COMPARISON OF ADAPTIVE MOMENT ESTIMATION OPTIMIZATION AND NESTEROV-ACCELERATED ADAPTIVE MOMENT ESTIMATION OPTIMIZATION ON CONVOLUTIONAL NEURAL NETWORK METHOD FOR FRUIT DETECTION

Ismail Didit Samudro[1], Andi Farmadi [2], Dwi Kartini[3], Dodon Turianto Nugrahadi [4], Muliadi [5]
[12345]Ilmu Komputer FMIPA ULM
A. Yani St. KM 36 Banjarbaru, South Kalimantan
Email: 1611016210009@mhs.ulm.ac.id

Abstract

Convolutional Neural Networks are often used in research to conduct training, validation, classification, prediction and detection of images using Deep Neural Network. Optimization algorithm is used to change the hyperparameter values in the Neural Network such as learning rate, optimization is needed to reduce losses and increase the accuracy of the model. Optimization algorithm that is widely used because of its good performance is Adam and Nadam optimization, but the learning rate setting still needs to be updated manually. In this research architecture that was based on VGG16 will be used, Learning Rate Scheduler is used in optimization to control the learning rate value by updating the learning rate value in each step during model training. In this study, a comparison of the optimization of Adam and Nadam was carried out when the Learning Rate Scheduler was used to update the learning rate value in model training and obtained prediction accuracy using Adam 98.85% and Nadam 95.02% and then obtained MAP model performance value using Adam 93.58%. and Nadam 75.28%.

Keywords: *Convolutional Neural Network, Optimization Algorithm, Learning Rate Scheduler, Deep Learning*

1. INTRODUCTION

Convolutional Neural Network (CNN) is a Machine Learning method that is widely used in various fields such as object classification, object detection, medical images, word processing and wave processing [2]. The main problem that often occurs in the use of CNN is the long training time, this is due to the selection of architecture, learning rate and optimization algorithms that are not appropriate in the case of the dataset to be processed [5].

It is widely known that the training time on the CNN method varies, it can be from a few hours, a few days and even several weeks [12]. It depends on the training set, architecture, optimization method used and the number of training epochs required. To speed up the training process, Graphic Processor Units (GPU) are usually used. There are various cases that stem from real problems where the dataset used is very large, it often happens that the GPU memory capacity used is not enough. The training set used cannot be processed entirely on the GPU and a mini-batch approach in the form of per-epoch training is used. Because of this, the number of training iterations per epoch varies depending on the size of the dataset and GPU memory [11].

In the journal by Dogo, et al. (2018) compared various Stochastic Gradient Descent optimization algorithms, namely SGD vanilla, SGD momentum, SGD

momentum + nesterov, RMSProp, Nadam, Adamax, Adam, AdaGrad, AdaDelta. The dataset used is MNIST in the form of dataset 1 - Cats and Dogs, dataset 2 - Fashion MNIST and dataset 3 - Natural Image. Experiments on the 1 Nadam dataset obtained the highest accuracy of 85.5% and reached convergence within 2.5 hours at 450 epochs. In dataset 2, Adam has the highest accuracy, which is 72% and Nadam below it at 71.2%, in this dataset, other algorithms have poor performance, such as AdaDelta which gets accuracy of 27.2% and RMSprop fails to get convergence. In dataset 3, Nadam gets the highest accuracy value, which is 91.3% and followed by AdaDelta and Adam, which scores below 90% and other optimization algorithms are below 80% [5], convergence is obtained at 140 epochs and 5 hours of training time. Based on the journal, it can be said that Nadam is the best optimization algorithm in the three experiments and produces the best performance from the accuracy value, but the training time is still quite slow.

The most important thing in order to speed up the training time is to determine the right learning rate value so that it can speed up training on large datasets. Determining the learning rate is difficult, a learning rate that is too small can cause very long training and slow convergence, while the use of a learning rate that is too large can hinder convergence and cause the loss function to fluctuate and even get stuck and fail to get the right accuracy value [11].

There is a new library on python that can help on controlling the learning rate, also known as Learning Rate Scheduler, it can be deployed with many optimization algorithm. The library succeeded in getting an increase in accuracy in the dataset and also the training time carried out became more short and takes only a few minutes from the usual few hours. It can be said that when controlling the learning rate with Adam using Learning Rate Scheduler, it can increase the accuracy value and speed up the training time carried out on the model that has been made.

This research was conducted in order to find out if the learning rate scheduler is carried out with the Nadam optimization algorithm whether it can produce a better accuracy value and shorter training time compared to Adam when the learning rate schedulerl is carried out. The dataset used is a dataset in the form of pictures of fruits that will be collected by researchers, the use of pictures of fruits due to the varied colors and various shapes and also easier data collection because this research requires quite a lot of image datasets.

2. RESEARCH METHODOLOGY

The research method that will be used in this research are comprised of fruit image data collection, data preprocessing, CNN architecture design, training and validation, evaluation of model accuracy, object detection and MAP value evaluation.

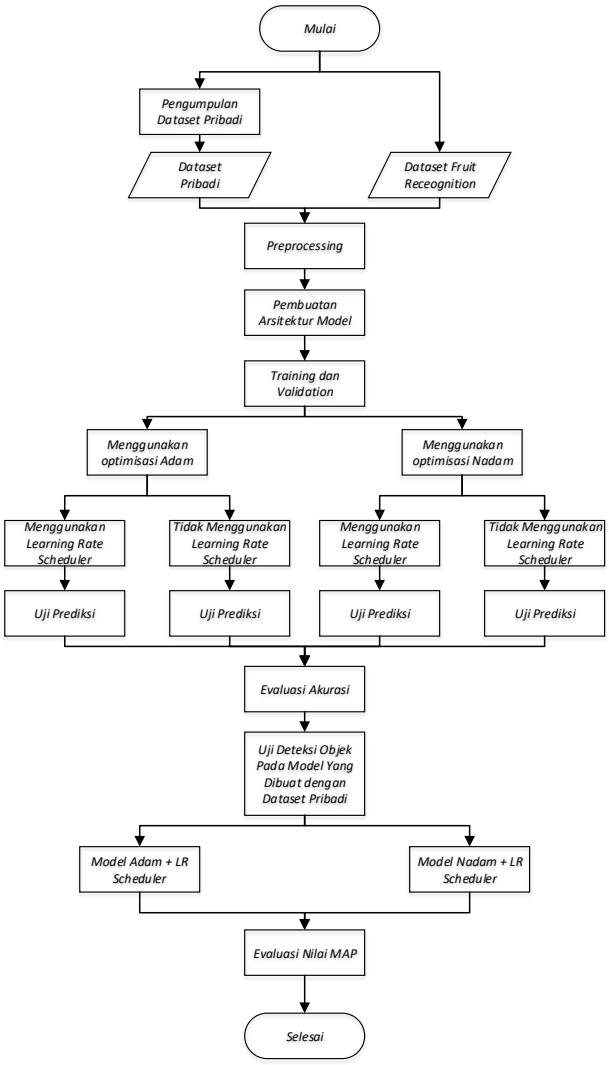

Figure 1 Research Methodology

2.1 Data Collection

Taking pictures for personal datasets is done in video format (.mp4) 30 FPS using a 13MP+5MP android smartphone camera. The image is taken from the top position but the fruit used is rotated and adjusted so that it can produce various image data. The collected video data is then converted into an image format (.png) according to the FPS of the video that has been taken. All image data that has been collected will be selected and deleted data which is a duplication and then resized to a size of 320x180 pixels.

Each fruit image that has been collected will be put into a folder with its own fruit label. There are 10 labels used, namely: [0] Apple, [1] StarFruit, [2] DragonFruit, [3] Guava, [4] Orange, [5] Kiwi, [6] Mango, [7] Melon, [8] Banana, [9] Tomato. The personal dataset and fruitrecognition dataset will be preprocessed again and resized to a size of 224x224 pixels before being trained. And for testing, apart from using a personal dataset, the fruitrecognition dataset created by Chris Gorgolewski will also be used.

2.2 Data Preprocessing

The preprocessing steps carried out for the modeling phase. Some preprocessing that necessary are image selection, duplication cleaning, changing the data format, resizing and image labelling.

2.3 CNN Architecture Design

The model is made with reference to the VGG16 architectural model. The architecture created has 10 output labels so it is necessary to make adjustments to the VGG16 architectural model, namely by lowering the filter on the convolution layer and also the dense layer value. The architecture created will use 16 filters at stage 1, 32 filters at stage 2, 64 filters at stage 3, 128 filters at stage 4, 256 filters at stage 5 and then it will be flattened so that it becomes a dimensional vector then a dense layer will be given. a value of 2 times the number of filters in the last layer to 512 which then ends with a softmax output with a value of 10 according to the number of labels.

2.4 Training and Validation

Modeling is done by testing and learning data on the collected fruit dataset. And to conduct experiments, researchers divide the dataset for training and prediction with a ratio of 80:20. Next, divide the training dataset to do training and validation with a ratio of 80:20. After the model is completed, the model will be trained using Adam and Nadam optimization and also training using the learning rate scheduler and also not using the learning rate scheduler on the Adam and Nadam standard settings.

2.5 Evaluation of Model Accuracy

Evaluation of model accuracy is done using python after the training process is complete. Evaluation of accuracy and also a prediction test using 20% of the data that has been divided and using the Multi Class Confusion Matrix technique to find the value of Prediction Accuracy, Precision, F1 Score and Recall.

2.6 Object Detection

Object detection performance testing is carried out using data from outside the dataset, object detection trials can only be performed on datasets labeled manually using labelImg so testing will only be performed on models created on private datasets.

2.7 MAP (Mean Average Precision) Value Evaluation

To find out the comparison of model performance, the MAP value will be calculated on the model that has been tested.

3. RESULTS AND DISCUSSION

3.1 Results

3.1.1 Collecting Data

he data was obtained using the rear camera of the cellphone with specifications of 13 megapixels plus 5 megapixels. The data is collected in video format (.mp4) with a size of 1980 pixels multiplied by 1080 pixels and the duration of video capture is at least 3 minutes with a Frame Per Second 30 (FPS), the video is captured on a clean background, the content of the video is adjusted to the needs of data collection. These data contain various fruits. The detail of the datasets is given in Table 1.

Table 1 Video detail

Number	Video File Name	Label of Fruits	Duration (Min : Sec)	Video FPS
1	Apel 1.mp4	Apple	03:18	5940
2	Apel 2.mp4	Apple	03:44	6720
3	Apel 3.mp4	Apple	03:08	5640
4	Belimbing.mp4	StarFruit	05:02	9060
5	BuahNaga 1.mp4	DragonFruit	03:54	7020
6	BuahNaga 2.mp4	DragonFruit	01:53	3390
7	JambuBiji 1.mp4	Guava	02:06	3780
8	JambuBiji 2.mp4	Guava	03:21	6030
9	Jeruk 1.mp4	Orange	02:29	4290
10	Jeruk 2.mp4	Orange	04:04	7320
11	Jeruk 3.mp4	Orange	02:29	4470
12	Kiwi.mp4	Kiwi	05:23	9690
13	Mangga.mp4	Mango	05:12	9360
14	Melon.mp4	Melon	05:03	9090
15	Pisang 1.mp4	Banana	01:05	1950
16	Pisang 2.mp4	Banana	02:09	3870
17	Pisang 3.mp4	Banana	04:11	7530
18	Tomat.mp4	Tomato	05:23	9690

The next step is preprocessing the data to ensure that the data is ready for the modeling process.

3.1.2 Image Selection and Duplication Cleaning

Data preprocessing is carried out before the training and data validation stages in order to simplify, improve results and ease the training process and data validation.

Table 2 Image Selection and Duplication Cleaning

Number	Label	Total Data	Total Duplication	Total Clean Data
1	Apple	18300	4572	13728
2	StarFruit	9060	1330	7730
3	DragonFruit	10410	1943	8467
4	Guava	9810	1072	8738

5	Orange	16080	5407	10673
6	Kiwi	9690	2996	6694
7	Mango	9360	4523	4837
8	Melon	9090	2685	6405
9	Banana	13350	3616	9734
10	Tomato	9690	2490	7200

Then the remaining images will be taken as many as 2100 images per label for a total of 21000 images. Each image will be included in its respective label folder.

3.1.3 Image Labelling

Data labeling is carried out in 3 stages, the first stage is carried out by collecting images according to the labels in their respective folders in order to facilitate data input. Later this folder will help in assigning numbers to each label before being made into a dataframe.

After being collected in one label folder, then stage 2 labeling is carried out using LabelImg which is one of the Python packages. LabelImg is downloaded from https://github.com/tzutalin/labelImg, labeling is done manually on each image and labels will be saved in image (.xml) format. Stage 2 labeling is done in order to increase the accuracy of data prediction in performing object detection.

Stage 3 labeling is carried out on Jupyter Notebook by assigning a number label to each image according to its respective label on the dataframe that has been created and then the data is randomized.

3.1.4 CNN Architecture

In order to get good CNN classification results, the architecture will follow the VGG-16 Architecture in modeling but use a smaller feature map to match the many labels and computer specifications used. The architecture created can be seen in Figure 2.

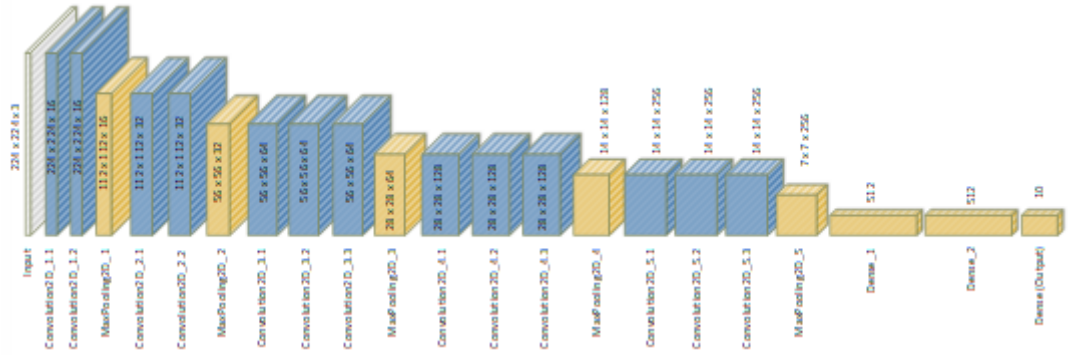

Figure 2 Architecture Based on VGG16

The filter size, stride, activation and padding used follow from the VGG-16 model, namely filter size with a size of 3x3, stride convolution 1, stride pooling 2, activation ReLu and the same padding. In accordance with VGG-16 in the architecture there will be 5 sets of Convolution layers which at the end of each Convolution set there is a Max Pooling layer, after that there are 2 Dense layers and ends with a Dense layer output with Softmax Activation.

3.1.5 Image Classification using CNN

Below are the result of the personal dataset that has gone through the training and validation process using Convolutional Neural Network from the test results with Learning Rate Scheduler and also the one that only uses the Standard Learning Rate (0.001) / without using the Learning Rate Scheduler.

Table 4 Personal Dataset Result

Model	Training		Validation		Prediction	Time (Sec)
	Accuracy	Loss	Accuracy	Loss		
Adam + LRS	97,78%	7,92%	99,35%	2,17%	98,85%	3782
Adam tanpa LRS	9,86%	230,28%	9,52%	230,30%	9,73%	3774
Nadam + LRS	93,60%	43,52%	95,68%	33,15%	95,02%	4756
Nadam tanpa LRS	10,31%	230,29%	10,33%	230,31%	9,61%	3729

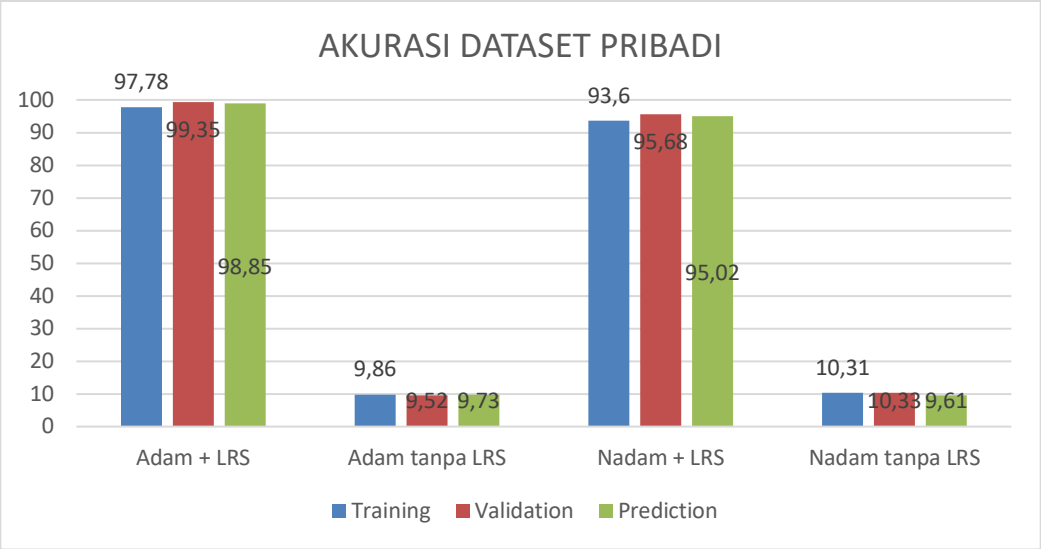

Figure 3 Personal Dataset Result

Below are the result of the fruitrecognition dataset that has gone through the training and validation process using Convolutional Neural Network from the test results with Learning Rate Scheduler and also the one that only uses the Standard Learning Rate (0.001) / without using the Learning Rate Scheduler.

Table 5 Fruitrecognition Dataset Result

Model	Training		Validation		Prediction	Time (Sec)
	Accuracy	Loss	Accuracy	Loss		
Adam + LRS	88,01%	35,38%	92,92%	22,22%	92,64%	3865
Adam tanpa LRS	9,68%	230,29%	9,52%	230,30%	9,73%	3827
Nadam + LRS	91,07%	54,08%	96,10%	20,18%	95,40%	4828
Nadam tanpa LRS	10,01%	230,28%	9,52%	230,30%	9,73%	3837

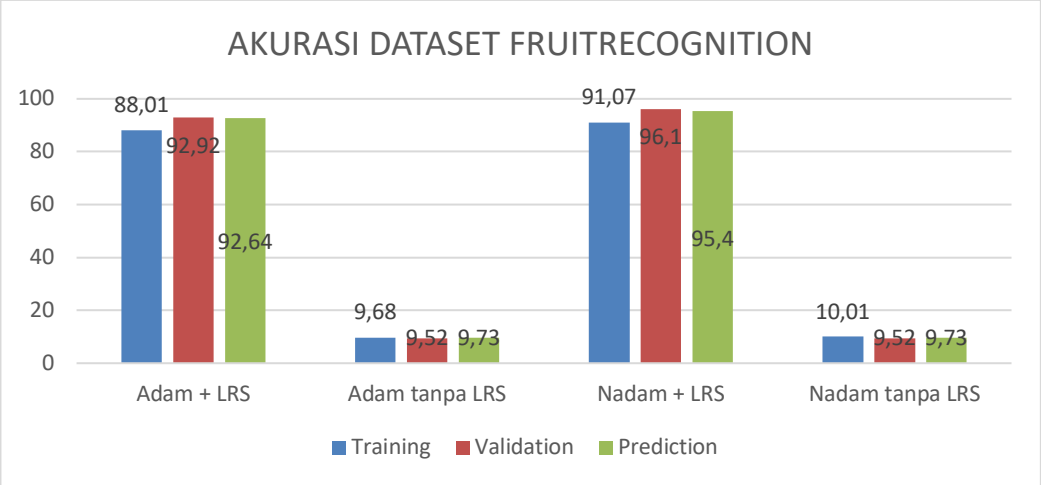

Figure 4 Fruitrecognition Dataset Result

3.1.6 Object Detection (MAP Evaluation)

After that, a search for the MAP value will be carried out on the model that was successfully trained using a private dataset, this is because the images in the private dataset have been manually labeled using the LabelImg package, this allows the trained model to detect objects. As explained earlier, the model that will calculate the MAP value is the first and third model, starting with determining the number of detections of True Positive, False Negative and False Negative in this process a confidence value will appear which represents how sure the model detects and finds the features of the fruit that have been trained, confidence values do not need to be considered to find the MAP value.

Table 6 MAP Value Result

No	Model	MAP
1	Pribadi + Adam + LRS	0.9358 (93.58%)
2	Pribadi + Nadam + LRS	0.7528 (75.28%)

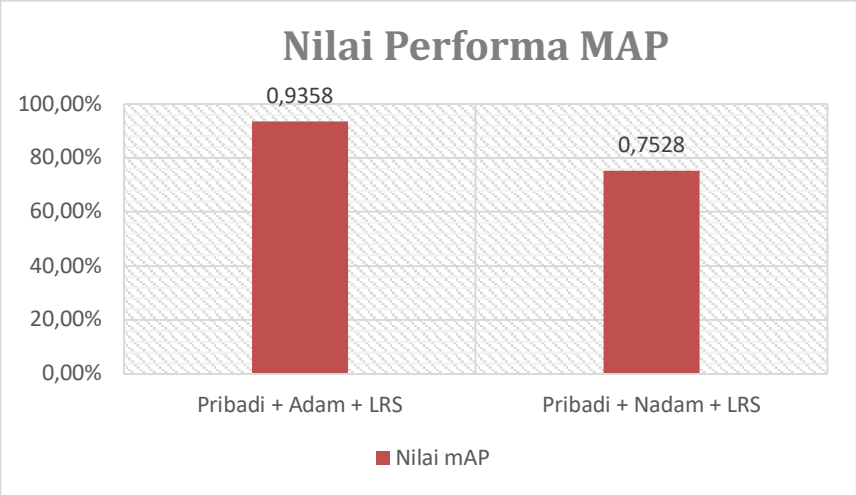

Figure 5 MAP Value Result

3.2 Discussion

The dataset used in fruit detection is a personal dataset collected by the researcher. The data is first collected in (.mp4) format per label taken at least 3 minutes in duration with a frame of 30 FPS, then converted to an image format (.png) which has a total of 114,840 image files (.png).

The data that has been collected then goes through the preprocessing stage to make it easier for the model to conduct training on the data, the preprocessing carried out includes checking for duplication, resizing, changing the format and labeling the data. Duplication checking is carried out according to research conducted in the research journal Cahyanti (2015) using python by converting image data into Hash MD5 form which is then compared to the hamming distance of each image. duplication removed. After checking for duplication of data, 2,100 images per label were selected for a total of 21,000 images. The selected data is then resized to a size of 320x180 pixels and the image file (.png) is converted to a format (.jpg). After the data is resized and put into a folder according to their respective labels, the data is then manually labeled with a bounding box using the LabelImg package. Data that has been labeled and placed in their respective folders can be used for training and validation.

After the data ratio has been divided, training can be carried out on the model, training will be carried out on 2 types of datasets, namely dataset 1 is a personal dataset and dataset 2 is a fruit recognition dataset created by Chris Gorgolewski. The two datasets have the same amount of data and labels so that comparisons can be made on the resulting models.

After that, a search for the MAP value will be carried out on the model that was successfully trained using a private dataset, this is because the images in the private dataset have been manually labeled using the LabelImg package, this allows the trained model to detect objects. As explained earlier, the model that will calculate the MAP value is the Adam with Learning Rate Scheduler and Nadam with Learning Rate Scheduler, starting with determining the number of detections of True Positive, False Negative and False Negative in this process a confidence value will appear which represents how sure the model detects and finds the features of the fruit that have been trained, confidence values do not need to be considered to find the MAP value.

The MAP performance produced by the two models in fruit detection is the model using Adam has a MAP value as high as 93.58% and the model using Nadam has a MAP value as high as 75.28%.

4. CONCLUSION

The CNN classification performed on the model that was trained using only Adam optimization with a learning rate of 0.001 resulted in a training accuracy value as high as 9.86%, with a training time of 3774 seconds (62 minutes) and prediction accuracy as high as 9.73%. Then the classification carried out on the trained model using only Nadam optimization with a learning rate of 0.001 resulted in a training accuracy value as high as 10.31%, with a training time of 3729 seconds (62 minutes) and prediction accuracy as high as 9.61%.

And the CNN classification performed on the model trained using Adam

optimization and learning rate control resulted in a training accuracy value as high as 97.78%, with a training time of 3782 (63 minutes), and a prediction accuracy as high as 98.85%. Classification carried out on the model trained using Nadam optimization and learning rate control resulted in a training accuracy value as high as 93.60%, with a training time of 4756 seconds (79 minutes) and prediction accuracy as high as 95.02%.

REFERENCES

[1] Albelwi, S., & Mahmood, A. (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. Entropy, 242(9).

[2] Arifin, A. Z., & Kurniati, W. (2002). Penggunaan Analisa Faktor Untuk Klasifikasi Citra Penginderaan Jauh Multispektral. JUTI: Jurnal Ilmiah Teknologi Informasi, 12(1).

[3] Arrofiqoh, E., & Harintaka. (2018). Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi. Geomatika, 61(24).

[4] Castelluccio, M., Poggi, G, Sansone, C., & Verdoliva, L. (2015). Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. arXiv, 1508.00092.

[5] Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B., & Aigbavboa, C. O. (2018). A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks. International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS).

[6] Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. IEEE Transactions on Image Processing.

[7] Dubey, S. R., Chakraborty, Soumendu., Roy, S. K., & Mukherjee, S. (2020). diffGrad: An Optimization Method for Convolutional Neural Networks. IEEE Transactions on Neural Networks and Learning Systems.

[8] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338.

[9] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv, 1412.6980.

[10] Muhaemen, M., Faisal, M. R., Nugrahadi, D.T., Farmadi, A., Herteno, R. (2020). Implementation Transfer Learning Convolutional Neural Network For Pothole Detection On Drone Video. Journal of Data Science and Software Engineering,1.

[11] Musso, D. (2020). Stochastic gradient descent with random learning rate. arXiv, 2003.06926.

[12] Poggio, T. A., Liao, Q., Banburski, A. (2020). Complexity control by gradient descent in deep networks. Nature, 11(1).

[13] Sakib, S., Ashrafi, Z., & Siddique, A. B. (2019). Implementation of Fruits Recognition Classifier using Convolutional Neural Network Algorithm for Observation of Accuracies for Various Hidden Layers. arXiv, 1904.00783.

[14] Zhao, Zhong-Qiu., Zheng, P., Xu, Shou-Tao., & Wu, X. (2019). Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11).