

DEEP LEARNING EM RADIOGRAFIAS DO TÓRAX PARA PREDIÇÃO DE DOENÇAS PULMONARES

Bruno Silva Daniel¹

Fabio De Bittencourt²

Resumo: Com o aparecimento do novo coronavírus e outras doenças pulmonares através dos anos, cada vez mais o número de tipos diferentes das mesmas crescem, tendo elas sintomas muito semelhantes, dificultando no diagnóstico. Por possuírem sintomas parecidos geralmente é necessário fazer testes em laboratório para diagnosticar a doença, entretanto esses testes podem demorar dias. O objetivo deste trabalho é analisar imagens de radiografia pulmonares, a fim de encontrar padrões que são característicos de cada tipo de doença analisada, para realizar uma predição a fim de contribuir para outros estudos na área. Para isso, os dados foram extraídos de repositórios públicos, com parte de seus dados baseados em um projeto para coleta de dados de doenças pulmonares, aprovado pela *University of Montreal's Ethics Committee*. Com o auxílio da linguagem de programação python foi possível criar dois modelos de redes neurais que realizam a classificação binária entre duas classes com um Recall acima de 90% nas classes de doenças pulmonares, e uma classificação de três classes distintas com o valor de recall de 96% para pneumonia e 93% para Covid-19.

Palavras-chave: Deep Learning, Doenças pulmonares, Classificação, Redes Neurais Artificiais, radiografia.

1 INTRODUÇÃO

No contexto de uma pandemia criada pela COVID-19, sendo ela uma doença pulmonar provocada pelo coronavírus, um vírus que se transmite com muita facilidade devido à maneira de seu contágio, já que o meio de transmissão é por meio de gotículas infectadas que podem ser propagadas via tosse, espirro, fala ou contato físico. Esse vírus causa aos enfermos sintomas muito semelhantes aos da gripe comum, resfriados e outras doenças pulmonares, sintomas esses como febre, fadiga, dor de cabeça e garganta, tosse seca dentre outros. Devido a essa similaridade, muitas pessoas ficam em dúvida se contraíram a doença, fazendo com que os hospitais e postos de saúdes fiquem sobrecarregados por enfermos, aumentando a hipótese de contaminação naquele meio.

¹ Graduando em Engenharia da Computação. Ano 2022/2. E-mail: bruno_silva_daniel@hotmail.com

² Professor de Data Science na Satc. E-mail: fabio.bittencourt@satc.edu.br

Conforme a EuroFarma [1], empresa farmacêutica multinacional, existem três métodos para testar a contaminação por COVID-19, o primeiro e mais comum é utilizando um *Swab* [2], uma haste, sendo inserida via nasal para a coleta e análise do material nasofaríngeo, segundo por imunocromatografia ou teste rápido onde é retirada uma gota de sangue do dedo do paciente, e por último o teste sorológico, feito em laboratório, coletado através da punção do sangue da veia do paciente, sendo que todos eles têm hipóteses de entregar resultados falso-negativo, dificultado o combate à doença já que a entrega de falsos negativos neste caso é algo relativamente perigoso, pois pode deixar uma pessoa contaminada exposta para a população aumentando o número de casos devido a fácil propagação do vírus.

Com o intuito de ajudar a identificar possíveis pacientes contaminados por essas doenças e doenças semelhantes, esse trabalho visa desenvolver pelas tecnologias de *machine learning* e visão computacional um modelo capaz de realizar uma predição através de imagens de raio-x do tórax do paciente e definir se ele possa estar contaminado com algum tipo de doença pulmonar, visando encontrar o melhor resultado possível utilizando técnicas de redes neurais, de modo a contribuir nos estudos nesta área, importante comentar que o intuito não é diagnosticar o paciente e funcionar como um método auxiliar, repassando as informações para um profissional com capacitação médica para que o mesmo, possa fazer o diagnóstico completo do enfermo.

2 REVISÃO BIBLIOGRÁFICA

Para o desenvolvimento do projeto, serão usados processos de análise e manipulação de dados, todos com base em conceitos nas áreas de inteligência Artificial e suas sub camadas conhecidas como “*Machine Learning*” e “*Deep Learning*”, com um foco maior na parte de “*Deep Learning*”, pois é nessa área onde o modelo de predição será criado. Os tópicos abaixo explicaram um pouco sobre o histórico e os conceitos utilizados dessas áreas.

A maioria dos conceitos abordados tem suas informações teóricas extraídas do Livro *Deep Learning (Adaptive Computation and Machine Learning*

series)[8], um livro escrito por Ian Goodfellow, Yoshua Bengio e Aaron Courville, o objetivo da obra conforme os autores é ajudar estudantes e pesquisadores a entrarem nas áreas de “*Machine Learning*” e “*Deep Learning*”, a versão utilizada foi a online que é uma versão completa e disponibilizada de graça.

Outras postagens e artigos, que também foram usados que contribuiriam para o projeto, estão devidamente referenciados, a maioria deles foram encontrados através da ferramenta de pesquisa do Google e sua ferramenta de pesquisa por imagens.

2.1 DOENÇAS RESPIRATÓRIAS

Doenças respiratórias são doenças que afetam o sistema respiratório, geralmente atacando sua principal parte o pulmão, alguns exemplos delas são bronquites crônicas, asma, pneumonia bacteriana e viral, rinite alérgica, sinusite, doença pulmonar obstrutiva crônica (DPOC) e as mais recentes doenças causadas a partir do novo coronavírus.

Segundo os dados divulgados pelo ministério da saúde brasileiro [22], foram mais de 34 milhões de casos de covid-19 confirmados desde o início da pandemia, tendo uma letalidade beirando a 2% com mais de 687 mil óbitos. Entretanto, a covid-19 não é a única doença pulmonar que se destaca no quesito óbitos, conforme as pesquisas divulgadas pela World Health Organization [21] as doenças respiratórias, crônicas foram a terceira maior causa de mortes no mundo (7,1%) no ano de 2017, ficando atrás de somente doenças cardiovasculares e neoplasias.

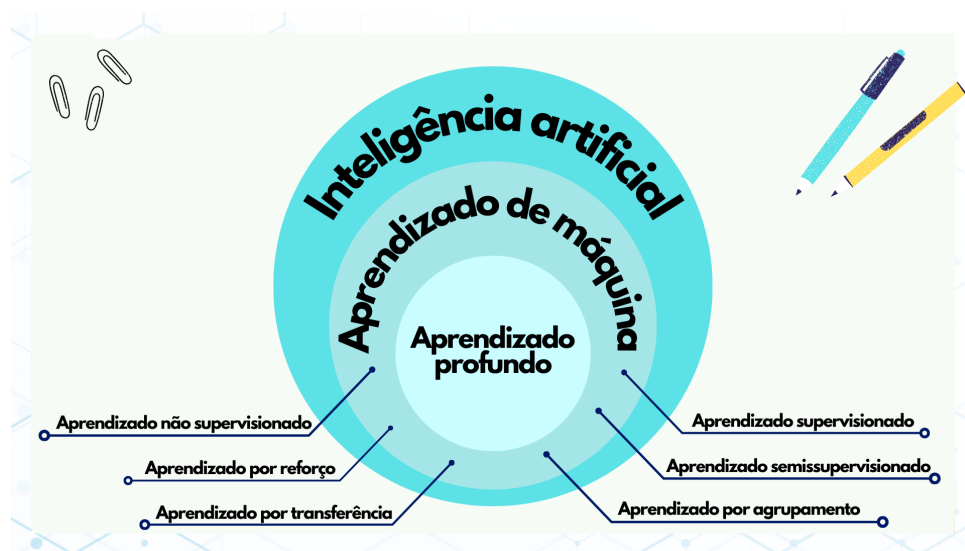
2.1 INTELIGENCIA ARTIFICIAL

Desde o início o conceito de IA pode-se ser atribuído a parte da Ciência da Computação que faz com que os sistemas ou máquinas demonstrem atributos, ou comportamentos que podem ser considerados parte de uma inteligência humana, tais como: aprender, adquirir conhecimentos, entender linguagens e resolver problemas. Sendo assim, o principal objetivo da IA, é a criação de modelos capazes

de executar essas funções e construir sistemas computacionais baseados nesses modelos.

O objetivo da IA pode ser considerado simultaneamente teórico, (a criação de teorias e modelos para a capacidade cognitiva), e prático, a implementação de sistemas computacionais baseados nestes modelos. Isso demonstra que a IA tem se destacado na busca por compreender a inteligência e por englobar diversos campos do conhecimento com o objetivo prático de simular a inteligência (LUGER, 2004).

Figura 1 – Subconjuntos da Inteligência Artificial.



Fonte: Visibilia (2020).

Inteligência artificial é a construção de um sistema que exhibe atributos e comportamentos inteligentes, já o termo “*Machine learning*” se refere sistemas projetados e criados para receber dados e informações e através desses dados descobrir tendências e padrões dentro deles, conseguindo categorizar e avaliar os dados recebidos por análises e treinamentos realizados. Um exemplo é um ‘*software*’ que analisa a entrada de uma imagem em uma plataforma de mídia social, verificando o conteúdo dos pixels das imagens traçando padrões conhecidos usando alguns métodos de “*machine learning*”, conseguindo uma resposta referente a identificação da imagem, tudo isso é capaz pelo treino realizado por base de dados que servem para “calibrar” o *software*. Tendo duas categorias de aprendizado, o supervisionado e o não supervisionado. No supervisionado é onde dados são

alimentados para a máquina com classificações já definidas, fazendo com que o trabalho dela seja identificar parâmetros semelhantes de cada tipo de categoria, a partir disso, ela consegue identificar novos dados com os parâmetros encontrados posteriormente. Já o não supervisionado identifica semelhanças nos dados e reage conforme a presença ou ausência de tais semelhanças em cada novo dado.

Já o “*Deep learning*” que é um subconjunto na IA que leva o conceito de processamento de alto nível de dados usando um grafo profundo com várias camadas de processamento, compostas de várias transformações lineares e não lineares, ele também significa aprender continuamente, ou seja, os sistemas criados através dele visam aprender com a realidade e ajustar o modelo conforme ele for aprendendo com novas informações e dados que ele recebe e analisa. As aplicações de “*Deep learning*” são geralmente utilizadas com redes neurais. Redes neurais são uma tentativa sintética de imitar o funcionamento e estrutura do cérebro humano, até mesmo nas conexões entre os nós cerebrais são estabelecidas.

Como mencionado anteriormente a inteligência artificial possui subconjuntos, sendo o “*Machine Learning*” que é qualquer método de máquina que aprenda com as experiências, já dentro deste mesmo subconjunto existe em uma área chamada de “*Deep learning*” que pode possuir várias redes neurais, a principal diferença dentre elas se encontra em como os dados de entrada são tratados onde no “*Machine Learning*” é necessário fazer um pré-processamento para encontrar os parâmetros que o modelo irá utilizar já no “*Deep Learning*” o modelo encontra os parâmetros, essas duas subcamadas que possibilitaram a criação desse projeto, utilizando dos métodos que elas fornecem é possível extrair informações e características dos dados de entrada utilizados e criar um modelo de classificação que retorne previsões. Como esse trabalho utiliza como dados de entrada imagem, será necessário entrar nos conceitos de visão computacional.

2.1.1 Como funciona a “visão” das máquinas?

Para o ser humano, quando observando um espaço ou cenário é possível nomear e identificar cores, formas e objetos utilizando de sua visão, conseguindo identificar padrões de uma maneira muito rápida, quase de uma maneira involuntária, o primeiro passo é visualizar o objeto, no segundo momento o cérebro

contornos. Passando por uma série de camadas, a rede retorna qual é a classe que mais se adéqua a imagem analisada, e em alguns casos gerando divisão de probabilidades como o exemplo de raças de animais (SUZANA, 2018). Esse conceito é importante, pois é graças a ele que a maquina pode diferenciar um paciente contaminado com uma doença pulmonar ou não.

2.3 Redes de convolução

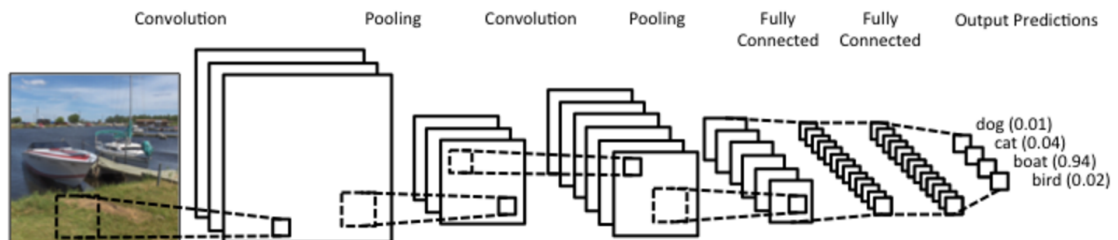
Neste trabalho foram utilizados aprendizados supervisionados, ou seja, as imagens alimentadas para a máquina já vem classificadas, então para o modelo entender o que cada classe tem em comum é necessário extrair dados e características, isso pode ser feito através de uma rede neural convolucional também conhecida como CNN (Convolutional Neural Network), que faz esse trabalho por filtros, sendo ela a mais indicada quando o assunto é classificação de imagens.

2.3.1 Redes neurais de convolução.

Uma imagem passa por uma série de camadas na rede e retorna uma saída, sendo ela uma possível divisão de probabilidades. O arranjo e estruturação dessas camadas e componentes tem um papel fundamental na construção de diferentes arquiteturas de rede, cada uma com suas vantagens e desvantagens.

Geralmente a arquitetura mais simples de uma CNN é montada de dois tipos de camadas, as de convolução e de *pooling*, e em seguida é utilizado uma ou mais camadas conectadas. Onde a camada de convolução é a responsável por extrair, bem como mapear o conteúdo da imagem, transformando-o em dado, já a camada de *pooling* é responsável, em reduzir as dimensões dos dados aplicando camadas de agrupamento. No caso das redes de convolução para segmentação de imagem, as camadas totalmente conectadas serão substituídas por uma camada final de convolução, sendo, portanto, chamadas de redes totalmente convolucionais.

Figura 3 - As camadas de uma CNN de classificação.



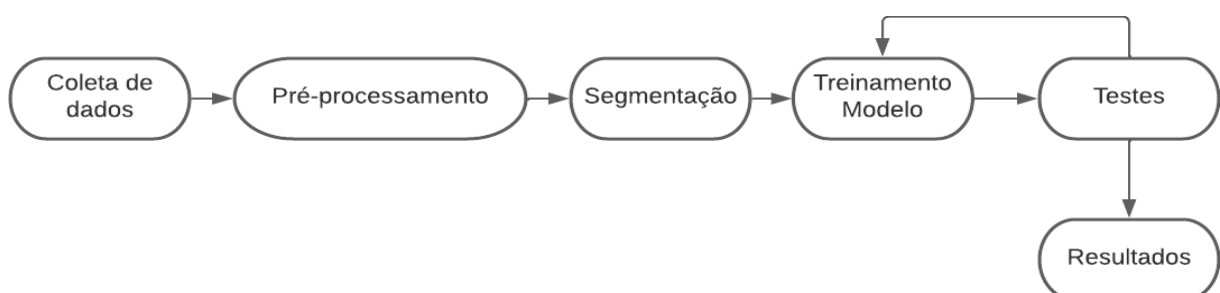
Fonte: Denny Britz, 2018

O reconhecimento de imagem é um clássico problema de classificação, e as Redes Neurais Convolucionais possuem um histórico de alta acurácia para esse problema. A primeira aplicação com sucesso de uma CNN foi desenvolvida por Yann LeCun em 1998 [20].

3. PROCEDIMENTOS EXPERIMENTAIS

Neste capítulo são apresentadas as etapas e procedimentos a serem seguidos neste trabalho, os materiais utilizados bem como técnicas de *deep learning* a serem aplicadas para tratativa, treinamento e teste do modelo desenvolvido conforme os dados utilizados. Os principais métodos utilizados na criação do modelo giram em torno da utilização de CNN, recebendo as imagens como entrada de dados passando elas entre as camadas e filtros para extração e manipulação de características e parâmetros.

Figura 4. Fluxograma Procedimento Experimental.



Fonte: Do autor (2021).

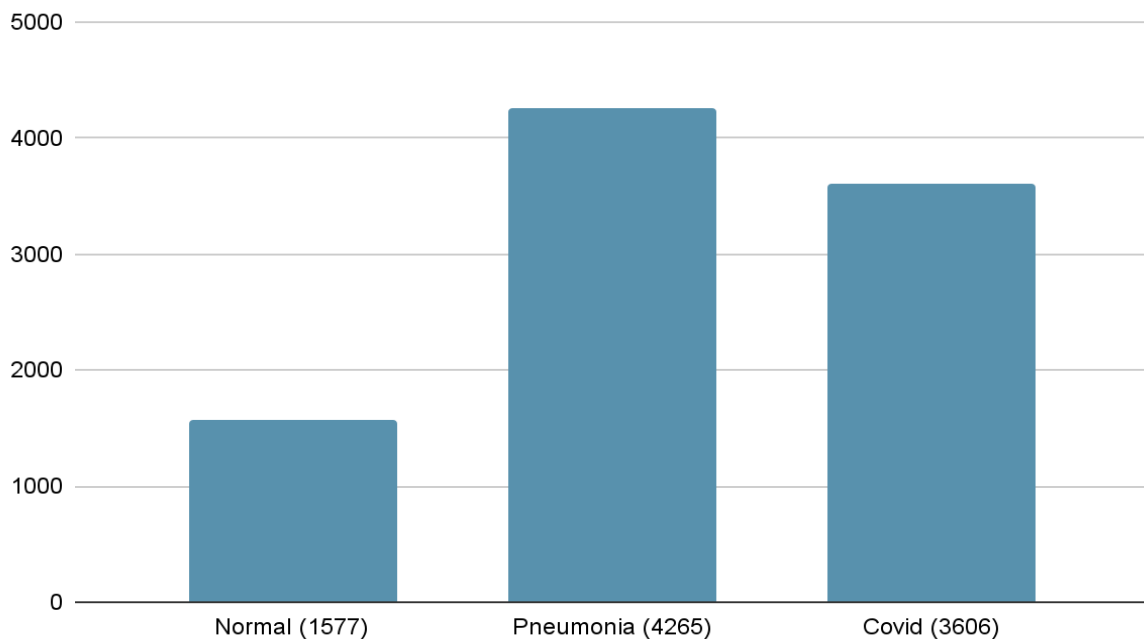
O procedimento se inicia com a coleta de dados, passando para o pré-processamento dos mesmo onde é feito uma tratativa e manipulação para melhor aproveitamento para o modelo, em seguida é realizada a segmentação com os dados de treino, apos isso os dados estão prontos para serem usados nos modelos, levando a criação e teste dos modelos e avaliação de resultados.

3.1 COLETA DE DADOS E PRÉ-PROCESSAMENTO

Para realização o projeto foi necessário fazer a coleta e processamento de imagens de radiografias do tórax humano, essas imagens foram coletadas e extraídas em grande maioria do site *Kaggle*[8], um site que funciona como repositório de base de dados para uso licenciado e gratuito, contendo diversas categorias de dados nas suas bibliotecas, as bases de dados escolhidas para esse trabalho foram manualmente mescladas juntando um total de três classes, pacientes com radiografias normais ou saudáveis, pacientes com covid e pacientes com pneumonia.

Além da coleta e junção dessas bases foi necessário um trabalho manual para remover imagens não ideais, seriam elas imagens onde existem anotações medicas, setas ou indicativos apontando para determinado lugar da imagem dentro outros fatores que poderiam interferir na hora de treinar o modelo, pois o mesmo poderia entender esses fatores como indicativos para fazer uma predição erronia. Também foi feito uma pre-processamento nas imagens, especificamente no tipo de entrada que as elas forneciam, os valores de entrada tem três parâmetros, os dois primeiros valores referentes ao seu tamanho e o terceiro referente a quantidade de canais de cores. Nesta etapa as imagens foram redimensionadas para os valores de 150x150, esse processo foi realizado afim, pois diminui muito o tempo de treino e processamento do modelo e aumenta sua atuação[23], além disso, as imagens foram modificadas para apenas um canal de cor, ou seja, preto e branco, já que todas elas eram radiografias clássicas sem contraste de cor ou máscaras de relevo não havia a necessidade de utilizar os três canais de cores.

Gráfico 1. Gráfico quantidade de imagens por categoria.



Fonte: Do autor (2021).

Bases de radiografias comprovadas disponíveis em domínios públicos são relativamente escassas, disponibilizando poucas opções, por isso a importância da criação manual dessa base, devido ao número de classes disponíveis na mesma. Importante destacar que outras classes foram encontradas além das citadas acima, entretanto, não representavam uma quantidade de dados relevante o suficiente para o desenvolvimento do projeto.

3.2 FERRAMENTAS UTILIZADAS

Para a criação dos modelos foi usado o *Colaboratory*[9] do *Google*, sendo ele uma aplicação Web que permite o usuário utilizar de máquinas virtuais para compilar e rodar códigos da linguagem de programação Python sem a necessidade de consumir recursos da máquina pessoal do usuário, tudo isso na praticidade de um navegador web, que pode ser acessado via desktop ou mobile, ela foi escolhida pela praticidade. Após criar um projeto e importar a base de dados escolhida, é possível utilizar uma integração do *Colaboratory* do *Google* com o *Google Drive* permitindo usar arquivos que estejam no mesmo, por praticidade a base de dados

foi colocado no mesmo, entretanto não é um procedimento necessário. É recomendado habilitar a opção de “*aceleração de hardware*” e selecionar a opção “GPU” na aba de configurações, essa opção permite a alocação de uma GPU caso disponível melhorando o processamento da aplicação. Também, foi utilizado o jupyter notebook com a versão 3.8 do python instalado em uma máquina que possuía uma placa de video da NVIDIA modelo GTX 1050 TI, essa ferramenta foi utilizada para realizar os testes dos modelos mais pesados e lentos.

Como comentado no parágrafo acima, a linguagem de programação escolhida foi Python, para o desenvolvimento do projeto é necessário importar algumas bibliotecas que trabalham com os conceitos abordados, sendo elas:

- Pandas [10]: oferece estruturas e operações para manipular tabelas numéricas e séries temporais;
- TensorFlow [11]: biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas. É um sistema para criação e treinamento de redes neurais para detectar e decifrar padrões e correlações, análogo à forma como humanos aprendem e raciocinam;
- Numpy [12]: suporta o processamento de grandes, multi-dimensionais arranjos e matrizes, com uma grande coleção de funções matemáticas de alto nível para operar sobre estas matrizes;
- Matplotlib [13]: é uma biblioteca de software para criação de gráficos e visualizações de dados em geral;
- Scikit-learn [14]: é uma biblioteca de aprendizado de máquina de código aberto;
- Keras[15]: biblioteca de rede neural de código aberto. Ele consegue rodar em cima de TensorFlow, Microsoft Cognitive Toolkit, R, Theano, ou PlaidML. Projetado para permitir experimentação rápida com redes neurais profundas, ele se concentra em ser fácil de usar, modular e extensível.
- OpenCV[24]: redimensionamento e pré-processamento de imagens.

3.3 SEGMENTAÇÃO

De modo a tornar o modelo mais robusto através do aumento de imagens na base de dados, foi utilizada uma prática de geração de imagens disponibilizada através de uma função do Keras chamada **ImageDataGenerator()**, essa função permite que seja aumentado o número de imagens em tempo real enquanto o modelo ainda está treinando, ela permite fazer vários tipos de alterações sendo possível aplicar transformações aleatórias em cada imagem de treinamento conforme ela seja passada para o modelo, dentre as alterações as mais utilizadas são alteração de brilho, virar imagem para determinadas posições ou posições aleatórias e realizar uma ampliação parcial ou completa. Optando por usar esta função ela não apenas torna o modelo em questão mais robusto, além disso, acaba por ter uma economia no uso de memória. As alterações utilizadas para esse modelo foram rotações, zoom e desorientação, resultante na imagem ter uma aparência de estar “puxada”.

3.4 TREINANDO OS MODELOS

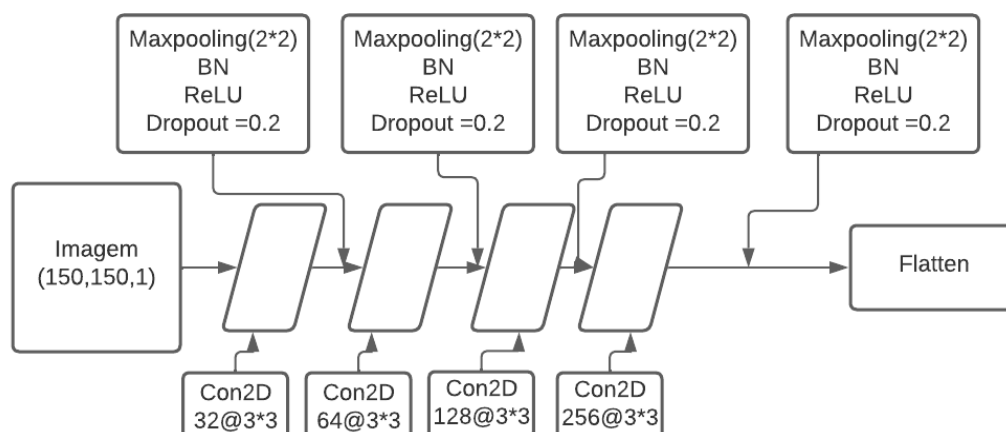
Após as imagens serem segmentadas seguiu-se para as criações de modelos de *Deep Learning*. Como o projeto trabalha com imagens foi escolhido utilizar redes neurais convolucionais (CNN), com esse método é possível criar camadas de filtros que extraem parâmetros das imagens passadas a ele, atribuindo determinados parâmetros a cada tipo de categoria informada. Além de utilização de diferentes camadas, as mesmas podem receber diversos tipos de fatores que influenciam na quantidade de parâmetros resultantes de cada modelo.

Com a base de dados separada foi realizado as criações de diferentes modelos com filtros e diversos parâmetros em cada um deles, com o intuito de identificar qual teria mais sucesso em classificar as imagens de entrada por cada categoria, desses modelos foram selecionados dois que obtiveram melhores desempenhos quando comparado aos outros.

3.4.1 PRIMEIRO MODELO

O primeiro modelo conta com uma arquitetura mais convencional utilizando de filtros menores nas primeiras camadas de convolução e aumentando gradativamente o valor dos mesmo nas camadas seguintes, nas subcamadas que são as camadas que ficam entre as camadas de convolução foram utilizados camadas de *Maxpooling*, o proposito dessa camada é que, uma vez que um recurso tenha sido encontrado pela camada de convolução, a localização exata não é tão importante quanto sua localização aproximada em relação resto dos elementos, fazer isso traz o benefício de haver muito menos recursos agrupados, portanto, isso ajuda a reduzir o número de parâmetros necessários nas camadas posteriores, também foram utilizados camadas de *dropout*, essas camadas são uma forma de garantir que o modelo seja robusto para a perda de qualquer evidência individual, camadas de normalização também foram adicionadas que tendem a reduzir os pesos e, assim, tornar a rede mais robusta para perder qualquer conexão individual na rede, essas duas camadas ajudam a evitar que o modelo fique sobre ajustado, que é quando o mesmo tem um bom desempenho no treino, mas performa mal nos teste, foram testadas ativações de ReLU, softmax e sigmoid onde as ativações do tipo ReLU com exceção da última camada tiveram melhores resultados.

Figura 5 - Arquitetura modelo 1.

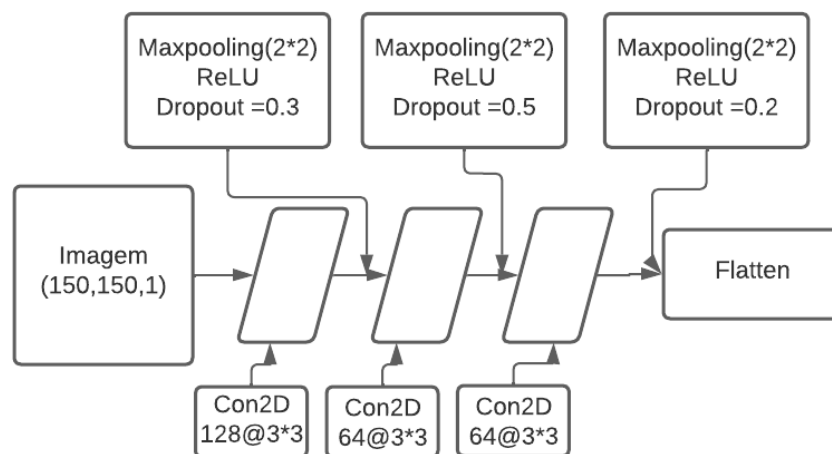


Fonte: Do autor (2021).

3.4.2 SEGUNDO MODELO

Este modelo utiliza de uma prática não tão comum, ele começa com um valor de filtro relativamente alto nas camadas de convolução e vai decrescendo com o passar das camadas subsequentes, isso faz com que sejam gerados mais parâmetros no início da rede neural fazendo com que o treinamento fique mais lento, por isso em questão de tamanho o modelo ficou menor do que o anterior para poupar processamento, foi escolhido essa prática, pois os dados de Covid tiveram melhores resultados quando iniciados com um número de filtros maior, este modelo também utiliza de camadas de *dropout* s e camadas de *Maxpooling* para diminuir possível sobre ajustamento.

Figura 6 - Arquitetura modelo 2.



Fonte: Do autor (2021).

3.5 TESTES MODELOS

A fim de determinar qual o modelo criado foi o mais efetivo na tarefa de classificação de imagens, foram utilizados de métodos que testam a acurácia e *loss*, esses valores são muito importantes para determinar se um modelo não está sub-ajustado, sendo quando ele não conseguiu relacionar os parâmetros encontrados com as categorias apresentadas, ou sobre ajustado que é quando o modelo ao invés de relacionar os parâmetros adquiridos com as categorias apresentadas, ele somente decora as imagens passadas a ele, e quando

apresentado uma nova imagem não vista antes ele tende a falhar na hora de identificar a mesma.

A acurácia é um valor dado em porcentagem, ela representa o desempenho geral do modelo, dentre todas as classificações que o modelo classificou quantas foram corretas, já o *loss* quanto menor o valor melhor o modelo, exceto se o modelo esteja sobre ajustado, esse valor é calculado nas etapas de treino e validação, ela é a interpretação de quanto bem o modelo está atuando nesses dois processos, diferente da acurácia, o *loss* não é um valor de porcentagem, ele é um sumário dos erros feitos em cada exemplo de treino ou de validação, também foram usados os valores de precisão que é a relação dentro as predições verdadeiras positivas e soma das predições verdadeiras positiva e falsas positivas, recall que demonstra o oposto que são as predições verdadeiras positivas e a soma das predições verdadeira positiva e falsa negativa e também o F1-score, ele demonstra uma média harmônica entre precisão e recall funcionando como meio-termo entre os dois.

Também foi utilizado de matrizes de confusão para visualização dos resultados, ela relaciona os valores preditos pelo modelo e seus verdadeiros resultados e cria essa matriz que neste caso contém cores quentes e frias, cada linha representa a instância de uma classe predita, enquanto cada coluna representa a instância da classe atual. Ela cria uma forma visual de relacionar os dados das predições feitas pelo modelo com a relação entre os valores mencionados acima.

4 RESULTADOS E DISCUSSÕES

Após o treino dos modelos foram coletados resultados entre classificações binárias dentre as classes de Pneumonia e Normal, e às classes de Covid e Normal. Além disso, foi coletado informações das predições de multi-classificação entre as três classes encontradas na base de dados criada.

Na classificação das classes Pneumonia e normal, o modelo com melhor atuação foi o primeiro modelo, o mesmo apresentou um *loss* de 0.33 e uma acurácia de 0.91.

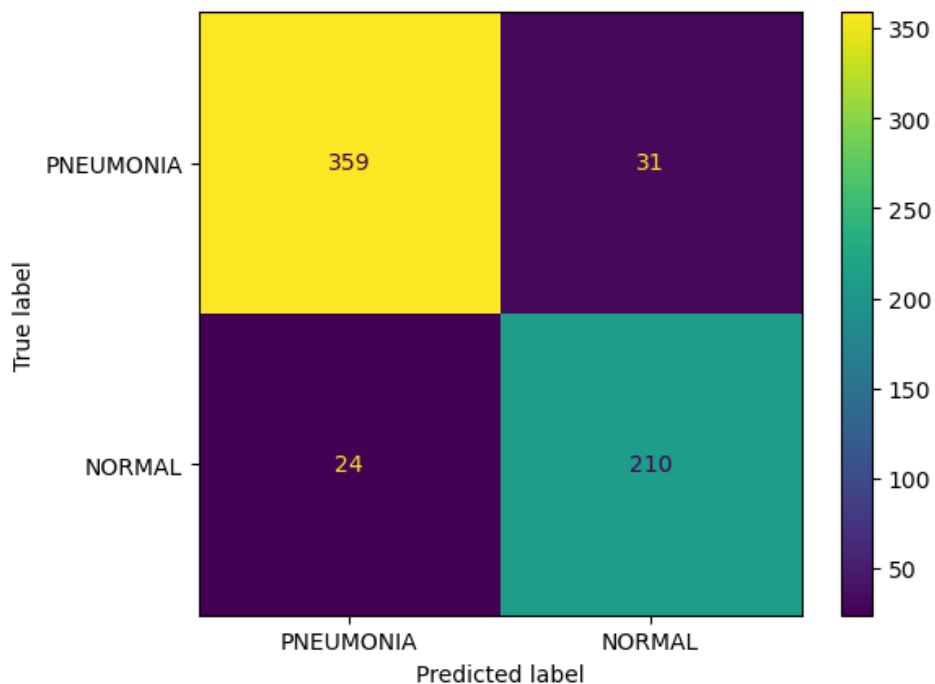
Tabela 1. Resultados Pneumonia e normal.

	Precisão	Recall	F1	Dados
Pneumonia	0.94	0.92	0.93	390
Normal	0.87	0.90	0.88	234
Acurácia			0.91	624

Fonte: Do autor (2021).

Como visto acima, a desempenho na classe de Pneumonia foi um pouco melhor do que na classe normal tanto em precisão, recall e acurácia, um dos motivos deve-se pela quantidade de dados treinados da classe normal ser menor que as demais classes. Entretanto, ambas as classes apresentaram resultados satisfatórios para a predição das classes informadas nos três quesitos de precisão, recall e F1-Score.

Figura 7. Matriz de confusão Pneumonia X Normal.



Fonte: Do autor (2021).

Acima é possível ver uma matriz de confusão desta comparação, linha em diagonal que cruza da esquerda para direita indica todos os resultados que o modelo acertou em questão de números, ou seja, os verdadeiros positivos, sendo bem semelhante com os resultados encontrados.

Já na classificação das classes Covid e normal, o modelo com melhor desempenho foi o segundo modelo, que utiliza dos filtros maiores no início e vai decrescendo com o tempo, o mesmo apresentou um *loss* de 0.34 e uma acurácia também de 0.91.

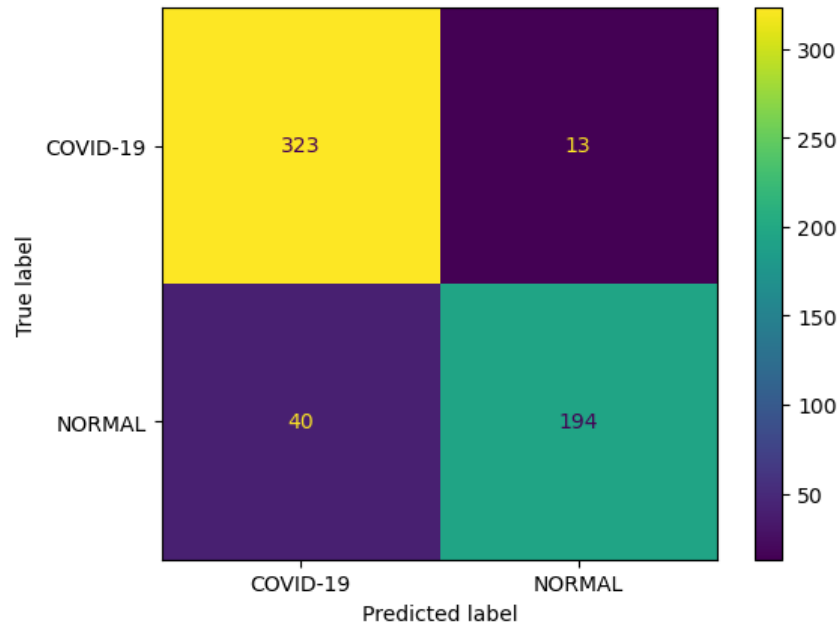
Tabela 2. Resultados COVID e normal.

	Precisão	Recall	F1	Dados
COVID	0.89	0.96	0.92	336
Normal	0.94	0.83	0.88	234
Acurácia			0.91	570

Fonte: Do autor (2021).

Como visto acima, a desempenho entre essas classes foi bem semelhante comparado com a primeira classificação, embora utilize de um modelo diferente. Neste caso o COVID teve uma precisão de 0.89 um pouco menor do que a outra classe, sendo ela de 0.94, entretanto a classe de doença pulmonar teve um Recall de 0.96, um valor muito importante, pois o pior caso possível desse projeto é quando ocorre um falso negativo nas classes de doenças, já o valor do Recall da classe normal foi um pouco menor, entretanto ainda assim satisfatório, ambas tiveram valores de F1-Score positivos com 0.92 para Covid e 0.88 para Normal.

Figura 8. Matriz de confusão Covid X Normal.



Fonte: Do autor (2021).

A matriz de confusão acima mostra as predições feitas pelo modelo, assim como na primeira comparação a mesma mostra valores satisfatórios dentre os resultados preditos pelo modelo e os valores verdadeiros.

A última comparação feita é entre as classes Pneumonia, Covid e normal, o modelo que demonstrou uma melhor desempenho também foi o segundo modelo com uma acurácia de 82%, pode se concluir que o modelo 2 apresenta melhores resultados quando utilizado com a classe de Covid comparado ao modelo 1.

Tabela 3. Resultados Pneumonia, COVID e normal.

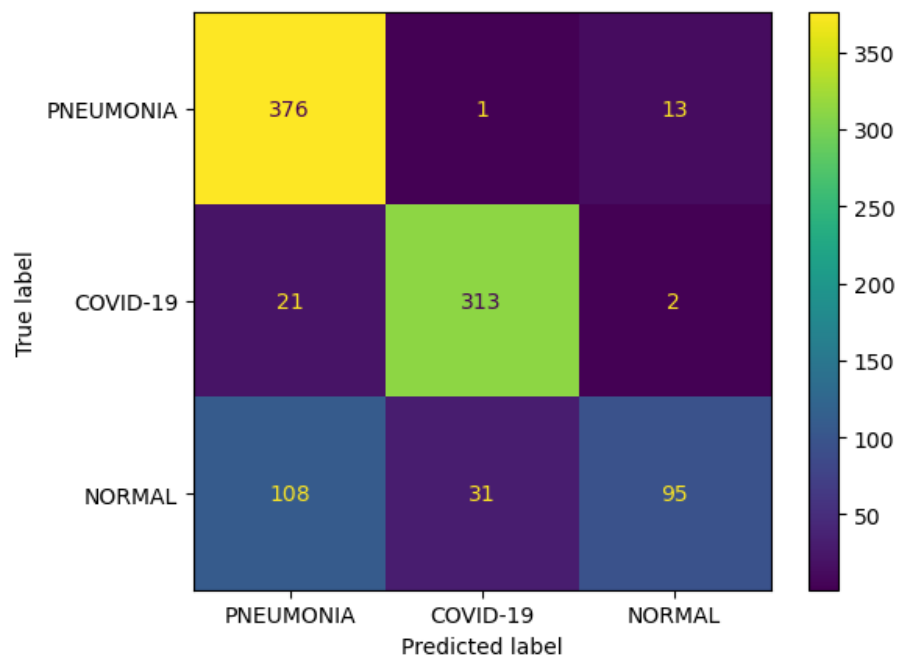
	Precisão	Recall	F1	Dados
Pneumonia	0.74	0.96	0.84	390
Covid-19	0.91	0.93	0.92	336
Normal	0.86	0.41	0.55	234

Acurácia			0.82	960
----------	--	--	------	-----

Fonte: Do autor (2021)

A classe que teve melhores resultados foi a de covid tento, uma precisão de 0.91 e um Recall de 0.93 resultando em uma média harmônica de 0.92, já a classe de pneumonia teve um valor de precisão abaixo das demais embora não ideal, ainda assim é um resultado positivo já que um falso positivo não é tão prejudicial do que um falso negativo para essa classe, em compensação ela teve um Recall alto de 0.96 que é o valor de mais importância como comentado acima resultando em uma média harmônica de 0.84, já a classe normal teve um valor de precisão de 0.86 sendo considerado um bom resultado, mas um valor de recall baixo de 0.41, embora esse valor seja baixo ele representa mais falsos positivos dessa classe o que não é o pior dos casos para esse modelo, devido ao valor baixo de recall a média harmônica desta classe ficou em 0.55.

Figura 9. Matriz de confusão Pneumonia X Covid X Normal.



Fonte: Do autor (2021)

Acima é possível ver a matriz de confusão dos resultados encontrados nesta última comparação, observando é possível ver uma relação do baixo valor de recall encontrado na classe normal, e também o valor um pouco menor da precisão de pneumonia.

5 CONCLUSÃO

O objetivo do projeto foi a utilização de *machine learning* e *deep learning* para auxiliar na classificação de doenças respiratórias pulmonares através de radiografias, criando modelos que analisam os parâmetros de determinado raio-x e classifica o mesmo de acordo com categorias.

Todo o desenvolvimento do projeto foi realizado por meio da linguagem de programação Python, além de utilizar de várias bibliotecas públicas e suas funções como opencv para filtragens e transformações da base de dados, *scikit-learn*, *keras* e Tensorflow para criação e testes com os modelos de *deep learning* utilizando da arquitetura CNN.

A conclusão deu-se pela criação de dois modelos de classificação de radiografia e seus resultados de 91% de acurácia em comparações binárias entre as classes pneumonia e normal, e as classes covid e normal, e 82% de acurácia em uma comparação de multi classificatória entre todas as classes citadas, a classe que teve pior atuação foi a normal, isso pode estar relacionada a quantidade de dados utilizados para esta classe ser menor quando comparada às demais classes.

Embora nem todos os resultados foram completamente satisfatórios, o projeto teve um resultado positivo de forma geral, em questão de melhorias o projeto, teve como um dos principais problemas como comentado acima a disparidade de classes no conjunto de dados trabalhado, o ideal para futuros trabalhos seria ter uma igualdade maior de classes. Outra melhoria seria na própria rede neural que pode ser melhorada com mais teste e refinamento alterando os parâmetros utilizados e os filtros.

Para trabalhos futuros é interessante usar novas classes conforme a disponibilidade dos dados, e também utilizar de máscaras na parte de pré-processamento das imagens, isolando os pulmões e retirando as demais informações das imagens.

REFERÊNCIAS

- [1] EUROFARMA. **Qual o exame mais confiável para COVID-19?**, Dasa. 2020. Disponível em: <<https://eurofarma.com.br/artigos/saiba-as-diferencas-entre-os-tipos-de-teste-de-covid-19/>>. Acesso em: 5 ago. 2022.
- [2] BBV. **SWAB para coleta de Vírus**, Bio Boa Vista. 2020. Disponível em: <https://bioboavista.com.br/swab-para-coleta-de-virus/?gclid=Cj0KCCQjw0oyYBhDGARIsAMZEUmuTDQtTKQN518tf1z7QNZiD76Ku3dOa_RJngkyWzZD5NTB21i5Hf0waAsX9EALw_wcB/>. Acesso em: 5 ago. 2022
- [3] EQUIPE DASA. **Diferenças entre os teste de Covid-19**, Dasa. 2020. Disponível em: <<https://dasa.com.br/blog/coronavirus/qual-o-exame-mais-confiavel-para-covid-19/>>. Acesso em: 5 set. 2021.
- [4] SRINGER. **Comparing COGNEX VisionPro Deep Learning**. 2021. Disponível em: <<https://link.springer.com/content/pdf/10.1007/s42979-021-00496-w.pdf>>. Acesso em: 23 set. 2021.
- [5] KAGGLE, **Your Machine Learning and Data Science Community 2022**. Disponível em: <<https://www.kaggle.com/>> Acesso em: 23 set. 2021.
- [6] PRAVEEN, **CoronaHack Chest X-ray Dataset 2021**. Disponível em: <<https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset>> Acesso em: 23 set. 2021.
- [7] EQUIPE VISIBILIA, **Aprendizado de Máquinas. Visibilia 2021**. Disponível em: <<https://visibilia.net.br/aprendizado-maquinas/>>. Acesso em: 23 set. 2021.
- [8] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning. [S.I.]: MIT Press**, 2016. Disponível em <<http://www.deeplearningbook.org>>. Acesso em: 03 out. 2021.
- [9] GOOGLE, **Colab Google**. Disponível em: <<https://colab.research.google.com>> Acesso em: 03 out. 2021.
- [10] **Pandas** . Disponível em: <<https://pandas.pydata.org/>> Acesso em: 18 abril. 2022.
- [11] **Tensorflow**. Disponível em: <https://www.tensorflow.org/resources/learn-ml?gclid=Cj0KCCQjwyYKUBhDJARIsAMj9IkE40ycI7YENGcJkGyfuVABCefTk5Hj8mDdWeArs1dv-BwRE4L_0IRkaAhQZEALw_wcB> Acesso em: 18 abril. 2022.

- [12] **Numpy**. Disponível em: <<https://numpy.org/>>
Acesso em: 18 abril. 2022.
- [13] **Matplotlib**. Disponível em: <<https://matplotlib.org/>>
Acesso em: 18 abril. 2022.
- [14] **Scikit-Learn**. Disponível em: <<https://scikit-learn.org/>>
Acesso em: 18 abril. 2022.
- [15] **Keras**. Disponível em: <https://keras.io/>>
Acesso em: 18 abril. 2022.
- [16] ANALYTICSVIDHYA. **Image Augmentation on the fly using Keras ImageDataGenerator!**. Disponível em:
<[https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/#:~:text=Keras%20ImageDataGenerator%20is%20a%20gem,up%20on%20the%20overhead%20memory!](https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/#:~:text=Keras%20ImageDataGenerator%20is%20a%20gem,up%20on%20the%20overhead%20memory!>)>
Acesso em: 17 abril. 2022. Utilizado na Página 17 e 18.
- [17] REALPYTHON, **Split Your Dataset With scikit-learn's train_test_split()**. Disponível em: <<https://realpython.com/train-test-split-python-data/>>
Acesso em: 18 abril. 2022. Utilizado na Página 16 e 17.
- [18] DENNY. **DENNY'S BLOG**. Disponível em:
<<https://dennybritz.com/posts/wildml/understanding-convolutional-neural-networks-for-nlp/>>
Acesso em: 20 agosto. 2022.
- [19] PAPERWITHCODE. **Paperwithcode**, Disponível em:
<<https://paperswithcode.com/>>. Acesso em: 20 agosto. 2022.
- [20] HISTORYOFDATASCIENCE. **Yann LeCun: An Early AI Prophet.**, Disponível em: <<https://www.historyofdatascience.com/yann-lecun/>>. Acesso em: 20 setembro. 2022.
- [21] WHO. **World Health Organization**, Disponível em:
<<https://www.who.int/data/gho/publications/world-health-statistics/>>. Acesso em: 20 setembro. 2022.
- [22] Ministério da Saúde. **Coronavírus Brasil**. Disponível em:
<<https://covid.saude.gov.br/>>. Acesso em: 20 setembro. 2022.
- [23] SERGIO SAPONARA. **Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance**. Disponível em:
<[https://link.springer.com/chapter/10.1007/978-3-030-95498-7_2#:~:text=Resizing%20images%20is%20a%20critical,training%20time%20for%20the%20architecture./](https://link.springer.com/chapter/10.1007/978-3-030-95498-7_2#:~:text=Resizing%20images%20is%20a%20critical,training%20time%20for%20the%20architecture./>)>. Acesso em: 02 novembro. 2022.



[24] OPEN CV. **OpenCV modules**. Disponível em: <<https://docs.opencv.org/4.x/>>. Acesso em: 20 setembro. 2022.