UNIVERSITY NAME

DOCTORAL THESIS

Thesis Title

Author:
John SMITH

Supervisor:
Dr. James SMITH

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

January 16, 2023

Abstract

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Abbreviations

For/Dedicated to/To my…

Chapter 1

Introduction

1.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

1.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

1.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

1.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec

pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 2

Chapter Title Here

Batch normalization algorithm is described as follows:

Require: : Minibatch activation values x : $\mathcal{B} = \{x_{1,\ldots,m}\}$; parameters to be learned γ ,β.
Ensure: : $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

1: $\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$

2: $\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$

3: $\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$

4: $y_i \leftarrow \gamma\hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$

Chapter 3

Chapter Title Here

3.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

3.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

3.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec

pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 4

Proposed Methodology I

In this chapter two methodologies for COVID19 detection is presented. Methodology I present light CNN architecture for COVID19 detection which does not require large computational power to run. Methodology I does not care about the scale of the pneumonia. While methodology II does.

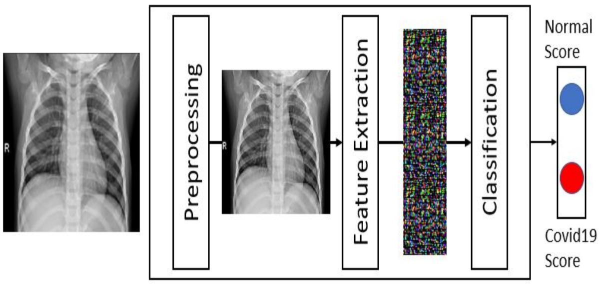

FIGURE 4.1: The phases of the proposed method I.

4.1 Methodology I

In this section, a proposed method I to detect COVID-19 disease from chest X-Ray images is presented. The proposed method exploits CNN model to classify the input chest X-Ray image to one of two categories; normal case or Covid-19 case. The proposed method I consists of three phases: preprocessing, feature extraction, and classification. The proposed method phases are shown in Fig.4.1.

4.1.1 Preprocessing Phase

The preprocessing phase is responsible for resizing and normalizing the input chest X-Ray images. The pre-processing phase is employed to maintain the numerical stability of the model and reduce the co-variance shift [1]. In addition, this phase leads the learning model of CNN model to reduce the required overhead to adapt to the different scales of different features of the input data. Reshaping size is determined empirically. The input chest X-Ray image is re-sized and then adapted and normalized to a normal distribution as follows:

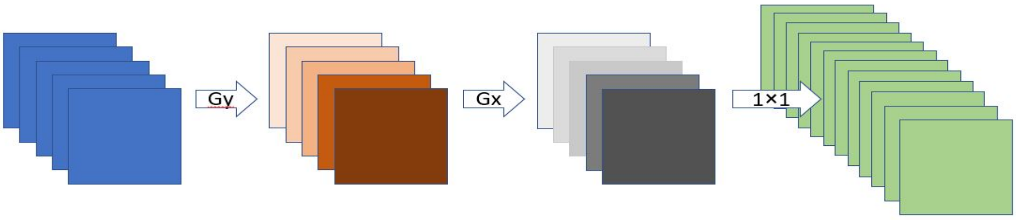

FIGURE 4.2: Separable convolution Gy and Gx have kernel size of $M \times 1$ and $1 \times M$. The combination of these kernels is approximately a $M \times M$ kernel and depth wise convolution are applied by a 1×1 convolution. The output depth is padded with zeros to have the same spatial size of Gy, Gx. Gy, Gx are performed channel wise.

$$Y := \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{4.1}$$

where μ and σ is the mean and standard deviation of chest X-Ray image (X), respectively.

After re-sizing the input chest X-Ray image, the input image is normalized to have a zero mean and unit standard deviation. Then, the image can be scaled and shifted with a normalization parameter which is determined and adapted by the training dataset during the training process according to the following equation:

$$Z := w_1 Y + w_2 \tag{4.2}$$

where w_1 and w_2 are a trainable parameter.

Unlike the normalization method presented in [2], the batch normalization process presented in this paper has z-score normalization parameter that is used in both training and validation phases.

4.1.2 Feature Extraction and Classification

CNN models achieved an outstanding success in image recognition [3]. This phase is responsible for extracting spatial features from the normalized chest X-Ray image using a tailored CNN model. This phase is based on learning the CNN model by the input preprocessed chest X-Ray images. The design of the tailored CNN model is described as follows:

1) Separable CNN kernels

Kernel separability[4] [5] is based on decomposing a 2D convolution kernel to linear combinations of two 1D vectors which leads to a large reduction in the total number of resulting parameters. For example, a 2D kernel of size 9×9 has a total number of $9^2 = 81$ trained parameters. Whereas in the case of separating this 2D kernel to linear combinations of two 1D vectors of sizes 9×1 and 1×9, this results in a total number of $9 + 9 = 18$ trained parameters. As a consequence, kernel separability reduces the

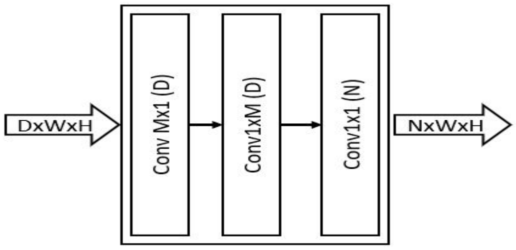

FIGURE 4.3: Separated Convolutional Layer
composed of three consecutive layers. The first Convolutional layer has a kernel size of $(M \times 1)$ and D convolutional neuron. The second layer operates in the same way as the first layer but it has a kernel of size $(1 \times M)$ and D convolutional neuron. The third layer is the convolutional layer with kernel of size (1×1) and number of convolutional neuron is N.

number of CNN model operations (such as the multiplication and the addition). A 2D kernel of $k \times k$ applied for 2D signal with spatial dimensions of $M \times N$ has a total number of $(N-4)(M-4) \times k^2$ operations but in case of applying kernel separability yields $2(N-4)(M-4)k$ operations. The flow of separated convolution operations are summarized in Fig. 4.2. Fig. 4.3 represents the structure, denoted by Separated Convolutional Layer, used in the proposed method with kernel size of $(M \times N)$ and satisfying the convolutional kernel separability. Separated Convolutional Layer is composed of three consecutive layers. The first convolutional layer has a kernel size of $(M \times 1)$ and the number of convolutional neuron and filters are equal to the number of channels as the input feature map and the convolution operations are performed in a channel wise. The second layer operates in the same way as the first layer but it has a kernel of size $(1 \times M)$. The third layer is the convolutional layer with kernel of size (1×1) and number of convolutional neuron is N. The collaboration of the three layers are connected to preform similarly to the convolutional layer with kernel size of $(M \times M)$ and number of neuron and filter are the same as N but with large difference in the performance.

2) Batch Normalization and Activation function

In the proposed method linear separable convolutional kernels are followed by a batch normalization and an activation function. Rectified Linear Unit (ReLU) [6] is a nonlinear activation that allows the network to fit and approximate highly non-linear datasets distribution. The proposed method employs the batch normalization which is described in [2].

Batch Normalization [2] reduces internal covariate shift produced as a result of moving between layers during the feedforward procedure [2]. Batch Normalization makes the loss landscape smoother and reduces the number of saddle points [7] which allows to use higher learning rates. Using a higher learning rate makes the network training faster[2]. Batch normalization reduces the vanishing gradient problem and exploding gradient problem as it makes the resulted activation scale independent from the trainable parameter scale[2]. Batch normalization has the effect of regularization

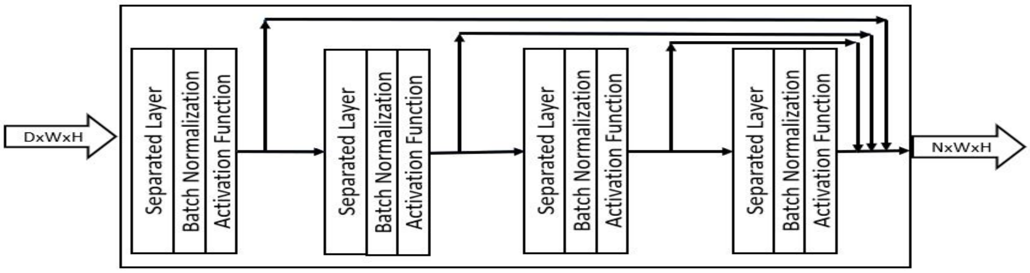

FIGURE 4.4: The stack of residual separated block (RSB) consists of four layer of separated convolutional layer each of which is followed by batch normalization and activation function.

because of the inherited randomness when selecting the batch sample[2] which help the generalization to unseen chest X-Ray image.

3) Deep and larger receptive field Network design

Deeper convolutional neural network design is a very important task for any image recognition task [8]. Training a deeper network is very expensive and has many challenges such as vanishing gradient problem, exploding gradient problem, and degradation problem [8]. Exploding gradient problem occurs when the gradient update becomes very large (approaching infinity) resulting in the network diversion. Vanishing gradient problem occurs when the gradient update becomes very small (approaching zero) resulting in preventing the parameter update for early layers[2] and preventing the network to learn new patterns. Batch normalization [2] and the use of ReLU activation function [9] alleviate these two problems.

The deep layers of CNN networks sometimes need to approximate the identity function which is not a simple task especially with the existence of a non-linear functions. Residual connection[8] overcomes this problem by using skip connection as shown in Fig. 4.4. Fig. 4.4 represents the building block layer of the feature extraction phase, denoted by stack of Residual Separated Block (RSB). RSB consists of four layers of separated convolutional layers, each layer is followed by a batch normalization and an activation function. It has an output of depth N where each sublayer produces an output of depth $N/4$ which is concatenated at the end of the layer to produce a depth N. RSB produces a feature map that includes both low level features and high level features.

Unlike the traditional neural network, which is fully connected to the previous layer, convolutional neural network is connected locally to a local region of the previous feature map. This introduces the concept of the network receptive field [10]. Receptive field should be large enough to capture large patterns in the input chest X-Ray image. Therefore, any consecutive convolutional layers in the proposed method without a pooling layer in between a larger kernel size is used in one of them. Residual Separated block, RSB, in Fig. 4.4 may have kernel sizes of 3, 5, 7, and 9, respectively. Fig. 4.5 Represent a complete CNN architecture.

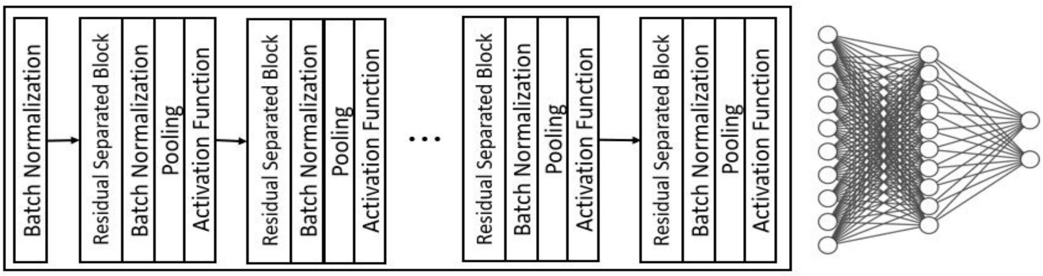

FIGURE 4.5: The complete proposed tailored CNN architecture.

Chapter 5

Proposed Methodology II

5.1 Methodology II

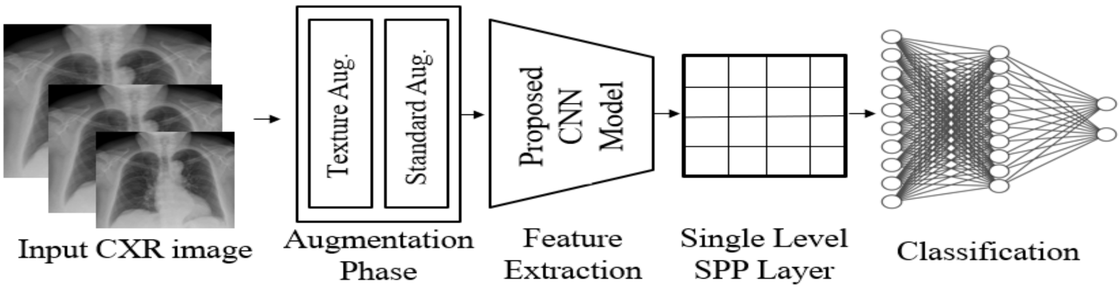

FIGURE 5.1: Proposed method for COVID-19 classification from CXR images.

The proposed system presented in this paper proposes a novel CNN micro-architecture model for learning scale-invariant features of CXR images and then classifies these features into normal or COVID-19 cases. Fig. 5.1 illustrates the proposed end-to-end pipeline of the proposed system. The proposed system depends on a novel Spatially weighted Atrous Spatial Pyramid Pooling (SWASPP) to extract multi-scale features of input CXR images. A novel attention model is then used to fuse the extracted multi-scale features and select the relevant scale features that the next CNN network should consider.

5.1.1 Data augmentation

The first phase of the proposed CXR classification system is data augmentation. Data augmentation is used to reduce the overfitting and artificially enlarge the training dataset [9]. The input CXR images are augmented using texture augmentation. Texture augmentation is performed by adding a multiplicative normally distributed

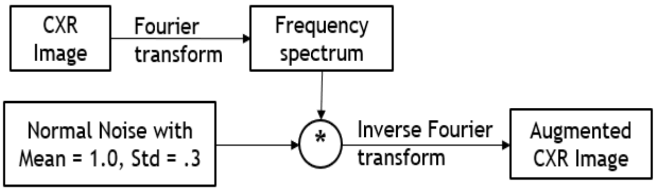

FIGURE 5.2: Texture Augmentation module

noises to the frequency spectrum of the input image. Noise is modeled using $\mathcal{N}(\mu = 1, \sigma = 0.3)$. Fig. 5.2 illustrates texture augmentation process. Fig. 5.3 shows the resultant CXR image. A standard augmentation such as random rotation, horizontal flipping, and vertical flipping are included in the augmentation process.

FIGURE 5.3: The resulting CXR image from Texture augmentation **left**: is the original image. **Right** is the augmented CXR Image

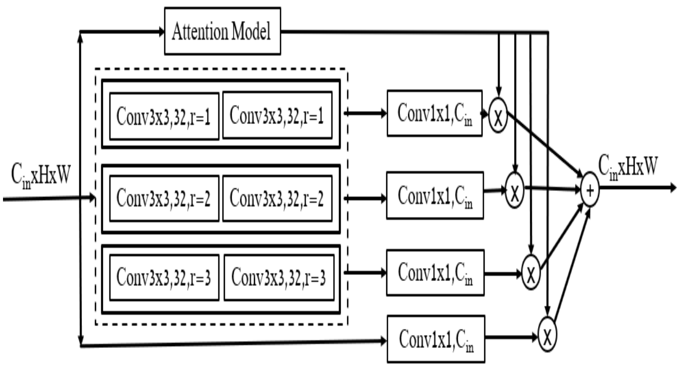

FIGURE 5.4: Spatially weighted atrous spatial Pyramid Pooling (SWASPP) interal layers within dashed square are parameter shared.

5.1.2 Spatially Weighted Atrous Spatial Pyramid Pooling

Atrous convolution is a powerful technique for adjusting the resolution of convolutional kernels. This allows to effectively enlarge the field-of-view of the kernel without increasing neither the number of kernel parameters nor the computational complexity of the convolution performance. A novel spatially weighted atrous spatial pyramid pooling (SWASPP) micro-architecture is presented. Fig. 5.4 shows the architecture structure. In Fig. 5.4, internal layers, bounded by dashed square, are parameter-shared and have different atrous rates. These layers are responsible for extracting multi-scale features. Sharing of the parameters enforce these layers to learn scale-invariant features. For a given input CXR image, three scales feature maps are produced. Each feature map corresponds to a particular scale.

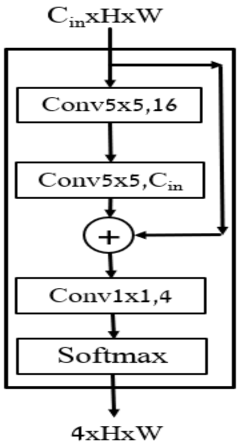

FIGURE 5.5: Attantion module structure used by SWASPP micro-architecture

To fuse the produced feature maps representing different scales of the input image, an attention module is added. Attention module can be thought as a pixel level classification of which scale does this spatial position belong. Fig. 5.5 illustrates the proposed attention module structure. Proposed attention module generates four heatmaps. The first three heatmaps correspond to the three scale feature maps while the remaining heatmap corresponds to the input feature map itself. These heatmaps are summed up to one (*i.e.*, for a spatial position (x, y), $\sum_{i=1}^{4} H(i, x, y) = 1$ where $H(i, x, y)$ is the i heatmap produced by the attention module). To make sure this property holds, softmax function is used.

The proposed mirco-architecture uses a pixel level weights produced by corresponding attention module rather than a single weight value for each scale. A single input CXR image may have multiple COVID-19 pneumonia scales which effectively lead to simply averaging the scale space when using single waight for each scale on scale space. In SWASPP, every convolution operation is followed by a BN and leakyReLU [9] non-linearity except the re-projection layers that used to project back to the input space. BN allows the use of larger learning rate[2] and makes network stable during training[2]. BN makes the loss landscape of the optimization proplem significantly smoother[7]. leakyRelu is used to reduce the vanishing gradient problem

[9]. A bottleneck is introduced within both the attention module and multi-scale feature extractor layer. A bottleneck in SWASPP is used to project the input feature map of dimension $C_{in} \times H \times W$ to $32 \times H \times W$ then re-project back to $C_{in} \times H \times W$. Multi-scale feature extraction is preformed on the projected dimension. Same logic is applied to the attention module where the input feature map is projected to a dimension of $16 \times H \times W$. This bottleneck allows the efficient use of model capacity and reduce the network computational complexity [11].

5.1.3 Proposed CNN Architecture

SWASPP is densely stacked [11] together as Fig. 5.6 illustrates. this kind of connectivity allows implicit deep supervisions as each layer is effectively connected to the last layer using shorter path also facilitate feature reuse [11]. Residual layers are easier to optimize if the required mapping is the identity mapping or simply near to it [8]. Densely stacked SWASPP is denoted by (DSWASPP). Convolutional part of proposed model consists of stacking six DSWASPP layers such that the first four layers are interconnected using maxpooling to reduce the spatial size and enlarge the Network receptive field. A single level Spatial Pyramid Pooling (SPP) [12] is added after to produce a fixed size feature vector for a variable size input. SPP layer divides the input feature map into $10 \times 10 = 100$ bins then performs a *max* for each bin as an aggregation function.

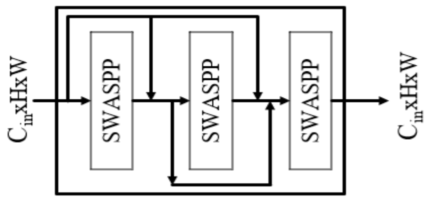

FIGURE 5.6: Densely connected SWASPP (DSWASPP): is a stack of densely connected SWASPP, such that the output of any SWASPP is Concatenated to the input of all next layers. All the three layers produce an output of dimension of $C_{in} \times H \times W$.

The fixed length feature vector produced by SPP is used as an input to dropout [13] layer. Dropout layer randomly sets the activation of to 0 with a probability of 0.5. Dropout prevents the overfitting and reduce complex co-adaptation between the neurons allowing them to learn better representation [13]. It allow implicit ensempling of exponential number of sampled thin network from the original network which enhance the network performance [13]. The result of dropout layer is used as input to the classification network. Classification network consists of a fully connected layers with a 3 Dense layers such that the output layer is 2-neuron for binary classification *i.e)* COVID19 or not. Table 5.1 shows the details of the proposed architecture.

Table 5.1: Proposed CNN architecture

Layer Name	**Proposed CNN Architecture of Methodology II**		
	Input Shape	***Output Shape***	***Param. Count***
Input layer	-	$1 \times 320 \times 320$	0
BatchNorm-1	$1 \times 320 \times 320$	$1 \times 320 \times 320$	2
DSWASPP-1	$1 \times 320 \times 320$	$32 \times 320 \times 320$	121,035
Maxpooling-1	$32 \times 320 \times 320$	$32 \times 160 \times 160$	0
DSWASPP-2	$32 \times 160 \times 160$	$64 \times 160 \times 160$	298,236
Maxpooling-2	$64 \times 160 \times 160$	$64 \times 80 \times 80$	0
DSWASPP-3	$64 \times 80 \times 80$	$128 \times 80 \times 80$	604,956
Maxpooling-3	$128 \times 80 \times 80$	$128 \times 40 \times 40$	0
DSWASPP-4	$128 \times 80 \times 80$	$128 \times 80 \times 80$	784,092
DSWASPP-5	$128 \times 80 \times 80$	$128 \times 80 \times 80$	784,092
DSWASPP-6	$128 \times 80 \times 80$	$128 \times 80 \times 80$	784,092
SPP-1	$128 \times 80 \times 80$	12800	0
Dropout-1	12800	12800	0
FC-1	12800	128	1,638,528
FC-2	128	128	16,512
FC-3	128	64	8,256
FC-4	64	2	130
Softmax	2	2	0
Total Number of Parameter			5,040,571

Any linear combination is followed by BN and leakyReLU nonlinearity excluding re-projection layer of the SWASPP modules

Chapter 6

Chapter Title Here

6.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

6.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

6.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

6.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec

pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 7

Chapter Title Here

7.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

7.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

7.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

7.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec

pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Appendix A

السيرة الذاتية لمقدم الرسالة

- **العنوان** : سملا مركز

Appendix B

موجز الرسالة

sfdsfdfsdf

Bibliography

[1] Yann LeCun et al. "Handwritten digit recognition with a back-propagation network". In: *Advances in neural information processing systems* 2 (1989).

[2] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[4] Roberto Rigamonti et al. "Learning separable filters". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 2754–2761.

[5] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[6] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[7] Shibani Santurkar et al. "How does batch normalization help optimization?" In: *Advances in neural information processing systems* 31 (2018).

[8] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[10] Wenjie Luo et al. "Understanding the effective receptive field in deep convolutional neural networks". In: *Advances in neural information processing systems* 29 (2016).

[11] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[12] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[13] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.