khazanah informatika

Jurnal Ilmu Komputer dan Informatika

The Advantage of Transfer Learning with Pre-Trained Model in CNN Towards Ct-Scan Classification

J.Pardede [1*], A.S.Purohita [2]
[1,2]Informatics
National Institute of Technology
Bandung
*jasman@itenas.ac.id

Abstract-Medical image classification plays significant role in the process of medical decisions making, especially during the difficult period of the pandemic. One method being considered good at such classification is Convolutional Neural Network, in which we use pre-trained model approach with transfer learning since the limitation of medical images may require optimal effort. Through this pre-trained model with transfer learning, the objective is to maximize the accuracy of classification and to push forward the training session throughout the comparison of both transfer learning and without transfer learning-based pre-train models. The first type provides average accuracy of 0.84 with approximate training time 0.54 hour while the latter shows the average result of accuracy as 0.74 with average training time 0.58 hour. As the result, the optimizations are 1.13x for accuracy and 1.1x for training time. EfficientNetV2 is one pre-trained model selected for this project, being exposed to both transfer learning and without transfer learning approach systems. The transfer learning version provides the superior accuracy as 0.88 and training time as 31 minutes - 50 seconds, showing the accuracy of 0.94 on validation and 0.88 on testing.

Keywords: CT-Scan, CNN, Classification, EfficientNetV2, Transfer Learning

Article info: *submitted September, 18, 2022, revised May, 15, 2023, accepted July, 18, 2023*

1. Introduction

There are clinical/medical characteristics being ably seen and analyzed by the experts, classified by using the CT-Scan images of the lungs belonging to COVID-19 patients. CT-Scan analysis requires both the ability and the presence of medical experts, moreover during the 2020 pandemic caused by COVID-19. The increasing number of COVID-19 cases creates the need of a method which immediately and precisely classifies COVID-19 patient data, such as CT-Scans functioning as a reference to the accuracy of the CPR test results. Through which, the doctors are able to veritably determine the cure towards the Covid-19 patients particularly ones having severe condition[1]. The Computerized CT-scan system is significant for the accurate diagnoses towards the patients, for example: the lung condition of ones suffering from Covid-19 [2].

Due to the limitation of the dataset and the difficulty of classifying medical data such as CT-scan Images of the lungs belonging to Covid-19 patients, Convolutional Neural Network (CNN) is selected as one convincing method to overcome the problem [3]. CNN is a subset of deep learning approach [4], having recently gained much of public attention because of its effectiveness in processing large amounts of data. CNN is one type of neural network often being used in creating an imagery data of an object, outperforming the conventional methods [5].

One approach being used in CNN is called transfer learning based pre-trained model. This model may overcome the existing problem of finding the best architecture with the ImageNet dataset applied standard test [6]. Transfer learning is the process of re-using pre-trained model weights which are trained in large datasets, usually in large-scale image classification tasks. Transfer learning method is also possibly modified to overcome other problems such as changing layers, updating parameters, and using weight or knowledge from the model to use in new datasets [7].

Nowadays, since AlexNet established several transfer learning based models have been existing and are often being used in classification: DenseNet, Inception, Xception, EfficientNet, MobileNet and others [6], [8]–[20]. An architecture has various layers having been trained in the ImageNet dataset. Therefore, its weights are considered sufficient. Each model has its own purpose of use and measurement. The use varies from classification to object detection and segmentation. One method named

Confusion Matrix, considerably accurate to measure a model's performance in classification [21]. In this study, Convolutional Neural Network (CNN) is used together with the desired architecture using ImageNet weight, expecting for the result will provide a shorter training time and exemplary accuracy based on the proposed validation approach.

2. Methods

a. Datasets

The observed dataset in this study is the CT-Scan image of the human lungs being categorized as healthy (Figure 1.a) and COVID-19 suffered lungs (Figure 1.b). The images were validated through previous studies [3]. In total, 2481 images are collected, consisting of 1252 healthy lung images and 1229 opposite ones. Prior to attempting the system, these categories are first divided into 3 parts: train, validation, and test data; whose ratio is 80:10:10 or 1952:244:244 in numbers, as shown in Table 1.

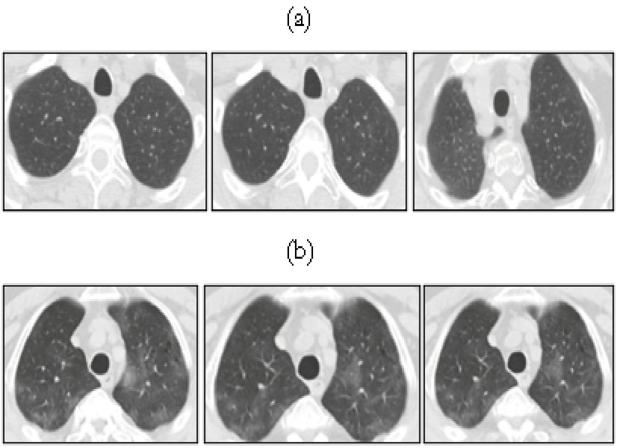

Figure 1. Datasets Normal (a) and Covid-19 (b) Lung CT-Scan

Table 1. Dataset Distribution

Category	Values		
	Train	*Validation*	*Test*
COVID-19	976	122	122
Healthy	976	122	122
Total	1952	244	244

b. Preprocessing and Augmentation

These three parts will then be implemented through several processes according to each part, two of which are the test and validation. The data will undergo preprocessing for rescaling the pixel value to [0, …, 255] and resizing the image to size 64X64. Furthermore, the training sessions, the data will also be augmented with several values as shown in Table 2. [22], [23]. The Values: First, Rotation up to 36 degrees, rotating the images' position between 0-36 degrees; Second, Brightness range from 0.8 to 1, transforming the brightness of the input image; Third, Horizontal - Vertical shift as 0-0.2, moving the image itself either vertically or horizontally; and Fourth, Flipping the image both horizontally and vertically based on line x and y.

Table 2. Preprocessing and Augmentation Values

Augmentation	Values
Rotation range	36
Brightness range	[0.8,…,1]
Height & Width shift range	[0,…,0.2]
Horizontal & Vertical flip	1
Rescaling	1./255
Model preprocessing	1

c. Deep Learning

Deep learning is a computational model in the learning layer for studying features in data. This model is used for improving outcomes of speech recognition tasks, dealing with visual object recognition and object detection tasks, also other domains such as drug discovery and genomics [4]. Deep learning studies the features of a large data set by using the back propagated algorithm to calculate values from parameters, based on the previous layer throughout utilizing the input, hidden, and output layers, as shown in Figure 2.

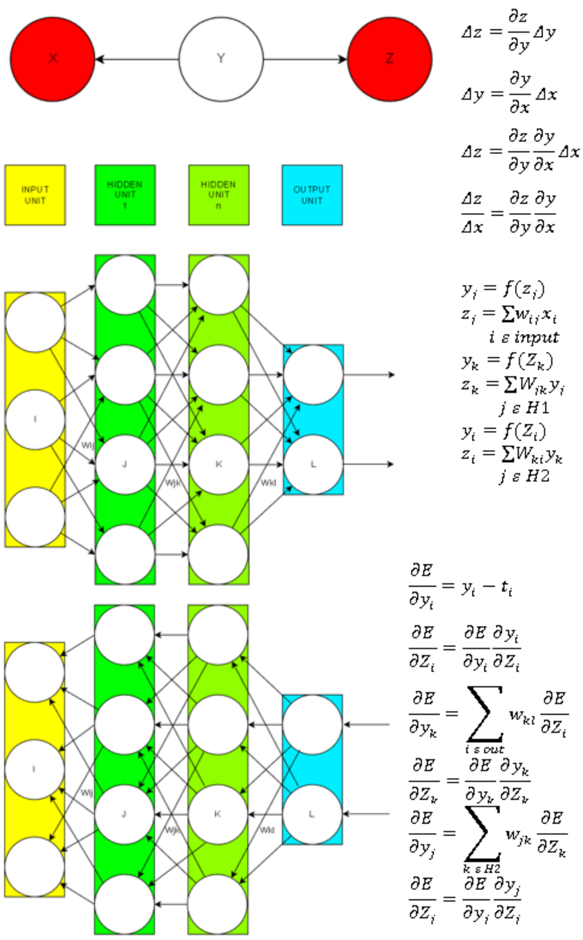

Figure 2. Multilayer Neural Network and Backpropagation [4]

d. Convolutional Neural Network (CNN)

Convolutional Neural Network is an artificial neural network which proceeds data originating from multiple array such as 3 2D-array-based color images, containing pixel intensity in those 3 color channels. Gray images consist of 1 2D array containing the average pixel intensity of 3 color channels becoming 1 color channel. Many data are formed from multiple array such as 1D signal and sequence, 2D images, 3D for videos. CNN preserves 4 major things in its working system: local connection, shared weight, pooling, and multilayer usage. Although the abundance of layers exist in Convolutional Neural Network, the common ones are convolutional, pooling, batch normalization, flatten, fully connected, and activation function layers [5][24]–[27], as shown in Figure 3.

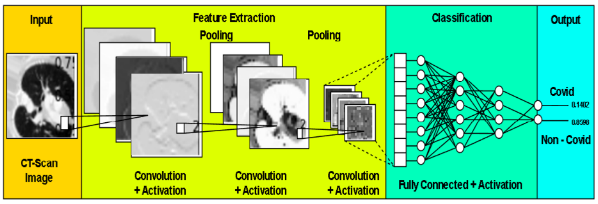

Figure 3. Convolutional Neural Network Structure[5]

During the training session, several hyper parameter values control the research: epoch, learning rate, batch size, optimizer, dropout rate, and activation [28]. We use 100 epochs for model training, 0.0001 for learning rate, 32 image batch size for each iteration, Adam as the optimizer, 0.8 for dropout connection and SoftMax for activation, being shown in Table 3.

Table 3. Hyperparameter Values

Hyperparameter	Values
Epoch	100
Learning rate	0.0001
Image size	64x64,Xception (71x71) InceptionResnetV2 (75x75)
Batch size	32
Optimizer	Adam
Dropout	0.8
Activation	SoftMax

e. Pre-trained Model and Transfer Learning

A Pre-Trained model is a model (a CNN architecture with weight, biases, and algorithms) that has been trained on a large dataset (ImageNet, CIFAR-10, MNIST, etc.), in case to solve a specific classification problem such as image classification, object detection, image captioning, the model already learned important feature from the data and already gone through the learning process, which can be computationally expensive and time-consuming [29]. There are several Pre-Trained Architecture (design, structure and operation in each layer), in this experiment 6 architecture are used; MobileNet, DenseNet, Xception, InceptionResNet, EfficientNet, and all EfficientNetV2 types, every model detail shown in Table 4.

Table 4. Pre-Trained Model Architectures

Architecture	Main Features	Top-1 Accuracy ImageNet (%)
EfficentNetV1	Compound Scaling	84.3
EfficentNetV2	SE Block, ConvLayer, Inverted Residual Block Combination, Model Scaling & Training Aware NAS	85.7
MobileNetV3	Depthwise Separable Convolution, Linear Bottleneck, Hard-Swish, Platform-Aware NAS & NetAdapt	75.2
Xception	Extreme Depthwise Separable Convolution	79
DenseNet	Dense Connectivity/ Dense Block	77.3
InceptionResNetV2	Inception Module, Residual Block, Scaling Residual	80.3

Transfer learning is a popular method as a part of deep learning approach which efficiently builds accurate models. This method enables the learning process of a pre-trained model to start from non-zero (some weight already embedded to architecture) and the basic patterns of the models, as at the same time, solving different problems. Through this way, we can have more benefits from the previous learning without starting from scratch using established weights [30], [31], shown in Figure 4.

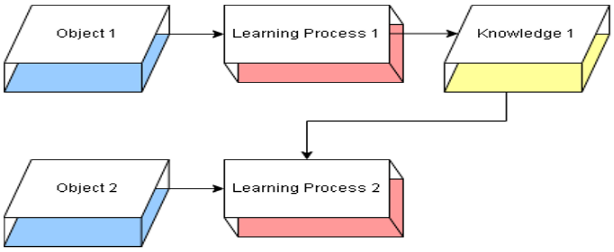

Figure 4. Transfer Learning Process

In the field of computerized working systems, transfer learning is usually expressed using pre-trained models. A pre-trained model is one being used in a large benchmark dataset to overcome the problems on which we focus. Therefore, because of the training cost of such models, the models from the published literature become a popular source to use (e.g., VGG, Inception, MobileNet, EfficientNet and others [8]–[13]).

Being shown in Figure 2, a comprehensive review towards the performance of a pre-trained model in dealing with computer vision problems using data from the ImageNet, becomes a reference of evaluation towards the performance of transfer learning model [32]. In this study, there is a proposed design process based on Figure 3 and Figure 4, from the image processing to the output layer, featuring pre-trained model-based extraction amid the two as illustrated in Figure 5. The whole process of input image processing to model testing is shown in Figure 5.

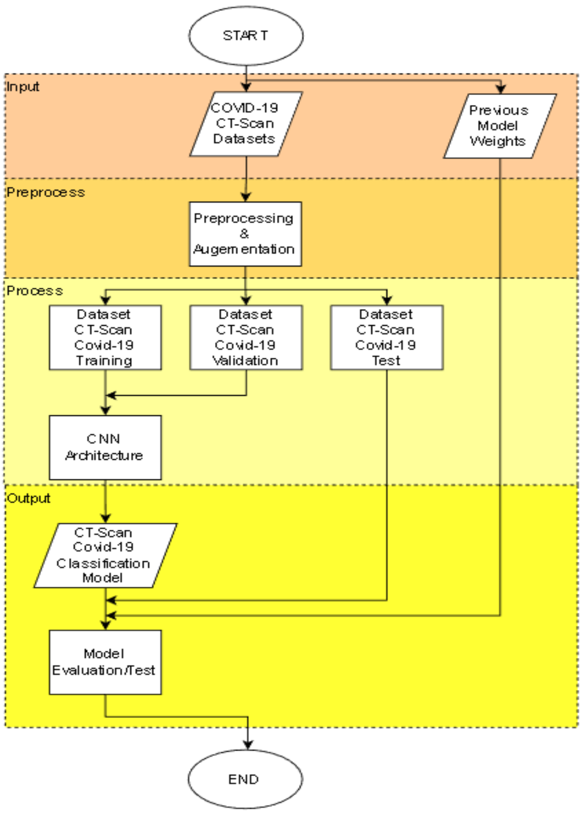

Figure 5. Experiment System Design Results

f. Model Evaluation

Measuring the performance of a model in the training, validating, and testing processes, requires a measurement metric consisting of accuracy, loss, and the training time[9]. The formulae of measuring the model's performance are depicted in equation 1-4[21];

$$accuracy = \frac{TP+TN}{Total\ Data} \quad (1)$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (4)$$

Where:
TP : True Positive
TN : True Negative
FP : False Positive
FN : False Negative

g. Experiment List

The Experimental Designs are created to assure the compatibility between the research results and its objectives, being shown in Table 5 (transfer learning effects). The Experiments are performed in both ImageNet weight and without ImageNet weight structures, the main difference is a model that used as details shown in Table 5.

Table 5. Experiment Scenario 1

Pre-Trained Model	Experiment 1	Experiment 2	Image Size
MobileNetV3L	Without ImageNet	ImageNet Weight	64x64
DenseNet201	Without ImageNet	ImageNet Weight	64x64
EfficientNetV2B0	Without ImageNet	ImageNet Weight	64x64
Xception	Without ImageNet	ImageNet Weight	71x71
EfficientNetV2B1	Without ImageNet	ImageNet Weight	64x64
InceptionResNetV2	Without ImageNet	ImageNet Weight	75x75
EfficientNetV2B2	Without ImageNet	ImageNet Weight	64x64
EfficientNetV2B3	Without ImageNet	ImageNet Weight	64x64
EfficientNetB7	Without ImageNet	ImageNet Weight	64x64
EfficientNetV2S	Without ImageNet	ImageNet Weight	64x64
EfficientNetV2M	Without ImageNet	ImageNet Weight	64x64
EfficientNetV2L	Without ImageNet	ImageNet	64x64

This training session is performed throughout Google Collaboratory applying Tesla T4 GPU with 12 GB RAM to the 6 models and the EfficientNetV2 models. Seven types of models appear, ranging from B0 to L, whose hyperparameter values are shown in Table 3. Testing the experiments in Table 5 will be performed based on the design in Figure 5. Each model has a different preprocessing because of the different input format. The results of both preprocessing and augmentation of every model are shown in figure 6.

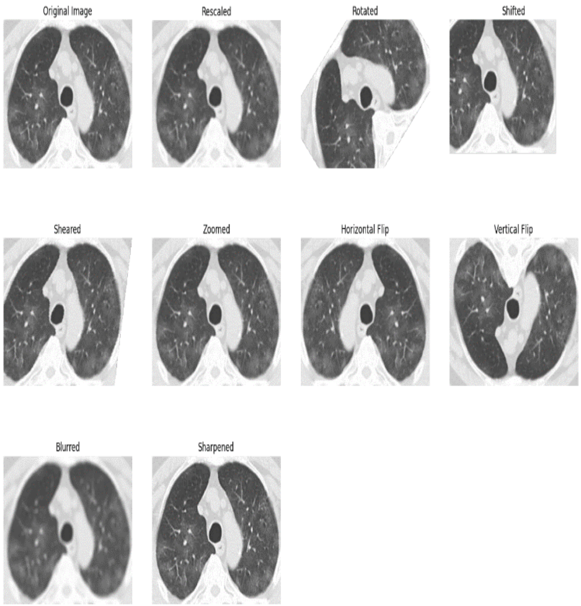

Figure 6. Image Preprocessing & Augmentation Results

3. Results and Discussion

The schematic of the test is implemented through the comparison between F1-Score training time and accuracy of both first and second conditions, each is orderly shown in Table 6-8. Tables 6-8 result from the models' performance towards validation data instead of the testing ones. However, these results are convincing to be documented as the complete analysis data.

First, we examine the values of the model validation results based on their accuracy, shown in Table 6. Table 6: The without ImageNet weight-based training model shows 0.82 best accuracy rate of EfficientnetV2L model classification and 0.73 average accuracy rate of none EfficentnetV2 based model. The ImageNet weight-based training model shows 0.92 best accuracy rate of EfficientnetV2L model classification results, similar with ones being shown, while the average accuracy of ImageNet weight-based models is 0.85. Based on the training sessions, inceptionresnetv2 model is stated as the successfully optimized one whose largest value is 0.15, with the average increase in accuracy of 0.12.

Second, the model is analyzed based on its F1 score, as shown in Table 7. Table 7, The without-ImageNet weight-based training model shows 0.85 as the best F1 score of the EfficientnetV2L model classification. The average F1-Score of the without-ImageNet weight-based model is 0.76. The ImageNet weight-based model shows the same best classification result of the EfficientnetV2L model same as one in Table 5. The F1-Score is 0.94 while the average accuracy of the ImageNet weight-based model is 0.85. Resulted from the training session, Mobilenetv3 is the most successfully optimized model whose largest value is 0.20; while the average increase in F1-Score is 0.08.

Table 6. Experiment 1 Results Based on Accuracy

Pre-Trained Model	Exp 1 Accuracy	Exp 2 Accuracy	Difference
MobileNetV3L	0.52	0.64	+0.12
DenseNet201	0.62	0.69	+0.07
EfficientNetV2B0	0.72	0.83	+0.11
Xception	0.73	0.77	+0.04
EfficientNetV2B1	0.72	0.86	+0.14
InceptionResNetV2	0.76	0.91	+0.15
EfficientNetV2B2	0.74	0.88	+0.14
EfficientNetV2B3	0.76	0.89	+0.13
EfficientNetV2S	0.76	0.89	+0.13
EfficientNetB7	0.77	0.91	+0.14
EfficientNetV2M	0.80	0.93	+0.13
EfficientNetV2L	0.82	0.94	+0.12

Table 7. Experiment 1 Results Based on F1-Score

Pre-Trained Model	F1-Score	F1-Score	Difference
MobileNetV3L	0.47	0.67	+0.20
DenseNet201	0.70	0.80	+0.10
EfficientNetV2B0	0.74	0.81	+0.05
Xception	0.67	0.73	+0.06
EfficientNetV2B1	0.77	0.82	+0.05
InceptionResNetV2	0.80	0.93	+0.13
EfficientNetV2B2	0.81	0.84	+0.03
EfficientNetV2B3	0.81	0.89	+0.08
EfficientNetV2S	0.83	0.89	+0.06
EfficientNetB7	0.84	0.93	+0.09
EfficientNetV2M	0.85	0.94	+0.09
EfficientNetV2L	0.85	0.94	+0.09

Finally, the model is analyzed based on its training duration as shown in Table 8. Table 8: the without-ImageNet weight training model shows 21 minutes 48 seconds as the fastest training through the MobileNetV3 model, while the average training time of such kind of training model is 36 minutes and 52 seconds. The ImageNet weight-based training model shows the fastest training duration through MobileNetV3 as 20 minutes and 40 seconds and then the seconds and average duration of the without-ImageNet weight-based model is 32 minutes and 18 seconds.

Table 8. Experiment 1 Results Based on Training Time

Pre-Trained Model	Without Transfer Learning	Using Transfer Learning
MobileNetV3L	0”21’48	0”20’40
DenseNet201	0”36’44	0”28’50
EfficientNetV2B0	0”26’11	0”23’29
Xception	0”38’29	0”36”51
EfficientNetV2B1	0”27’34	0”24”58
InceptionResNetV2	0”44”36	0”36”14
EfficientNetV2B2	0”29’05	0”27’35
EfficientNetV2B3	0”31”20	0”30”03
EfficientNetV2S	0”36”23	0”31”23
EfficientNetB7	0”46”20	0”42”21
EfficientNetV2M	0”43”21	0”34”38
EfficientNetV2L	0”59”33	0”50”44

4. Conclusion

All these experiments suggest that the ImageNet-weight-based transfer learning approach is able to directly optimize the accuracy and F1-score of the models and push forward the training process, caused by the pre-trained models with transfer learning that has built-in weights and biases so that it can be slightly better at recognizing important feature in the image.

Furthermore, we may state that EfficientNetV2 is the best model to perform towards the CT-scan Classification, applying both transfer learning and non-transfer learning methods. Its accuracy rates are 0.82 and 0.94 and training durations are 59 minutes 33 seconds to 50 minutes 44 seconds. Transfer learning ImageNet weight pre-trained models are evidently able to optimize CNN in CT-scan classification tasks: Training Duration, F1-score, and Accuracy; surpassing the non-transfer learning models. The final results: The optimized training duration reaches 1.1x in average while the optimized accuracy reaches 1.13x in average, surpassing the non-transfer learning pre-trained models' abilities.

Due to the limitations of researchers in this research and many opportunities for future research, there are several things that can be improved based on this research, especially after conducting research about pre-trained model with fine-tuning and optimization fine-tuning using hyperparameter tuning, adding analyzing feature to the model for deeper understanding how the models work, such as CAM, GRAD-CAM, GRAD-CAM++, LRP, and etc.

References

[1] C. Huang *et al.*, “Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China,” *The Lancet*, vol. 395, no. 10223, pp. 497–506, Feb. 2020, doi: 10.1016/S0140-6736(20)30183-5.

[2] Z. Y. Zu *et al.*, “Coronavirus Disease 2019 (COVID-19): A Perspective from China,” *Radiology*, vol. 296, no. 2. Radiological Society of North America Inc., pp. E15–E25, Aug. 01, 2020. doi: 10.1148/radiol.2020200490.

[3] E. Soares, P. Angelov, S. Biaso, M. H. Froes, and D. K. Abe, “SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification,” medRxiv, p. 2020.04.24.20078584, Jan. 2020, doi: 10.1101/2020.04.24.20078584.

[4] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[5] Y. Lecun, K. Kavukcuoglu, and C. Farabet, “Convolutional Networks and Applications in Vision,” 2010. [Online]. Available: http://www.cs.nyu.edu/

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” 2012. [Online]. Available: http://code.google.com/p/cuda-convnet/

[7] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A Survey on Deep Transfer Learning,” Aug. 2018, [Online]. Available: http://arxiv.org/abs/1808.01974

[8] A. Howard *et al.*, “Searching for mobilenetv3,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision,* pp. 1314-1324, 2019.

[9] M. Tan and Q. v. Le, “EfficientNetV2: Smaller Models and Faster Training,” Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.00298

[10] M. Tan and Q. v. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.06993

[12] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” 2017. [Online]. Available: www.aaai.org

[13] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” Oct. 2016, [Online]. Available: http://arxiv.org/abs/1610.02357

[14] Md Zahangir Alom, et al., “AlexNet,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018,

[15] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861,* 2017.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, & L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 4510-4520, 2019.

[17] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *International Conference on Machine Learning,* pp. 448-456, June 2015.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2818-2826, 2016.

[19] C. Szegedy *et al.*, “Going Deeper with Convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1-9, 2015.

[20] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[21] J. Pardede and A. S. Purohita, “Hyperparameter Search for CT-Scan Classification Using Hyperparameter Tuning in Pre-Trained Model CNN With MLP,” in *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, Oct. 2022, pp. 1–8. doi: 10.1109/ICOSNIKOM56551.2022.10034878.

[22] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[23] R. Balestriero, L. Bottou, and Y. LeCun, “The Effects of Regularization and Data Augmentation are Class Dependent,” Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.03632

[24] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[25] G. Hinton, Srivasta Nitish, and Swersky Kevin, “RMSProp,” 2013.

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

[27] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” Oct. 2017, [Online]. Available: http://arxiv.org/abs/1710.05941

[28] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” Jun. 2012, [Online]. Available: http://arxiv.org/abs/1206.5533

[29] X. Han *et al.*, “Pre-trained models: Past, present and future,” *AI Open*, vol. 2, pp. 225–250, 2021, doi: https://doi.org/10.1016/j.aiopen.2021.08.002.

[30] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra, “Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection,” *International Journal of Intelligent Systems and Applications*, vol. 13, no. 2, pp. 52–61, Apr. 2021, doi: 10.5815/ijisa.2021.02.04.

[31] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning Transferable Architectures for Scalable Image Recognition,” Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.07012

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Computer Vision and Pattern Recognition,” in *CVPR 2009, IEEE Conference* on, 20-25 June 2009, IEEE, 2009.