x)$ e $Softmax(x)$ geraram o melhor resultado quanto a acurácia, além disso, o tempo de treinamento médio sofreu pouca variação neste comparativo.

Após a implementação e ajuste da rede neural, os parâmetros da MLP foram incluídos na CNN que por sua vez, foi configurada para ter 2 camadas de convolução, sendo que cada uma dessas camadas passa por um processo de *maxpooling* e esta foi aplicada para um conjunto de dados de COVID-19.

A base de dados COVID-19 é formada de diferentes conjuntos de dados disponíveis publicamente e disponibilizado no repositório *Kaggle* [25]. Esse conjunto possui imagens de raios X de tórax para casos positivos de COVID-19, pneumonia viral e casos saudáveis.

Foi utilizado essa de base de dados na sua ultima versão, contendo 3.616 imagens de COVID-19 no qual a maioria das imagens (2.743) são coletadas do conjunto de dados BIMCV-COVID19 [26], 10.192 imagens de casos normais e 1.345 de pneumonia viral. Porém, para esse estudo foi utilizado somente imagens de COVID-19 e saudáveis, mostrado na Fig. 3.

O número de imagens aproveitadas de casos COVID-19 e saudáveis foram ambas de 3.520, visto que, o conjunto de dados possuía alguns ruídos que poderiam afetar o resultado final, como mostra a Fig. 4. Sendo assim, foi empregado *undersampling* como procedimento para diminuição dos dados, que consiste em reduzir de forma aleatória os exemplos da classe majoritária.



(a) Raio X de caso saudável

(b) Raio X de caso COVID-19

Figura 3. Imagens de caso positivo de COVID-19 e saudáveis retiradas da base de dados [25]

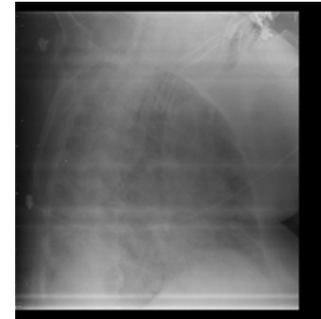


Figura 4. Radiografia indefinida [25]

As imagens de radiografias foram redimensionadas para serem aplicadas na rede com intuito de reduzir parâmetros. Elas foram redimensionadas para 150x150 pixels, além de serem normalizadas através de (6)

$$X = \frac{X - \text{mean}(X)}{\text{std}(X)}, \quad (6)$$

onde, X representa o vetor de dados da imagem, $\text{mean}()$ é média aritmética e $\text{std}()$ é o desvio padrão.

Dito isso, o treinamento foi feito utilizando ADAM como método de otimização com intuito de melhorar a performance do algoritmo e um tamanho de lote, escolhido empiricamente, de 32. Ainda foi empregado 15 filtros convolucionais de tamanho 3x3 em cada camada de convolução e um número máximo de 4 épocas.

Para aplicação dos outros métodos utilizados (SVM, RF e GBM) a base de dados também foi redimensionada para 150x150 pixels além de ser normalizada e vetorizada. Os hiperparâmetros selecionados para esses algoritmos foram obtidos a partir de trabalhos presentes na literatura sobre classificação de imagens, uma vez que esses métodos servirão apenas de comparação com a CNN implementada.

Por fim, foi usado o método estatístico validação cruzada com k partições (em inglês, *k-fold cross validation*) para todos os outros métodos utilizados. Esse método é comumente usado na literatura para estimar a capacidade de modelos de aprendizado de máquina, medindo o desempenho do modelo de forma geral.

VI. RESULTADOS

Para o experimento com a base de dados COVID-19, utilizamos algumas métricas de avaliação, considerando-se que, para este problema em específico, a principal métrica a ser avaliada é a sensibilidade (*recall*). Essa métrica, é usada em situações em que os falsos negativos são mais importantes do que os falsos positivos, ou seja, o modelo é treinado com enfoque especial na classificação correta de pacientes com COVID-19, visando errar o mínimo possível. Isso foi definido tendo em vista uma interpretação na vida real de uma instância de falso negativo, onde iria resultar em uma suposição equivocada de que o paciente não está infectado com a doença, o que poderia acarretar um diagnóstico tardio além de contribuir para a propagação do vírus.

Dessa forma, a precisão é menos relevante para este trabalho, visto que, a gravidade em classificar pacientes saudáveis como se tivessem a doença é algo menos crítico, uma vez que os procedimentos para tratamento da doença não são debilitantes considerando-se o que a literatura demonstra até a data deste estudo. Entretanto, do ponto de vista estatístico, um resultado ruim quanto à métrica precisão pode representar um baixo valor de acurácia para o modelo, indicando falsamente que o modelo não classifica bem pacientes com a doença.

A Tabela III, mostra o desempenho do algoritmo usando *k-fold cross validation*. Neste trabalho, utilizou-se $k = 10$ em todos os experimentos, a fim de reduzir o viés.

Tabela III
DESEMPENHO OBTIDO PELA CNN PROPOSTA

Métricas	Média	Desvio Padrão	Mínimo	Máximo
Acurácia	89,60%	3,72%	84,09%	93,76%
Precisão	81,19%	6,79%	69,60%	88,35%
<i>Recall</i>	97,58%	1,83%	94,23%	99,35%
F1-Score	88,38%	4,55%	81,00%	93,00%
AUC	90,15%	3,11%	86,06%	93,75%
Tempo de execução (h)	71,76	0,06	71,69	71,83

Através da Tabela III, verifica-se que os resultados do experimento são promissores, considerando-se a métrica avaliada e usando um método de validação robusto e consolidado na literatura.

O banco de dados também foi submetido aos classificadores *Random Forest*, *Gradient Boosting* e SVM, com o intuito de relacionar a capacidade dos algoritmos em encontrar resultados que sejam comparáveis à CNN implementada.

Para o RF, os parâmetros foram coletados a partir do trabalho de [27], em conjunto com uma rede pré treinada, a Resnet 50, para extração de características. Entretanto, o classificador RF neste estudo usou a técnica de *bootstrap aggregating - bagging*, que é projetado para melhorar a estabilidade e a precisão dos algoritmos de aprendizado de máquina, além de reduzir a variância. Dessa forma, usando uma profundidade de árvores igual a 20 e o número de árvores na floresta igual a 100, este método produziu o melhor desempenho com relação à métrica precisão, além de apresentar o menor custo computacional.

A Tabela IV mostra os resultados obtidos pelo *Random Forest* considerando-se as métricas utilizadas.

Tabela IV
DESEMPENHO OBTIDO PELO *Random Forest*

Métricas	Média	Desvio Padrão	Mínimo	Máximo
Acurácia	86,59%	5,79%	78,83%	93,89%
Precisão	84,81%	8,69%	75,43%	98,43%
<i>Recall</i>	90,33%	3,28%	83,52%	96,02%
F1-Score	87,29%	5,18%	79,78%	93,75%
AUC	86,60%	5,80%	78,83%	93,89%
Tempo de execução (s)	68,15	1,61	67,13	69,48

A Tabela V mostra os resultados encontrados utilizando-se a SVM. Os parâmetros para este classificador foram obtidos do trabalho de [28] onde a SVM foi implementada juntamente com uma rede convolucional pré-treinada, empregada para classificar casos positivos de COVID-19. O *kernel* linear foi usado na SVM.

Tabela V
DESEMPENHO OBTIDO PELA SVM

Métricas	Média	Desvio Padrão	Mínimo	Máximo
Acurácia	79,93%	5,54%	70,17%	88,30%
Precisão	79,51%	6,61%	70,05%	89,84%
<i>Recall</i>	81,09%	5,40%	70,45%	88,63%
F1-Score	80,31%	5,42%	70,25%	88,25%
AUC	79,91%	5,55%	70,17%	88,25%
Tempo de execução (min)	32,61	3,22	29,67	35,98

A Tabela VI mostra os resultados obtidos utilizando-se o GBM. Os parâmetros utilizados para este classificador são os mesmos empregados no trabalho de [29], no qual foi usado o algoritmo *Stochastic Gradient Boosting* para melhorar o desempenho de um modelo de classificação. O GBM foi aplicado à base usando uma profundidade de árvores igual a 20, número de árvores na floresta igual a 100 e uma taxa de aprendizagem de 0,1.

Tabela VI
DESEMPENHO OBTIDO PELO *Gradient Boosting*

Métricas	Média	Desvio Padrão	Mínimo	Máximo
Acurácia	75,76%	5,44%	67,61%	82,67%
Precisão	76,03%	7,13%	66,66%	86,16%
<i>Recall</i>	76,16%	4,82%	70,45%	86,93%
F1-Score	75,95%	4,94%	68,61%	82,20%
AUC	75,76%	5,44%	67,61%	82,67%
Tempo de execução (h)	5,23	0,09	5,31	5,15

Ao analisar a Tabela VI, nota-se que o tempo de execução do algoritmo é alto em comparação ao SVM e *Random Forest*. Diante disso, como o principal objetivo desse método é reduzir o viés dos modelos simples, o GBM acaba levando bastante tempo de execução, visto que, o treinamento dos modelos são feitos de maneira gradual, aditiva e sequencial.

Observando as tabelas IV e V, percebe-se que o modelo usando SVM obteve um resultado inferior a *Random Forest* e CNN. Isso pode ter ocorrido devido à dependência da

SVM com relação à escolha do *kernel*, além de possivelmente precisar de um ajuste mais refinado para a base de dados utilizada. Em contrapartida, a RF busca a melhor característica em um subconjunto aleatório das características, criando uma grande diversidade e gerando melhores modelos, obtendo resultados mais satisfatórios [17]. Entretanto, ao comparar os resultados obtidos utilizando a CNN e a RF pode-se perceber que a RF converge com um baixo tempo de execução, contudo, obtém resultados inferiores à CNN. Isso ocorre pois as redes convolucionais tendem a aprenderem certas características importantes para a classificação. As *Random Forests* tendem a levar em consideração a classe majoritária com um peso maior, devido à forma classificação das árvores.

Nesse sentido, as redes neurais convolucionais predominam em classificar e distinguir casos saudáveis e casos positivos de COVID-19 em imagens de raio x. A Fig. 5 e a Fig. 6 mostram, respectivamente, a matriz de confusão e a curva ROC do melhor resultado obtido pela CNN.

		Real	
		Sim	Não
Predito	Sim	VP = 311	FP = 41
	Não	FN = 3	VN = 349

Figura 5. Matriz de confusão do melhor resultado

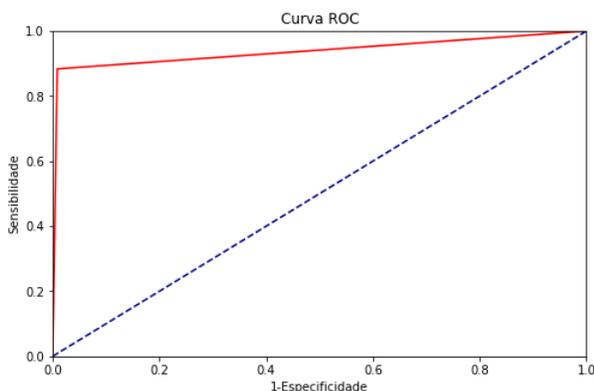


Figura 6. Curva ROC do melhor resultado

Na Fig. 6, a linha pontilhada azul representa uma resposta aleatória do modelo e a linha vermelha indica o modelo gerado pelo algoritmo proposto. É possível perceber através desse gráfico que o modelo consegue ter uma boa separabilidade devido dado o seu valor de AUC. Isto mostra que o algoritmo obtém boa separabilidade entre casos saudáveis e casos positivos de COVID-19.

A Tabela VII mostra o desempenho obtido em trabalhos recentes da literatura para classificar imagens de COVID-19 com diferentes CNN's e diferentes bancos de dados.

Tabela VII
COMPARAÇÃO COM O ESTADO-DA-ARTE

Artigos	Técnicas	Banco de dados	Performance
[10]	CNN (ChexNet) + Correção gamma	COVID-19 <i>dataset</i> (3616 COVID-19, 8851 Saudáveis e 6012 Não-COVID)	Acurácia: 96,29%, precisão: 96,28%, sensibilidade: 96,29%, F1-Score: 96,28% e especificidade: 97,27%
[7]	CNN (MobileNet v2)	<i>dataset</i> com 224 COVID-19, 714 Pneumonia (bacteriana e viral) e 504 Saudáveis	Sensibilidade: 98,66%, precisão: 96,78% e especificidade: 96,46%
[8]	CNN (ResNet-101)	154 COVID-19 e 5.828 imagens não rotuladas extraído do conjunto ChestX-ray14	AUC: 0.82, sensibilidade: 77,3%, acurácia: 71,9% e especificidade: 71,8%
[9]	CNN (Inception V3, Xception e ResNeXt)	<i>dataset</i> com 1583 imagens Saudáveis, 576 COVID-19 e 4273 de Pneumonia	Acurácia: 97,97% e sensibilidade: 92,00%
CNN proposta	CNN com 2 camadas de convolução	COVID-19 <i>dataset</i> (3520 COVID-19 e 3520 Saudáveis)	AUC: 90,15%, sensibilidade: 97,58%, precisão: 81,19%, acurácia: 89,60% e F1-Score: 88,38%

O trabalho de [25] é o único trabalho que utiliza um banco de dados com um grande número de imagens. Todavia, mesmo com uma rede convolucional profunda e com aprimoramento de imagem, a sensibilidade alcançada foi de 96,29%, sendo inferior ao resultado obtido no presente estudo. Ademais, o método estatístico usado para validar a eficiência foi o *k-fold cross validation* com 5 *folds*, o que mostra um resultado menos robusto comparado ao realizado neste trabalho.

Analisando o estudo de [7], pode-se observar que o *Recall* atingindo foi de 98,66%, um valor maior ao que foi obtido neste trabalho. É possível observar também que o método de treinamento utilizado é bastante robusto, visto que o método de *Transfer Learning* armazena o conhecimento adquirido em outras aplicações. Por outro lado, o banco de dados usado conta com uma pequena amostra de dados, contribuindo para o desempenho dos modelos que sofrem menos com relação ao custo computacional.

Os trabalhos de [9] e [8] também utilizam poucas imagens positivas para COVID-19, isto acarreta em um desbalanceamento de dados, e consequentemente, maior chance de o treinamento sofrer *overfitting* e priorizar a classe majoritária, conforme podemos perceber pelos valores obtidos para a métrica sensibilidade, que foram nitidamente inferiores aos valores obtidos pela nossa abordagem.

Apesar de não utilizar as imagens de casos positivos para pneumonia não oriunda de COVID-19, a CNN proposta consegue diferenciar bem radiografias de pacientes com COVID-19 e pacientes saudáveis. Além disso, a COVID-19 pode evo-

luir para pneumonia, provocando síndrome respiratória aguda grave, sendo assim, mesmo o modelo identificando alguma anomalia presente nas radiografias e classificando como caso positivo de COVID-19, permite que o paciente seja isolado para uma checagem mais minuciosa, tendo uma probabilidade baixa de infectar outra pessoa.

Por fim, é certo dizer que um diagnóstico não é feito somente a partir de uma radiografia do tórax, mas o modelo proposto pode influenciar na tomada de decisão, propiciando medidas de contenção da COVID-19, dado que a rede convolucional permite uma rápida resposta.

VII. CONCLUSÃO

É possível concluir que o uso de *deep learning* com CNNs, é capaz de ter efeitos significativos na extração de características e detecção automática de imagens de raios X relacionadas à pacientes com COVID-19.

Dessa forma, com base nos resultados obtidos usando uma Rede Neural Convolucional, foi possível classificar casos positivos de COVID-19, tendo uma sensibilidade de 97,58%. Este modelo pode ser útil como ferramenta para classificação, auxiliando na triagem rápida do paciente e prevenindo a propagação do vírus.

Os resultados obtidos neste estudo são confiáveis e generalizáveis, uma vez que foram treinados e validados com um grande conjunto de dados, sendo comparáveis a diversos estudos recentes e outros algoritmos de classificação do estado-da-arte.

Para trabalhos futuros, é relevante empregar técnicas de segmentação, onde a região de interesse poderia ser destacada, eliminando partes irrelevantes para o treinamento e assim, simplificando a imagem para uma melhor análise. Além disso é interessante melhorar a velocidade de treinamento da CNN utilizando paralelismo, com o objetivo de diminuir o tempo de execução do algoritmo, uma vez que desta forma o esforço computacional pode ser dividido para cada núcleo do processador ou unidade de processamento gráfico (em inglês, *Graphics Processing Unit - GPU*).

AGRADECIMENTOS

Os autores agradecem à UFVJM por todo o suporte prestado no desenvolvimento deste trabalho.

REFERÊNCIAS

- [1] F. He, Y. Deng, and W. Li, "Coronavirus disease 2019: What we know?" *Journal of medical virology*, vol. 92, no. 7, pp. 719–725, 2020.
- [2] P. M. de Medeiros Carvalho, M. M. Moreira, M. N. A. de Oliveira, J. M. M. Landim, and M. L. R. Neto, "The psychiatric impact of the novel coronavirus outbreak," *Psychiatry research*, vol. 286, p. 112902, 2020.
- [3] S. K. Brooks, R. K. Webster, L. E. Smith, L. Woodland, S. Wessely, N. Greenberg, and G. J. Rubin, "The psychological impact of quarantine and how to reduce it: rapid review of the evidence," *The lancet*, vol. 395, no. 10227, pp. 912–920, 2020.
- [4] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 826–834, 1983.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

- [6] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [7] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 2, pp. 635–640, 2020.
- [8] M. Z. Che Azemin, R. Hassan, M. I. Mohd Tamrin, and M. A. Md Ali, "Covid-19 deep learning prediction model using publicly available radiologist-adjudicated chest x-ray images as training data: preliminary findings," *International Journal of Biomedical Imaging*, vol. 2020, 2020.
- [9] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, "Deep learning based detection and analysis of COVID-19 on chest x-ray images," *Applied Intelligence*, vol. 51, no. 3, pp. 1690–1700, Oct. 2020. [Online]. Available: <https://doi.org/10.1007/s10489-020-01902-1>
- [10] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104319, 2021.
- [11] L. G. L. Fernandes *et al.*, "Previsão de séries de tempo: redes neurais artificiais e modelos estruturais," 1995.
- [12] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, 2015, vol. 25.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [14] Phung and Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, p. 4500, 10 2019.
- [15] L. G. Hafemann, L. S. Oliveira, P. R. Cavalin, and R. Sabourin, "Transfer learning between texture classification tasks using convolutional neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.
- [16] A. Eloyr and A. Nunes, "Support vector machines," 2020. [Online]. Available: <https://lamfo-unb.github.io/2020/07/04/SVM/>
- [17] J. Friedman, T. Hastie, R. Tibshirani *et al.*, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.
- [18] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.
- [19] G. Duarte, "Gradient boostings parte 1: Métodos de ensemble learning," 2020. [Online]. Available: <https://datarisk.io/gradient-boostings-parte-1-metodos-de-ensemble-learning>
- [20] H. Singh, "Understanding gradient boosting machines," 2018. [Online]. Available: <https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab>
- [21] A. Santra and C. J. Christy, "Genetic algorithm and confusion matrix for document clustering," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 1, p. 322, 2012.
- [22] D. Nogare, "Performance de machine learning - matriz de confusão," 2020. [Online]. Available: <http://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao>
- [23] S. Narkhede, "Understanding auc-roc curve," *Towards Data Science*, vol. 26, pp. 220–227, 2018.
- [24] F. e. a. Pedregosa, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [25] T. Rahman, "Covid-19 radiography database," 2020. [Online]. Available: <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>
- [26] BIMCV-COVID19, "Datasets related to covid19's pathology course," 2020. [Online]. Available: <https://bimcv.cipf.es/bimcv-projects/bimcv-covid19>
- [27] R. Mostafiz, M. S. Uddin, N.-A. Alam, M. M. Reza, and M. M. Rahman, "Covid-19 detection in chest x-ray through random forest classifier using a hybridization of deep cnn and dwt optimized features," *Journal of King Saud University-Computer and Information Sciences*, 2020.
- [28] P. K. Sethy, S. K. Behera, P. K. Ratha, and P. Biswas, "Detection of coronavirus disease (covid-19) based on deep features and support vector machine," 2020.
- [29] L. Lin, W. Yue, and Y. Mao, "Multi-class image classification based on fast stochastic gradient boosting," *Informatica*, vol. 38, no. 3, 2014.