

Comparative Analysis of CNN Performance on Segmented and Unsegmented COVID-19 Chest X-Rays

Candidate IDs 42903, 44568, and 50995

1. Introduction

The COVID-19 pandemic has placed an enormous strain on existing medical systems due to rapid proliferation of the disease across the globe and the pressing need to quickly and easily identify contagious individuals. The gold standard method of diagnosing COVID-19 is reverse-transcription polymerase chain reaction (RT-PCR) tests. However, certain health-care systems lack sufficient numbers of these tests, leading to reporting and containment issues. AI applications provide an alternative method of identification via chest X-ray to aid diagnosing COVID-19 in patients.

One of the rising areas in AI research is the convolutional neural network (CNN) model, which applies convolutional layers to images to extract important information about the image. Wang et al developed a CNN model to detect COVID-19 in chest X-rays (Wang et al., 2020). In their study, they find their COVID-Net model generates a high positive predictive value (PPV) for patients with COVID-19 (98.9%); however, the model could be further improved by collecting even more data to improve the PPV and further train the model. Similarly, Krizhevsky et al researched how CNN can be used to assign images to different classes (Krizhevsky et al., 2017). While they find their CNN model generates error rates that are better than other methods (37.5% for top-1, 17.0% for top-5), they identify many areas for improvement. These include leveraging different techniques to prevent overfitting, using unsupervised pre-training, and maintaining depth in the model's layers to enhance the network's performance.

While CNN architecture is useful for accepting chest X-ray images of different sizes, the pooling and dense layer steps make it challenging to extract enough characteristics to categorise the pixels and differentiate these images (Cai et al., 2020). Therefore, researchers have begun implementing segmentation to focus on relevant features and improve overall accuracy of the model. Segmentation divides an input image into a variety of components, allowing the model to delve into specific elements of the image while maintaining pertinent information, such as the pixel location (Minaee et al., 2022).

In their study on the randomised generative adversarial network (RANDGAN) to detect the presence of COVID-19 in chest X-rays, Motamed et al find that implementing segmentation helps the model focus on the chest X-ray image itself, preventing the model from being influenced by other unrelated markers (Motamed et al., 2021). Segmentation has been found effective in other applications, for example in breast cancer detection, where it improved the accuracy of the model by 7% (Mehta et al., 2018). Segmentation has recently gained an increase in usage in medical imaging research as it has been proven effective when used

with multiple diagnostic techniques, including heart ultrasounds and brain MRI to grade brain tumours (Yang et al., 2018; Shalev-Shwartz and Zhang, 2013).

While segmentation has been found highly effective in improving accuracy in these models, there are many clinical fields where segmentation has yet to be explored. Therefore, we proposed to investigate the effect of segmentation on a convolutional neural network (CNN) model used to classify the presence of COVID-19 in chest X-rays.

2. Problem Formulation

In order to analyse the effect of segmentation, the chest X-ray data was preprocessed and run through the CNN model, as described in the Proposed Solution section below. CNN performance is heavily reliant on the structure of the input data, which can be impacted by methods such as segmentation. While the general CNN code simply pulls pre-developed layers from the keras library, each step involves complex mathematics to extract information on the input image, generate output images, and optimise the loss function. The general mathematical framework for each step in the CNN model code is described below.

1. Conv2D Layer

The task for generating the Conv2D layer can be formulated as follows (Skalski):

$$(f * g)(x, y) = \sum m \sum n * f(m, n) * g(x - m, y - n)$$

f : the input image

g : the kernel/filter function

(x, y) : the location in the output image

(m, n) : the indexes and rows of the result matrix

The Conv2D layer uses a 2-dimensional filter (or kernel) to produce a tensor of outputs, multiplying and summing across each element in the input image to result in an output image. Convolution is performed at each location along the input image, in this case each chest X-ray, and can be repeated with several different filters to pull different features from the image. In the case of our model, 128 different filters were used in this step. Each output image was then passed through ReLU, which is a non-linear activation function, before being passed on to the next layer. The goal of convoluting is to extract information on the low-level features of the image.

2. MaxPooling2D Layer

The task for generating the MaxPooling2D layer can be formulated as follows (Luo et al., 2020; Alajrami et al., 2020):

$$f'(x, y) = \max(f(i, j))$$

f' : output image

f : input image

$i \in [\text{xstride}:\text{xstride}:\text{pool_size}]$
 $j \in [\text{ystride}:\text{ystride}:\text{pool_size}]$
pool_size: the size of the pooling window
stride: the distance between each pooling window

The goal of pooling is to reduce dimensionality, simplifying the model and making it easier to differentiate small distortions.

In the case of our model, the pool size was set to 2x2 pixels. The maximum pooling output shape can be represented as follows (Khosla):

$$\left\lceil \frac{(n_n - f + 1)}{s} \right\rceil * \left\lceil \frac{(n_w - f + 1)}{s} \right\rceil * [n_c]$$

n_n : height of feature map
 n_w : width of feature map
 n_c : number of channels in feature map
 f : size of filter
 s : stride

3. Dropout Layer

The task for generating the Dropout layer can be formulated as follows (Yeoh et al., 2018; Wager et al., 2013; Keras):

$$f'(x, y) = f(x, y) * \text{mask}$$

f : input image
 mask : a binary matrix of the same shape as the input data

The dropout layer is implemented to prevent overfitting in the model. In this step, some of the inputs in the model are randomly set to 0, while those that are not are scaled up by $\frac{1}{1 - \text{dropout rate}}$ via the mask array. This layer is commonly used during training, but skipped during testing so that all inputs are incorporated.

4. Dense Layer

The task for generating the Dense layer can be formulated as follows (Freire et al., 2022):

$$y = \Phi(Wx + b)$$

y : output vector
 W : weight matrix
 Φ : nonlinear activation function
 b : bias vector

The dense layer connects each neuron with all other neurons from the previous layer

with weight w_{ij} . This becomes helpful in learning the high-level features of the input image. The number of real multiplications (RM) of the dense layer is equivalent to the number of features in the input vector (n_i) multiplied by the number of neurons in the layer (n_n): $RM_{denselayer} = n_n n_i$

These initial four steps are repeated with 315,058 parameters to develop the full model used. Once this is complete, we move onto the compile step.

5. Compile

The compile feature is used to construct the general learning model for CNN. In our model, the Adam optimizer was used as the optimizer, the sparse categorical cross entropy was used as the loss function, and ‘accuracy’ was used for the metrics.

The Adam optimizer uses a gradient descent function to train the neural network, adjusting the learning rate of the model using the moving average of the gradient. The formula for the Adam optimizer is described below (Kingma and Ba, 2014):

$$\Theta_t = \Theta_{t-1} - \alpha \left(\frac{m_t}{(\sqrt{v_t} + \epsilon)} \right) * g_t$$

Θ : parameter

t : time step

α : learning rate

$\sqrt{v_t}$: moving average of squared gradient

m_t : moving average of the gradient

ϵ : small constant

g_t : gradient at time step t

The formula for the sparse categorical cross entropy is described below (Zhang and Sabuncu, 2018):

$$L = \frac{-1}{n[\sum(y_i * \log(p_i))]}$$

L : loss

y_i : true label (for i^{th} data point)

p_i : probability for the predicted true label (for i^{th} data point)

When the cross entropy function is run, the parameters are adjusted until the true label is predicted with a high probability, such that the loss is minimised. Through implementing this loss function, the Adam optimizer, and accuracy metrics, we are able to construct the general learning model for CNN.

3. Proposed Solution

Our approach will assess the effectiveness of CNN classification models at differentiating COVID-19 chest X-rays based on segmented and unsegmented image data sets.

CNN is a supervised learning model, and in this case will be used to classify images into COVID-19 positive and COVID-19 negative sets. One of the risks of CNN models is in the

expected generalisation loss, meaning the ability of the model to generalise and adapt to previously unseen data sets. The CNN model will attempt to minimise the generalisation loss using empirical risk minimization to minimise loss on the observed sample.

Additionally, while some models only use two data sets, the training and test sets, for model training and performance assessment, CNN uses three data sets to evaluate the model. The first of these is the training set, used to train the candidate model. The second is the validation set, which is used to evaluate the candidate models and choose the best one. Finally, the test set is used to evaluate the best model based on the validation set output, and report the chosen model’s performance. The inclusion of the additional data set, the validation set, also helps to minimise the expected generalisation loss discussed above.

Computational complexity for CNN is set by the algorithm $O(n)$, with n being the number of pixels in the image under review. A larger image may provide more precision - for example, more detail to learn from and distinguish based on but will result in a linear increase in computational complexity. Similarly, there is a linear increase in computational complexity for higher resolution images or an increase in image quantity.

There are several new developments in CNN models to reduce computational complexity and training and retraining time, for example the 1D-FALCON model introduced by Maji and Mullins (Maji and Mullins, 2018). The proposed 1D-FALCON model consists of an approximation stage and a fast arithmetic stage. In the approximation stage, each convolutional layer is approximated to reduce computational complexity and then decomposed into rank-1 filter banks. Then, the fast arithmetic stage applies a modified version of the Toom-Cook algorithm to compute the 1D convolution for a chosen set of distinct data points to reduce the number of complex operations. The 1D-FALCON model reduces the computational complexity for an $m * n$ pixel image from $O(mn)$ for standard 2D CNN convolution to $O(m + n)$.

In this study, we will segment the chest X-ray from our datasets in order to assess changes in performance of the CNN models. We applied pre-trained weights for the segmentation model provided by Motamed et al. to produce the masks (Motamed et al., 2021). Motamed et al first trained a segmentation model using chest X-rays dataset from Montgomery County’s Tuberculosis screening program which had already been labelled by expert radiologists (Motamed et al., 2021; Jaeger et al., 2012). These images were then resized uniformly and converted to grayscale (pixel values between 0 and 1). These normalised images were used to train a U-NET based model (Motamed et al., 2021). The team applied transfer learning with the Montgomery County dataset as the source and their COVID chest X-ray dataset as the target domains for segmenting their target chest X-rays. Finally, the resulting segmentation masks were corrected by trained radiologists. By applying a pre-trained mask to our segmentation model, we believe that we will see a significant improvement in the CNN classification model when classifying the segmented versus non-segmented dataset.

4. Numerical Experiments

Our CNN classification model is based on the deep learning analysis on detection of COVID cases by Sana Shaikh and Omkar Patil (Shaikh and Patil). The dataset used for this analysis is COVIDx CXR-3, a public open source data set of COVID-19 chest X-rays (Zhao et al.). The database has been approved by the University of Montreal’s Ethics Committee.

This data set contains 29,986 images from 16,648 patients, 15,994 of which are COVID-19 positive X-rays and 13,992 of which are COVID-19 negative X-rays. The images used in this data set were taken from multiple multinational public sources and thus vary in terms of rotation, exposure, resolution, and dimensions. Since these images are taken from varying data sets and are of varying quality, we believe that they are a fair approximation of a real-world implementation of this model in the field.

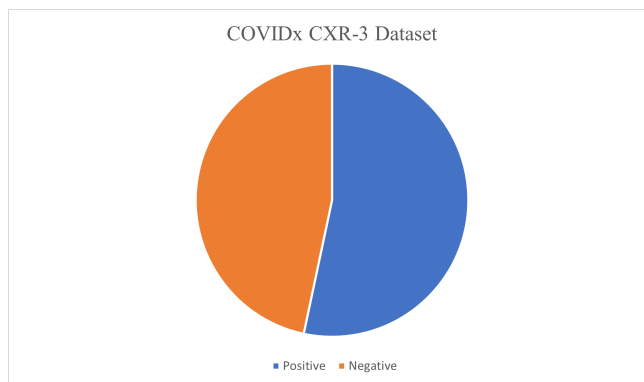


Figure 1: Distribution of positive and negative cases of COVIDx CR3 dataset

Segmentation was first applied to the images using the pre-trained weights described above in the Proposed Solution (Motamed et al., 2021).



Figure 2: Example chest X-ray from COVIDx dataset (left), its corresponding mask (middle), its segmented image (right)

For both classification analyses of the segmented and unsegmented datasets, the set of chest X-rays was processed and standardised to consistent dimensions and pixel values were normalised between 0 and 1.

For both analyses, the data was split into train, test, and validation sets using the following parameters: 90% training (26,987 images), 10% validating (2,999 images), and a random sample of 400 images in the test set. The model contains 315,058 parameters, all of which are trainable.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 68, 68, 128)	3584
max_pooling2d (MaxPooling2D)	(None, 34, 34, 128)	0
dropout (Dropout)	(None, 34, 34, 128)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	73792
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	36928
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 16)	200720
dropout_2 (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 2)	34

=====
Total params: 315,058
Trainable params: 315,058
Non-trainable params: 0

Figure 3: Model design used for analysis of segmented and unsegmented COVID-19 X-ray data sets

The classification models were evaluated and compared based on precision, recall, and f-1 score. Additionally, model accuracy and loss was traced over time through each epoch. We first assess the performance of the CNN classification model using unsegmented data. Overall, the model was not able to discriminate between COVID-19 positive and COVID-19 negative cases on the unsegmented data. The model predicted every image in the training data set as a positive case. This resulted in precision and accuracy scores of 0.5349, equivalent to the ratio of positive scans in the training set. Furthermore, the model did not significantly improve as it was trained. Early stopping occurred after epoch 12, based on the patience value cap of 10. This indicates that there was no improvement in model performance after the first epoch.

The performance of the model on the unsegmented data indicates that it is not robust enough to differentiate between COVID-19 positive and negative chest X-rays and thus would not be a good candidate for deployment in real-world scenarios.

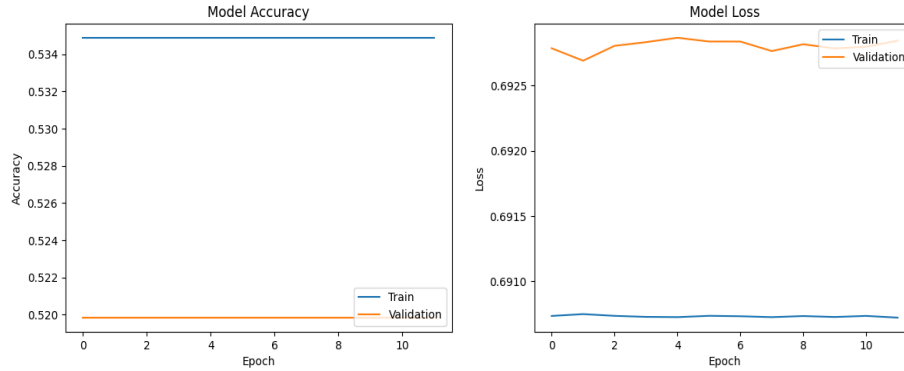


Figure 3: Graph summarising the history of accuracy (left) and loss (right) of the non-segmented dataset during the classification process

	Training Data	Validation Data	Test Data
Specificity	0.5349	0.5198	0.50000
Accuracy	0.5349	0.5198	0.50000
Precision	0.6970	0.6841	0.6667

Table 1: Classification report for train, validation and test data from the non-segmented images

Next we assess the performance of the CNN classification model using segmented data. The CNN model performed significantly better on segmented data as opposed to the unsegmented data set. This model had an accuracy score of 0.9857 and a precision score of 0.9815 on the training data. There was a slight decrease in model performance after the model validation and selection stage, with an accuracy score of 0.715 and a precision score of 0.9057 on the test data. Despite these dips in performance the model is still performing significantly better on the segmented data as opposed to the unsegmented data. We observe fluctuations in model accuracy and loss around epochs 10-20 but by epoch 30 the model has stabilised. This model also stopped early at epoch 40, indicating that there was no improvement in model performance after epoch 30.

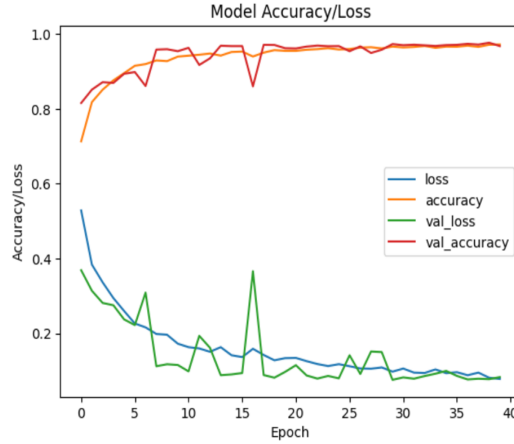


Figure 4: Graph summarising the history of accuracy and loss of the segmented dataset during the classification process

	Training Data	Validation Data	Test Data
Specificity	0.9785	0.9625	0.9500
Accuracy	0.9857	0.9677	0.7150
Precision	0.9815	0.9656	0.9057

Table 2: Classification report for train, validation and test data from the segmented images

Given the high precision score of this model, it is feasible that the model accuracy would further improve with additional training data or k-fold cross validation. Regardless,

the model still performs well under the current training conditions and gives indications that it would be reliable in a real-world implementation.

5. Conclusion

In this paper, we compared the performance of CNN classification models on segmented and non-segmented sets of chest X-ray images from COVIDx-CXR-3 dataset. This dataset contains images from multiple sources with varying quality which we believe reflect real-life diagnostic qualities and challenges. We used a pre-trained weight provided by Motamed et al. to generate segmentation masks (Motamed et al., 2021). The result of the CNN classification on segmented data proved to be significantly more accurate and precise. Thus, our study emphasised the importance of utilising segmentation in medical imaging in order to train models and improve diagnostic accuracy as previously demonstrated by multiple studies (Aggarwal et al., 2011; Sharma and Aggarwal, 2010).

As Aggarwal et al noted, accurate segmentation of medical images can be a labour intensive and time-consuming task as it requires both computational power and human expertise in order to create the most appropriate image contouring (Aggarwal et al., 2011). In our study, we were not able to train our own segmentation model due to the lack of human expertise, such as a trained radiologist. In future research, we would like to collaborate with expert radiologists, specifically those studying COVID-19 and other respiratory illnesses, in order to create the most appropriate segmentation model instead of relying on a pre-trained weight.

Furthermore, since the classification CNN model is a supervised learning model it requires a large number of labelled data for all classes (negative and positive) in order to learn to classify new data appropriately. In a situation where a new disease has emerged, the task of gathering enough labelled data to train the model might prove to be difficult. Therefore, recent studies have suggested that a generative adversarial neural network (GANs) might be the better solution than a traditional supervised learning such as CNN. Multiple studies using GANs for the task of classification have showed its semisupervised or unsupervised detection of abnormalities to be superior to CNN. These studies have shown that GAN models are only required to be trained on normal pathology in order to detect images with new or abnormal pathologies, and require significantly less labelled data to achieve an equivocal performance to the traditional CNN classification model (Madani et al., 2018a,b; Yi et al., 2019).

In summary, there is a substantial opportunity to implement AI solutions in disease identification and diagnosis, and we currently have many tools at our disposal to ensure that these models perform accurately and reliably. Further testing and development is needed to ensure that these models perform well against previously unseen diseases and to ensure that they are accessible to those communities and areas that are most heavily impacted by novel pathologies.

References

- P Aggarwal, R Vig, and S Bhandoria et al. Role of segmentation in medical imaging: A comparative study. *International Journal of Computer Applications*, 29:54–61, 2011.
- E Alajrami, BAM Ashqar, and BS Abu-Nesser et al. Handwritten signature verification using deep learning. *International Journal of Academic Multidisciplinary Research*, 3(12): 39–44, 2020.
- L Cai, J Gao, and D Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, 8(11), 2020.
- PJ Freire, S Srivallapanondh, and A Napoli et al. Computational complexity evaluation of neural network applications in signal processing. *arXiv*, 2022.
- S Jaeger, A Karargyris, and S Antani et al. Detecting tuberculosis in radiographs using combined lung masks. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4978–4981, 2012.
- Keras. Dropout layer keras.io. Technical report. (https://keras.io/api/layers/regularization_layers/dropout/).
- S Khosla. Cnn — introduction to pooling layer. URL <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>.
- DP Kingma and J Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.
- A Krizhevsky, I Sutskever, and GE Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- X Luo, X Tu, and Y Ding Y et al. Expectation pooling: an effective and interpretable pooling method for predicting dna–protein binding. *Bioinformatics*, 36(5):1405–1412, 2020.
- A Madani, M Moradi, and A Karargyris et al. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. *IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 1038–1042, 2018a.
- A Madani, JR Ong, and A Tibrewal et al. Deep echocardiography: data-efficient supervised and semi-supervised deep learning towards automated diagnosis of cardiac disease. *npj Digital Medicine*, 1(59), 2018b.
- P Maji and R Mullins. On the reduction of computational complexity of deep convolutional neural networks. *Entropy (Basel)*, 20(4):305, 2018.
- S Mehta, E Mercan, and J Bartlett et al. Y-net: Joint segmentation and classification for diagnosis of breast biopsy images. *Med Image Comput Assist Interv*, 2018.
- S Minaee, Y Boykov, and F Porikli et al. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

- S Motamed, P Rogalla, and F Khalvati. Randgan: Randomized generative adversarial network for detection of covid-19 in chest x-ray. *Scientific Reports*, 11(1), 2021.
- S Shaikh and O Patil. Detection of covid positive cases using dl. URL <https://www.kaggle.com/code/sana306/detection-of-covid-positive-cases-using-dl/notebook>.
- S Shalev-Shwartz and T Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.
- N Sharma and LM Aggarwal. Automated medical image segmentation techniques. *Journal of Medical Physics*, 35(1):3–14, 2010.
- P Skalski. Gentle dive into math behind convolutional neural networks. URL <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9>.
- Stefan Wager, Sida Wang, and Percy S. Liang. Dropout training as adaptive regularization. Technical report, NIPS’13: 26th International Conference on Neural Information Processing Systems, 2013.
- L Wang, ZQ Lin, and A Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10, 2020.
- Y Yang, L-F Yan, and X Zhang et al. Glioma grading on conventional mr images: a deep learning study with transfer learning. *Frontiers in Neuroscience*, 12(804), 2018.
- YJ Yeoh, T Morie, and H Tamukoh. An efficient hardware-oriented dropout algorithm. *Neurocomputing*, 427:191–200, 2018.
- X Yi, E Walia, and P Babyn. Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58, 2019.
- Z Zhang and MR Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv*, 2018.
- A Zhao, A Wong, and H Gunraj et al. Covid-net open source initiative - covidx cxr-3 dataset. URL https://www.kaggle.com/datasets/andyczhao/covidx-cxr2?select=competition_test.