Detection of COVID-19 Cases from X-ray Images Using Capsule-based Network

Donya Ashtiani Haghighi[1], Amirali Baniasadi[1]

[1]Department of Electrical and Computer Engineering, University of Victoria, Victoria, Canada
d.ashtiani.h@gmail.com
amiralib@uvic.ca

2023 International Conference on Multimedia Computing, Networking and Applications (MCNA) | 979-8-3503-3927-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/MCNA59361.2023.10185856

ABSTRACT- Coronavirus (Covid-19) disease has spread abruptly all over the world since the end of 2019. Computed Tomography (CT) scans and X-ray images are used to detect this disease. Different Deep Neural Network (DNN)-based diagnosis solutions has been developed, mainly based on Convolutional Neural Networks (CNNs), to accelerate identification of covid-19 cases. however, CNNs lose important information in intermediate layers and require large datasets. In this paper, capsule network (CAPSNET) is used. Capsule network performs better than CNNs for small datasets. an accuracy of 0.9885, F1-score of 0.9883, precision of 0.9859, recall of 0.9908 and area under the curve (AUC) of 0.9948 are achieved on the capsule-based framework with hyperparameter tuning. Moreover, different dropout rates are investigated to decrease the overfitting. Accordingly, a dropout rate of 0.1 shows best results. finally, we remove one convolution layer and decrease the number of trainable parameters to 146,752, which is a promising result.

KEYWORDS

Capsule Network, Dropout, Hyperparameter Tuning

I. INTRODUCTION

Since the end of 2019, the novel COVID-19 disease has spread very fast all around the world. One of the crucial actions to avoid the spread of infection in the body is early detection of the diagnosed cases. COVID-19 infection can be revealed as lung infection. In order to detect any type of lung infection, computed tomography (CT) and chest X-ray (CXR) images are mainly used [1]. To help doctors, researchers are developing different methods for early detection of coronavirus infections. The most commonly used methods rely on artificial intelligence and deep learning solutions. Convolutional neural networks (CNNs) are typically used for image classification. However, CNNs face challenges when it comes to recognizing the pose, texture, and deformations of the whole or part of an image [2]. This is mostly due to the lost information in the pooling layers. Although these layers add translation-invariance to the network, CNNs are still unable to handle rotation and scale-invariance without explicit data augmentation. Therefore, to improve performance and compensate for the drawbacks, CNNs require a large amount of data in particular for highly complex tasks. Also, due to adversarial attacks such as pixel perturbations, CNNs [3] could suffer from wrong classifications [4]. Capsule Network (CapsNet) were introduced to overcome the aforementioned limitations of CNNs.

CapsNet encodes the spatial information of the image and generates vectors for different categories. The relationship between the different levels of features such as edges, blobs, the whole objects are determined using the iterative Dynamic Routing algorithm. The other advantage of CapsNets is the robustness over affine transformations applied to input images, and it performs better than CNNs without training on a particular set of transformations. Capsule Networks also perform better on detecting overlapping images [2].

In this work, we investigate how Capsule Networks can help with detecting COVID-19 cases. To this end, we evaluate different models and compare different metrics.
The remaining sections in this paper are structured as follows. In Section 2, we review related work. Section 3 gives an overview of Capsule Network. Section 4 presents the dataset, capsule-based architecture, hyperparameter tuning, dropout method, and metrics used in this research. Section 5 reports experimental results. Finally, Section 6 makes concluding remarks.

II. RELATED WORK

In [5], A. Abbas et al. proposed a new CNN model, DeTraC. DeTraC employs a backbone architecture for feature extraction from images, decomposition part for training by the SGD optimizer, and class composition for classification. M. Siddhartha and A. Santra introduced COVIDLite. This design uses the depth-wise separable convolutional neural network (DSCNN) for the classification of CXR images [6]. K. Hammoudi et al. [7] designed a set of CNN models based on established architectures to detect normal, viral pneumonia, and bacterial pneumonia classes. They also provided an estimator for the infection rate from the predictions. M. Rahimzadeh and A. Attar introduced a modified deep convolutional neural network in [8]. Authors used extracted features from two models (Xception and ResNet50V2) and then a convolutional layer and a classification layer to detect infected cases. I. D. Apostolopoulos et al. [9] introduced a method to extract features from MobileNet, which employs a global pooling layer, a fully connected layer, and a classifier for image classification. These models are trained with different methods such as fine-tuning, transfer learning, and training from scratch. A. I. Khan et al. [10], propose CoroNet based on Xception. They use a dropout and two fully connected layers to classify X-ray images into four classes: normal, bacterial pneumonia, viral pneumonia, and COVID-19 positive. S. Karakanis and G. Leontidis [11], develop a unique CNN architecture without transfer learning for binary classification and multiclass classification of chest X-ray images. P. Kumar Sethy et al. [12], use different pretrained architectures on the ImageNet dataset such as AlexNet, DenseNet201, GoogleNet, InceptionV3, ResNet18, ResNet50, ResNet101, VGG16, VGG19, XceptionNet, and InceptionResNetV2 [13] to extract the deep features. D. N. Le et al. [14], use a Gaussian filter before feature extraction for preprocessing raw data. They also present a depth-wise separable convolution neural network (DWS-CNN) to extract the features from X-ray images. Finally, they apply a deep support vector machine (DSVM) for classification.

III. CAPSULE NETWORK

A capsule is a vector of neurons. In this method, there are a group of capsules, and each capsule includes neurons related to different information about any detectable object. This information reflects position, rotation, scale, and so on. The main elements of a Capsule Network are primary capsules and higher layer capsules. We review these elements below [2].

A. Primary Capsules

In the first stage of Capsule Networks, a series of convolutional layers are used to extract an array of feature maps from the input image. The extracted features are reshaped into vectors. To keep the length of each vector equal to one, a nonlinear Squash function is used [2]. This function can be described as follows:

$$d_j = \frac{\|p_j\|^2}{1+\|p_j\|^2}\frac{p_j}{\|p_j\|} \tag{1}$$

$$p_j = \sum_i a_{ij}\hat{v}_{ji}$$

$$\hat{v}_{ji} = A_{ij}\hat{v}_i \tag{2}$$

d_j is the vector output of capsule j and p_j is its input. The input capsule $p_j$can be calculated by multiplying a_{ij} which is determined by the Dynamic Routing algorithm (DR) and the prediction vector $\hat{v}_{ji}$. Prediction vector $\hat{v}_{ji}$ is the prediction output of the next capsules and the production of the previous capsule's output $\hat{v}_i$ and a weighted matrix A_{ij}. A_{ij} is the affine transformation matrix. The Dynamic Routing algorithm is used to infer the prediction outputs from the primary capsules by finding the corresponding coefficient for each of the primary capsules which presents a suitable routing between the input capsules and the capsules in the next layer. These coefficients are calculated by DR algorithm at every iteration during the training process based on the agreement between the primary capsules.

Note that in Capsule Network the number of output capsules equals to the number of classes in the classification problem. Each element of these vectors corresponds to an instantiation parameter of the image (e.g. size, skewness, etc). The length of each vector is the existence probability of an entity in an image. The Squash function facilitates normalizing the length of vectors.

B. Higher Layer Capsule

Capsule Network uses the dynamic routing algorithm which is based on the agreement between inputs. This method calculates the coupling coefficient a_{ij} using the softmax function:

$$a_{ij} = \frac{e^{c_{ij}}}{\sum_k e^{c_{ik}}} \tag{3}$$

Where c_{ij} is the log probability. The initial value of c_{ij} at the routing by agreement process is zero. In this process, the log probability c_{ij} is updated based on the agreement between d_j and $\hat{v}_i$, which is calculated as follows:

$$c_{ij} = d_j\hat{v}_i \tag{4}$$

Capsule Network uses the reconstructed image to improve accuracy. To reconstruct the input image, it uses an additional decoder network which recreates the input image by minimizing the squared error between the reconstructed image and the input image. There are three fully connected layers in this decoder, two rectified linear unit-activated units, and then the sigmoid-activated layer to decode the output vectors.

C. Loss Function

Capsule Network uses a separate loss function called margin loss L_{cap} for providing intraclass compactness and interclass separability in each digit capsule *t*. Therefore, high margin loss specifies that the prediction of each category in each capsule is not correct [15]. It is defined as

$$L_{cap} = W_t \max(0, n^+ - \|v_t\|)^2 + \lambda(1-W_t)\max(0, \|d_t\| - n^-)^2 \tag{5}$$

IV. COVID-19 CAPSULE NETWORK

The COVID-19 radiography database includes chest X-ray images for COVID-19 positive cases along with

Normal and Viral Pneumonia images. This dataset includes 3616 COVID-19 positive cases with 10,192 Normal, and 1345 Viral Pneumonia 224 * 224 one channel images. As the main goal of this project is to detect positive COVID-19 cases (binary classification), the labels for COVID-19 positive images are one (positive) and for Viral Pneumonia and Normal images are zero (negative).

All the neurons corresponding to these capsules are divided by $1 - p$.

We use the Adam optimizer, different types of learning rate, 100 epochs, and a batch size of 16 are used. We use 80% of the dataset for training and 20% for validation. Finally, we evaluate the trained model using different metrics such as accuracy, loss, precision, recall, and f1-score. In this paper, we report our improvements in comparison with COVIDcaps as the baseline [2].

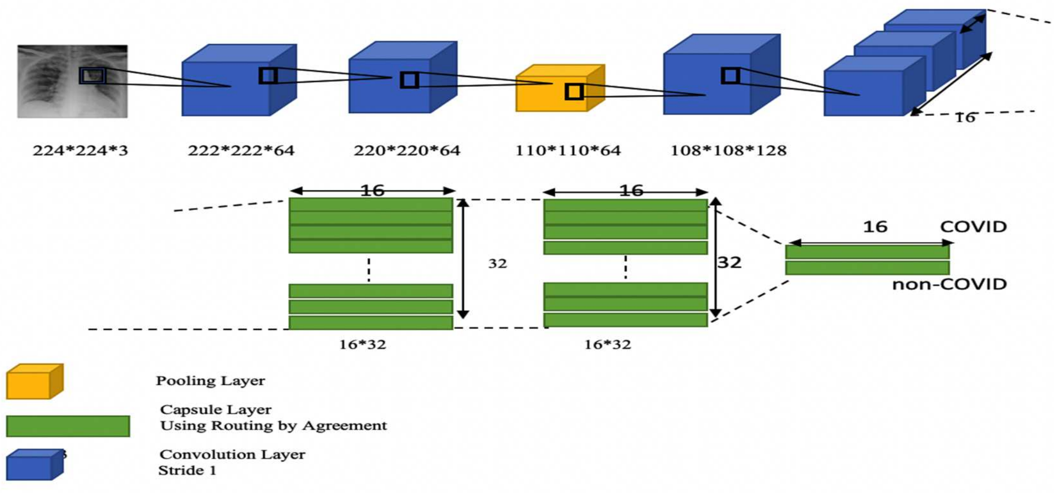


Fig. 1. Capsule-based Framework

The architecture of the Covid-19 Capsule Network, which can be found in Fig.1, includes four convolutional layers and three capsule layers. In the first layer, there are two convolutional blocks with 64 3*3 kernels followed by a 2*2 average pooling layer. Then, convolutional layers with 128 kernels are added as the third and fourth layers. The output of the fourth layer is reshaped for the first capsule layer. Then, capsule dropout with rate 0.1 was used to decrease overfitting. In this architecture, there are three capsule layers to apply the routing by agreement process. Finally, for the last capsule layer. The network contains the instantiation parameters of the positive and negative COVID-19 classes. The probability of each class is estimated by the length of these two capsules.

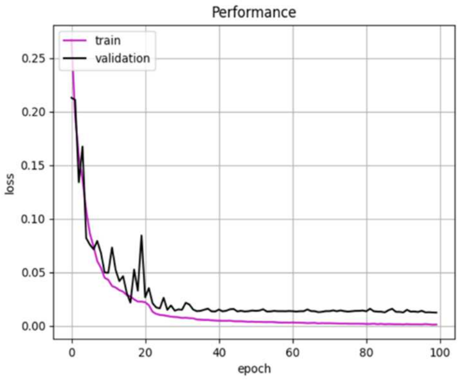
We also used the *Plateau Learning Rate scheduler* to reduce the learning rate in the absence of any improvement in a selected metric for a specified number of epochs (*patience* values). It can be seen that different "*patience*" values (the number of epochs before any drop in the value of learning rate) can affect the training process.

We use a capsule dropout method to reduce overfitting of the network by regularizing the training process and randomly dropping some of the primary capsules. We then follow the DR algorithm using the remaining capsules. In this method instead of dropping the elements inside vectors, we drop the whole capsule to avoid any changes in the direction of the vector. If p is the drop probability, the primary capsules are scaled by $1 / p - 1$ during the training.

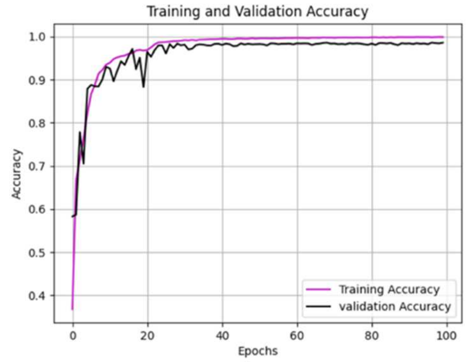
Fig 2: Covid-19, Normal and Viral Pneumonia images

V. RESULTS AND DISCUSSION

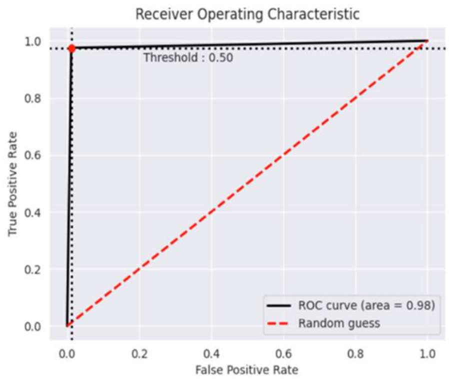
As we report in Table 1, we achieve the lowest loss of 0.0092, best accuracy of 0.9885, f1-score of 0.9883, precision of 0.9859, recall of 0.9908 and Area Under the Curve (AUC) of 0.9948 when using Plateau learning rate scheduler and margin loss function. Moreover, among the different dropout rates as one of hyperparameters which were used to reduce overfitting, dropout rate of 0.1 shows best results (Fig. 3). Finally, by removing one capsule layer and having far less number of trainable parameters in comparison to the main architecture we employ only 146,752 of learning parameters



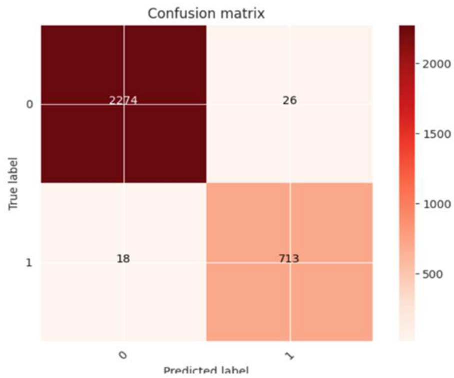
(a)



(b)



(c)



(d)

Fig. 3. (a) Loss, (b) accuracy, (c) ROC and AUC, (d) Confusion Matrix for margin loss, Plateau scheduler and Dropout rate 0.1

Tabel 1. Results obtained from the proposed Capsule-based framework

	Loss	**F1Score**	**Accuracy**	**Precision**	**Recall**	**ROC-AUC**	**No of trainable parameters**
Main [15]	0.0247	0.9547	0.9617	0.9393	0.9706	0.9935	295,488
Plateau dropout	**0.0092**	**0.9883**	**0.9885**	**0.9859**	**0.9908**	0.9948	295,488
Reduced Layers	0.0125	0.9816	0.9818	0.9774	0.9858	**0.9959**	**146,752**

I. CONCLUSION

In this paper, we investigated Capsule Network's performance in detection of COVID-19 cases from X- ray images. We modified the architecture by removing convolutional layer to reduce the number of training parameters and increase the run speed.

REFERENCES

[1] A. Rehman, T. Saba, U. Tariq, and N. Ayesha, (2021) "Deep learning-based COVID-19 detection using CT and X-Ray images: current analytics and comparisons", IT Professional, Vol. 23, No. 3, pp 63–68.

[2] S. Sabour, N. Frosst, and G. E. Hinton, (2017) "Dynamic routing between capsules", http://arxiv.org/abs/1710.09829

[3] J. Su, D. V. Vargas, and K. Sakurai, (2019) "One pixel attack for fooling Deep Neural Networks", IEEE Transactions on Evolutionary Computation, Vol. 23, No. 5, pp 828–841.

[4] J. Su, D. V. Vargas, and K. Sakurai, (2019) "Attacking convolutional neural network using differential evolution", IPSJ Transactions on Computer Vision and Applications, vol. 11, No. 1, p. 1.

[5] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, (2021) "Classification of COVID-19 in chest X-ray images using DeTraCdeep convolutional neural network", Applied Intelligence, Vol. 51, No. 2, pp 854–864.

[6] M. Siddhartha and A. Santra, (2020) "COVIDLite: A depth-wise separable deep neural network with white balance and CLAHE for detection of COVID-19", http://arxiv.org/abs/2006.13873

[7] K. Hammoudi et al., "Deep learning on chest x-ray images to detect and evaluate Pneumonia cases at the era of COVID-19", https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

[8] M. Rahimzadeh and A. Attar, (2020) "A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2", Informatics in Medicine Unlocked, Vol. 19, p. 100360.

[9] I. D. Apostolopoulos, S. I. Aznaouridis, and M. A. Tzani, (2020) "Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to Pulmonary diseases", Journal of Medical and Biological Engineering, Vol. 40, No. 3, pp 462–469.

[10] A. I. Khan, J. L. Shah, and M. M. Bhat, (2020) "CoroNet: a deep neural network for detection and diagnosis of COVID-19 from chest x-ray images", Computer Methods and Programs in Biomedicine, Vol. 196, p. 105581.

[11] S. Karakanis and G. Leontidis, (2021) "Lightweight deep learning models for detecting COVID-19 from chest x-ray images," Computers in Biology and Medicine, Vol. 130, p. 104181.

[12] P. Kumar Sethy, S. Kumari Behera, P. Kumar Ratha, and P. Biswas, (2020) "Detection of coronavirus disease (COVID-19) based on deep features and Support Vector Machine", www.preprints.org

[13] U. Ozkaya, S. Ozturk, and M. Barstugan, (2020) "Coronavirus (COVID-19) classification using deep features fusion and ranking technique", https://arxiv.org/abs/2004.03698

[14] D. N. Le, V. S. Parvathy, D. Gupta, A. Khanna, J. J. P. C. Rodrigues, and K. Shankar, (2021) "IoT enabled depth-wise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification", International Journal of Machine Learning and Cybernetics, Vol. 12, No. 11, pp 3235–3248

[15] A. F. M. Saif, T. Imtiaz, S. Rifat, C. Shahnaz, W.-P. Zhu, and M. O. Ahmad, "CapsCovNet: A modified Capsule Network to diagnose COVID-19 from multimodal medical imaging," IEEE Transactions on Artificial Intelligence, vol. 2, no. 6, pp. 608–617, Aug. 2021, doi: 10.1109/tai.2021.3104791.