Paper Jurnal/Prosiding

by Trismayanti Dwi Puspitasari

Light-weight MobileNet for Fast Detection of COVID-19

Muhammad Hafidh Firmansyah
School of Computer Science, Kyungpook
National University, Republic of Korea
hafidh@knu.ac.kr

Seok-Joo Koh
School of Computer Science, Kyungpook
National University, Republic of Korea
sjkoh@knu.ac.kr

Wahyu Kurnia Dewanto
Program Studi Manajemen Informatika
Politeknik Negeri Jember, Indonesia
wahyu@polije.ac.id

Trismayanti Dwi Puspitasari
Program Studi Teknik Informatika,
Politeknik Negeri Jember, Indonesia
trismayanti@polije.ac.id

Abstrak— The machine learning models based on Convolutional Neural Networks (CNNs) can be effectively used for detection and recognition of objects, such as Corona Virus Disease 19 (COVID-19). In particular, the MobileNet and Single Shot multi-box Detector (SSD) have recently been proposed as the machine learning model for object detection. However, there are still some challenges for deployment of such architectures on the embedded devices, due to the limited computational power. Another problem is that the accuracy of the associated machine learning model may be decreased, depending on the number of concerned parameters and layers. This paper proposes a light-weight MobileNet (LMN) architecture that can be used to improve the accuracy of the machine learning model, with a small number of layers and lower computation time, compared to the existing models. By experimentation, we show that the proposed LMN model can be effectively used for detection of COVID-19 virus. The proposed LMN can achieve the accuracy of 98% with the file size of 27.8 Mbits by replacing the standard CNN layers with separable convolutional layers. (*Abstrak*)

Keywords— MobileNet; light-weight; SSD; CNN; COVID-19;

INTRODUCTION

Nowadays, the Corona Virus Disease 19 (COVID-19) has become a global pandemic issue since January 2019 [1]. Many methods have been developed to diagnose the patients who may get an infection with the virus by using a swab test or rapid test. With the help of the embedded device, the virus can be detected much faster. Furthermore, a variety of object detection methods can be used to obtain faster and more accurate results.

The Convolutional Neural Networks (CNNs) have become popular since 2012, and one of the famous CNN models is MobileNet [2]. MobileNet and its variants have so far been proposed, such as SqueezeNet [3], and Inception [4]. To achieve higher accuracies, the models need to be more in-depth and more complex. MobileNet does not use the standard convolutions. It instead uses the depth-wise separable convolutions which have a small number of parameters, and MobileNet will be used as a classifier. For object detection, the Single Shot multi-box Detector (SSD)[5] has been widely used because of the simplicity of the architecture. Therefore, the combination of MobileNet and SSD may be used to improve the accuracy of object detection on COVID-19, as shown in the Thin MobileNet [6] with SSD. However, the conventional combination of MobileNet or Thin MobileNet and SSD tends to increase the model size and computation cost, which makes it less efficient to deploy the object detection model on embedded devices.

Based on the observations, in this paper, we propose a Light-weight MobileNet (LMN) architecture that can be used to increase the model accuracy and to reduce the model size. The proposed LMN model is based on the ReLU [7] and 2D separable convolution. These techniques are helpful to resize the model up to four times the traditional combination of MobileNet and SSD. In the proposed scheme, the accuracy of object detection will not be dropped, while reducing the networks size by using these techniques. The proposed LMN model can achieve more than 90% accuracy for object detection of COVID-19 viruses.

As for the challenge of a small size of dataset for COVID-19 object detection, we employed a proper technique, called the transfer learning from CIFAR-10 [8] dataset to COVID-19 dataset. It is noted that the CIFAR-10 dataset consists of ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The CIFAR-10 dataset can help to fill the knowledge domain gap for the LMN model, and the transfer learning from CIFAR-10 will give the high accuracy of COVID-19 object detection model.

The proposed LMN model can be used to effectively identify the COVID-19 virus or its variation by reducing the inference time for object detection. The main contribution of this paper can be summarized as follows:

- We proposed a small and efficient model that can run on an embedded device, and the accuracy of our model is about 89% on the CIFAR10 dataset with ten classes as fire sparks on the training;
- Our model can detect and classify the COVID-19 viruses based on the cell images. The proposed LMN model can be used as a pre-trained model for another type of viruses. In the experiments, we use the three kinds of virus dataset [9]: 1) COVID-19, 2) Rotavirus, and 3) Influenza. The proposed model can run smoothly on the small embedded device, such as Raspberry Pi or mobile phones. The proposed model can be used to help the medical

staff to check faster and to get higher accuracy even with the small computing devices.

This paper is organized as follows. In Related Works Section, we discuss the related works, including MobileNet, Thin MobileNet, Transfer Learning, and SSD. Proposed LMN Architecture Section describes the proposed LMN architecture. Experimentation And Performance Analysis Section discusses the experimentation and the performance analysis of the proposed LMN model with the existing models. In Conclusion Section, we conclude this paper.

RELATED WORKS

A. MobileNet

MobileNet [2] has a unique characteristic by using the depth-wise separable convolutions. MobileNet consists of two hyper-parameters: 1) resolution multiplier to decrease the resolution of images, 2) width multiplier to slim the network. Those two hyper-parameters can make the computation cost reduced by eight to nine times, and the model size will get much smaller. It is noted that the original MobileNet consists of 28 layers. Batch Normalization [10] and ReLU are used after the convolutional layer. In addition, MobileNet has the Average Pooling layer, Fully Connected layer, and Softmax classifier.

Table 1 shows the architecture of MobileNet with all layers for classification. For object detection, MobileNet gives a good accuracy with SSD combination. MobileNet becomes a classifier accelerator with SSD as an object detection model. The MobileNet architecture consists of 2D convolutional layer and 2D depth-wise separable convolutional layer. Two types of strides are used: Stride 1 (s1) and Stride 2 (s2). Stride represents the element-wise shift displacement of a kernel over an input along a particular axis [11]. Stride 1 will move one filter at a time, and Stride 2 will move two filters at a time. The resolution of the image used on this model is 224 x 224 pixels, where the 2D convolutional layer only receives an image with the resolution 224x224 pixels. The resolution of the input file will be reduced until 1x1 on the last layer.

TABEL I.   MOBILENET ARCHITECTURE FOR CLASSIFICATION

No	Type / Stride	Output Shape	No	Type / Stride	Output Shape
1	Conv/s2	224 x 224 x 3	15	Conv dw/s1	14 x 14 x 512
2	Conv dw/s1	112 x 112 x 32	16	Conv dw/s1	14 x 14 x 512
3	Conv/s1	112 x 112 x 32	17	Conv dw/s1	14 x 14 x 512
4	Conv dw/s2	112 x 112 x 64	18	Conv dw/s1	14 x 14 x 512
5	Conv/s1	56 x 56 x 64	19	2 x Conv/s1	14 x 14 x 512
6	Conv dw/s1	56 x 56 x 128	20	Conv/s1	14 x 14 x 512
7	Conv/s1	56 x 56 x 128	21	Conv/s1	14 x 14 x 512
8	Conv dw/s2	56 x 56 x 28	22	Conv dw/s2	14 x 14 x 512
9	Conv/s1	28 x 28 x 12	23	Conv/s1	7 x 7 x 512
10	Conv dw/s1	28 x 28 x 256	24	Conv dw/s2	7 x 7 x 1024
11	Conv/s1	28 x 28 x 256	25	Conv/s1	7 x 7 x 1024
12	Conv dw/s2	28 x 28 x 256	26	Avg Pool/s1	1 x 1 x 1024
13	Conv/s1	14 x 14 x 256	27	FC / s1	1 x 1 x 1024
14	Conv dw/s1	14 x 14 x 512	28	Softmax/s1	1 x 1 x 1000

B. Thin MobileNet

Thin MobileNet [6] is a fork of MobileNet. It uses separable convolutions rather than depth-wise separable convolutional layers. It also changes the ReLU non-linear unit into the Drop-Activation function. It introduces Random Erasing [12] to increase the accuracy during the training. Meanwhile, Random Erasing is data augmentation. The system selects a random area on images, then it erases the pixel of that area and substitutes with a random value.

Figure 1 shows the difference between MobileNet and Thin MobileNet, in which Thin MobileNet has a simpler block. Thin MobileNet cannot be combined with SSD networks because Random Erasing makes the training more challenging to reach high accuracy.
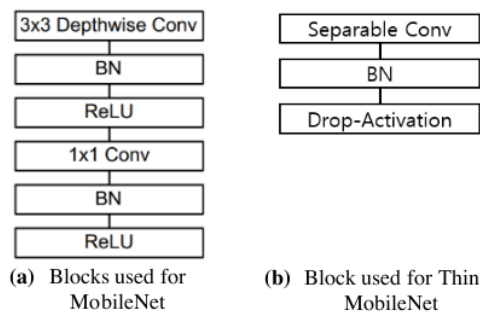

(a) Blocks used for MobileNet

(b) Block used for Thin MobileNet

Figure 1. Comparison of convolutional blocks for MobileNet and Thin MobileNet

C. Transfer Learning for Visual Categorization

Transfer Learning for Visual Categorization [13] has the three types of knowledge transfer: 1) source domain features, 2) source domain features, and the corresponding labels, 3) parameters of the learned source domain models. For COVID-19 object detection, we use CIFAR-10 dataset knowledge on the first step. The model can identify the COVID-19 dataset, not from prior knowledge. Transfer learning has the two classes for knowledge transfer: 1) Weights will rank instance-based transfer learning, training sample from source domain on the target domain; 2) Feature-based transfer learning uses a standard feature that transfers information from the source domain to the target area.

D. Single Shot multi-box Detector (SSD)

Single Shot multi-box Detector (SSD) [5] is a first deep neural network for object detector using a bounding box. It has a good accuracy for object detection using the

VOC2007 dataset [14]. SSD can improve the speed because it eliminates the subsequent bounding box proposals and the features resampling stage. The core role of SSD is to predict category scores and box offsets. There is some auxiliary structure on the system to provide detections with the following key features: 1) Multi-scale feature maps for detection; 2) Convolutional predictors for prediction; and 3) Default boxes and aspect ratios. SSD comes with Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG-16) [15] as a feature extractor on the top layer, as shown in Figure 2.
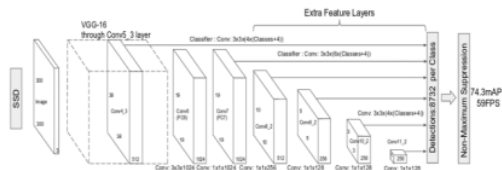

Figure 2. SSD and VGG-16 architecture.

E. *MobileNet with SSD*

The MobileNet with SSD [16] is a combination of object detection and recognition tasks that can be implemented on the autonomous mobile robot with a camera. Figure 3 shows the combination of MobileNet and SSD. This combination can be helpful to increase the accuracy of the object detection. MobileNet can extract the feature on the input images. The original size of the image will be reduced into 300x300 pixels. SSD is used as extra feature layer for object detection. The result of this combination can achieve good accuracy, but it still tends to give the big file size.
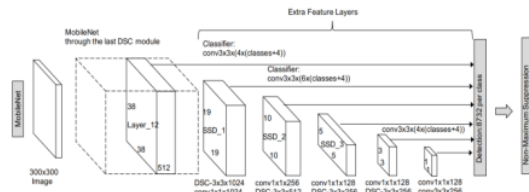

Figure 3. MobileNet with SSD for classfication and object detector

PROPOSED LMN ARCHITECTURE

In this section, we describe the LMN architecture by modifying the drop activation and random erasing on Thin MobileNet structure into ReLU activation non-linear unit. Drop activation and random erasing cannot run properly on a combination of Thin MobileNet and SSD. Thus, we change the SSD 2D convolution layer into 2D separable convolution. This modification makes the SSD model size much smaller without decreasing the accuracy. We use Tensorflow [17], and Keras framework [18] to build the LMN model. We also choose Adam [19] as an optimizer for the proposed LMN model.

A. *LMN Architecture Overview*

In the proposed LMN architecture, the training process consists of the three stages, as shown in Figure 4. Stage 1

(LMN without SSD) will train the dataset for classification. Stage 2 (Transfer Learning) is a bridge between LMN without SSD into LMN with SSD. Stage 3 (LMN with SSD) will train the dataset for the object detection model. Based on Figure 4, the four steps to produce an object detection model can be summarized as follows.

1) Training for classification on LMN without SSD using the CIFAR10 dataset.
2) The first transfer learning is divided into the two parts:
   a) The result will go through LMN with SSD for VOC object detection model.
   b) The result will go through LMN without SSD for virus classification model.
3) These two models will be implemented on the second transfer learning stage.
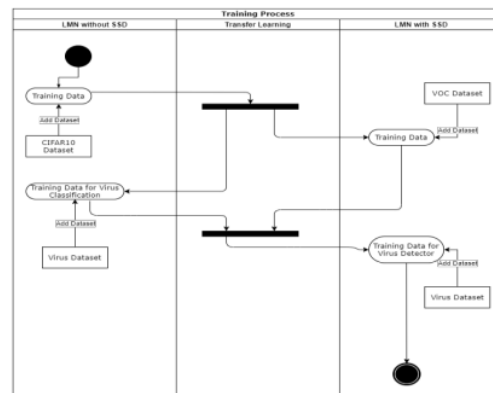4) The last model will feed into LMN with SSD for virus object detection model.

Figure 4. Transfer learning process on LMN training model

Table 2 compares the differences between the existing Thin MobileNet architecture and the proposed LMN architectures (LMN with SSD, LMN without SSD). The transfer learning can be applied to the proposed LMN architectures, differently from Thin MobileNet. The proposed LMN is also applied for virus dataset as well as CIFAR10. In addition, the proposed LMN can be used for object detection and classification, whereas Thin MobileNet can be used only for object classification.

TABEL II.　　COMPARISON OF THIN MOBILENET AND PROPOSED LMN MODEL

Type	Type of MobileNet		
	Thin MobileNet	*LMN with SSD*	*LMN without SSD*
Transfer Learning	Impossible	Possible	Possible
Dataset	CIFAR10	VOC Virus	CIFAR10, Virus
Output	Classification	Object Datection	Classificatin

Figure 5 compares the functional blocks used for the existing Thin MobileNet and the proposed LMN architecture. With the separable convergence layer and

Batch Normalization (BN) layer, the proposed LMN architecture uses the ReLU activation, instead of the drop activation. This is because the drop activation is not well working on the LMN network, and it tends to make the model very hard to achieve reasonable accuracy.
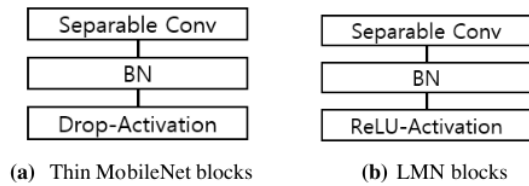

(a) Thin MobileNet blocks   **(b)** LMN blocks

Figure 5. Difference of funtional blocks for Thin MobileNet and LMN

B. LMN without SSD for Object Classification

The LMN without SSD is used for classification. Table 3 shows the 8 layers of the proposed LMN architecture for classification. We note that totally 28 layers are employed in the existing MobileNet architecture, as shown in Table 2.

TABEL III.   LMN ARCHITECTURE FOR CLASSIFICATION

No	Type / Stride	Output Shape	No	Type / Stride	Output Shape
1	Input Layer	32x32x3	6	Separable conv2d/s2	4x4x512
2	Conv2d/s2	16x16x32	7	Separable conv2d/s2	4x4x1024
3	Separable conv2d/s1	16x16x32	8	Global average pool/s1	1x1x1024
4	Separable conv2d/s2	8x8x128	9	FC and Softmax/s1	1x1x10
5	Separable conv2d/s2	8x8x256			

Figure 6 shows the training process of LMN without SSD for object classification. For virus classification, we only need to perform a single transfer learning for the CIFAR10 training dataset. Through this transfer learning process, the LMN without SSD model can produce a virus classification model. This LMN model will be used again on the LMN with SSD training stage. This training process on LMN without SSD is helpful to increase the accuracy of LMN model.
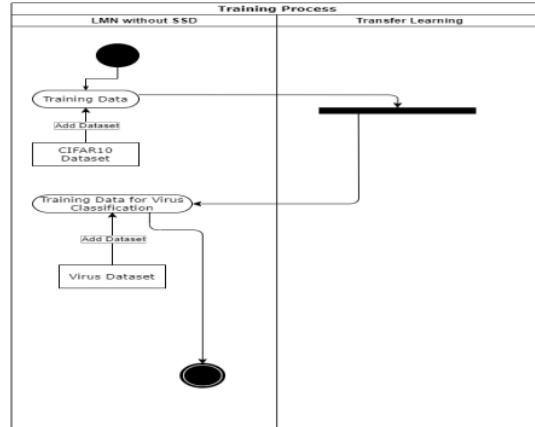

Figure 6. Training process on LMN without SSD

C. LMN with SSD for Object Classification

For object detection, we employ the combination of LMN with SSD. The VOC dataset will be used for initilatization of training, and the COVID-19 dataset is used to produce a model for object detection of COVID-19 viruses. We use the LMN structure, as described in the previous section, whereas the SSD networks will be changed by applying the first 2D convolution layer, and then the rest of 2D convolutional layer will be replaced with the 2D separable convolution. This modification is done to reduce the file size after combination of LMN with SSD networks.

Figure 7 shows the combination of LMN with SSD on the streamed-line neural network, in which LMN acts as a classifier to increase the training process time and for identification of the object. The associated results will feed to the SSD networks for object detection process, which will produce the output on the image.
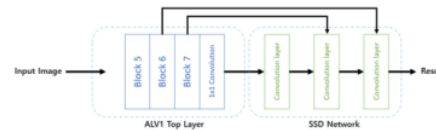

Figure 7. Combination of LMN with SSD for object detection

The proposed LMN with SSD model is designed to improve the MobileNet with SSD model. Table 4 summarizes the differences in layers between MobileNet with SSD and LMN with SSD. In the proposed LMN with SSD, the 2D separable convolution layer is employed to factorize a convolution kernel into the two smaller kernels, and to keep the file size small, while maintaining the high accuracy of the model.

TABEL IV.   DIFFERENCES IN LAYERS BETWEEN MOBILENET WITH SSD AND LMN WITH SSD

No	Type	MobileNet with SSD	LMN with SSD
1	2D Convolution Layer	Used for all layers	Used only for th/e top layer

No	Type	MobileNet with SSD	LMN with SSD
2	2D Separable Convolution Layer	Not Used	Used on each layer
3	File Size	75 Mb	9 Mb

(a) MobileNet (existing)   **(b)** LMN (proposed)

Figure 9. Object detection results for existing MobileNet and proposed LMN

EXPERIMENTATION AND PERFORMANCE ANALYSIS

A. *Experimentations for COVID-19 Virus Classification and Detection*

Based on the proposed LMN architecture, we performed the two kinds of experimentations for COVID-19 viruses: classification and detection. Virus Classification is a step taken after the transfer learning stage on cross-domain knowledge transfer due to a limited dataset. CIFAR-10 dataset is very helpful to continue the training by using the transfer learning. After the LMN model has obtained enough knowledge to do classification, we feed the virus dataset into the LMN network so as to classify three kinds of viruses; influenza, rotavirus, and COVID-19. We set the rotation on 90°, 180°, and backflip on each picture with Adam as an optimizer and the default learning rate for our training.

Figure 8 shows the virus image that we will use as dataset, in which we choose only three classes to test how much good our model is in identifying the three type of the viruses with small dataset.
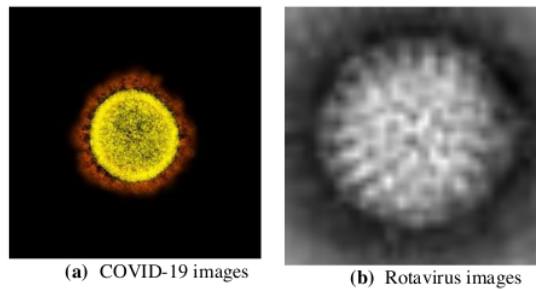

(a) COVID-19 images   **(b)** Rotavirus images

Figure 8. Virus images from Transfer Electron Microscope (TEM)

Detection of COVID-19 viruses (objects) is the main objective of this paper. For object detection, we combine the LMN classifier and SSD object detector model. Due to the limited dataset, COVID-19 has good accuracy in object detection. COVID-19 dataset consists of 1,200 images using a bounding box technique.

Figure 9 shows the object (virus) detection results for the existing MobileNet and proposed LMN architectures. In the figure, we see that the proposed LMN architecture can give higher accuracy for COVID-19 object detectors than the existing MobileNet architecture.
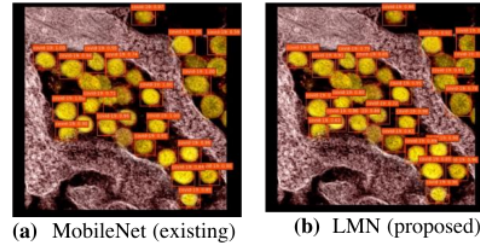
B. Performance Comparison

1) LMN Architectures without SSD using CIFAR-10 Dataset

For performance analysis, we first trained the model from scratch by using the CIFAR-10 dataset which consist ten class, 50,000 train images and 10,000 test images without transfer learning technique. By experimentation, we have trained the model using Adam as optimizer, and 200 epochs on each 1563 steps.

Figure 10 shows the accuracy and loss after LMN model has been trained. From the figure, we see that the proposed LMN architecture gives the accuracy of 86%, whereas the existing MobileNet architecture provides the accuracy of 84.3%. We can also see that the proposed LMN architecture provides lower losses than the existing MobileNet architecture. It seems that this performance benefit comes from the elaborately designed LMN and SSD with a small file size.
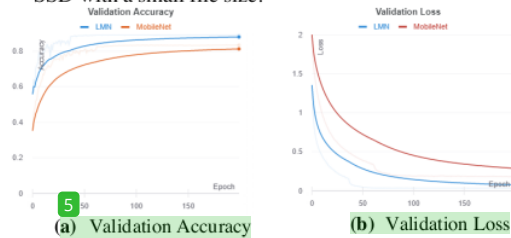

(a) Validation Accuracy   **(b)** Validation Loss

Figure 10. Validation accuracy and loss for LMN without SSD .

Table 5 compares the performance with CIFAR-10 dataset for the three candidate schemes: MobileNet, Thin MobileNet, and proposed LMN without SSD. From the table, we see that the proposed LMN model provides higher accuracy, smaller model size, and smaller number of parameters than the existing MobileNet and Thin MobileNet architectures.

TABEL V. COMPARISON OF PERFORMANCE FOR CIFAR-10 DATASET.

Model	CIFAR-10 Accuracy	File Size (Mb)	No. of Parameters
MobileNet	84.30%	39.1	3,239,114
Thin MobileNet	85.61%	9.9	814,826
LMN	**86%**	**8.8**	**722,922**

69

2) LMN Architectures without SSD using CIFAR-10 Dataset

We now analyze the performance of the proposed LMN without SSD for COVID-19 dataset. In this section, the LMN architecture was implemented by using the transfer learning method for the COVID-19 dataset.

Table 6 compares the performance with the COVID-19 dataset for MobileNet and LMN. Note that Thin MobileNet cannot be applied to the COVID-19 dataset, since it fails to apply transfer learning. Figure 11 shows the accuracy and loss after the LMN with SSD model has been trained. From the table and figure, we see that the proposed LMN architecture gives the accuracy of 98%, whereas the existing MobileNet architecture provides the accuracy of 94%. Overall, the proposed LMN architecture provides higher accuracy and lower losses, with smaller file sizes, than the existing MobileNet architecture.

TABEL VI.   COMPARISON OF PERFORMANCE FOR COVID-19 DATASET

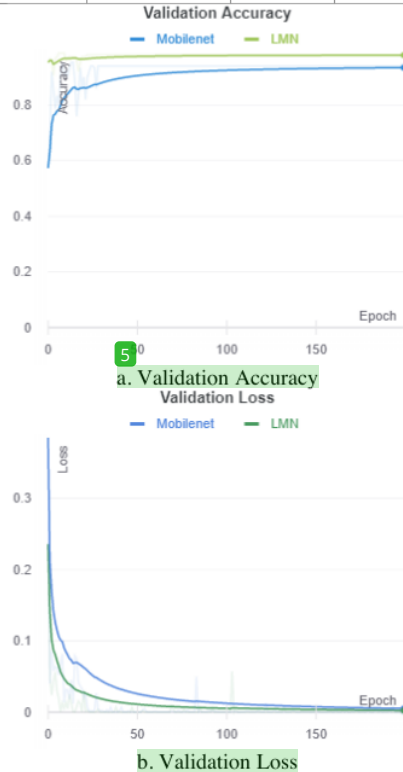| Model	CIFAR-10 Accuracy	File Size (Mb)	No. of Parameters
MobileNet	94%	70.3	5,854,403
LMN	98%	27.8	2,310,125

a. Validation Accuracy

b. Validation Loss

Figure 11. Validation accuracy and loss for LMN with SSD

3) LMN Architecture with SSD using COVID-19 Dataset

In this section, we analyze the object detection performance of the proposed LMN architecture with SSD

networks by using the COVID-19 dataset. It is noted that the transfer learning model gives a big impact on the object detection training accuracy.

Table 7 shows the performance comparison for MobileNet and LMN. The mean Average Precision (mAP) is used as the performance metric, which is typically used for computer vision object detection. From the table, we see that the proposed LMN model can reach 0.68 mAP, whereas the existing MobileNet gives 0.52 mAP. We can also see that the LMN architecture requires smaller file size and smaller number of parameters than the existing MobileNet architecture.

TABEL VII.   PERFORMANCE COMPARISON FOR OBJECT DETECTION OF COVID-19 VIRUSES

Model Type	COVID-19 mAP	File Size (Mb)	No. of Parameters
MobileNet	0.52	75.2	6,230,132
LMN	0.68	6	1,442,004

In this paper, we proposed a light-weight MobileNet (LMN) architecture for detection of COVID-19 viruses, which can be used to improve the accuracy of the machine learning model, with a small file size. The proposed architecture is featured by the replacement of the standard CNN layer with separable convolutional layers, differently from the existing MobileNet and SSD schemes. From experimentations, we see that the proposed LMN can achieve higher accuracy with smaller file size and smaller number of parameters, compared with the existing MobileNet and Thin MobileNet architectures. In the proposed LMN model, it is also possible to use the transfer learning method, which can be useful for a small dataset to train a model. For future study, the proposed LMN model needs to be implemented on real embedded devices.

CONCLUSION

In this paper, we proposed a light-weight MobileNet (LMN) architecture for detection of COVID-19 viruses, which can be used to improve the accuracy of the machine learning model, with a small file size. The proposed architecture is featured by the replacement of the standard CNN layer with separable convolutional layers, differently from the existing MobileNet and SSD schemes. From experimentations, we see that the proposed LMN can achieve higher accuracy with smaller file size and smaller number of parameters, compared with the existing MobileNet and Thin MobileNet architectures. In the proposed LMN model, it is also possible to use the transfer learning method, which can be useful for a small dataset to train a model. For future study, the proposed LMN model needs to be implemented on real embedded devices. With the small file size on LMN with SSD can reach up to 6 Mb, and 1,442,004 no. of parameters.

REFERENCE

[1] W. H. Organization, "Coronavirus disease 2019 (COVID-19): situation report, 30," World Health Organization.

[2] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional

Neural Networks for Mobile Vision Applications," Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861.

[3] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.07360.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.07261.

[5] W. Liu *et al*., "SSD: Single Shot MultiBox Detector," Dec. 2015, doi: 10.1007/978-3-319-46448-0_2.

[6] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, Oct. 2019, pp. 280–285, doi: 10.1109/UEMCON47517.2019.8993089.

[7] S. Dittmer, E. J. King, and P. Maass, "Singular Values for ReLU Layers," no. 2, pp. 1–12, 2018, [Online]. Available: http://arxiv.org/abs/1812.02566.

[8] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 (Canadian Institute for Advanced Research)," [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[9] Kylberg, "Texture Dataset v. 1.0," 2011, Accessed: Jul. 15, 2020. [Online]. Available: http://www.cb.uu.se/~gustaf/texture/.

[10] E. Sari and V. P. Nia, "Batch Normalization in Quantized Networks," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.14214.

[11] J. Yepez and S. B. Ko, "Stride 2 1-D, 2-D, and 3-D Winograd for Convolutional Neural Networks," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 28, no. 4, pp. 853–863, 2020, doi: 10.1109/TVLSI.2019.2961602.

[12] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Jul. 2017, doi: 10.1609/aaai.v34i07.7000.

[13] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015, doi: 10.1109/TNNLS.2014.2330900.

[14] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385.

[16] A. Driaba, A. Gordeev, and V. Klyachin, "Recognition of various objects from a certain categorical set in real time using deep convolutional neural networks," *CEUR Workshop Proc.*, vol. 2500, pp. 4–9, 2019.

[17] M. Abadi *et al*., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.04467.

[18] F. Chollet and others, "Keras." GitHub, 2015, [Online]. Available: https://github.com/fchollet/keras.

[19] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980.

Paper Jurnal/Prosiding

PRIMARY SOURCES

1	Submitted to Institut Teknologi Nasional Malang Student Paper	**2**%
2	Submitted to Metropolia Ammattikorkeakoulu Student Paper	**2**%
3	Debjyoti Sinha, Mohamed El-Sharkawy. "Thin MobileNet: An Enhanced MobileNet Architecture", 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2019 Publication	**1**%
4	dspace.bracu.ac.bd:8080 Internet Source	**1**%
5	Kazuhiko Kakuda, Tomoyuki Enomoto, Shinichiro Miura. "Nonlinear Activation Functions in CNN Based on Fluid Dynamics and Its Applications", Computer Modeling in Engineering & Sciences, 2019 Publication	**1**%
6	Diaa Salama Abdelminaam, Fatma Helmy Ismail, Mohamed Taha, Ahmed Taha, Essam	**1**%

H. Houssein, Ayman Nabil. "CoAID-DEEP: An Optimized Intelligent Framework for Automated Detecting COVID-19 Misleading Information on Twitter", IEEE Access, 2021
Publication

7   D. P. Kim, S. J. Koh. "Analysis of Handover Latency for Mobile IPv6 and mSCTP", ICC Workshops - 2008 IEEE International Conference on Communications Workshops, 2008
Publication   1 %

8   www.irjet.net
Internet Source   1 %

| Exclude quotes | On | Exclude matches | < 1% |
| Exclude bibliography | On | | |