Pneumonia and COVID-19 Classification in Chest X-rays Using Faster Region-Based Convolutional Neural Networks (Faster R-CNN)

Hanan J Farhat (✉ hananfarhat@hotmail.com)

University of Saint Joesph in Beirut <https://orcid.org/0000-0002-0781-1151>

Georges E. Sakr

Virgil Systems Inc., Canada

Rima Kilany

University of Saint Joesph in Beirut

Tarek Smayra

Hotel-Dieu Hospital, Saint Joseph University

Iyad Mallak

Hotel-Dieu Hospital, Saint Joseph University

Research Article

Keywords: COVID-19, Faster R-CNN, Chest X-ray, Deep Learning , Medical Imaging

Posted Date: January 27th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1285679/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Pneumonia and COVID-19 Classification in Chest X-rays Using Faster Region-Based Convolutional Neural Networks (Faster R-CNN)

Hanan Farhat^{a,**}, George E. Sakr^{b,1}, Rima Kilany^{a,2}, Tarek Smayra^{c,3}, Iyad Mallak^{c,4}

^aSaint Joseph University of Beirut, Mar Roukos, Beirut, Lebanon

^bVirgil Systems Inc., Canada

^cRadiology Department, School of Medicine, Hotel-Dieu Hospital, Saint Joseph University, Beirut, Lebanon.

ARTICLE INFO

Article history:

Funding: This project has been funded with the joint support from the National Council for Scientific Research in Lebanon and the St. Joseph University of Beirut.

Keywords: COVID-19, Chest X-rays, Deep learning, Medical Imaging, Faster-RCNN

ABSTRACT

Chest X-rays have been a subject of deep learning applications on medical images for the high expectations of their evidence in assessing pulmonary diseases. Whereas the arising of COVID-19 or 2019 novel coronavirus in December 2019 have prioritized research on pulmonary diseases diagnosis and prognosis, especially using artificial intelligence (AI) and Deep Learning (DL). For this sake, we extend the work on detecting pneumonia using Faster Region-Based Convolutional Neural Networks (Faster R-CNN) by applying Faster R-CNN to the detection of Pneumonia and COVID-19 in Chest X-ray images using several datasets involving COVID-19 images. Different combinations of training scenarios in addition to internal and external testing at different objectiveness thresholds, epochs counts, and lengths yielded variant results. Our results comply with the state of the art of Faster-RCNN in pneumonia detection but do not show promising results in COVID-19 detection as a standalone model. Future work may emphasize introducing segmentation to the model pipeline, adding a secondary classification stage, and even engaging other medical data to improve the performance and constitute a robust Faster R-CNN-based prediction model.

© 2022

1. Introduction

Coronavirus disease (COVID-19) is an infectious disease that was first detected in China - Wuhan in December 2019 and is caused by a newly discovered coronavirus. Most infected people recover without special treatment but people with underlying medical problems and the elderly are more likely to develop serious illness[1]. COVID-19 cases show several symptoms including dry cough, fever, shortness of breath, weakness, and others. In hospitalized cases, 75% of the complications include pneumonia [2].

Coronavirus and a variety of other organisms -including bacteria and fungi- can cause pneumonia infection, which is an inflammation in the air sacs of a lung or of both lungs. It can range

from mild to life-threatening especially for infants, people with health problems, and the elderly [3].

Medical imaging -of different types- is used to help medical doctors assess pulmonary diseases. For example, Ultrasound is used for the diagnosis of pediatric pulmonary diseases and is reviewed by Heuvelings et al. [4]. Moreover, computed tomographies (CT) are important for prognosis and diagnosis of lung cancer that they became a subject of automatic detection research reviewed by Zhang et al. [5]. However, chest X-rays (CXR) are widely available, effective, emit less radiations than other medical images, and are affordable, which makes them a reasonable target for automated Artificial Intelligence (AI) assistants in cases of viral diseases such as COVID-19.

AI researchers' response to the coronavirus pandemic was immediate as they applied their experiences and methodologies to assist in combating the virus by speeding up the diagnosis during its spread [6]. And as deep learning (DL) is the most famous form of AI[7], it had its share on many levels, including pulmonary medical imaging [8].

Faster Regional Convolutional Neural Networks is an object

*Corresponding author: hanan.farhat@net.usj.edu.lb

**ORCID: <https://orcid.org/0000-0002-0781-1151>

¹email:gsakr@virgilsystems.com

²email:rima.kilany@usj.edu.lb

³email:smayra@icloud.com

⁴email:iyad.mallak@gmail.com

detection model that includes a region proposal network (RPN) which shares the full-image convolutional features with a detection network. It performed outstandingly in several competitions, and thus its use was expanded to many different fields [9].

In this paper, the work done by Ismail et al. [10] who applied Faster Regional Convolutional Neural Networks (Faster R-CNN) to CXR classification, was extended using different datasets and adding COVID-19 pneumonia to some of the experiments scenarios. Different combinations of subsets of Chestxray14[11], RSNA Pneumonia[12], CheXpert[13], the COVID-19 image data collection by Dr. Joseph Paul Cohen et al[14], and the dataset of SIIM-FISABIO-RSNA COVID-19 Detection Kaggle Competition[15] were used. To implement the model, existing python libraries were used, and the evaluation of the model's performance was done by calculating and analyzing the accuracies, precision, sensitivity, and specificity. The assessment also involved comparison with similar work, and with medical representatives.

This paper is organized as follows. Section 2 describes Faster R-CNN architecture and paradigm in the light of deep learning and convolutional neural networks, and highlights the related work. Section 3 describes the methodology used to carry out this work. Section 4 displays and discusses the results, and finally, section 5 provides the conclusion and the future work.

2. Background

This section describes briefly the architecture of Faster R-CNN model and highlights applications of the model in different domains.

2.1. Model Architecture

Our experiment is held to study the performance of Faster R-CNN algorithm in the detection of pneumonia including COVID-19 pneumonia from medical chest images, specifically chest X-rays. Faster R-CNN is an add-up of the region proposal network (RPN) and the Fast R-CNN, where Fast R-CNN [16] is an updated version of R-CNN [17].

The R-CNN inputs an image to the network, then extracts region proposals using selective search [18] which combines segmentation and exhaustive search. After that, it computes the features for each proposal using transfer learning and CNN, then uses a support vector machine (SVM) to classify each of the proposals.

Fast R-CNN aimed to speed up the detection process by disregarding selective search. In fact, the input image is associated with ground-truth bounding boxes. Feature maps are extracted and region-of-interest (ROI) pooling is applied, by which the ROI feature vector is obtained. After that, two sets of fully convolutional layers (FC) are used to output through softmax layer the classification results, and through a regression layer the localizing bounding boxes of predicted proposals.

Yet, the Fast R-CNN still consumed too much time for region proposals, and thus, Faster R-CNN was proposed. It mainly incorporates region proposals into Fast R-CNN such that a preliminary convolutional neural network produces a feature map,

an RPN proposes regions of interest and their corresponding scores based on the produced feature map, and finally, the region proposals along with the feature map are forwarded to an ROI pooling layer and fully connected layers respectively. The output is in the form of bounding boxes coordinates (localization of target) and classes to which the images belong. This criterion shares convolutional features of improved quality with a detection network, nearly diminishes the ROI generation time, and improves the frame rates to become almost real-time.

Being introduced in 2015 by Ren et al. [9], Faster R-CNN have achieved state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012 and MS COCO datasets. Moreover, it earned the first-place winning entries in several tracks in ImageNet Large Scale Visual Recognition Challenge 2015 (ILSVRC) localization competition and COCO 2015 segmentation competitions. Thus, it became an interesting model for a variety of application fields.

2.2. Related Work

The promising algorithm usages can be categorized at different related levels: medical/non-medical usage, and pneumonia/covid-19 detection.

2.2.1. Non-Medical Usage

Faster R-CNN has been experimented for many non-medical objectives, like the detection of objects in the wild by Chen et al. [19], the detection of faces [20], that of facial expressions [21], and even fire smoke detection in the wild-land forest [22].

2.2.2. Non-Medical Usage: COVID-19 emphasis

For COVID-19 pandemic resistance, it was used on many levels other than direct disease diagnosis. For instance, it was experimented on person detection for monitoring social distancing by Pun et al. [23]. Faster R-CNN algorithm was also tested by Yadav [24] on face mask detection to assure the safety guidelines are being preserved. In an advanced approach, Faster R-CNN was tested by Wu et al. [25] for vehicle detection used to fulfill the transportation ban inside Wuhan where COVID-19 first occurred.

2.2.3. Medical Usage

Faster R-CNN was used in object detection from malaria images by Hung and Carpenter [26]. Another example of its usage was for liver segmentation by Tang et al. [27], and also for heart segmentation by Xu et al. [28]. Moreover, it was used for brain tumors detection by Ezhilarasi and Varalakshmi [29].

2.2.4. Medical-Images Usage: Pneumonia/COVID-19 emphasis

For direct diagnosis of pulmonary diseases using medical images, an approach using faster R-CNN was done by Ismail et al. [10] for chest X-rays classification, achieving 62% accuracy. Besides, Yao et al. [30] implemented a model that is based on Faster-RCNN model with the integration of several other components that were proved to improve the performance of the algorithm, achieving 38.02% and 39.23% mean Average Precision (mAP) using ChestX-ray14[11] dataset and that of the Radiological Society of North America (RSNA)[12] respectively.

Emphasizing on covid-19 detection, Tahir et al. [31] use Faster-RCNN to forecast covid-19 through chest X-rays and registration slips of admitted patients and achieves 87% mean Average Precision (mAP). Another approach is by Podder et al. [32] who use Mask-RCNN rather than Faster-RCNN to detect COVID-19 through frontal views of chest X-rays achieving 96.98% accuracy. Shibly et al. [33] developed COVID faster R-CNN that use chest X-rays and achieves outstanding results: 97.36% accuracy, 97.65% sensitivity, and precision of 99.28%.

Another approach targets tuberculosis, pneumonia, and covid-19 detection all together by Mahajan et al. [34] using Faster-RCNN for lung isolation and then DenseNet for classification, followed by fine-tuning, classifying by a multi-level perceptron structure, then a calibration.

3. Methodology

This section describes the methodology used to prepare, implement, and test Faster R-CNN on pneumonia and Covid-19 classification using chest X-rays.

3.1. Data Preparation

In order to experiment Faster R-CNN, the chest X-rays were imported, pre-processed and categorized into relevant training, and testing datasets. Datasets were created using different combinations of four publicly available datasets:

- Dataset of RSNA Pneumonia Detection Challenge (2018) [35]
- Chestxray14 dataset [11]
- CheXpert dataset [13]
- COVID-19 image data collection by Cohen et al. [14]
- SIIM-FISABIO-RSNA COVID-19 Detection Kaggle Competition [15]

RSNA pneumonia dataset constitutes of 26984 unique chest X-rays in DICOM format, from which 3000 X-rays are allocated for testing, while the rest are labelled for training. Training images labels include whether the image is diseased (pneumonic) or not, and corresponding bounding boxes that evidence the label. Out of the training images, 9555 are classified as diseased. As for Chestxray14 dataset, it contains 112120 unique chest X-rays of PNG format. However, the labeling covers only 880 diseased images. Fourteen different diseases labels are used, one of which is the main concern; pneumonia. The third dataset is CheXpert, which is made up of 223,414 chest radiographs of 65,240 patients. The images are of JPG format, and are labeled according to 14 classes (No findings in addition to 13 findings types). The disadvantage of this dataset is that it lacks bounding boxes localizing the findings. The final dataset used is a subset of an earlier version of Cohen et al. [14], which is a collection in progress, gathering chest X-rays covering 20 different findings including pneumonia and covid-19. This dataset contains images with different types, sizes and qualities. Finally, the dataset of SIIM-FISABIO-RSNA

Fig. 1. Visualization of a resized Chest X-ray from CheXpert dataset [13]

COVID-19 Detection Kaggle Competition [15] is divided into two parts: a training dataset that consists of 6,334 chest scans in DICOM format, and a hidden dataset that is roughly of the training dataset scale. The images are organized according to study level and image level. At the study level, the class of the case is identified as Negative for Pneumonia, Typical COVID-19 appearance, Indeterminate appearance, and Atypical COVID-19 appearance. Whereas at the image level, the bounding boxes for the detected opacities are given along with the correct prediction label for the provided bounding box. The dataset was available also in other formats by the competitors.

Images count: The total number of CXRs used in each dataset was highly dependant on the initial count of available images in public datasets, and sometimes on the availability of the necessary annotations (specifically coordinates of bounding boxes locating symptoms of pneumonia) at the time of performing this experiment.

Images type and size: In order to unify the images types of the built datasets, DICOM images of RSNA dataset were converted to PNG images using dcm2pnm [?] utility of DCMTK OFFIS DICOM Toolkit, which converts DICOM images to PGM/PPM, PNG, TIFF or BMP. Moreover, PIL(Python Imaging Library) or pillow library, which is a python library, was used to convert JPG images to PNG type, and to resize all to best fit 1024x1024 pixels keeping the initial ratio of image, and filling the remaining space with black color if exists. An example of resized image from CheXpert dataset is in Figure 1.

COVID-19 images labels: In the case of COVID-19 images from [14], the images of better resolution fit with the criterion were picked. However, the annotations were missing, so an expert in Radiology located the symptoms of pneumonia caused by COVID-19. The expert was provided with VGG Image

Table 1. Chest X-rays count and sources for training datasets (Tr+ means the training set is additive)

	Dataset	Normal	Pneumonic	Tr +
Tr 1	RSNA	8800	9555	
Tr 2	ChestXray14	100	100	Tr 1
Tr 3	Covid-19 collection		38	Tr 1
	CheXpert	38		
Tr 4	Covid-19 collection		38	Tr 2
	CheXpert	38		
Tr 5	Kaggle	1822	4172	

Annotator Version 3 (VIA3) [36] and 97 CXRs (the available CXRs at that time), of which 37 are of lower priority, because their sizes are smaller than the rest of the images. The expert disregarded the CXRs of unacceptable resolution for diagnosis, and annotated the 47 kept images. Throughout annotation, each CXR would possibly contain more than one located region of interest, which gets denoted by "alveolar opacity", that might be caused by pneumonia or other pulmonary diseases, and is not pneumonia itself. The expert clarified that an alveolar opacity has undefined contours and can rarely be included in a square shape, that's why the regions of interest were traced emphasizing on the highest densities rather than the whole opacity area.

Training datasets: For training the algorithm, 5 datasets were prepared as shown in table 1. The first training dataset (Tr 1) was composed of 8800 CXRs of normal people, and 9555 of people diagnosed with pneumonia from RSNA dataset, while the second training dataset (Tr 2) was composed of the first training dataset in addition to 200 CXRs from ChestXray14 dataset, where half of them was for pneumonic people and the rest are for normal ones. The first dataset acted as the gold standard, whereas the second dataset was created to train the algorithm with CXRs from hybrid sources with different qualities. The third and fourth training datasets (Tr 3 and Tr 4) were both made up of 38 CXRs of normal people from CheXpert dataset and 38 CXRs of people diagnosed with COVID-19 from aforementioned COVID-19 collection. However, Tr 3 combined its CXRs with Tr 1, while Tr 4 added up CXRs of Tr 2. The last training dataset denoted by Tr 5 was made up of 1822 normal and 4172 COVID-19 pneumonic chest X-rays consisting of the three labels: atypical, typical and indeterminate.

Testing datasets: As for testing datasets, five datasets were constructed, and are described in table 3. The first testing dataset (Te 1) is composed of 100 CXRs from RSNA dataset, where half of them correspond to pneumonic people. The second testing dataset (Te 2) adds up to Te 1 a total of 40 CXRs from ChestXray14 dataset, of which 20 CXRs are labelled as pneumonic. Thirdly, Te 3 constitutes of 100 CheXpert CXRs, with half of them classified as pneumonic. The fourth testing dataset (Te 4) is composed of 6 normal CXRs and 6 pneumonic ones from CheXpert dataset, in addition to 6 COVID-19 labelled CXRs from COVID-19 collection. Finally, the fifth dataset is made up of 102 normal images and 198 COVID-19 labeled as typical, atypical and indeterminate.

Table 2. Chest X-rays count and sources for testing datasets (Te+ means that the testing set is additive)

	Dataset	Normal	Pneumonic	Te +
Te 1	RSNA	50	50	
Te 2	ChestXray14	20	20	Te 1
Te 3	CheXpert	50	50	
Te 4	Covid-19 collection		6	
	CheXpert	6	6	
Te 5	Kaggle	102	198	

Table 3. Training and Testing experiments

Scenario	Tr	Te	
1	Tr 1	Te 1	Internal Testing
2	Tr 2	Te 2	
3	Tr 1	Te 3	External Testing
4	Tr 2	Te 3	
5	Tr 3	Te 4	Internal Testing
6	Tr 4	Te 4	
7	Tr 5	Te 5	Internal Testing

3.2. Model Building and Training

The model was reproduced using Python language, based on the publicly available keras implementation of Faster R-CNN [37]. For the training process, Anaconda virtual environment is used given the requirements of the implementation installed, which are Keras, tensorflow, open-cv, sklearn, H5py and Numpy. The experiments were executed with RTX 2080Ti GPU and 64GB RAM at 1e-5 learning rate. The anchor box scales were defined as 64, 128 and 256. Besides, the anchor ratios were [1,1], [1,2] and [2,1]. For region proposal network, the minimum overlap used was 0.3 with maximum reaching 0.7, while minimum and maximum overlap values for the classifier were 0.1 and 0.5 respectively. The weights of the hyper-parameters of the model were updated only when loss was less than previous loss value.

The Faster R-CNN model trained is VGG16-based as in [9], and is trained in many scenarios described in table 4. Five training scenarios (Tr 1, Tr 2, Tr 3, Tr 4, Tr 5) were obtained by changing the training dataset used. The whole model initial hyper-parameters were inherited from the pre-trained model hyper-parameters using ImageNet dataset [38].

The experiment was held in two parts (refer to table 4):

1. For scenarios 1-6, the total count of epochs is 50, given 250 iterations per epoch
2. For scenarios 1-8, epochs count varies (50, 250, and 400), and epoch length varies (250, 100, 16)

3.3. Model Testing and Validation

For testing and evaluation, the scenarios performed are described in table 3, and detailed in table 4. The first four scenarios were by adopting the first two training scenarios and

testing each against two testing datasets: the first was an internal dataset that constituted of images originating from the same training dataset, and the other one was an external dataset that constituted of images from different dataset than that of the training scenario. As for the following two testing scenarios, they internally test -using Te4- the algorithm trained by the third and fourth training datasets each at a time. The last two scenarios adopt the fifth training dataset and the same dataset augmented respectively, both tested internally.

Then, the accuracy, precision, sensitivity and specificity were calculated for testing results given that TP is true positives, FP is false positives, TN is true negatives, and FN is false negatives, Precision was calculated by:

$$\frac{TP}{FP + TP}$$

Sensitivity was calculated by:

$$\frac{TP}{FN + TP}$$

Specificity was calculated by:

$$\frac{TN}{TN + FP}$$

And accuracy was calculated by:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

The testing was done in two parts with respect to the training process as follows:

1. Test using several objectiveness thresholds (0.7, 0.75, 0.8), and compare to a senior trainee in radiology for training part 1
2. Test using a fixed objectiveness threshold equal to 0.75 for training part 2

where the objectness threshold "measures the membership to a set of object classes verses the background" [9].

Finally, the experiments' performance was compared to that of state-of-the-art.

4. Results and Discussion

As two experiments were held, one targeting different objectiveness thresholds, and another one emphasizing on epochs length and count, analysis will be held in two steps. Furthermore, it is important to differentiate between the internal and external testing when analyzing the testing results of the trained models with different training datasets. The internal dataset - used to test the model- is composed of images from the same nature and source of the training dataset used, while the external dataset is composed of images from a source that is different from that of the training datasets. The difference in the source might also mean a difference in the image quality as well as a difference due to the genetic behavior of different nations.

4.1. Experiment with Fixed Epochs Count and Epochs Length

For the first experiment, 50 epochs were executed, each of 250 epoch length at different objectiveness thresholds: 0.7, 0.75 and 0.8. The results are displayed in table 5. Scenarios 1-6 are detailed in table 4.

4.1.1. Performance between Internal and External Testing

At 0.7 objectness threshold, testing against datasets internally, the accuracy of the model of the second scenario that is based on images from two sources (RSNA and Chest X-Ray14), with a larger count of images, performed better than that of the first scenario based on images from one single source (RSNA). When tested against external datasets (CheXpert), both third and fourth scenarios recorded less accuracy values than the first and second scenarios. However, the training dataset with the unique source performed better than the dual-sourced training dataset, against the internal testing case.

The fifth and sixth testing scenarios use Te4 that is composed of images from CheXpert and Covid-19 collection, to evaluate the algorithm trained by Tr3 and Tr4 consisting of images from CheXpert and Covid-19 collection, though adding Tr1 to Tr3, and Tr2 to Tr4. Both experiments are considered internal testing, and perform worse than all four previous scenarios, with scenario 6 performing better than that of scenario 5. This could be caused by the small count of COVID-labeled images provided and that of testing images.

Precision, sensitivity and specificity are calculated only for binary datasets which exclude the fifth and sixth scenarios. The same analysis applies to precision. Yet, with respect to sensitivity, the second scenario performed better than the first scenario. On top, scenarios 3 and 4 performed the best, while the third scenario beat the fourth scenario. For specificity, second scenario came first, followed by the first scenario (27% and 25% respectively), and finally third and fourth scenarios (4% each).

In general, as objectness threshold increased, the accuracy increased, the precision slightly increased, the specificity noticeably increased, and the sensitivity decreased.

Therefore, the second scenario performed better than the first scenario, which is explained by the increased count of images, and although it is not a very evident increase (only 1%), the higher contrast of the added images implied the performance improvement. However, that was on the level of internal testing. Moving to external testing, the model trained by a unique-sourced dataset was found to be more robust (the third scenario is better than the fourth). Adding covid-19 images, and similar to the internal testing in the first two scenarios, the sixth scenario performed better than that of the fifth scenario.

In comparison to a health representative's performance, the trainee was asked to classify the chest X-rays of the testing datasets without prior knowledge of their diagnosis. It is noticeable that the trainee's accuracy is higher than the model's best accuracy at each scenario by an average of 26%. The sensitivity of the trainee's performance decreases significantly at scenarios 3 and 4, while the specificity notably decreases at 5th and 6th scenarios containing COVID-19 images. Yet, the model is much faster given that the run time for scenario 8 testing of 300 images is 03:06.599128 minutes.

Table 4. Training and Testing Scenarios with images counts and sources

Experiment		Training Data		Testing Data		Testing scope
1	Tr 1	RSNA	Te 1	RSNA	Internal Testing	
		18355		100		
2	Tr 2	ChestXray14 + RSNA	Te 2	ChestXray14 + RSNA	Internal Testing	
		18355 + 200		40 + 100		
3	Tr 1	RSNA	Te 3	CheXpert	External Testing	
		18355		100		
4	Tr 2	ChestXray14 + RSNA	Te 3	CheXpert	External Testing	
		18355 + 200		100		
5	Tr 3	Covid-19 + CheXpert + Tr 1	Te 4	Covid-19 + CheXpert	Internal Testing	
		38 + 38 + 18355		6 + 12		
6	Tr 4	Covid-19 + CheXpert + Tr 2	Te 4	Covid-19 + CheXpert	Internal Testing	
		38 + 38 + 18555		6 + 12		
7	Tr 5	Kaggle	Te 5	Kaggle	Internal Testing	
		1822		300		
8	Tr 5	Kaggle Augmented	Te 5	Kaggle	Internal Testing	
		5994		300		

Table 5. Testing results at 50 epochs and 250 epoch length compared to healthcare representative

Objectness Threshold	Accuracy			Precision			Sensitivity			Specificity		
	0.7	0.75	0.8	0.7	0.75	0.8	0.7	0.75	0.8	0.7	0.75	0.8
1	0.54	0.636364	0.656566	0.518987	0.597015	0.653061	0.836735	0.816327	0.653061	0.254902	0.46	0.66
Trainee		0.933824			0.987179			0.905882			0.980392	
2	0.58	0.63	0.676768	0.5625	0.608696	0.672727	0.865385	0.807692	0.72549	0.270833	0.4375	0.625
Trainee		0.903409			0.92233			0.913462			0.888889	
3	0.51	0.49	0.58	0.505155	0.494382	0.555556	0.98	0.88	0.8	0.04	0.1	0.36
4	0.434343	0.494949	0.454545	0.465909	0.5	0.461538	0.82	0.74	0.48	0.040816	0.244898	0.428571
Trainee		0.68			0.95			0.38			0.98	
5	0.3333	0.178	0.1777	-	-	-	-	-	-	-	-	-
6	0.3333	0.29	0.4	-	-	-	-	-	-	-	-	-
Trainee		0.833333			0.888889			0.888889			0.666667	

4.1.2. Comparing to Related Work

Comparing to similar previous work, it is possible to infer that all the results comply with each other. Table 6 states the values of accuracies, specificity, sensitivity and precision of two similar experiments published by Ismail et al. [10] and Shibly et al. [33], in addition to this model's results where the values are calculated by averaging the highest ones in table 5 with equal weights for all experiments.

In the work done by Ismail et al. [10], they perform the experiment on 200 images from ChestXray14 dataset, and classify them into pathological and normal images, which means that this experiment is an expansion of theirs on image count level, and focusing on pneumonia and pneumonia caused by COVID-19 virus rather than general pathologies. However, this models' performance is within a difference ranging from 3% to 13% considering accuracy, precision, specificity and sensitivity. Especially that the first and second scenarios that include pneumonia and normal-labeled images as in the aforementioned

work, almost have the same accuracy (63%) with that of Ismail et al. [10].

On another level, Shibly et al. [33] results are relatively high with respect to our model. Accuracy, sensitivity and precision values are 97.36%, 97.65% and 99.28% respectively, using two datasets containing images labeled as normal, Covid-19 and non-Covid pneumonia, with a count of 13800 images and 5370 images for the first and second datasets respectively. Their evaluation is performed on 10-fold validation technique. This could be due to the difference in training dataset, which is almost triple the size of dataset used in scenario 8, and may also be due to the difference in images quality and labeling certainty.

4.2. Experiment with Fixed Objectiveness Threshold

Fixing the objectiveness threshold at 0.75, comparison was done at two levels: epochs count (table 7) and epochs length (table 8).

Fixing epochs length at 250 as suggested by the initial Code developers, epochs count was varied between 50, 250 and 400,

Table 6. Comparison of the unweighted average performance of table 5 to that of work done by Ismail et al. [10] and Shibly et al. [33]

	Ismail et al. [10]	Shibly et al. [33]	Un-weighted averages of results:		
Objectness Threshold	Unspecified	Unspecified	0.7	0.75	0.8
Accuracy	62%	97.36%	45.41%	45.321%	49.1%
Sensitivity	72%	97.65%	87.553%	81.10%	66.46%
Specificity	54.39%	Unspecified	15.16%	25.54%	51.839%
Precision	54.39%	99.28%	51.3116%	55%	58.57%

Table 7. Testing results using a fixed objectiveness threshold equal to 0.75 and 400 epochs for training scenarios using different epochs count and different epoch lengths

Scenario	Epoch length	Accuracy
1	250	0.51
	100	0.65
	16	0.72
2	250	0.3214
	100	0.3357
	16	0.3928
3	250	0.51
	100	0.57
	16	0.51
4	250	0.45
	100	0.47
	16	0.43
5	250	0.3333
	100	0.4444
	16	0.3333
6	250	0.2222
	100	0.2777
	16	0.27777
7	250	0.3266
	100	0.29333
	16	0.29333

Table 8. Testing results using a fixed objectiveness threshold equal to 0.75 and epoch length equal to 250 for training scenarios using different epochs count and different epoch lengths

Scenario	Epochs count	Accuracy
1	50	0.73
	250	0.78
	400	0.51
2	50	0.2214
	250	0.7642
	400	0.3214
3	50	0.51
	250	0.48
	400	0.51
4	50	0.46
	250	0.46
	400	0.45
5	50	0.1666
	250	0.25
	400	0.3333
6	50	0.2777
	250	0.1666
	400	0.2222
7	50	0.29333
	250	0.31333
	400	0.3266
8	250	0.29333

and results are displayed in table 7. It is noticeable that at epoch count 250, accuracies are either the best or within a 8% range at maximum. The first and second scenarios performed best of all scenarios at 250 epochs count (78% and 76.42% respectively). Testing on external datasets in scenarios 3 and 4 yielded a 30% drop in accuracy. Introducing COVID-19 images increased the drop to 25% and 16% in scenarios 5 and 6, while Kaggle dataset introduction resulted with a slightly better accuracy (31%). Augmenting the images of the Kaggle dataset did not improve the accuracies as shown in the eighth scenario.

Fixing epochs count at 400, epoch length was varied between the following three values: 250, 100, and 16, and results are displayed in table 8. For the first and second scenarios, the accuracy was best at epoch length equal to 16, but the first scenario resulted with better accuracies, suggesting that the addition of the RSNA dataset created a distortion to the model instead of fine-tuning it. However, maybe the epochs count was too much for the second scenario such that the best model wasn't found. In external testing, scenarios 3 and 4 were best at epoch length equal to 100 with a slight variation when the epoch length is increased or decreased. Introducing COVID-19 images, scenarios 5 and 6 accuracies drop in comparison with the previous scenarios, similar to the case of Kaggle dataset in scenario 7 which constitutes of a larger number of images of 4-label classes (Typical, Atypical, Normal, Indeterminate).

Summing all the epochs Count/epochs lengths combinations, we conclude that the combination of 250 epochs and 250 epoch length makes the model at its best. The maximum accuracy achieved is 78% in the first scenario, emphasizing on pneumonia only, and testing on an internal dataset. When applying external testing, the accuracies are no longer acceptable, which disregards the generalizability of the model. Neither applying the model on COVID-19 images using small datasets nor using large ones is evident enough to train the model as should.

5. Conclusion and Future Work

Like our work -in scenarios 1 and 2- and Ismail et al. [10]'s work show close performance, especially with a difference in dataset sources and combinations, images counts, classes, testing strategies(internal and external), and objectness thresholds, it is the time for the questions: what matters the most? Is it to detect as much pneumonic cases as possible? or the main concern is for the trueness of the pneumonic detections? or for that of the normal detections? The emphasis would be on the sensitivity, the precision, or the specificity respectively for each of the aforementioned questions. If more pneumonic cases are to-be predicted, the 70% objectness threshold would be recommended. On another side, 80% objectness threshold would be better if the trueness of either pneumonic or normal detections. For all cases, it is recommended to use multi-sourced training dataset when testing is internal, and one-sourced training dataset for external testing.

Different approaches used Faster-RCNN as a part of a model, such as [34] who used it for lung isolation before the classification process. Also [31] forecast-ed COVID-19 applying Faster-RCNN and other models but included registration slips

along with the chest X-rays. [30] adopted Faster-RCNN but integrated it with another networks and models to improve its performance achieving mAP of 39.23%. A high accuracy of 96.98% was recorded by [32] who adopted Mask-RCNN rather than Faster-RCNN to detect COVID-19 from chest X-rays.

Based on our work, using Faster-RCNN as a standalone model for classification of Pneumonia from chest X-rays is promising. On the other side, it is not efficient for classification of COVID-19 pneumonia. Nevertheless, integrating it with other models, adopting Mask-RCNN, introducing segmentation, or adding patient input beside the medical images could assist in producing better results in COVID-19 and Pneumonia forecasting.

6. Declarations

6.1. Funding

This project has been funded with the joint support from the National Council for Scientific Research in Lebanon and the St. Joseph University of Beirut.

6.2. Conflicts of Interest

Not applicable

6.3. Availability of Data

Not applicable

6.4. Code Availability

Not applicable

References

1. Organization, W.H.. Coronavirus. ??? URL: https://www.who.int/health-topics/coronavirus#tab=tab_1.
2. Wiersinga, W.J., Rhodes, A., Cheng, A.C., Peacock, S.J., Prescott, H.C.. Pathophysiology, transmission, diagnosis, and treatment of coronavirus disease 2019 (covid-19): a review. *Jama* 2020;.
3. MayoClinic, . Pneumonia. 2020. URL: <https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>.
4. Heuvelings, C.C., Bélar, S., Familusi, M.A., Spijker, R., Grobusch, M.P., Zar, H.J.. Chest ultrasound for the diagnosis of paediatric pulmonary diseases: a systematic review and meta-analysis of diagnostic test accuracy. *British medical bulletin* 2019;.
5. Zhang, G., Jiang, S., Yang, Z., Gong, L., Ma, X., Zhou, Z., Bao, C., Liu, Q.. Automatic nodule detection for lung cancer in ct images: A review. *Computers in biology and medicine* 2018;.
6. Vaishya, R., Javaid, M., Khan, I.H., Haleem, A.. Artificial intelligence (ai) applications for covid-19 pandemic. *Diabetes Metabolic Syndrome: Clinical Research Reviews* 2020;14(4):337 – 339. URL: <http://www.sciencedirect.com/science/article/pii/S1871402120300771>. doi:<https://doi.org/10.1016/j.dsx.2020.04.012>.
7. Hatcher, W.G., Yu, W.. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* 2018;6:24411–24432.
8. Shi, F., Wang, J., Shi, J., Wu, Z., Wang, Q., Tang, Z., He, K., Shi, Y., Shen, D.. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE Reviews in Biomedical Engineering* 2020;:1–1.
9. Ren, S., He, K., Girshick, R., Sun, J.. Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*. 2015:91–99.

10. Ismail, A., Rahmat, T., Aliman, S.. Chest x-ray image classification using faster r-cnn. *MALAYSIAN JOURNAL OF COMPUTING* 2019;4(1):225–236.
11. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE; 2017:3462–3471.
12. Rsn pneumonia detection challenge. ??? URL: <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data>.
13. Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghighi, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: *Proceedings of the AAAI Conference on Artificial Intelligence*; vol. 33. 2019:590–597.
14. Cohen, J.P., Morrison, P., Dao, L.. Covid-19 image data collection. *arXiv 200311597* 2020;URL: <https://github.com/ieee8023/covid-chestxray-dataset>.
15. Lakhani P Mongan J, E.A.. The 2021 siim-fisabio-rsna machine learning covid-19 challenge: Annotation and standard exam classification of covid-19 chest radiogra. 2021. URL: <https://osf.io/532ek/>.
16. Girshick, R.. Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*. 2015:1440–1448.
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014:580–587.
18. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.. Selective search for object recognition. *International journal of computer vision* 2013;104(2):154–171.
19. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.. Domain adaptive faster r-cnn for object detection in the wild. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018:3339–3348.
20. Jiang, H., Learned-Miller, E.. Face detection with the faster r-cnn. In: *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE; 2017:650–657.
21. Li, J., Zhang, D., Zhang, J., Zhang, J., Li, T., Xia, Y., Yan, Q., Xun, L.. Facial expression recognition with faster r-cnn. *Procedia Computer Science* 2017;107:135–140.
22. Zhang, Q.x., Lin, G.h., Zhang, Y.m., Xu, G., Wang, J.j.. Wildland forest fire smoke detection based on faster r-cnn using synthetic smoke images. *Procedia engineering* 2018;211:441–446.
23. Pun, N.S., Sonbhadra, S.K., Agarwal, S.. Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques. *arXiv preprint arXiv:200501385* 2020;.
24. Yadav, S.. Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence ???;.
25. Wu, C., Yuan, J., Ru, L., Chen, H., Du, B., Zhang, L.. A measurement of transportation ban inside wuhan on the covid-19 epidemic by vehicle detection in remote sensing imagery. *arXiv preprint arXiv:200616098* 2020;.
26. Hung, J., Carpenter, A.. Applying faster r-cnn for object detection on malaria images. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017:56–61.
27. Tang, W., Zou, D., Yang, S., Shi, J.. Dsl: Automatic liver segmentation with faster r-cnn and deeplab. In: *International Conference on Artificial Neural Networks*. Springer; 2018:137–147.
28. Xu, Z., Wu, Z., Feng, J.. Cfun: Combining faster r-cnn and u-net network for efficient whole heart segmentation. *arXiv preprint arXiv:181204914* 2018;.
29. Ezhilarsi, R., Varalakshmi, P.. Tumor detection in the brain using faster r-cnn. In: *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on*. IEEE; 2018:388–392.
30. Yao, S., Chen, Y., Tian, X., Jiang, R.. Pneumonia detection using an improved algorithm based on faster r-cnn. *Computational and Mathematical Methods in Medicine* 2021;2021.
31. Tahir, H., Iftikhar, A., Mumraiz, M.. Forecasting covid-19 via registration slips of patients using resnet-101 and performance analysis and comparison of prediction for covid-19 using faster r-cnn, mask r-cnn, and resnet-50. In: *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. 2021:1–6. doi:10.1109/ICAECT49130.2021.9392487.
32. Podder, S., Bhattacharjee, S., Roy, A.. An efficient method of detection of covid-19 using mask r-cnn on chest x-ray images. *AIMS Biophysics* 2021;8(3):281–290.
33. Shibly, K.H., Dey, S.K., Islam, M.T.U., Rahman, M.M.. Covid faster r-cnn: A novel framework to diagnose novel coronavirus disease (covid-19) in x-ray images. *medRxiv* 2020;.
34. Mahajan, K., Sharma, M., Vig, L., Khincha, R., Krishnan, S., Niranjana, A., Dash, T., Srinivasan, A., Shroff, G.. Covid diagnosis: Deep diagnosis of covid-19 patients using chest x-rays. In: *International Workshop on Thoracic Image Analysis*. Springer; 2020:61–73.
35. Shih, G., Wu, C.C., Halabi, S.S., Kohli, M.D., Prevedello, L.M., Cook, T.S., Sharma, A., Amorosa, J.K., Arteaga, V., Galperin-Aizenberg, M., et al. Augmenting the national institutes of health chest radiograph dataset with expert annotations of possible pneumonia. *Radiology: Artificial Intelligence* 2019;1(1):e180041.
36. Dutta, A., Gupta, A., Zissermann, A.. VGG image annotator (VIA). 2016. URL: <http://www.robots.ox.ac.uk/~vgg/software/via/>; version: X.Y.Z, Accessed: INSERT DATE HERE. Bardool, K. *kardool/keras - frCNN*. 2021. URL: ; original-date: 2017-10-31T20:11:10Z.
37. ImageNet. 2021. <http://image-net.org/download-images>.