The Effect of Batch Size and Epoch on Performance of ShuffleNet-CNN Architecture for Vegetation Density Classification

by Novita Sari

The Effect of Batch Size and Epoch on Performance of ShuffleNet-CNN Architecture for Vegetation Density Classification

Yuslena Sari
Doctoral Department of Agricultural Science, Department of Information Technology, Universitas Lambung Mangkurat
yuzlena@ulm.ac.id

Yudi Firmanul Arifin
Faculty of Forestry, Universitas Lambung Mangkurat
yudifirmanul@ulm.ac.id

Novitasari Novitasari*
Faculty of Engineering, Universitas Lambung Mangkurat
novitasari@ulm.ac.id

Mohammad Reza Faisal
Department of Compter Science, Universitas Lambung Mangkurat
reza.faisal@ulm.ac.id

ABSTRACT

Land cover is an important factor in geographic analysis, ranging from physical geography studies, approaches to sustainable planning to environmental analysis. Vegetation analysis according to the Indonesian National Standard (SNI 7645:2014) is classified based on density. In the technical aspect to obtain information related to vegetation, this can be done using remote sensing. Remote sensing uses two data to obtain information, namely satellite data and Unmanned Aerial Vehicle (UAV) data. This study use UAV data with shooting locations in the Liang Anggang Protected Forest in classifying land cover. The method used is Convolutional Neural Network. This research is used the ShuffleNet v2 architecture on the CNN method. the comparison of epochs starts from 10, 25, 50, 75 and 100, which were tested with a number of different batch sizes showing the results that epoch 75 got higher results than the number of other epochs. The results obtained using epoch 75 on batch size 32 obtained 91.1% results, while the number of batch sizes 64 using epoch 75 totals obtained 91% results. Comparison of the results of batch size obtained different results where the results using batch size 32 were higher than batch size 64. By using the *ShuffleNet* v2 model with batch size 32, further testing was carried out using the matrix available in the Python programming language scikit-learn library, and the highest result was obtained at epoch 75 with a yield of 97.44%.

CCS CONCEPTS

• **Computing methodologies**; • **Artificial intelligence**; • **Image Processing and Pattern Recognition**;

KEYWORDS

Keywordword: CNN, classification, class, ShuffleNet

ACM Reference Format:
Yuslena Sari, Yudi Firmanul Arifin, Novitasari Novitasari*, and Mohammad Reza Faisal. 2022. The Effect of Batch Size and Epoch on Performance of ShuffleNet-CNN Architecture for Vegetation Density Classification. In *7th International Conference on Sustainable Information Engineering and Technology 2022 (SIET '22), November 22, 23, 2022, Malang, Indonesia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3568231.3568239

SIET '22, November 22, 23, 2022, Malang, Indonesia

ACM ISBN 978-1-4503-9711-7/22/11…$15.00
https://doi.org/10.1145/3568231.3568239

1 INTRODUCTION

Land cover is important information regarding the current status and current trends of land changes, especially in peatlands to evaluate the impact of changes, support the formulation and implementation of peatland management policies [1]. Land cover is a parameter of land drought index in peatlands [2]. Studying the condition of peatland surface vegetation cover requires an effective and efficient monitoring method. The algorithm that interprets the visual image is the key to determining the vegetation density class. Considerable work to detect the presence of vegetation from imagery, and classify vegetation density has been carried out in the fields of robotics, remote sensing, and computer vision.

In robotics, vegetation detection is dominated by the use of a vegetative index which usually consists of the ratio between color pixel values and/or near-infrared (NIR) images [3–5]. The downside of this approach is that it often requires special hardware because infrared and image capture require close distances. Beyond the vegetative index, researchers have explored the transformation of raw color information (RGB) to a more suitable color space. Philipp and Rath [6] found that the Lab, Luv, and HSV color spaces were very effective in distinguishing between vegetation and non-vegetation. Vegetation cover with the development of data technology from remote sensing has substantially increased the resolution and accuracy of vegetation cover detection [7], [8]. Disadvantages of remote sensing data processing can be time consuming, data is not always available, requires training, and special software.

The methods used in determining vegetation density include visual interpretation methods, pixel-based digital classification methods and object-based classification methods [9]. The use of data mining methods is an area of interest for land cover analysis researchers. Classification methods have been widely used in land classification. Machine learning has many conventional classification methods such as Support Vector Machine (SVM), Naive Bayes (NB), Convolutional Neural Network (CNN), etc. CNN is one of the most outstanding classification methods because of its accuracy in terms of classification and feature extraction [10].

Convolutional Neural Network (CNN) can learn human-level solutions for specific visual tasks. This method has been used especially in remote sensing image analysis tasks, including object detection in imagery, image recording, scene classification, segmentation, object-based image analysis, land use and land cover classification [11]. Convolutional Neural Network (CNN) is one of the Deep Learning methods that has recently developed. This method has proven effective in pattern recognition and object classification [12]. Previous studies that have carried out land cover classification using the CNN method yielded satisfactory accuracy results of 73% - 98% [13–17].

Convolutional Neural Network (CNN) has many popular architectures used, for example LeNet5 (1998), AlexNet (2012), ZFNet (2013), GoogleNet (2014), ResNet (2015), FractalNet (2016), ShuffleNet (2018) and other architectures. other. Previous research has done a comparison on the use of architecture on CNN in the field of classification. The architecture that has been compared shows the advantages and disadvantages of each, for the architecture that is widely used in the field of image classification and is quite recent and has been compared with several other architectures is ShuffleNet. ShuffleNet is a very efficient CNN architecture and has a better accuracy rate [17]. Research that has used the ShuffleNet architecture and has made comparisons with other architectures such as GoogleNet, DenseNet, MobileNet, Xception, IGCV2, EffNet V1, EffNet V2, IoTNet-3-5 and ResNet50 in the classification process states that the ShuffleNet architecture improves accuracy from 82% - 98% with less memory usage and faster processing time [11], [18–26].

The field of deep learning in conducting land cover classification is widely mentioned and applies the CNN method. The feature extraction used in this research is GLCM. This research uses the ShuffleNet architecture on the CNN method. This research is conducted in the Liang Anggang Protected Forest area, Banjarbaru area block 1 with targeted data collection for a month. The selection of this research location was based on observations made during the observation and survey of the block 1 area where the land use pattern for the block 1 area based on DishutProv in 2017, an area of 479 hectares of block 1 area is filled with land such as agriculture, plantations, roads and settlements and an area of 494 hectares full of weeds.

2 LITERATURE REVIEW

2.1 Color Space

The number of possible colors is denoted by M0 in the color space where the color space is very large on the order of hundreds of thousands of colors. The framework assumes that the number of distinctive colors in an image should be limited. Therefore, a discrete procedure is required to extract the set M limited to only typical colors denoted by L from the entire color space. A discrete process with L would be:

$$L = \{L(1), \ldots, L(M)\}.$$

In this work, we use the classic conversion of the RGB color space to the set of Gray Levels, also called Intensity.

2.2 RGB

RBG image is a color image whose representation is more complex and varied. The RBG image consists of a 3-dimensional array, M x N x 3. M and N are the dimensions of the length and width of the image, while 3 is the number of color channels. There are 3 color channels, namely red (R), green (G), and blue (B). each color channel contains an 8-bit value, which indicates a red, green and blue scale between 0 and 255. Each pixel has 3 color channel components to represent color. The combination of 3 8-bit color channels produces 24-bit numbers so there are 16,777/216 color combinations. An alternative representation is to use 32 bits per pixel by inserting a 4th channel called the alpha channel. This channel is used to measure the transparency per pixel which is commonly used in image editing. RGB image with R, G, and B components.

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is an algorithm used in data processing with a grid structure, one of which is a two-dimensional image and is also capable of processing high-dimensional data such as video. Convolutional Neural Network (CNN) is the development of Multiplayer Perceptron (MLP) so that it is designed to process two-dimensional data. CNN is included in the type of Deep Learning because of its high network depth and widely applied to image data. CNN works similar to a standard neural network. The difference is that CNN uses a two-dimensional kernel or the height dimensions of each unit in the CNN layer to be convoluted. Convolution is an operation of linear algebra which multiplies the matrix of the filter on the image to be processed.

2.4 *ShuffleNet Architecture*

ShuffleNet is a neural network architecture that is very efficient and has a better speed of accuracy due to the nature of the network itself which is designed for small networks, so that the computing process tends to be light on even low-power devices [27]. The ShuffleNet network is designed for small networks with a bottleneck design principle, then applied to mobile devices with limited computing power, this network has an efficient architecture by adding the map channels feature to provide more data that can work on very small networks [28]. ShuffleNet is a compute efficient CNN architecture, designed for mobile devices with 10-150 MFLOP (mega floating-Point operations per second) computing power. ShuffleNet uses Pointwise group convolution and channel shuffle to reduce computational costs while maintaining accuracy [21].

2.5 *Epoch*

Epoch is when the entire dataset has gone through the training process on the Neural Network until it is returned to the beginning for one round, because one epoch is too large to be fed into the computer, therefore we need to divide it into small units (batches).

Figure 1: Proposed method flow

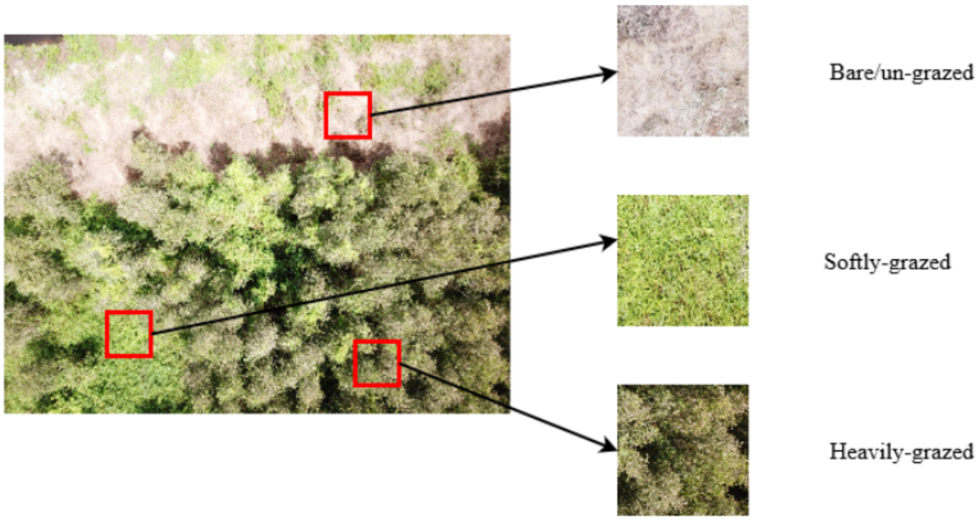

Figure 2: Pre-processing Data

Previously, an epoch comparison was made to determine the accuracy and validation results of each training with a different number of epochs. The purpose of this epoch comparison is to get the best model. The number of epochs compared starts from 10, 25, 50, 75 and 100 [26], [28–30]. The use of epoch greatly affects the resulting accuracy [31]. Epoch can improve accuracy and the resulting accuracy is stable, therefore it is necessary to use the right epoch in training data so that the accuracy obtained is maximum [31].

2.6 *Batch Size*

Batch Size is the number of data samples that are propagated to the Neural Network. Example: if we have 100 datasets and our batch size is 5 then this algorithm used the first 5 data samples from the 100 data we have (1st, 2nd, 3rd, 4th, and 5th) then distributed or trained by the Neural Network until it is finished then take back the second 5 data samples from 100 data (6th, 7th, 8th, 9th, and 10th), and so on until the 20th 5th data sample (1005=20).

3 METHODOLOGY

This research has a methodological flow starting from image acquisition, preprocessing, classification, and validation. The flow is depicted in Figure 1.

3.1 Data Acquisition

The data used for this research was obtained from drone image data. The drone brand used is DJ Mavix Pro, with a height of 20 meters above the object. Vegetation cover image data was taken from the Liang Anggang Protection Forest Block I Banjarbaru. The data used is 3000 data with 1000 classes for un-grazed or bare classes, 1000 for softly-grazed classes, and 1000 for heavily-grazed classes.

3.2 Pre-Processing

Pre-processing data is one of the important activities in doing work related to image analysis. Image conditions and segments that differ from class are discarded to reduce image preprocessing time. First, draw the snippet for the required section of a class. Then, removing unwanted portions of the image reduces the initial processing time [25]. Figure 2 shows the results of preprocessing.

3.3 Classification

Image classification is a process of compiling, sorting or grouping a pixel in several classes based on a criterion or object category, each pixel in each class is assumed to have homogeneous characteristics [32]. The purpose of the classification process is to extract the spectral patterns contained in the image. The classification used in this study is a Supervised Classification with the Convolutional Neural Network method using the ShuffleNet architecture.

3.4 CNN Models

Convolutional Neural Network is a deep learning method where the algorithm works by taking input images and rolling them with filters or kernels to extract features. An NxN image is wrapped with the FxF filter and the convolution operation learns this feature. It slides the window after each operation and its features are studied by the feature map. Only the CNN neural network can process the grid structure data, for example, two-dimensional images. Layer convolution is an operation of linear algebra which produces a matrix of filters in the image to be processed. One type of many layers that can be owned in a network is usually called a convolution layer process.

The image entered into the CNN classification model that has been made through the fit model stage will produce an output that is calculated using the optimized weights. Almost all CNN Architectures follow the same general design principle of applying convolution layers sequentially to the input, periodically down sampling (Max pooling) the spatial dimensions while increasing the number of feature maps. In addition, there are also fully connected layers, activation functions and loss functions (e.g., cross entropy or softmax).

However, among all CNN operations, the convolution layer, pooling layer, and fully connected layer are the most important. The Convolutional layer is the first layer where CNN can extract features from the image. Since pixels are only related to adjacent and close pixels, convolution makes it possible to maintain relationships between different parts of an image. Convolution is filtering an image with a smaller pixel filter to reduce the image size without losing the relationship between pixels. When applying convolution to a 7x7 image using a 3x3 filter in 1x1 steps (1-pixel shifts at each step), we get a 5x5 output.

3.5 *ShuffleNet* v2

ShuffleNet v2 is a sophisticated net architecture. In each block, half of the channel passes through the block and immediately joins the next block, although this can be considered as a kind of reuse feature, it reduces nonlinearity and expressibility to some extent. The initial convolutional layer (Conv1) is a type of CNN layer consisting of 24 channel filters measuring 3x3 with a step value (stride) 2. The next stages are stages 2, 3 and 4, each of which consists of 2 stages of shuffle with their respective strides. 2 and 1 respectively as in Figure 2.5. The second shuffle stage of each stage 2, 3 and 4 is repeated up to 2, 4, and 2 times. GConv is a convolution layer based on points with a 1x1 filter and ends with BN normalization. Depth Wise Convolution (DWConv) is a 3x3 convolution with stride 2 and terminated by BN normalization [26].

The performance process of the Shuffle channel starts from a convolution group, each of which receives convolutional results from another group, so that each group on the Shuffle channel consists of a combination of channels from different groups which results in strengthening the correlation. The features generated from the previous group can be divided into subgroups, so that the next convolution stage can be divided into different combinations. It is implemented efficiently and elegantly by the shuffle unit.

3.6 Performance

Performance results from this study were measured based on accuracy, F1 Score, Precision, Recall, Hamming Loss, Jaccard Similarity, Matthews Correlation Coefficient, and Zero One Loss. The accuracy equation is as shown in (1).

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{1}$$

And the formula to calculate precision is as shown in (2).

$$\text{Precision} = \frac{(TP)}{(TP + FP)} \tag{2}$$

The formula to calculate recall is shown in (3).

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \tag{3}$$

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

The formula to calculate F1 Score is shown in (4).

$$\text{F1 Score} = \frac{2\ x\ precision\ x\ recall}{precision + recall} \tag{4}$$

The formula to calculate Hamming Loss is shown in (5).

$$\text{Hamming Loss} = \frac{1}{nL}\sum_{i=1}^{n}\sum_{j=1}^{L}\left[I\left(y_j^{(i)} \neq \hat{y}_j^{(i)}\right)\right] \tag{5}$$

- Hamming Loss computes the proportion of incorrectly predicted labels to the total number of labels.
- For a multilabel classification, we compute the number of False Positives and False Negative per instance and then average it over the total number of training instances.

Where:
n : Number of training examples
$y_j^{(i)}$: true labels for the ith training example and jth class
$\hat{y}_j^{(i)}$: predicted labels for the ith training example and jth class
The formula to calculate Jaccard Similarity is shown in (6).
Jaccard Similarity

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

Where:
J = Jaccard Distance
A = set 1
B = set 2

The formula to calculate Matthews Correlation Coefficient is shown in (7).

$$\text{MCC} = \frac{(TP\ X\ TN) - (FP\ X\ FN)}{\sqrt{(TP + FP)\ (TP + FN)\ (TN + FP)\ (TN + FN)}} \tag{7}$$

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

Lastly, the formula to calculate Zero One Loss is shown in (8).

$$\text{L0/1 (h)} = \frac{1}{nL}\sum_{i=1}^{n}\delta_{h(x_i)\neq y_i}\ ,\ where\ \delta_{h(x_i)\neq y_i} = \begin{cases} 1, & \text{if h } (x_i) \neq y_i \\ 0, & \text{o.w.} \end{cases} \tag{8}$$

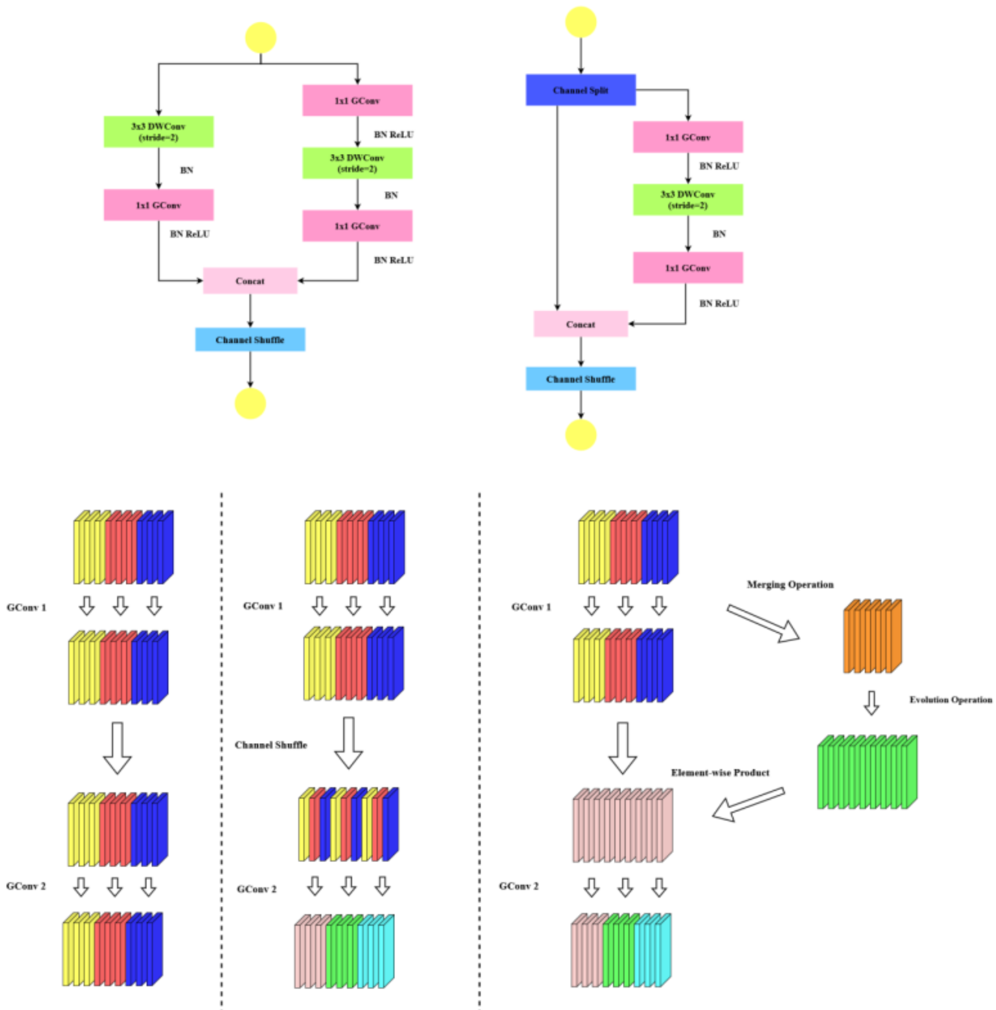

Figure 3: ShuffleNet Architecture Performance Process [26]

4 RESULT AND DISCUSSION

4.1 Dataset

Research data collected in the form of 3000 (three thousand) image data, each of which consists of 1000 data for each class, namely 1000 data for images with un-grazed labels, 1000 data for images with softly-grazed labels, and 1000 data for heavily-grazed or tightly labeled image. Of the 3000 data is divided randomly between training data and test data, with a percentage of 70% for training data and the remaining 30% for test data. So that we get 2100 training data and 900 test data. From each data, train accuracy and validation accuracy tests are carried out.

4.2 Image Pre-Processing Implementation

The image pre-processing stage is carried out using the library contained in the Python programming language. The library used is TORCHVISION. In the torchvision library there is a transforms2 module which is used for transformations to make image processing easier. The steps taken in the pre-processing image are resizing or changing the image size to 256 x 256. Followed by center cropping, which is trimming the image in the middle to a predetermined size. Furthermore, the image is converted into a tensor form, which is a multidimensional data arrangement. Finally, the image is normalized with the aim of increasing accuracy in the image recognition process.

Table 1: First test attempt

No.	*Number of Epoch*	*Batch Size*	*Train Loss*	*Train Accuracy*	*Validation Loss*	*Validation Accuracy*
1.	10	64	0.461	0.8047	0.590	0.821
2.	25	64	0.105	0.9642	0.8345	0.834
3.	50	64	0.025	0.9905	0.4006	0.89
4.	75	64	0.016	0.9947	0.5079	0.91
5.	100	64	0.015	0.9967	0.5909	0.893

Table 2: Second test attempt

No.	*Number of Epoch*	*Batch Size*	*Train Loss*	*Train Accuracy*	*Validation Loss*	*Validation Accuracy*
1.	10	32	0.31812	0.86381	0.33423	0.8578
2.	25	32	0.12782	0.95523	0.35935	0.8589
3.	50	32	0.09076	0.96523	0.40089	0.87
4.	75	32	0.03325	0.99047	0.31051	0.911
5.	100	32	0.00320	0.99809	0.50444	0.898

4.3 Convolutional Neural Network Implementation (CNN)

In deep learning, CNN is a type of neural network that is used for feature extraction and image classification. The model used on CNN is ShuffleNet v2 which is an optimization of ShuffleNet v1. The use of ShuffleNet v2 in the Python programming language using the torchvision library. The input parameters used in ShuffleNet v2, namely: the number of classes that exist in the image data, the size of the image, and the type of network which are used to select how many channels are used in the ShuffleNet v2 model.

4.4 Accuracy Test

ShuffleNet v2 model neural network testing was conducted to test training data and test data. The data is divided into 70:30 to get train loss, train accuracy, validation loss, and validation accuracy. Loss on train and validation is used in machine learning optimization. Loss is performed to assess how well the ShuffleNet v2 Model performs in training and testing. Loss is the number of errors that occur in each training and validation set, which can be interpreted as how bad or good the model is when iterations are run. While the accuracy is used to measure the accuracy of the model in performing the classification (prediction) [33–35].

The results of train loss, train accuracy, validation loss, and validation accuracy can be seen in the table 1.

There are hyper-parameters in ShuffleNet v2, some of them are Number of Epoch and Batch Size. The Number of Epoch is the number of times the training data set is displayed to the neural network during training. While the batch size is the number of samples that were processed before the model was updated. The number of epochs and batch size were increased until the validation accuracy decreased.

Based on the table 2, the highest accuracy results are obtained at the number of epochs of 75. Compared to the previous table, the comparison of batch sizes that has been carried out, batch size 32 with the number of epochs of 75 has the highest accuracy, seen in the validation accuracy of 91.1%. Batch 32 gives better results than batch 64, this experiment explains that smaller batches are able to give better results than larger batches. By using the ShuffleNet v2 model with a batch size of 32, further testing was carried out using the matrix available in the Python programming language scikit-learn library. From the library, look for the values of accuracy, f1 score, precision, recall, hamming loss, jaccard score, Matthew Corrcoef, and zero one loss. Where in the confusion matrix, accuracy is the correct prediction value of the entire data. Precision is a true positive predictive value compared to the overall positive predicted result. Recall is a true positive predictive value compared to all true positive data. The F1 score is a weighted comparison of the average precision and recall.

Then the haming loss which is part of the incorrectly predicted label. Jaccard or often called Jaccard similarity is a value used to compare the similarity between a set of labels and the predicted result set. Then, the Matthews correlation coefficient which can be interpreted in machine learning as a measure of classification quality. Finally, the zero one loss value is generally used to find the loss value in the classification. From there, the following evaluation results were obtained:

Based on the table above, the highest result is still the 75th epoch with an accuracy value of 97.44%.

5 CONCLUSION

Based on the research results obtained from testing by applying the CNN method, it was found that the comparison of epochs starting from 10, 25, 50, 75 and 100, which were tested with different batch sizes showed the result that epoch 75 got higher results than the number of other epochs. The results obtained using epoch 75 on batch size 32 obtained 91.1% results, while the number of batch sizes 64 using epoch 75 totals obtained 91% results. Comparison of the results of batch size obtained different results where the results using batch size 32 were higher than batch size 64. By using the

Table 3: Third test attempt

Epoch	*Accuracy*	*F1 Score*	*Precision*	*Recall*	*Haming Loss*	*Jaccard Similarity*	*Matthews Correlation Coefficient*	*Zero One Loss*
10	0.81888	0.81208	0.85605	0.81888	0.18111	0.70256	0.75132	0.18111
25	0.91555	0.91443	0.92951	0.91555	0.08444	0.84807	0.88113	0.08444
50	0.97111	0.97108	0.97109	0.97111	0.02888	0.94432	0.95668	0.02888
75	**0.97444**	**0.97436**	**0.97450**	**0.97444**	**0.02555**	**0.95031**	**0.96177**	**0.02555**
100	0.97	0.96993	0.97042	0.97	0.03	0.94221	0.95527	0.03

ShuffleNet v2 model with batch size 32, further testing was carried out using the matrix available in the Python programming language scikit-learn library, and the highest result was obtained at epoch 75 with a yield of 97.44%.

ACKNOWLEDGMENTS

This research was supported by funding from the DRTPM Research Program of the Ministry of Education, Culture, Research, and Technology of Indonesia. Main Contract Number: 026/E5/PG.02.00.PT/2022, Derivative Contract Number: 042/UN8.2/PG/2022

REFERENCES

[1] X. Giam, L. P. Koh, H. H. Tan, J. Miettinen, H. T. W. Tan, dan P. K. L. Ng, "Global extinctions of freshwater fishes follow peatland conversion in Sundaland," *Front. Ecol. Environ.*, 2012, doi: 10.1890/110182.
[2] N. Novitasari, J. Sujono, S. Harto, A. Maas, dan R. Jayadi, "Drought index for peatland wildfire management in central kalimantan, indonesia during el niño phenomenon," *J. Disaster Res.*, vol. 14, no. 7, hal. 939–948, 2019, doi: 10.20965/jdr.2019.p0939.
[3] J. Gai, L. Tang, dan B. Steward, "Plant localization and discrimination using 2D+3D computer vision for robotic intra-row weed control," 2016. doi: 10.13031/aim.20162460814.
[4] D. V. Nguyen, L. Kuhnert, dan K. D. Kuhnert, "Spreading algorithm for efficient vegetation detection in cluttered outdoor environments," *Rob. Auton. Syst.*, 2012, doi: 10.1016/j.robot.2012.07.022.
[5] S. Haug, A. Michaels, P. Biber, dan J. Ostermann, "Plant classification system for crop /weed discrimination without segmentation," *2014 IEEE Winter Conf. Appl. Comput. Vision, WACV 2014*, hal. 1142–1149, 2014, doi: 10.1109/WACV.2014.6835733.
[6] I. Philipp dan T. Rath, "Improving plant discrimination in image processing by use of different colour space transformations," *Comput. Electron. Agric.*, 2002, doi: 10.1016/S0168-1699(02)00050-9.
[7] J. Miettinen, C. Shi, dan S. C. Liew, "Two decades of destruction in Southeast Asia's peat swamp forests," *Front. Ecol. Environ.*, 2012, doi: 10.1890/100236.
[8] B. Trisakti, A. Julzarika, U. C. Nugroho, D. Yudhatama, dan Y. Lasmana, "Can the Peat Thickness Classes Be Estimated From Land Cover Type Approach?," *Int. J. Remote Sens. Earth Sci.*, vol. 14, no. 2, hal. 93, 2018, doi: 10.30536/j.ijreses.2017.v14.a2677.
[9] M. Guo, J. Li, C. Sheng, J. Xu, dan L. Wu, "A review of wetland remote sensing," *Sensors (Switzerland)*. 2017. doi: 10.3390/s17040777.
[10] J. Yu dan L. Zhao, "A novel deep CNN method based on aesthetic rule for user preferential images recommendation," *J. Appl. Sci. Eng.*, vol. 24, no. 1, hal. 49–55, 2020, doi: 10.6180/jase.202102_24(1).0006.
[11] G. W. Lindsay, "Convolutional neural networks as a model of the visual system: Past, present, and future," *J. Cogn. Neurosci.*, vol. 33, no. 10, hal. 2017–2031, 2021, doi: 10.1162/jocn_a_01544.
[12] H. Aljuaid, N. Alturki, N. Alsubaie, L. Cavallaro, dan A. Liotta, "Computer Methods and Programs in Biomedicine Computer-aided diagnosis for breast cancer classification using deep neural networks and transfer learning," vol. 223, 2022, doi: 10.1016/j.cmpb.2022.106951.
[13] Z. Xu, K. Guan, N. Casler, B. Peng, dan S. Wang, "A 3D convolutional neural network method for land cover classification using LiDAR and multi-temporal Landsat imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 144, hal. 423–434, 2018, doi: 10.1016/j.isprsjprs.2018.08.005.
[14] E. Miranda, A. B. Mutiara, Ernastuti, dan W. C. Wibowo, "Forest classification method based on convolutional neural networks and sentinel-2 satellite imagery," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 19, no. 4, hal. 272–282, 2019, doi: 10.5391/IJFIS.2019.19.4.272.
[15] S. M. Hamylton *dkk.*, "Evaluating techniques for mapping island vegetation from unmanned aerial vehicle (UAV) images: Pixel classification, visual interpretation and machine learning approaches," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 89, no. March, hal. 102085, 2020, doi: 10.1016/j.jag.2020.102085.
[16] Y. Hu, Q. Zhang, Y. Zhang, dan H. Yan, "A deep convolution neural network method for land cover mapping: A case study of Qinhuangdao, China," *Remote Sens.*, vol. 10, no. 12, hal. 1–17, 2018, doi: 10.3390/rs10122053.
[17] S. Watanabe, K. Sumi, dan T. Ise, "Identifying the vegetation type in Google Earth images using a convolutional neural network: a case study for Japanese bamboo forests," *BMC Ecol.*, vol. 20, no. 1, hal. 1–14, 2020, doi: 10.1186/s12898-020-00331-5.
[18] S. Pan *dkk.*, "Land-cover classification of multispectral LiDAR data using CNN with optimized hyper-parameters," *ISPRS J. Photogramm. Remote Sens.*, vol. 166, 2020, doi: 10.1016/j.isprsjprs.2020.05.022.
[19] A. M. Censi, D. Ienco, Y. J. E. Gbodjo, R. G. Pensa, R. Interdonato, dan R. Gaetano, "Attentive Spatial Temporal Graph CNN for Land Cover Mapping from Multi Temporal Remote Sensing Data," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3055554.
[20] A. K. Rangarajan dan H. K. Ramachandran, "A fused lightweight CNN model for the diagnosis of COVID-19 using CT scan images Enhanced Reader," *Automatika*, 2022.
[21] X. Zhang, X. Zhou, M. Lin, dan J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, hal. 6848–6856, 2018.
[22] N. Ma, X. Zhang, H. T. Zheng, dan J. Sun, "Shufflenet V2: Practical guidelines for efficient cnn architecture design," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11218 LNCS, hal. 122–138, 2018, doi: 10.1007/978-3-030-01264-9_8.
[23] G. Losapio *dkk.*, "Lightweight and efficient convolutional neural networks for recognition of dolphin dorsal fins," hal. 68–72, 2020.
[24] Z. Wang dan L. Ma, "SYOLO: An Efficient Pedestrian Detection," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 768, no. 7, 2020, doi: 10.1088/1757-899X/768/7/072067.
[25] G. Liu *dkk.*, "I3d-shufflenet based human action recognition," *Algorithms*, vol. 13, no. 11, 2020, doi: 10.3390/a13110301.
[26] R. Gaetano, D. Ienco, K. Ose, dan R. Cresson, "A two-branch CNN architecture for land cover classification of PAN and MS imagery," *Remote Sens.*, vol. 10, no. 11, 2018, doi: 10.3390/rs10111746.
[27] Q. L. Zhang dan Y. Bin Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021, vol. 2021-June. doi: 10.1109/ICASSP39728.2021.9414568.
[28] N. L. Husni *dkk.*, "Real-time littering activity monitoring based on image classification method," *Smart Cities*, vol. 4, no. 4, 2021, doi: 10.3390/smartcities4040079.
[29] A. Anand, S. Rani, D. Anand, H. M. Aljahdali, dan D. Kerr, "An efficient CNN-based deep learning model to detect malware attacks (CNN-DMA) in 5G-IoT healthcare applications," *Sensors*, vol. 21, no. 19, 2021, doi: 10.3390/s21196346.
[30] A. Y. Saleh dan E. Liansitim, "Palm oil classification using deep learning," *Sci. Inf. Technol. Lett.*, vol. 1, no. 1, 2020, doi: 10.31763/sitech.v1i1.1.
[31] T. Karim *dkk.*, "Cnn applied in public transport for the protection against the covid-19 spread," *Int. J. Eng. Trends Technol.*, vol. 69, no. 10, 2021, doi: 10.14445/22315381/IJETT-V69I10P205.
[32] J. Song, S. Gao, Y. Zhu, dan C. Ma, "A survey of remote sensing image classification based on CNNs," *Big Earth Data*, vol. 3, no. 3, 2019, doi: 10.1080/20964471.2019.1657720.
[33] S. Salman dan X. Liu, "Overfitting Mechanism and Avoidance in Deep Neural Networks," 2019.

[34] R. C. Hidayatullah dan S. Violina, "Convolutional Neural Network Architecture and Data Augmentation for Pneumonia Classification from Chest X-Rays Images," *Int. J. Innov. Sci. Res. Technol.*, vol. 5, no. 2, 2020.

[35] R. A. Pramunendar, D. P. Prabowo, D. Pergiwati, Y. Sari, P. N. Andono, dan M. A. Soeleman, "New workflow for marine fish classification based on combination features and CLAHE enhancement technique," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 4, hal. 293–304, 2020, doi: 10.22266/IJIES2020.0831.26.

The Effect of Batch Size and Epoch on Performance of ShuffleNet-CNN Architecture for Vegetation Density Classification

ORIGINALITY REPORT

17%	12%	12%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

1%

★ Rahul Chauhan, Kamal Kumar Ghanshala, R.C Joshi. "Convolutional Neural Network (CNN) for Image Detection and Recognition", 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		