



# CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR



Maram Mahmoud A. Monshi <sup>a,b,\*</sup>, Josiah Poon <sup>a</sup>, Vera Chung <sup>a</sup>, Fahad Mahmoud Monshi <sup>c</sup>

<sup>a</sup> School of Computer Science, The University of Sydney, Camperdown, NSW, 2006, Australia

<sup>b</sup> Department of Information Technology, Taif University, Taif, 26571, Saudi Arabia

<sup>c</sup> Radiology and Medical Imaging Department, King Saud University Medical City, Riyadh, 12746, Saudi Arabia

## ARTICLE INFO

### Keywords:

Chest X-Ray  
Convolutional neural network  
COVID-19  
Data augmentation  
Hyperparameters

## ABSTRACT

To mitigate the spread of the current coronavirus disease 2019 (COVID-19) pandemic, it is crucial to have an effective screening of infected patients to be isolated and treated. Chest X-Ray (CXR) radiological imaging coupled with Artificial Intelligence (AI) applications, in particular Convolutional Neural Network (CNN), can speed the COVID-19 diagnostic process. In this paper, we optimize the data augmentation and the CNN hyperparameters for detecting COVID-19 from CXRs in terms of validation accuracy. This optimization increases the accuracy of the popular CNN architectures such as the Visual Geometry Group network (VGG-19) and the Residual Neural Network (ResNet-50), by 11.93% and 4.97%, respectively. We then proposed CovidXrayNet model that is based on EfficientNet-B0 and our optimization results. We evaluated CovidXrayNet on two datasets, including our generated balanced COVIDcxr dataset (960 CXRs) and the benchmark COVIDx dataset (15,496 CXRs). With only 30 epochs of training, CovidXrayNet achieves state-of-the-art accuracy of 95.82% on the COVIDx dataset in the three-class classification task (COVID-19, normal or pneumonia). The CovidXRayNet model, the COVIDcxr dataset, and several optimization experiments are publicly available at <https://github.com/MaramMonshi/CovidXrayNet>.

## 1. Introduction

Coronavirus disease 2019 (COVID-19), caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), became a global pandemic in less than four months after first appearing in December 2019 in Wuhan, China. It has since reached 127.34 million confirmed cases and over 2.78 million deaths worldwide, as of March 30, 2021 [1]. This caused devastating issues in public health and the global economy. COVID-19 patients may have one or more of the following symptoms: fever, cough, sore throat, headache, fatigue, muscle pain, and shortness of breath [2]. Early detection of positive COVID-19 cases is the most critical factor in slowing the spread of this pandemic.

The golden standard for diagnosing COVID-19 patients is the reverse transcriptase-polymerase chain reaction (RT-PCR) testing, which detects SARS-CoV-2 through collected respiratory specimens of nasopharyngeal or oropharyngeal swabs [3]. However, RT-PCR testing is time-consuming, laborious, and shows poor sensitivity [4]. Alternatively, chest radiography imaging, including computed tomography (CT) or chest X-ray, may be examined by a radiologist to inspect any

visual indicators linked to SARS-CoV-2 [5]. Whereas CT scans have greater image details, CXR images are more accessible, portable, and offer rapid triaging. CXR imaging is more accessible in most healthcare systems than CT scanners that require expensive equipment and maintenance. The portability of the CXR system reduces the risk of COVID-19 transmission by performing the exams within the isolation room, which is not possible with the fixed CT scanners. Importantly, CXR allows rapid triaging of suspected COVID-19 cases in most affected countries like the USA, Spain, and Italy where they have run out of both capacity and PC-RCT testing supplies [6]. Combining laboratory results with radiological image features can speed the process of COVID-19 detection.

Artificial Intelligence (AI) applications coupled with chest radiological imaging can speed the COVID-19 diagnosing process. Deep Learning (DL), in particular, enables AI-based models to achieve accurate results without manual feature extraction [7]. For example, a Convolution Neural Network (CNN), which is a supervised DL approach, has recently gained popularity among the research community of AI in medicine. For COVID-19 detection from chest x-ray images, CNN produced the best classification accuracy compared to other classification

\* Corresponding author. School of Computer Science, The University of Sydney, Camperdown, NSW, 2006, Australia.

E-mail address: [mmon4544@uni.sydney.edu.au](mailto:mmon4544@uni.sydney.edu.au) (M.M.A. Monshi).

**Table 1**  
Dataset of COVID-19 CXR.

Dataset	Description
Figure 1 COVID-19 Chest X-Ray Dataset Initiative <sup>a</sup>	56 CXR, metadata & clinical notes
ActualMed COVID-19 Chest X-Ray Dataset Initiative <sup>b</sup>	239 CXR, metadata & clinical notes
covid-19-ct-cxr <sup>c</sup> [18]	263 CXR and relevant text
COVID-19 image data collection <sup>d</sup> [19]	654 CXR, metadata & clinical notes
COVID-19 radiography database <sup>e</sup>	219 COVID-19, 1341 normal & 1345 Pneumonia CXR
COVIDx <sup>f</sup> [20]	13917 CXR for training & 1579 CXR for testing

- <sup>a</sup> <https://github.com/agchung/Figure1-COVID-chestxray-dataset>.  
<sup>b</sup> <https://github.com/agchung/Actualmed-COVID-chestxray-dataset>.  
<sup>c</sup> <https://github.com/ncbi-nlp/COVID-19-CT-CXR>.  
<sup>d</sup> <https://github.com/ieee8023/covid-chestxray-dataset>.  
<sup>e</sup> <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>.  
<sup>f</sup> <https://github.com/lindawangg/COVID-Net>.

techniques, such as Artificial Neural Network (ANN), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) [8].

Typically, the CNN model is created by combining one or more of the following: a convolution layer, a pooling layer, and a fully connected layer that extracts features from the input, minimizes the size for computational performance, and classifies an image, respectively.

Simultaneously, the CNN model adjusts its internal parameters to achieve a specific task, like classifying chest X-rays [9,10]. The performance of such CNN models can be improved in various ways, including optimizing data augmentation and CNN hyperparameters.

### 1.1. Data augmentation

A method that artificially inflates the original training set  $S$  with label preserving transformations is a data augmentation method. It can be mapped as  $\varphi : S \mapsto T$ , where  $T$  is the augmented set of  $S$ . The label preserving transformation means that if image  $x \in S$ , then  $\varphi(x) \in T$  [11]. Hence, the artificially enlarged training set is defined as  $S' = S \cup T$ , where  $S'$  consists of  $S$  and the corresponding transformations denoted by  $\varphi$ . Resizing, flipping, and zooming are examples of data augmentation methods.

Data augmentation improves CNN performance [12], prevents over-fitting [11], and is easy to implement [13]. Training a CNN on limited data, such as COVID-19 data, inhibits its ability to generalize results to unseen data due to the over-fitting issue. However, inflating the dataset by using data augmentation methods adds more invariant cases and thus prevents over-fitting. In addition, generic methods are easy to implement and computationally inexpensive. Several recent works have proved the benefit of data augmentation in improving CNN-based models for various DL applications [12]. [11]. [13]. However, limited existing methods specifically address data augmentation in

**Table 2**  
Models for detecting COVID-19 from CXR.

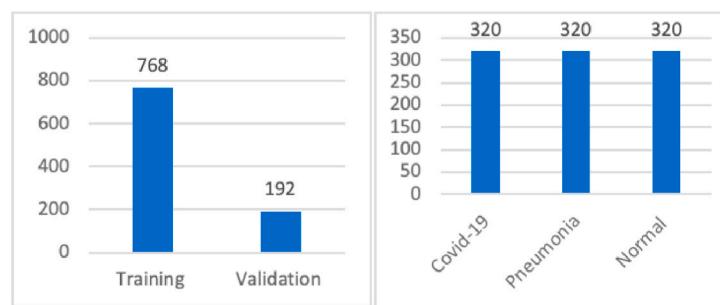
Classification	Model	Acc (%)	Repositories/Datasets	COVID-19	Pneumonia	Normal
Binary	COVIDx-Net [21]	90.00	COVID-19 image data collection	25	-	25
	CovXNet [22]	97.40	Guangzhou Medical Center in China & Sylhet Medical College in Bangladesh	305	-	305
	ResNet-50 [23]	98.00	COVID-19 image data collection & Kaggle	50	-	50
	DarkCovidNet [24]	98.08	COVID-19 image data collection & ChestXray-14	125	-	500
Multi-class	VGG-16 [25]	83.68	COVID-19 image data collection & Radiological Society of North America (RSNA)	215	533	500
	DarkCovidNet [24]	87.02	COVID-19 image data collection & ChestXray-14	125	500	500
	CovXNet [22]	90.30	Guangzhou Medical Center in China & Sylhet Medical College in Bangladesh	305	305-Viral 305-Bacterial	305
	COVID-Net [20]	93.30	COVIDx	53	5526	8066
	MobileNet-v2 [26]	94.72	COVID-19 image data collection, Radiological Society of North America (RSNA), Radiopaedia, Italian Society of Medical & Interventional Radiology (SIRM) & Germany dataset	224	700	504
	CNN-SVM [27]	95.33	COVID-19 image data collection, COVID-19 radiography database & Germany dataset	127	127	127

**Table 3**  
Data augmentation for detecting COVID-19 from CXR.

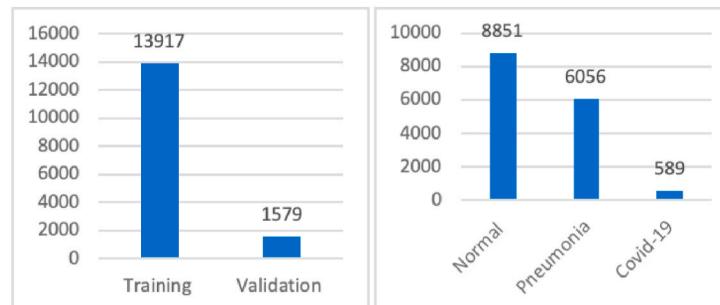
Model	Software	Norm.	Size	Flip	Rotate	Zoom	Light	Extra
VGG-16 [25]	Keras, Tensorflow	-	220*220	HORIZ	15	85-115%	-	shear transformation mixup: 0.1
DarkCovidNet [24]	fastai v1, Pytorch	yes	256*256	-	-	-	-	default values of fastai
CovXNet [22]	Keras, Tensorflow	yes	uniform	-	30	0.2	-	rescale: 1/255 shift: 0.1
COVID-Net [20]	Keras, Tensorflow	yes	480*480	HORIZ	yes	yes	-	intensity shift
MobileNet-v2 [26]	-	-	200*266	-	-	-	-	background: 1:1.5

**Table 4**  
CNN hyperparameters for detecting COVID-19 from CXR.

Model	CNN	Pretrained	Optimizer	Learning Rate	Loss Function	Epoch	Batch
VGG-16 [25]	VGG-16	yes	Adam	1e-4 3e-3	cross entropy	100	8
DarkCovidNet [24]	YOLO DarkNet-19					100	32
CovXNet [22]	CovXNet			1e-3		70	128
COVID-Net [20]	COVID-Net			2e-4 & lr policy		22	64
MobileNet v2 [26]	MobileNet v2			-		10	64



(a) COVIDcxr (A random split is performed [with a fixed seed] by setting 20% of the data for the validation set.)



(b) COVIDx (The validation set consists of 100 COVID-19, 594 pneumonia, and 885 normal cases, in a fixed manner.)

**Fig. 1.** Dataset distribution.

detecting COVID-19 from chest x-rays. A shortcoming of existing studies is the limited amount of data augmentation methods evaluated. As such, this is the investigative scope of our paper because data augmentation leads to positive results when training CNN on limited data but only with suitable augmentation techniques for each dataset [14].

As there is an endless array of mappings  $\varphi(x)$ , we examine common data augmentation methods, including resizing, flipping, rotating, zooming, warping, lighting and normalizing. Our investigative space was determined by consultations with practical radiologist and research on common techniques in the literature. From a radiologist's perspective, the use of portable devices, that minimize the infection control issues of COVID-19 result in low-quality CXRs and incorrect rotation. From the literature's perspective, researchers tend to apply resizing, zooming, warping, and lighting to increase the number of cases to handle the issue of limited COVID-19 data.

### 1.2. CNN hyperparameters

CNN hyperparameter optimization, on the other hand, aims to find the optimal combination of values that must be selected for a given dataset before the training starts in a reasonable amount of time (e.g., the number of epochs). Deep learning practitioners aim to identify such values through automatic software, such as Optuna [15], or through a trial and error method. For example, Nishio et al. [16] utilize Optuna to implement Bayesian optimization in segmenting the lungs from severely abnormal CXRs.

### 1.3. Contribution

The main contribution of this study is the implementation of the CovidXrayNet model, which improves the detection rate of COVID-19 from CXRs, by means of optimizing the data augmentation pipeline and CNN hyperparameters. To the best of the authors' knowledge,

CovidXrayNet is one of the first models that demonstrates the effects of data augmentation pipelines on CXR quality while also investigating several CNN hyperparameters. This in turn may significantly enhance the accuracy of CNN in diagnosing COVID-19. In addition, we introduce COVIDcxr, a balanced and complete dataset that consists of CXRs and the associated tabular data.

In this paper, we use a three-class classification ("COVID-19", "pneumonia", "normal") because these three automatic predictions can help doctors to quickly triage patients for RT-PCR testing for COVID-19 diagnosis confirmation and choose the suitable treatment plan based on the presence and cause of infection (i.e., COVID-19 infection or non-COVID-19 infection). We investigate data augmentation on the COVID-19 CXR classification task to observe the differences between them in terms of the model's accuracy. We also explain and visualize the chosen data augmentation techniques on CXR, (including resizing, flipping, rotating, zooming, warping, lighting, and normalizing) to understand what happens behind the scenes.

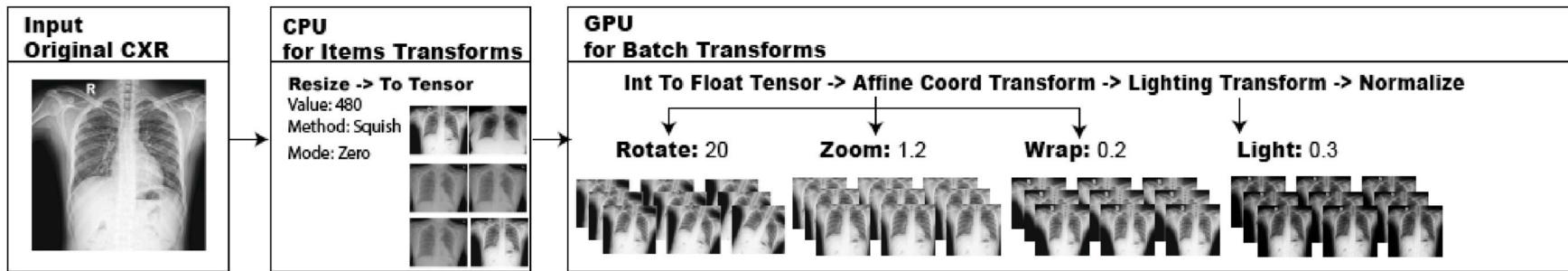
## 2. Related work

A growing number of research publications have demonstrated the compelling ability of deep learning with CNNs to automatically detect COVID-19 from chest X-ray images.

### 2.1. Dataset

**Table 1** outlines the public datasets of COVID-19 CXR. Currently, the largest and most popular dataset among researchers is COVIDx. However, COVIDx is unbalanced as the number of cases in the COVID-19 class (589) is far less than pneumonia (6,056) and no-finding (8,851). This may cause a sharp increase and decrease in the loss values while training a DL model. To address this issue, Bridge et al. [17] proposed the generalized extreme value (GEV) as an alternative to the common

## 1. Data Augmentation Pipeline



## 2. Training

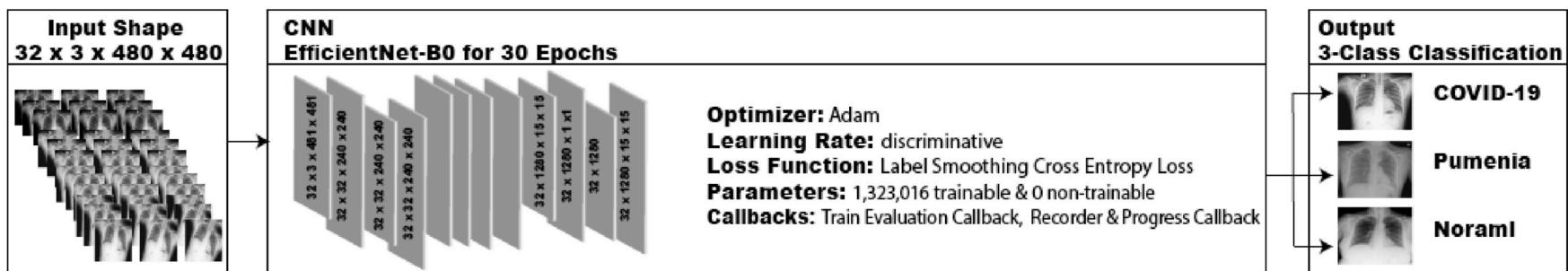


Fig. 2. CovidXrayNet structure.

**Table 5**

Pipeline for Data Augmentation on CXR. For each independent parameter, we trained ResNet-18 on COVIDcxr for 30 epochs to examine the effects of various transformers on COVID-19 CXR classification.

Independent Parameter	Resize		Rotate	Zoom	Wrap	Light	Extra	(%)		
	Size	Method						Acc	AUC	F1
<b>Resize</b>	224*224	crop	0	0	0	0	none	78.12	90.84	78.04
		pad						79.68	90.66	79.39
		squish						74.47	88.63	74.47
	256*256	crop						79.16	92.12	79.12
		pad						76.04	90.62	75.55
		squish						78.12	90.15	78.22
	480*480	crop						80.72	94.42	80.65
		pad						82.81	<b>94.67</b>	82.86
		<b>squish</b>						<b>83.85</b>	94.14	<b>83.95</b>
	512*512	crop						80.72	93.35	80.78
		pad						78.64	93.22	78.62
		squish						77.08	92.67	77.22
<b>Rotate</b>	480*480	squish	0	0	0	0	none	83.85	94.14	83.95
			10					85.93	95.73	85.95
			20					<b>86.45</b>	<b>96.48</b>	86.56
			30					86.45	95.97	<b>86.58</b>
			50					84.89	95.72	85.03
<b>Zoom</b>	480*480	squish	0	1	0	0	none	83.85	94.14	83.95
			1.2					<b>85.41</b>	<b>95.86</b>	<b>85.48</b>
			1.3					82.29	95.77	82.37
			1.4					84.37	95.60	84.45
			1.5					81.25	95.29	81.21
<b>Warp</b>	480*480	squish	0	0	0	0	none	83.85	94.14	83.95
			0.1					84.37	95.36	84.42
			0.2					<b>85.41</b>	96.33	<b>85.50</b>
			0.3					84.89	<b>96.34</b>	84.94
			0.4					82.81	95.34	82.97
<b>Lighting</b>	480*480	squish	0	0	0	0	none	83.85	94.14	83.95
			0.1					81.77	93.12	81.93
			0.2					83.85	94.34	83.91
			0.3					<b>85.41</b>	95.10	<b>85.46</b>
			0.4					82.81	95.34	82.97
<b>Flip (dihedral)</b>	480*480	squish	0	0	0	0	flip	83.85	<b>95.69</b>	83.81
							mixup	83.33	94.88	83.29
							erase	80.72	94.11	80.91
							norm	<b>83.85</b>	94.14	<b>83.95</b>
<b>Mixup (0.4)</b>	480*480	squish	20	1.2	0.2	0.3	flip	81.77	95.70	81.69
			20	1.2	0.2	0.3	mixup	82.81	95.86	82.48
			20	1.2	0.2	0.3	flip, norm	81.77	95.70	81.69
			20	1.2	0.2	0.3	norm	<b>88.02</b>	<b>96.20</b>	<b>88.14</b>
<b>Erasing (random)</b>	480*480	squish	20	1.2	0.2	0.3	flip	81.77	95.70	81.69
			20	1.2	0.2	0.3	mixup	82.81	95.86	82.48
			20	1.2	0.2	0.3	flip, norm	81.77	95.70	81.69
			20	1.2	0.2	0.3	norm	<b>88.02</b>	<b>96.20</b>	<b>88.14</b>
<b>Normalize (Imagenet)</b>	480*480	squish	20	1.2	0.2	0.3	flip	81.77	95.70	81.69
			20	1.2	0.2	0.3	mixup	82.81	95.86	82.48
			20	1.2	0.2	0.3	flip, norm	81.77	95.70	81.69
			20	1.2	0.2	0.3	norm	<b>88.02</b>	<b>96.20</b>	<b>88.14</b>
<b>Multiple Param (pipeline)</b>	480*480	squish	20	1.2	0.2	0.3	flip	81.77	95.70	81.69
			20	1.2	0.2	0.3	mixup	82.81	95.86	82.48
			20	1.2	0.2	0.3	flip, norm	81.77	95.70	81.69
			20	1.2	0.2	0.3	norm	<b>88.02</b>	<b>96.20</b>	<b>88.14</b>

sigmoid activation function. They proved that GEV distribution improves the performance of COVID-19 classification from unbalanced datasets.

These COVID-19 CXR datasets are constantly updated with new images added by researchers around the world. Nevertheless, none of these datasets provides complete metadata for all patients.

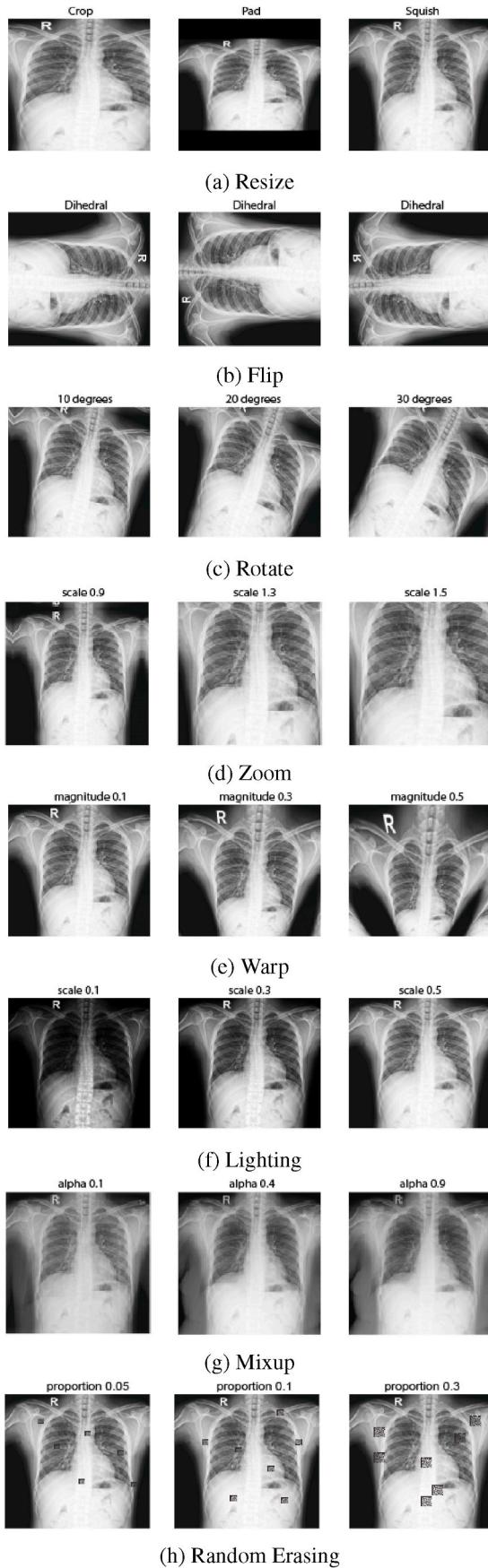
## 2.2. Models

Table 2 summarizes the proposed CNN based models in the literature, which can be grouped into binary classification (i.e., COVID-19 or normal), and multi-class classification (COVID-19, pneumonia or normal).

For the binary classification, COVIDX-Net [21] achieves 90% with only 25 CXRs for COVID-19 and 25 CXRs for normal patients. Another balanced dataset with 305 cases in each class was used to train the CovXNet model, resulting in 97.40% accuracy [22]. With a combination of ResNet50, InceptionV3, and Inception-ResNetV2, Narin et al. [23] model achieves 98% accuracy. The 50 healthy CXRs in this study, however, belong to children (one to five years olds) from a Kaggle repository [28]. Using a larger but unbalanced dataset of 1125 images, DarkCovidNet [24] achieves 98.08% accuracy. Beyond the classification task, Wang et al. [29] localize the pulmonary location coordinates of COVID-19 (i.e., left lung, right lung, or both [bi-pulmonary]), using a residual attention network [30].

For the three-class problem, Nishio et al. [25] achieve 83.68% accuracy using a VGG-16 based model with a combination of data augmentation methods. By starting with a real-time object detection system, named “you only look one” (YOLO), which is based on the Darknet-19 [31] classifier, DarkCovidNet achieves 87.02% accuracy [24]. However, this result could be biased due to the small number of COVID-19 cases (125), compared to 500 pneumonia cases and 500 normal cases. To compensate for this issue, Mahmud et al. [22] transfer training from a large dataset of normal cases and viral/bacterial pneumonia cases to a small balanced COVID-19 dataset, achieving 90.3% accuracy for their CovXNet model. Further, COVID-Net [20] leveraged the generative synthesis [32] to determine the optimal design, where COVID-19 sensitivity and positive predictive value (PPV) are at or above 80%. Conversely, Oh et al. [33] proposed a patch-based CNN method that may handle the issue of small datasets, as it uses only 11.6 million trainable parameters on COVIDx. Note that the COVID-Net team is releasing enhanced versions of COVID-Net through a GitHub repository, and this paper refers to the COVID-Net-CXR3-B version. Apostolopoulos and Mpesiana [26] achieve a 93.48% performance by transferring the learning of MobileNet v2 [34]. They conclude that MobileNet v2 is better than VGG-19 [35] for this particular COVID-19 classification task, as it has the fewest instances of False Negatives. Furthermore, Sethy et al. [27] add a support vector machine (SVM) to classify the features obtained from CNN models and achieved 95.33% accuracy.

Table 3 and Table 4 outline the data augmentation and the CNN



**Fig. 3.** Visualizing Data Augmentation Effects on CXR. The CXR is for a 25-year-old COVID-19-positive female taken from the COVID-19 image data collection.

hyperparameters in recent proposed models, respectively. Nishio et al. [25] show that combining multiple data augmentation techniques is more effective than only using one or not using any data augmentation in detecting COVID-19 from CXRs. They utilize a random search [36] to select the optimal VGG-16 hyperparameters and data augmentation methods, including conventional method and mixup [37]. This resulted in an increase in their model's accuracy from the initial 78.72%–83.68%. However, this approach of hyperparameter tuning is hard to achieve with complex networks such as EfficientNet [38] due to the large number of trainable parameters.

In terms of optimizing CNN hyperparameters, existing models used pre-trained architectures on ImageNet, Adam optimizer [39], epochs that ranged from 10 to 100, and a batch size of 8, 32, 64 or 128. Notably, several proposed architectures apply few arbitrary transformers to the X-rays based on random choices rather than well-justified motives. For instance, Ozturk et al. [24] apply the default values in the fastai v1 library. However, selecting the optimal CNN hyperparameters and data augmentation methods improves the robustness of CNN models [13].

### 3. Proposed model

#### 3.1. Dataset

We trained CovidXrayNet on two datasets called COVIDcxr and COVIDx (refer to Fig. 1). Both datasets contain three classes of CXR: COVID-19 viral infection, pneumonia (i.e., non-COVID-19 infection such as viral and bacterial), and normal (i.e., no infection).

COVIDcxr is the dataset of COVID-19 that we have generated from two open-source repositories: ChestX-Ray14 [40] and COVID-19 image data collection [19] with the associated tabular data (i.e., gender, sex, and view) for each patient. It is comprised of 960 CXR images. Our aim was to create a balanced, unbiased, and complete COVID-19 CXR dataset. We randomly selected 320 no-finding and 320 pneumonia from ChestX-Ray14, and 320 COVID-19 from the COVID-19 image data collection, along with complete metadata. There are 568 male and 392 female cases, and the average age of these subjects is about 56 years.

COVIDx [20] is the largest public dataset in terms of presented positive COVID-19 cases. It includes 15,496 CXRs generated from five public datasets, where three of them—COVID-19 Image Data Collection, Figure 1 COVID-19 Chest X-Ray Dataset Initiative, and ActualMed COVID-19 Chest X-Ray Dataset Initiative can be downloaded from the GitHub repository, and two datasets—RSNA Pneumonia Detection challenge dataset and COVID-19 radiography database can be obtained from Kaggle. Note that COVIDx is expanding on a regular basis with the addition of new patient records for training while maintaining the same test dataset for consistency. We employed COVIDx-v3 in this research.

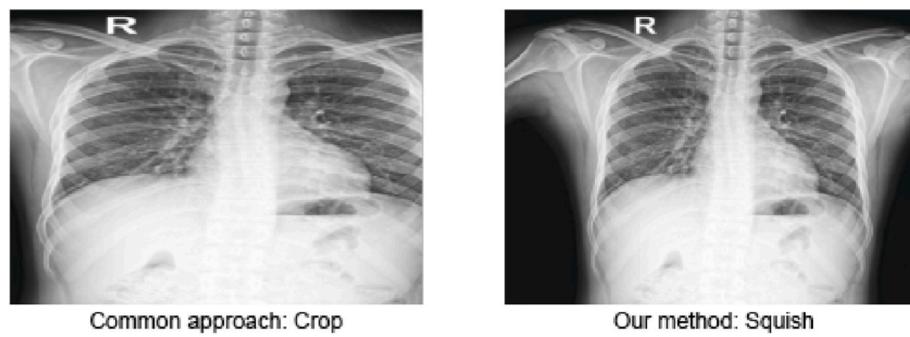
#### 3.2. CovidXrayNet architecture

The overall structure of our proposed CovidXrayNet model, which classifies a CXR into either “COVID-19”, “normal”, or “pneumonia”, is presented in Fig. 2. Before feeding the CXRs to the pre-trained EfficientNet-B0 along with the optimized CNN hyperparameters, we performed several augmentation techniques on the data.

##### 3.2.1. Data augmentation

First, we performed several deliberate data augmentations, based on extensive experiments on the COVIDcxr dataset using ResNet18 as it can be seen in Table 5. Fig. 3 plots all transformer techniques against each other to observe the differences between them.

At the item transformation level, we resized each CXR to 480x480



**Fig. 4.** Resizing Method. We propose to squish a 480\*480 pixel CXR rather than cropping it to maintain important CXR details from the edges of the image.

**Table 6**

CNN Architectures on COVIDx and COVIDcxr. We trained the popular CNN architectures on both datasets for 30 epochs using the optimized data augmentation pipeline.

CNN	Dataset	Acc (%)	AUC (%)	MCC (%)	Precision (%)	Recall (%)	F1 (%)
VGG-16	COVIDcxr	80.73	94.68	72.29	82.03	81.35	80.53
VGG-19		84.90	95.67	77.74	85.31	85.26	84.92
ResNet-18		85.94	96.72	79.40	86.84	86.31	86.14
ResNet-34		79.69	94.91	70.02	80.26	80.03	79.70
ResNet-50		82.81	95.90	75.31	84.90	83.29	83.12
<b>EfficientNet-B0</b>		<b>88.02</b>	-	<b>82.01</b>	<b>87.98</b>	<b>88.03</b>	<b>88.00</b>
VGG-16	COVIDx	93.41	98.70	87.74	94.40	89.41	91.61
VGG-19		93.60	98.55	88.06	95.29	85.53	89.24
ResNet-18		93.29	98.86	87.48	95.03	86.73	90.05
ResNet-34		94.74	99.10	90.19	95.85	89.95	92.53
ResNet50		95.12	99.22	90.92	96.08	91.76	93.72
<b>EfficientNet-B0</b>		<b>95.69</b>	-	<b>92.01</b>	<b>96.24</b>	<b>94.76</b>	<b>95.48</b>

pixels by squishing the CXR on the horizontal axis on the Central Processing Unit (CPU). This constricts the ribcage towards the center while keeping all the parts of the CXR. Our method is different from the common approach in the literature, which resizes each CXR to the same aspect ratio to set the smallest dimension to a specified size and then arbitrarily crops it on the other dimension, as illustrated in Fig. 4. This cropping method may erase important CXR details from the edges of the image. Resizing all CXRs to a fixed size is a prerequisite data augmentation for classifying them using CNN.

At the batch transformation level, we applied a group of optimized augmentation parameters on a Graphical Processing Unit (GPU) to minimize the number of computation and lossy operations. We used pipeline to compose the best transformers' values together. A series of experiments on the COVIDcxr dataset, with a fixed seed, was used to find the best combination of choices and orders of data augmentation that ensured the ResNet-18 gave the best accuracy, as recorded in Table 5.

We applied a random rotation with a maximum of 20° and 75% probability to overcome the incorrect rotation of some of the acquired images. Such low-quality CXRs are the result of using portable devices that minimize the infection control issues of COVID-19 [41]. In addition, it is not uncommon, especially for anteroposterior (AP) supine CXRs, for the patient to be rotated, which makes interpretation difficult. In addition to rotating CXRs, we also applied zooming, warping, and lighting as we relied on data augmentation to handle the issue of limited COVID-19 data through increasing the number of cases [12] and, hence, preventing overfitting. With a 75% probability, we zoomed the CXRs by a scale of 1.2, lighted by a scale of 0.3, and warped by a magnitude of 0.2. Warping and lightening augmentation may handle situations when patients face the X-ray device at different angles and in various lighting rooms. We attempted to apply the random erasing [42] and the mixup [37] techniques, however, we did not notice improved performance.

### 3.2.2. CNN architectures and hyperparameters

Second, we replaced the head of EfficientNet-B0 with a head suitable

for the three-class classification and trained it for 30 epochs. To compensate for the small dataset, we perform transfer learning with the pre-trained weights from ImageNet. Then, we fine-tuned EfficientNet-B0 using one NVIDIA Tesla V100. EfficientNet scales the CovidXrayNets' width and depth according to the size of 480x480 pixels, which results in substantially less computational power used and fewer parameters with high performance compared to other CNN architectures.

Table 6 presents the performance of the optimized data augmentation on two datasets COVIDcxr (small and balanced dataset) and COVIDx (large and unbalanced dataset) using the benchmark deep neural network architectures including: VGG-16, VGG-19 [35], ResNet-18, ResNet-34, ResNet-50 [43], and EfficientNet-B0 [38]. Among various CNN architectures, EfficientNet-B0 accomplishes the best results in classifying COVID-19 from the COVIDcxr and COVIDx datasets based on various evaluation metrics such as accuracy, precision, recall and F1 scores. Please refer to Section 3.3 for more details about these evaluation metrics.

EfficientNet introduced a new and simple compound scaling technique to scale the number of layers  $\alpha$ , the number of channels  $\beta$ , and the number of pixels  $\gamma$  in an image, which represent the CNN width, depth, and resolution, respectively [38], as depicted in Eq. (1). This technique uses a compound coefficient  $\varphi$ , which defines the amount of available resources to determine how to scale  $\alpha$ ,  $\beta$ , and  $\gamma$ . The constraint  $(\alpha \cdot \beta^2 \cdot \gamma^2) \approx 2$  is applied in order to make sure that the total floating-point operations per second (FLOPS) do not exceed  $2^\varphi$ . CovidXrayNet is based on the baseline network EfficientNet-B0, where the optimal values are  $\alpha = 1.2$ ,  $\beta = 1.1$ , and  $\gamma = 1.15$ . Using this multi-objective neural architecture search, we optimize both accuracy and FLOPS. Although the original EfficientNet-B0 uses the standard input size 224x224, it perfectly handles the 480x480 CXR pixels.

**Table 7**

Optimizing CNN hyperparameters using COVIDcxr. For each independent parameter, we trained several architectures on COVIDcxr to examine the effects of various hyperparameters on the accuracy of COVID-19 CXR classification.

CNN	Epoch	Batch Size	Loss Function	Acc (%)	MCC (%)	F1 (%)
VGG-16	10	32	Cross Entropy	77.08	68.15	76.26
	20			77.60	66.71	77.43
	30			80.73	72.29	80.53
	40			83.33	75.51	83.31
	30	8		<b>85.42</b>	<b>79.20</b>	<b>85.43</b>
		16	Label Smoothing	84.38	76.61	84.27
		32		79.17	69.62	78.91
	VGG-19	10		78.65	68.38	78.89
	20			82.81	74.25	82.96
	30			84.90	77.74	84.92
ResNet-18	40			84.38	76.66	84.35
	30	8		84.90	78.36	84.96
		16		82.81	74.90	82.74
		32		<b>85.42</b>	<b>78.30</b>	<b>85.51</b>
	10	32	Cross Entropy	81.25	73.69	81.25
	20			82.29	74.21	82.45
	30			<b>85.94</b>	<b>79.40</b>	<b>86.14</b>
	40			85.42	78.16	85.37
	30	8		81.25	73.56	81.39
ResNet-34	16			82.29	74.20	82.37
		32	Label Smoothing	84.38	76.95	84.46
	10	32		81.25	72.10	81.20
	20			81.25	71.93	80.91
	30			79.69	70.02	79.70
	40			81.25	71.94	81.23
	30	8		<b>86.46</b>	<b>80.12</b>	<b>86.54</b>
		16		85.94	79.00	85.87
		32		83.85	76.08	83.85
	ResNet-50	10	32	Cross Entropy	81.77	73.18
EfficientNet-B0	20		84.90	77.32	84.93	
	30		82.81	75.31	83.12	
	40		85.42	78.12	85.45	
	30	8	86.46	80.49	86.52	
		16	<b>86.98</b>	<b>80.84</b>	<b>87.16</b>	
		32	Label Smoothing	83.85	76.21	84.05
	10	32		83.33	75.36	83.65
	20			84.38	76.67	84.41
	30			88.02	82.01	88.00
	40			85.42	78.10	85.42
EfficientNet-B0	30	8		88.02	82.06	87.89
		16		86.98	80.45	86.99
		32		<b>88.54</b>	<b>82.83</b>	<b>88.62</b>

$$\text{depth: } d = \alpha^\varphi$$

$$\text{width: } w = \beta^\varphi$$

$$\text{resolution: } r = \gamma^\varphi$$

$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

**Table 8**

Optimizing CNN hyperparameters using COVIDx. For each independent parameter, we trained several architectures on COVIDx to examine the effects of various hyperparameters on the accuracy of COVID-19 CXR classification.

CNN	Epoch	Batch Size	Loss Function	Acc (%)	MCC (%)	F1 (%)
VGG-16	10	32	Cross Entropy	92.08	85.20	86.99
	20			93.35	87.56	90.10
	30			93.41	87.74	91.61
	40			94.24	89.25	91.99
	30	8		93.86	88.56	91.03
		16	Label Smoothing	<b>94.30</b>	<b>89.38</b>	<b>92.00</b>
		32		94.05	88.88	91.35
	VGG-19	10		92.53	86.04	87.29
	20			93.98	88.77	91.57
	30			93.60	88.06	89.24
ResNet-18	40			93.29	87.46	88.72
	30	8		94.49	89.73	92.14
		16	Label Smoothing	<b>94.93</b>	<b>90.56</b>	<b>92.79</b>
		32		93.79	88.40	90.10
	ResNet-18	10		93.10	87.08	88.43
	20			93.60	88.06	90.07
	30			93.29	87.48	90.05
	40			93.86	88.53	90.87
	30	8		94.17	89.11	91.17
		16		<b>94.43</b>	<b>89.60</b>	<b>92.49</b>
ResNet-34	32			94.30	89.35	91.58
	10	32	Cross Entropy	94.05	88.89	91.41
	20			94.62	89.97	93.32
	30			94.74	90.19	92.53
	40			94.43	89.63	93.38
	30	8		94.87	90.44	92.43
		16	Label Smoothing	<b>95.31</b>	<b>91.28</b>	<b>94.50</b>
		32		94.62	89.96	92.50
	10	32		94.93	90.55	92.62
	20			94.81	90.34	93.37
	30			95.12	90.92	93.72
ResNet-50	40			94.81	90.35	93.14
	30	8		93.03	87.01	91.99
		16	Label Smoothing	<b>95.57</b>	<b>91.76</b>	<b>95.35</b>
		32		95.12	90.91	93.36
	10	32		95.69	91.99	94.52
	20			95.19	91.02	93.38
	30			95.69	92.01	95.48
	40			95.00	90.72	95.00
	30	8		94.68	90.16	93.25
		16		95.38	91.40	94.88
EfficientNet-B0	32			<b>95.82</b>	<b>92.24</b>	<b>96.16</b>

**Table 9**

Comparing our Optimised Data Augmentation Pipeline and CNN Hyperparameters with Benchmark. Both papers used VGG19 and ResNet50 on the COVIDx dataset but with different transformers and hyperparameters.

CNN	Paper	Parameters (M)	Acc (%)	AUC (%)	MCC (%)	F1 (%)
VGG-19	COVID-Net [20]	20	83.00	-	-	-
	CovidXrayNet		<b>94.93</b>	<b>98.69</b>	<b>90.56</b>	<b>92.79</b>
ResNet-50	COVID-Net [20]	25	90.60	-	-	-
	CovidXrayNet		<b>95.57</b>	<b>99.29</b>	<b>91.76</b>	<b>95.35</b>

Furthermore, we have studied various CNN hyperparameters on COVIDcxr and COVIDx, including the loss function, the number of epochs, and the batch size, as demonstrated in Table 7 and Table 8, respectively. Based on this trial and error method, we selected the optimal hyperparameters for EfficientNet-B0 on the COVIDx dataset including the label smoothing [44] of the cross-entropy loss function, 30 epochs, and a batch size of 32. Label smoothing for our three-class problem is presented in Eq. (2), where  $(1 - \varepsilon)$  is the prediction of the

**Table 10**

Comparing CovidXrayNet with Benchmark. All models are based on three-class COVID-19 classification. COVID-Net and CovidXrayNet employed the COVIDx dataset.

Model	Acc (%)	MCC (%)	Precision (%)	Recall (%)	F1 (%)
DarkCovidNet [24]	87.02	-	89.96	-	87.37
COVID-Net [20]	93.30	-	-	-	-
MobileNet v2 [26]	93.48	-	-	-	-
CovidXrayNet	<b>95.82</b>	<b>92.24</b>	<b>96.93</b>	<b>95.43</b>	<b>96.16</b>

correct class, and  $\epsilon$  is the prediction of the other two classes. In this formula,  $ce(x)$  denotes the standard cross-entropy loss of  $x$ ,  $\epsilon$  is a small positive number,  $i$  is the correct class, and  $N$  is the number of classes. This regularization technique improved CovidXrayNet performance and robustness by computing cross entropy with a weighted mixture of the hard targets from the COVIDx dataset using the uniform distribution.

$$\text{cross entropy loss} = (1 - \epsilon) ce(i) + \epsilon \sum \frac{ce(j)}{N} \quad (2)$$

We fine-tuned CovidXrayNet using the one-cycle policy [45] and discriminative learning rates [46]. Equation (3) defines this discriminative fine-tuning technique, where CovidXrayNet's parameters  $\theta$  are split into  $\{\theta^1, \dots, \theta^L\}$  and the learning rates  $\eta$  are split into  $\{\eta^1, \dots, \eta^L\}$  at time step “ $t$ ” for the number of layers “ $L$ ”. Using this function, we start with a learning rate of  $2e - 3$ , then automatically adjust this value for both COVIDx and COVIDcxr datasets, where the gradient of the CovidXrayNet's objective function is  $\nabla_{\theta} j$ .

$$\theta_t^l = \theta_{t-1}^l - \eta^l \times \nabla_{\theta} j(\theta) \quad (3)$$

### 3.3. CovidXrayNet evaluation

We computed the accuracy, macro average precision, macro average

recall, and macro F1 score of the CovidXrayNet in distinguishing between the three classes (“COVID-19”, “pneumonia”, “normal”). Equations (4)–(7) explain these metrics for a generic class  $k$ , where  $TP$  refers to True Positives classifications,  $FN$  denotes False Negatives classifications,  $TN$  presents True Negatives classification, and  $FP$  means False Positives classifications. In the Macro approach, all classes are considered as basic elements of the calculation [47] (i.e., each class has the same weight in the average regardless of its size).

$$\text{Accuracy} = \frac{\sum_{k=1}^K \frac{TN_k + TP_k}{TN_k + TP_k + FN_k + FP_k}}{K} \quad (4)$$

$$\text{Precision}_{Macro} = \frac{\sum_{k=1}^K \frac{TP_k}{TP_k + FP_k}}{K} \quad (5)$$

$$\text{Recall}_{Macro} = \frac{\sum_{k=1}^K \frac{TP_k}{TP_k + FN_k}}{K} \quad (6)$$

$$F1_{Macro} = 2 \times \frac{\text{Precision}_{Macro} \times \text{Recall}_{Macro}}{\text{Precision}_{Macro}^{-1} + \text{Recall}_{Macro}^{-1}} \quad (7)$$

Moreover, we used the Area Under the Receiver Operating Characteristic Curve (AUC) [48], and Matthews correlation coefficient (MCC) [49]. AUC for multi-class is defined in Eq. (8), where  $AUC(c_i)$  is the area under the class reference ROC curve for the positive class  $c_i$ . This implementation of AUC score is simple and fast but it is sensitive to class distributions and error costs. MCC, on the other hand, is a good indicator of total unbalanced prediction models as defined in Eq. (9), where “ $c$ ” represents all correctly predicted cases, “ $s$ ” represents all cases, “ $p_k$ ” is the number of instances that class “ $k$ ” was predicted to be, and “ $t_k$ ” is the number of instances that class “ $k$ ” truly occurred. Since accuracy depends mostly on the number of samples in each class, CNN-based models perform seemingly well in the imbalanced datasets, such as COVIDx. This may result in an inaccurate conclusion. Therefore, the combination of multiple evaluation matrices should be the criterion for selecting the best model.

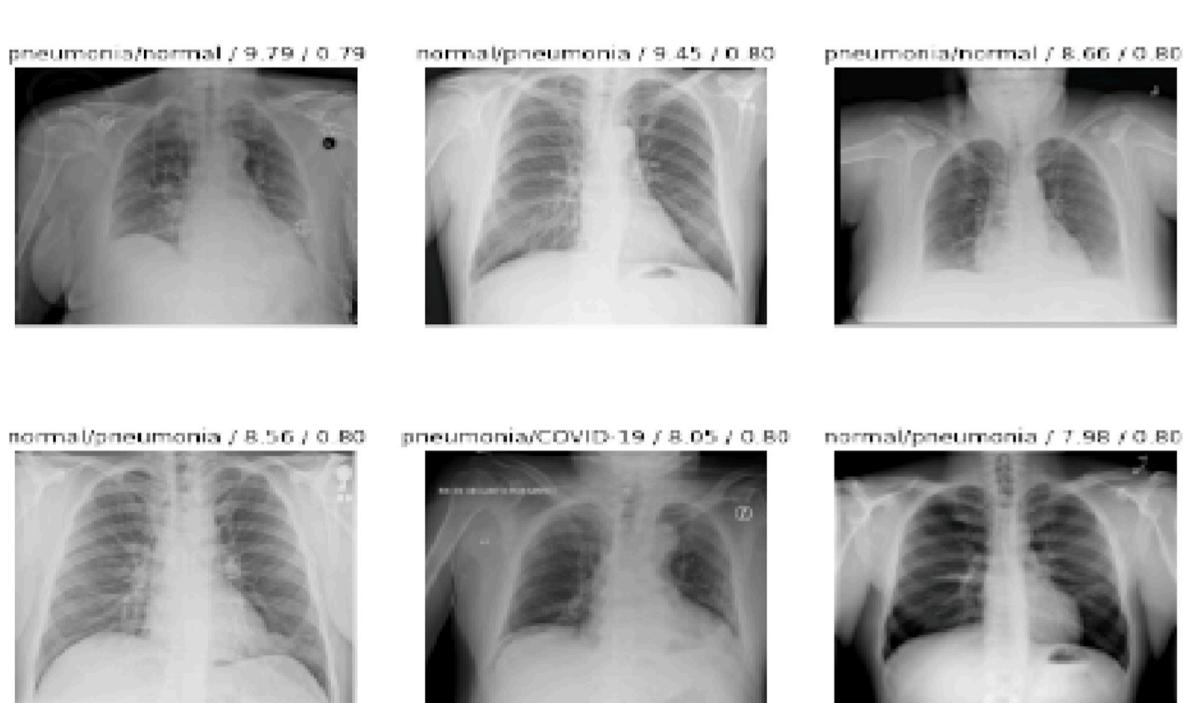
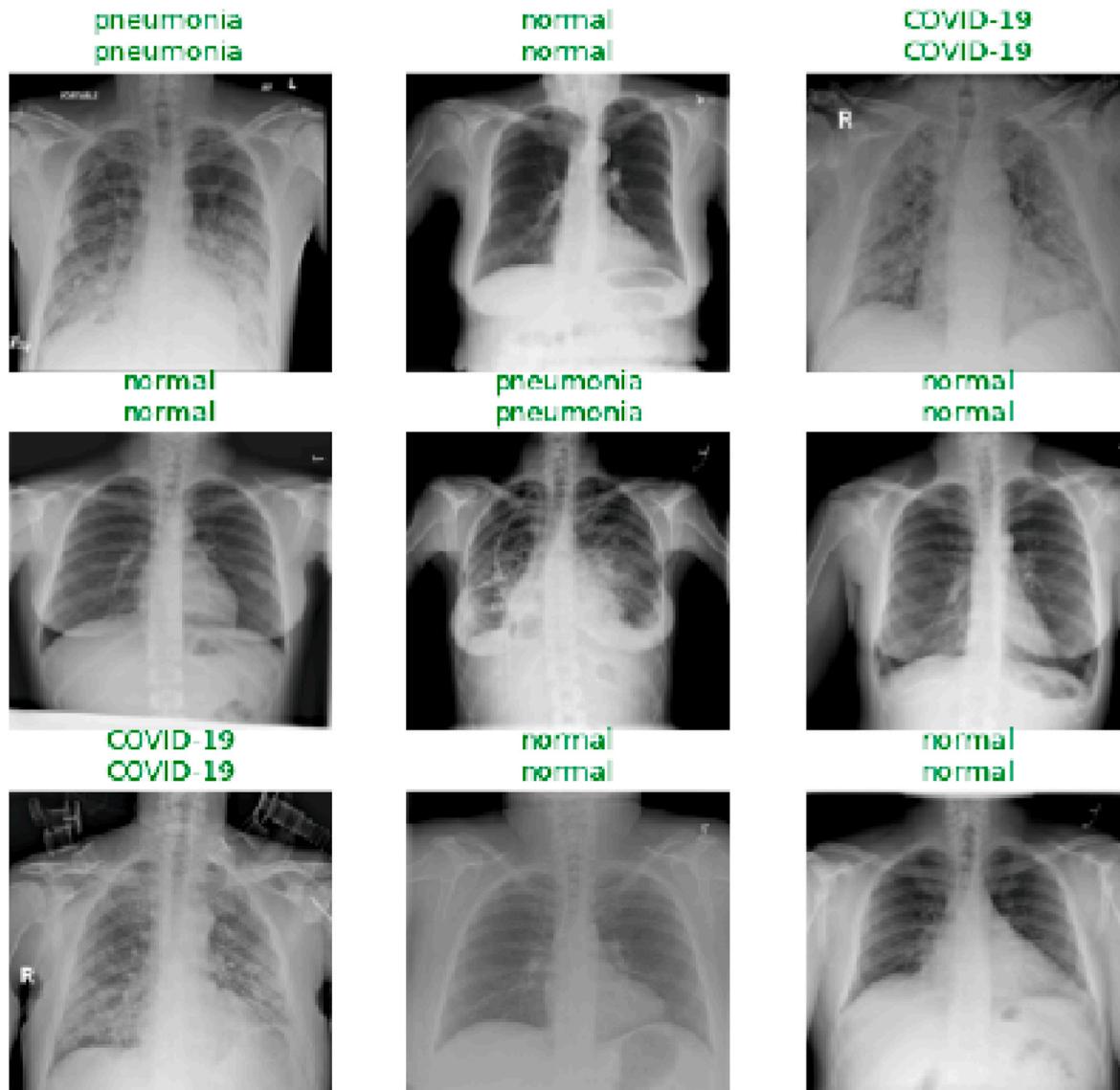


Fig. 5. Top prediction errors generated by CovidXrayNet on COVIDx test dataset.



**Fig. 6.** Randomly generated results for CovidXrayNet on COVIDx test dataset.

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \times p(c_i)$$

$$MCC = \frac{c \times s - \sum_k^K P_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2)(s^2 - \sum_k^K t_k^2)}}$$

## 4. Experiment

### 4.1. Implementation

We used PyTorch software [50], fastai library [51], an n1-highmem-8 (8 vCPUs, 52 GB memory) machine, and one NVIDIA Tesla V100 GPU. Fastai is a deep learning library that enables the implementation of CovidXrayNet with its unique ability to join several transformers inside a pipeline that manages the minimum number of computations and lossy operations.

### 4.2. Result

#### 4.2.1. Quantitative evaluation

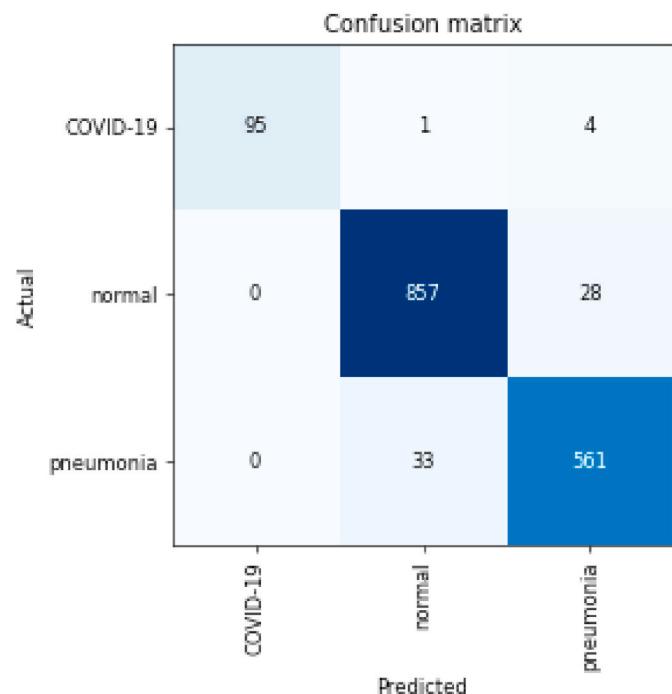
In order to evaluate our proposed data augmentation pipeline, we

(8) compared the reported results of VGG-19 and ResNet-50 in the COVID-Net paper [20] with our results on the COVIDx dataset as recorded in Table 9. With only 30 epochs of learning cycles, the accuracy of VGG-19 increased by 11.93%, while the accuracy of ResNet-50 improved by 4.97%. This clearly indicates the effect of our proposed method on enhancing the accuracy of COVID-19 classification from CXRs.

Table 10 compares CovidXrayNet to other studies in the literature that are based on three-class classification. We achieved better accuracy (95.82%) over the remainder of the models, including DarkCovidNet (87.02%), COVID-Net (93.30%), and MobileNet v2 (93.48%). Further, the F1 score for CovidXrayNet (96.16%) is higher than DarkCovidNet (87.37%) and the precision score of CovidXrayNet (96.93%) is better than DarkCovidNet (89.96%). Significantly, the overall sensitivity of CovidXrayNet is 95.43%. Our reported results are reproducible. We have used the same test dataset as COVID-Net.

#### 4.2.2. Qualitative evaluation

We have ensured the robustness of CovidXrayNet by sharing its top prediction errors and actual labels with expert radiologists (refer to Fig. 5). CovidXrayNet classified four COVID-19 patients as pneumonia. Since COVID-19 is a subset of pneumonia diseases, the diagnosis is correct but the interpretation is not. For this reason, CovidXrayNet can



**Fig. 7.** Confusion matrix for CovidXrayNet on COVIDx test dataset.

only offer a second opinion to the radiologist in the clinical setting.

## 5. Discussion

The rapid spread of the COVID-19 pandemic along with the limited number of RT-PCR test kits and qualified radiologists, has necessitated the need for accurate automated detection systems. CXR is one of the main imaging methods that are fast, non-invasive, affordable, and possibly able to be completed at bedside to monitor the progression of COVID-19 infection. However, radiologists with expertise in CXR interpretation may not be available at every institution.

### 5.1. Optimization in deep learning

We aim to implement an AI model, CovidXrayNet, that can identify COVID-19 infection based on CXRs. CovidXrayNet optimizes data augmentation to enable CNN models to observe visual features that are not noticeable to the radiologist's eye. With data augmentation, CNN models will generalize better results. However, the implications of choosing efficient and effective augmentation techniques depend on the dataset at hand. Using CXR with COVID-19 datasets, we performed a separate search phase that was computationally expensive. Recent work, such as RandAugment [52] and AutoAugment [53], suggests removing the need for a search phase to reduce the parameter space for data augmentation. However, incorrect choices in the COVID-19 classification task may lead to erasing or diluting vital features.

Notably, individual data augmentation methods yielded a minor increased task performance as seen in Table 5. For example, the optimal warping value improves the classification task accuracy by only 1.56%. However, a combination of these optimized methods (i.e., our proposed data augmentation pipeline and CNN hyperparameters) has increased the performance significantly, as can be seen in Table 9. It increases the accuracy of the popular CNN architectures such as VGG-19 and ResNet-50, by 11.93% and 4.97%, respectively.

We find that EfficientNet-B0 performs well for COVID-19 CXR classification with the following data augmentation pipeline: squishing the CXR to 480x480 pixels, rotating by 20°, zooming by 1.2 scale, warping by 0.2 magnitude, lighting by 0.3 scale, and normalizing. Also, the label

smoothing cross-entropy loss function, at the batch size of 32 with 30 epochs, increases the accuracy of CovidXrayNet on the COVIDx dataset. EfficientNet is rapidly becoming the deep learning practitioners' choice over ResNet for many classification tasks. It allows practitioners to use the minimum FLOPS while achieving the best possible accuracy by compound scaling the network's depth, width, and input resolution.

### 5.2. Limitation and future direction

While CovidXrayNet performs well as a whole (see Fig. 6), it mis-identified four patients with COVID-19 as having pneumonia, and one patient with COVID-19 as being normal (refer to the confusion matrix in Fig. 7). However, it is important to limit the number of missed COVID-19 patients to be isolated as well as the number of false-positive COVID-19 patients to avoid unnecessary burden for the clinical sites. Therefore, CovidXrayNet is still at a research stage and is not suitable for direct clinical diagnosis. It can be built upon and optimized with additional data augmentation and better CNN hyperparameters.

Without conducting a proper clinical study, the achieved accuracy of CovidXrayNet (95.82%) on the COVIDx dataset does not indicate that CovidXrayNet is sufficient for detecting COVID-19 from CXR. Our aim is to empower this research wave through our optimized data augmentation pipeline and CNN hyperparameters. Therefore, we are releasing the source code of CovidXrayNet to enable researchers to reproduce the results and experiment on different datasets.

As there is an endless array of transformation, our work evaluates common augmentation techniques in the CXR classification literature (i.e., resize value, resize method, rotate, zoom, warp, light, flip and normalize), recent proposed methods (i.e., mixup and random erasing), and combinations of these methods. Future research can enhance our model with de-noising or segmentation steps. In addition, the proposed data augmentation pipeline was tested only on the three-class classification task ("COVID-19", "normal" or "pneumonia"). Researchers may investigate the effects of the proposed technique on detecting other common CXR observations including atelectasis, cardiomegaly, consolidation, edema, enlarged cardiomedastinum, fracture, lung lesion, lung opacity, pleural effusion, pleural other, pneumonia and pneumothorax.

Designing a fair testing protocol could be highly challenging. Different datasets were merged with large differences among them in order to respond to the global challenge of quickly identifying COVID-19 [19]. COVIDx and COVIDcxr datasets were collected from public sources. They were also indirectly collected from hospitals and physicians. For the COVIDx, we tested our model in the official split recommended by the COVIDx paper to allow for future comparison. For the COVIDcxr dataset, we will release the dataset generation scripts. Future research should assess the validity of the available testing protocol by validating the COVID-19 CXRs with clinical experts and determining the ground truth.

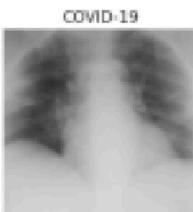
COVIDcxr is suitable for building a single neural network based on both images (CXR) and tabular data (sex, age, and view), as can be seen in Fig. 8. However, we did not observe better performance for such a model than a linear model with embedding. Even though a multi-modal network, with multiple input modalities, receives more information, it is often prone to over-fitting [54]. Future research may explore training multi-modal classification networks based on the COVIDcxr dataset using various CNN architectures and hyperparameters.

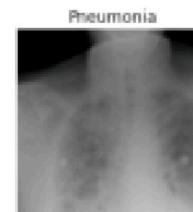
## 6. Conclusion

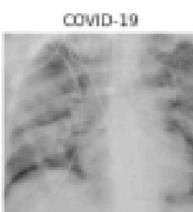
We have demonstrated that optimizing data augmentation and CNN hyperparameters result in outstanding effects on the automatic extraction of features from CXR related to the diagnosis of COVID-19. CovidXrayNet only requires 30 learning cycles to process a CXR yet achieves 95.82% accuracy on the COVIDx dataset.

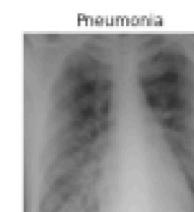
```
mixed_dl.show_batch()
```

	sex	view	age	finding
0	M	PA	41.000000	Pneumonia
1	F	AP	67.000000	No Finding
2	M	PA	75.000000	COVID-19
3	M	PA	21.000001	Pneumonia
4	M	PA	79.000000	No Finding
5	M	PA	68.000000	No Finding
6	M	PA	75.000000	No Finding
7	M	AP Supine	21.000001	COVID-19





**Fig. 8.** Data Loader from COVIDcxr that Combines both Tabular Data and CXR.

#### Declaration of competing interest

The authors declare no competing interests.

#### Acknowledgement

This material is based upon work supported by Google Cloud COVID-19 research credits program.

#### References

- [1] WHO coronavirus (COVID-19) dashboard — WHO coronavirus disease (COVID-19) dashboard [Online]. Available: <https://covid19.who.int/>.
- [2] T. Singhal, A review of coronavirus disease-2019 (COVID-19), Indian J. Pediatr. (2020) 1–6.
- [3] W. Wang, et al., Detection of SARS-CoV-2 in different types of clinical specimens, Jama 323 (18) (2020) 1843–1844.
- [4] C.P. West, V.M. Montori, P. Sampathkumar, COVID-19 testing: the threat of false-negative results, in: In Mayo Clinic Proceedings, vol. 95, Elsevier, 2020, pp. 1127–1129, 6.
- [5] M.-Y. Ng, et al., Imaging profile of the COVID-19 infection: radiologic findings and literature review, Radiology: Cardiothoracic Imag. 2 (1) (2020), e200034.
- [6] G.D. Rubin, et al., The Role of Chest Imaging in Patient Management during the COVID-19 Pandemic: a Multinational Consensus Statement from the Fleischner Society, Chest, 2020.
- [7] Y. LeCun, Y. Bengio, G. Hinton, “Deep learning,” *nature* 521 (7553) (2015) 436–444.
- [8] M. Ahsan, M. Based, J. Haider, M. Kowalski, COVID-19 detection from chest X-ray images using feature fusion and deep learning, Sensors 21 (4) (2021) 1480.
- [9] X. Wang, Y. Peng, L. Lu, Z. Lu, R.M. Summers, Tienet: text-image embedding network for common thorax disease classification and reporting in chest x-rays, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9049–9058.
- [10] M.M.A. Monshi, J. Poon, V. Chung, Convolutional neural network to detect thorax diseases from multi-view chest X-rays, in: International Conference on Neural Information Processing, Springer, 2019, pp. 148–158.
- [11] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.
- [12] S. Calderon-Ramirez, et al., Correcting Data Imbalance for Semi-supervised Covid-19 Detection Using X-Ray Chest Images, 2020 *arXiv preprint arXiv:2008.08496*.
- [13] L. Taylor, G. Nitschke, Improving Deep Learning Using Generic Data Augmentation, 2017 *arXiv preprint arXiv:1708.06020*.
- [14] L.F. Rodrigues, M.C. Naldi, J.F. Mari, Comparing convolutional neural networks and preprocessing techniques for HEp-2 cell classification in immunofluorescence images, Comput. Biol. Med. 116 (2020) 103542.
- [15] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: In Proceedings of the 25th ACM

- SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2623–2631.
- [16] M. Nishio, K. Fujimoto, K. Togashi, Lung segmentation on chest X-ray images in patients with severe abnormal findings using deep learning, *Int. J. Imag. Syst. Technol.* (2020). ISSN: 0899-9457.
- [17] J. Bridge, Y. Meng, Y. Zhao, Y. Du, M. Zhao, R. Sun, Y. Zheng, Introducing the GEV activation function for highly unbalanced data to develop COVID-19 diagnostic models, *IEEE J. Biomed. Health Inform.* 24 (10) (2020) 2776–2786.
- [18] Y. Peng, Y.-X. Tang, S. Lee, Y. Zhu, R.M. Summers, Z. Lu, COVID-19-CT-CXR: a Freely Accessible and Weakly Labeled Chest X-Ray and CT Image Collection on COVID-19 from Biomedical Literature, 2020 *arXiv preprint arXiv:2006.06177*.
- [19] J.P. Cohen, P. Morrison, L. Dao, K. Roth, T.Q. Duong, M. Ghassemi, COVID-19 Image Data Collection: Prospective Predictions Are the Future, 2020 *arXiv preprint arXiv:2006.11988*.
- [20] L. Wang, Z.Q. Lin, A. Wong, Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, *Sci. Rep.* 10 (1) (2020) 1–12.
- [21] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar, Covidx-net: A Framework of Deep Learning Classifiers to Diagnose Covid-19 in X-Ray Images, 2020 *arXiv preprint arXiv:2003.11055*.
- [22] T. Mahmud, M.A. Rahman, S.A. Fattah, CovXNet: a multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization, *Comput. Biol. Med.* 122 (2020) 103869.
- [23] A. Narin, C. Kaya, Z. Pamuk, Automatic Detection of Coronavirus Disease (Covid-19) Using X-Ray Images and Deep Convolutional Neural Networks, 2020 *arXiv preprint arXiv:2003.10849*.
- [24] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya, Automated detection of COVID-19 cases using deep neural networks with X-ray images, *Comput. Biol. Med.* (2020) 103792.
- [25] M. Nishio, S. Noguchi, H. Matsuo, T. Murakami, Automatic classification between COVID-19 pneumonia, non-COVID-19 pneumonia, and the healthy on chest X-ray image: combination of data augmentation methods, *Sci. Rep.* 10 (1) (2020) 1–6.
- [26] I.D. Apostolopoulos, T.A. Mpesiana, Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks, *Phys. Eng. Sci. Med.* 1 (2020).
- [27] P.K. Sethy, S.K. Behera, Detection of coronavirus disease (covid-19) based on deep features, *Preprints 2020030300* (2020) 2020.
- [28] P. Mooney, Chest X-Ray Images (Pneumonia), tanggal akses, 2018. *Online*, <http://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>.
- [29] Z. Wang, Y. Xiao, Y. Li, J. Zhang, F. Lu, M. Hou, X. Liu, Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays, *Pattern Recogn.* (2020) 107613.
- [30] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164.
- [31] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger,, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
- [32] A. Wong, M.J. Shafiee, B. Chwyl, F. Li, Ferminets: Learning Generative Machines to Generate Efficient Neural Networks via Generative Synthesis, 2018 *arXiv preprint arXiv:1809.05989*.
- [33] Y. Oh, S. Park, J.C. Ye, Deep learning COVID-19 features on CXR using limited training data sets, *IEEE Trans. Med. Imag.* (1–1) (2020).
- [34] A.G. Howard, et al., Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017 *arXiv preprint arXiv:1704.04861*.
- [35] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 *arXiv preprint arXiv:1409.1556*.
- [36] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, *J. Mach. Learn. Res.* 13 (2) (2012).
- [37] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: beyond Empirical Risk Minimization, 2017 *arXiv preprint arXiv:1710.09412*.
- [38] M. Tan, Q.V. Le, “Efficientnet, Rethinking Model Scaling for Convolutional Neural Networks, 2019 *arXiv preprint arXiv:1905.11946*.
- [39] D.P. Kingma, J. Ba, “Adam, A Method for Stochastic Optimization, 2014 *arXiv preprint arXiv:1412.6980*.
- [40] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, R.M. Summers, Chestx-ray8: hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2097–2106.
- [41] J. de Moura, et al., Deep Convolutional Approaches for the Analysis of Covid-19 Using Chest X-Ray Images from Portable Devices, *medRxiv*, 2020.
- [42] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: *AAAI* (2020) 13 001–13 008.
- [43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
- [45] L.N. Smith, “A Disciplined Approach to Neural Network Hyper-Parameters: Part 1-learning Rate, Batch Size, Momentum, and Weight Decay, 2018 *arXiv preprint arXiv:1803.09820*.
- [46] J. Howard, S. Ruder, Universal Language Model Fine-Tuning for Text Classification, 2018 *arXiv preprint arXiv:1801.06146*.
- [47] M. Grandini, E. Bagli, G. Visani, Metrics for Multi-Class Classification: an Overview, 2020 *arXiv preprint arXiv:2008.05756*.
- [48] T. Fawcett, An introduction to ROC analysis, *Pattern Recogn. Lett.* 27 (8) (2006) 861–874.
- [49] J. Gorodkin, Comparing two K-category assignments by a K-category correlation coefficient, *Comput. Biol. Chem.* 28 (5–6) (2004) 367–374.
- [50] N. Ketkar, Introduction to pytorch, in: In Deep Learning with python, Springer, 2017, pp. 195–208.
- [51] J. Howard, S. Gugger, “Fastai, A layered API for deep learning, *Information* 11 (2) (2020) 108.
- [52] E.D. Cubuk, B. Zoph, J. Shlens, Q.V. Le, “RandAugment, Practical automated data augmentation with a reduced search space, in: In Proceedings Of the IEEE/CVF Conference On Computer Vision And Pattern Recognition Workshops, 2020, pp. 702–703.
- [53] S. Lim, I. Kim, T. Kim, C. Kim, S. Kim, Fast Autoaugment, 2019 *arXiv preprint arXiv:1905.00397*.
- [54] W. Wang, D. Tran, M. Feiszli, What makes training multi-modal classification networks hard?, in: In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12695–12705.