JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website Jurnal : http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN : 2085-3688; e-ISSN : 2460-0997

Early Detection of Deforestation through Satellite Land Geospatial Images based on CNN Architecture

Nor Kumalasari Caecar Pratiwi[1*], Yunendah Nur Fu'adah[2], Edwar[3]
[1,2,3] School of Electrical Engineering, Telkom University
[1,2,3] 01 Telekomunikasi Street, Bandung, Jawa Barat 40257, Indonesia
*Corresponding email: caecarnkcp@telkomuniversity.ac.id

Received 09 March 2021, Revised 30 March 2021, Accepted 08 May 2021

Abstract — This study has developed a CNN model applied to classify the eight classes of land cover through satellite images. Early detection of deforestation has become one of the study's objectives. Deforestation is the process of reducing natural forests for logging or converting forest land to non-forest land. The study considered two training models, a simple four hidden layer CNN compare with Alexnet architecture. The training variables such as input size, epoch, batch size, and learning rate were also investigated in this research. The Alexnet architecture produces validation accuracy over 100 epochs of 90.23% with a loss of 0.56. The best performance of the validation process with four hidden layers CNN got 95.2% accuracy and a loss of 0.17. This performance is achieved when the four hidden layer model is designed with an input size of 64 × 64, epoch 100, batch size 32, and learning rate of 0.001. It is expected that this land cover identification system can assist relevant authorities in the early detection of deforestation.

Keywords – land cover classification, CNN, AlexNet, deforestation detection

I. INTRODUCTION

Numerous researchers and scientists conducting a study onto land cover classification through satellite imagery in order to monitor deforestation. Deforestation is the process of reducing natural forests for logging or converting forest land to non-forest land. Between 2000 and 2012, at least 2.3 million km^2 of trees are significantly cut worldwide, equating to 2×10^5 km^2 yearly [1]. Deforestation has dramatically altered land cover in Indonesia over the last thirty years, posing a threat to biodiversity and many plant and animal species [2]. Deforestation has several adverse effects, including habitat destruction and biodiversity loss, decreased water quality, air pollution, and greenhouse gas emissions that contribute to climate change [3]. Kalimantan, known as the Earth's lungs, the overall pattern of deforestation rates has been declining from 1990 to 2015, but the rate of deforestation in West and North Kalimantan tends to be larger [4].

An investigation regarding converting low to high resolution images in the process of classifying land satellite images by Reddy and Parvathy [5] is one of the studies related to land cover classification that has been carried out. This study categorize a large amount of data into various classes with bolster vector machine and Artificial Neural Network method. In contrast, 2d convolution methods, histogram generation, and edge detection are used in the localization, segmentation, and feature extraction processes. The findings of this study do not include any information about the method's performance. The analysis of the percentage of decline in water supplies and a rapid increase in buildings in each region is all that is described in the research study.

Devi and Chib are also developing a study related to satellite image classification by using a perceptron neural network (PNN) [6]. Principal Components Analysis (PCA) is used to extract essential features. The color palette is used to extract major significant characteristics since color contributesto identifying features in satellite images. Water, soil, and vegetation are the three types of objects under investigation. The classification results were investigated for 16 × 16 and 8 × 8 block sizes, with training accuracy of 100% in both cases using a collection of 150 known images. The overall classification accuracy results for three classes were 87.3% for 16 × 16 block-sized and 96,9% for 8 × 8 block-sized, using 150 test/unknown images.

In 2018, Fedoseev conducted a research project to develop an appropriate multi-stage methodology for the conceptual classification of hyper-spectral satellite images [7]. In this study, they used two hyperspectral images: Indian Pines and Pavia University. The results show that the SVM-RBF classifier can perform well; however, to accommodate the spectral and spatial relationships between images, a k-means ++ segmentation algorithm should be applied. On the two testing images, the selected methods increased classification accuracy by up to 11% (for the localized training set) and 42.5% (for the random training set).

Al-Ghrairia, Abedb, Fadhilc, and Naserd evaluate the characterization of satellite images using remote sensing based on color features [8]. In their study, for each image block, the color moment's features, mean, standard deviation, and skewness are extracted as a vector and stored in a 2D array. The K-Means algorithm is used to group these features and select the most appropriate clusters within the resulted features. When classes are determined by spectral distinctions inherent in the data, the K–means clustering algorithm based on the moment features classification method is effective. It is observed that overall classification accuracy is enhanced into 92.12% in classifying the class of rivers, agriculture area, buildings with vegetation, buildings without vegetation, and bare lands.

Studies related to the classification of satellite images using a convolutional neural network have been carried out by Rai and his colleagues in 2020 [9]. In their investigation, the PCA method was used to reduce dimensions of fused images, and CNN was applied as a classifier. Snow, water, cloud, vegetation, and urban are the five categories that make up the entire dataset. The calculation of the kappa coefficient and other accuracy metrics is used to evaluate the proposed system's performance. The results showed that the kappa coefficient value was 0.8, and the overall accuracy was 94.5%.

Kadhim and Abed have used CNN to classify satellite imagery from the SAT4, SAT6, and UC Merced Land datasets [10]. There are six classes in the SAT4 and SAT6 datasets, while The UC Merced Land dataset is divided into 21 separate classes. There are four CNN architectures investigated in this study; AlexNet, VGGNet-19, GoogleNet, and ResNet. The study's findings show that the ResNet architecture provides the best system performance, with the accuracy of 95.8%, 94.1%, and 98 % for each dataset, respectively.

Using the CNN approach through Google Earth images, Watanabe and colleagues described the vegetation categories (Japan Bamboo forest and Non-Bamboo Forest) [11]. Sanyo Onoda, Ide, and Isumi, three separate bamboo forest locations, were used for the analysis. More than 90% of targets can be detected correctly using trained models. As a result, when opposed to traditional machine learning approaches, CNN recognition has higher accuracy.

Miranda and colleagues published the results of a study that used the CNN approach to classify forests based on the Sentinel-2 Satellite imagery [12]. The main forests classification divided into three classes, those are primary dry forest, secondary dry forest, and plantation forest. The results showed that by using CNN with image features such as NDVI, Brightness, GLCM homogeneity, and Rectangular fit, the classification method yielded a high overall accuracy of 97.66%. Compared to GBT, which has an overall accuracy of 95.50%, there was a small increase in overall accuracy.

This study aims to establish a system capable of classifying the EuroSat images into eight different classes: Forest, Highway, Industrial, Pasture, Residential, River, Sea-Lake, and Vegetation. The results of this study should be able to substitute the review process of satellite image classification studies. The system uses a convolutional neural network (CNN) approach to classify objects in real-time. The CNN architecture used in previous studies was quite complex. As shown by the number of hidden layers, we propose a structure assisted by a simple structure in this analysis. The suggested model is a simple CNN with four hidden layers and fully connected layers, followed by a comparative analysis of Alexnet architecture. The testing process is carried out in several scenarios to evaluate the hyperparameter's impact that can produce the best system performance. Input size, epoch, batch size, and learning rate are all hyperparameters that will be evaluated. It is expected that this land cover identification system can assist relevant authorities in early detection of deforestation.

II. RESEARCH METHODS

A. Dataset

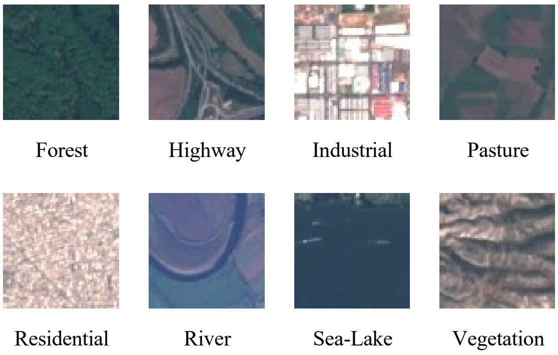

Fig.1. Examples of Images from the Dataset for Each Class

EuroSat land geospatial images were used to build the dataset for this study [13]. The dataset consists of overall 21500 images, divided into eight classes. This dataset came from the Sentinel-2 satellite. The amount of 75% from the total 21500 images will be used as training data, while the remaining will be used as testing data.

B. Proposed Method

Like multilayer perception machines and support vector machines, traditional machine learning techniques rely on limited configurations to manage a small number of samples and computing units [14]. When the target objects have complex classification problems, this traditional machine learning are obviously insufficient.

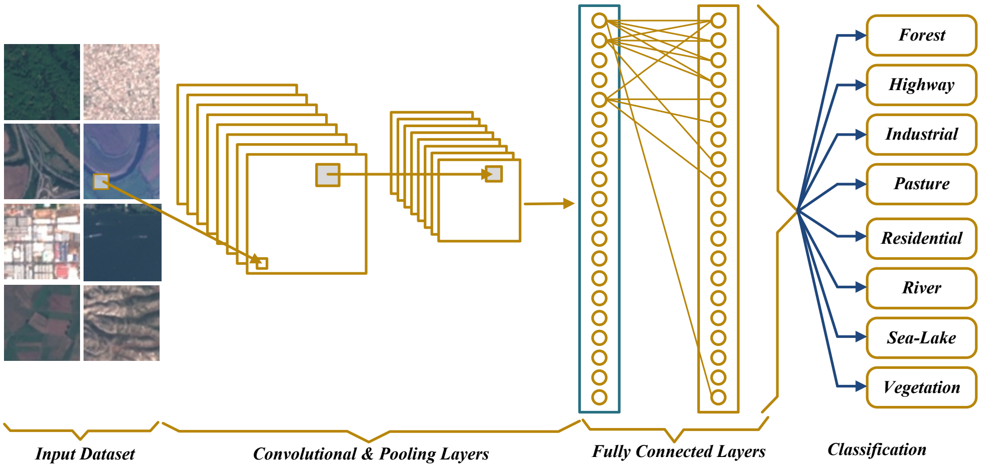

Fig.2. The System General Scheme

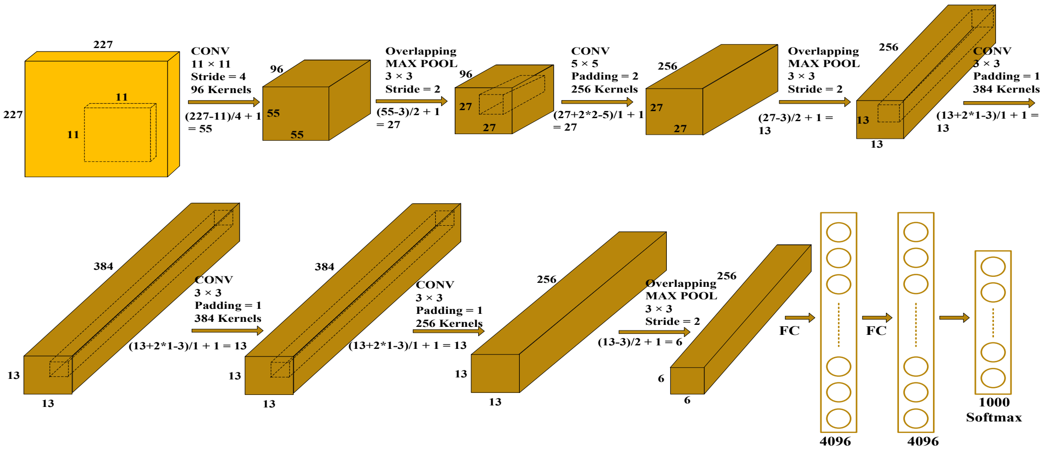

Fig.3. The AlexNet Architecture

The architectures to be used are Alexnet and Simple CNN model with four hidden layers and fully connected layers. AlexNet is the architecture used in a paper published by Alex Kriszhevsky in 2012 [15]. In the ImageNet LSVRC-2010 contest, Alexnet used to classify 1.2 million images into 1000 different classes. Alexnet was a massive breakthrough in the field of machine learning and computer vision for visual classification and recognition purposes, and it led to the introduction of an explosion in interest in deep learning [16].

AlexNet contains eight learned layers, divided into the first five convolutional layers followed by three fully connected layers. The output of the final fully-connected layer is fed into a 1000-way softmax, which is then distributed among the 1000 class labels. Only the kernel maps in the previous layer are bound to the kernels of the second, fourth, and fifth convolutional layers. Both kernel maps in the second layer are bound to the kernels of the third convolutional layer. The neurons in the fully connected layers are all connected to the neurons in the layer before them. Any convolutional and fully connected layer's output is subjected to the ReLU non-linearity. The input image with the size of 224 × 224 × 3 will be filtered in the first convolutional layer with 96 kernels of size 11 × 11 × 3 and stride of 4 pixels. Then, the output will be filtered with 256 kernels of size 5 × 5 ×48 on the second convolutional layer. The output of the second convolutional layer will be connected to the third convolutional layer with 384 kernels of size 3 × 3 × 256. The fourth and fifth convolutional layer has 384 kernels of size 3 × 3 × 192 and 256 kernels of size 3 × 3 × 192, respectively.

The second proposed architecture is a simple CNN model, with four hidden layers and a fully connected layer. Any convolutional and fully connected layer's output is subjected to the ReLU activation. The input image will be filtered in the first until fourth convolutional layer with the number of kernels is 8, 16, 32, and 64, respectively. Each hidden layer used Maxpooling2D with a pool and stride size of 2 × 2. The fully connected layer using a dropout value of 0.5 and Softmax activation.

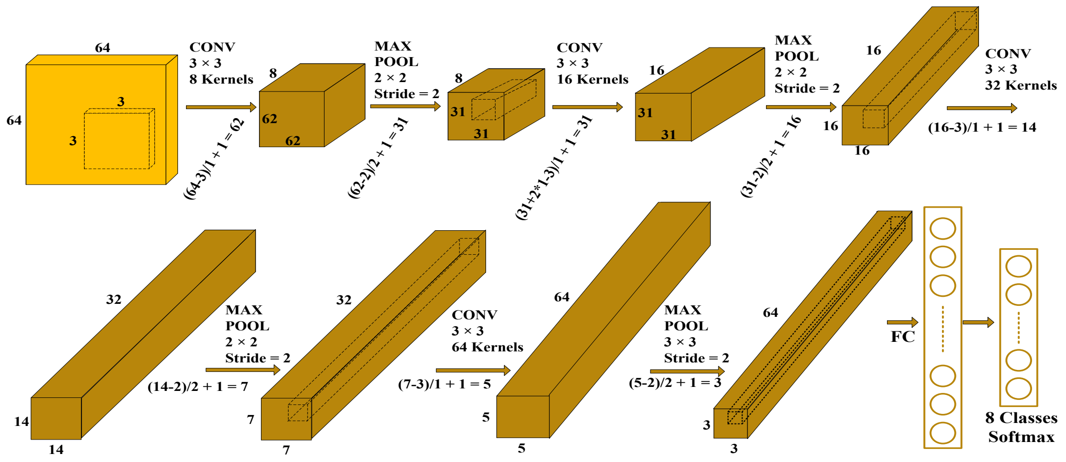

Fig.4. The CNN Architecture with Four Hidden Layer and a Fully Connected Layer

C. Performance Evaluation

The accuracy, precision, recall, and f1-score values are used to evaluate the system performance. The data on the confusion matrix is used to visualize these values. Each element in the confusion matrix represents the number of predictions made by the model that could be classified as true or false.

Table 1. Confusion Matrix of Eight Classes Classification

		Predicted							
		Forest	***Highway***	***Industrial***	***Pasture***	***Residential***	***River***	***Sea-Lake***	***Vegetation***
Actual	Forest	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
	Highway	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
	Industrial	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
	Pasture	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
	Residential	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
	River	x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}
	7Sea-Lake	x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}
	Vegetation	x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}

From the confusion matrix, we can evaluate the value of Total of False Negative (TFN), Total of False Positive (TFP), Total of True Negative (TTN), and Total of True Positive (TTP) for each class [17]:

$$\mathrm{TFN_i} = \sum_{j=1,\, j\neq i}^{n} x_{ij} \qquad (1)$$

$$\mathrm{TFP_i} = \sum_{j=1,\, j\neq i}^{n} x_{ji} \qquad (2)$$

$$\mathrm{TTN_i} = \sum_{j=1,\, j\neq i}^{n} \sum_{k=1,\, k\neq i}^{n} x_{jk} \qquad (3)$$

$$\mathrm{TTP_{all}} = \sum_{j=1}^{n} x_{jj} \qquad (4)$$

We can calculate the accuracy, precision, recall, and f1-score of the system by using (1) – (4).

$$\text{Precision} = \frac{\mathrm{TTP_{all}}}{\mathrm{TTP_{all}} + \mathrm{TFP_i}} \qquad (5)$$

$$\text{Recall} = \frac{\mathrm{TTP_{all}}}{\mathrm{TTP_{all}} + \mathrm{TFN_i}} \qquad (6)$$

$$\text{f1-Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \qquad (7)$$

$$\text{Accuracy} = \frac{\mathrm{TTP_{all}} + \mathrm{TTN_{all}}}{\text{Total Number of Testing Entries}} \qquad (8)$$

III. RESULTS

We performed the testing process with several scenarios to evaluate the impact of the hyperparameter that is capable of producing the best system performance.

The first scenario is carried out to investigate the effect of input size on output performance. At least three different input sizes are considered: 32x32, 64x64, and 128x128. Other hyperparameter values were made consistent when evaluating the effect of the input sizes, epoch 50, batch size 32, and a learning rating of 0.001. For both architectures, below are the evaluation results.

Table 2. The Accuracy and Loss Over Various Input Sizes

Architecture	Input Size	Accuracy (%)	Loss
Alexnet	32 × 32	77.95%	0.59
	64 × 64	78.77%	0.69
	128 × 128	82.97%	0.56
4 Hidden Layers CNN	32 × 32	89.82%	0.30
	64 × 64	95.20%	0.16
	128 × 128	94.69%	0.17

The test results show that the Alexnet architecture produces the best accuracy when the system receives input with a size of 128 × 128. In contrast, the 4 hidden layers of CNN architecture produce the highest accuracy by using input with 64 × 64. Since the system loses some valuable data whenever the input content is smaller, the performance suffers the consequences, and the alexnet architecture demonstrates this.

The first scenario accomplishment will serve as a reference point for the following scenario, which will examine the impact of the epoch hyperparameter on system performance. Epoch refers to a single loop of forwarding and backward propagation through the entire training dataset. Four different epoch values will be tested for comparison purposes, increasing from 10, 25, 50, to 100. Meanwhile, the batch size and learning rate hyperparameters hold 32 and 0.001, respectively, as in the previous scenario. One training epoch refers to the learning algorithm's single pass through the training dataset, which consisted of randomly selected "batch size" groups of samples.

Table 3. The Accuracy and Loss Over Various Epoch

Architecture	Input Size	Accuracy (%)	Loss
Alexnet	10	71.90%	0.74
	25	84.16%	0.57
	50	83.10%	0.53
	100	85.65%	0.72
4 Hidden Layers CNN	10	89.02%	0.31
	25	92.59%	0.22
	50	94.20%	0.18
	100	95.25%	0.17

For both Alexnet and 4 Hidden Layer CNN architectures, an epoch of 100 generates the highest accuracy, with 85.65% and 95.20%, respectively. In the following scenario, the impact of differences in batch size on system performance used an epoch of 100, and a learning rate of 0.001. Meanwhile, the tested batch size value varies from 16, 32, 64 to 128.

Table 4. The Accuracy and Loss Over Various Batch Size

Architecture	Input Size	Accuracy (%)	Loss
Alexnet	16	79.68%	0.98
	32	85.67%	1.06
	64	86.59%	0.91
	128	87.27%	0.74
4 Hidden Layers CNN	16	94.83%	0.24
	32	95.20%	0.17
	64	94.75%	0.18
	128	94.53%	0.16

The Alexnet architecture produces the best accuracy when the system using a batch size of 128. In contrast, the four hidden layers CNN architecture produces the highest accuracy by using a batch size of 32. The batch size indicates how many samples will be taken from the training dataset and estimate the error gradient before updating the model weights.

Table 5. The Accuracy and Loss Over Various Learning Rate

Architecture	Input Size	Accuracy (%)	Loss
Alexnet	0.01	13.97%	2.07
	0.001	88.07%	0.73
	0.0001	90.23%	0.56
4 Hidden Layers CNN	0.01	82.79%	0.51
	0.001	95.20%	0.17
	0.0001	90.17%	0.26

According to the overall testing results, Alexnet architecture generates the best performance while using 128 × 128 input data, a hyperparameter epoch value of 100, batch size 128, and a learning rate of 0.0001. The validation accuracy of the system is 90.23%, with a loss of 0.56.

Figure 5 indicates that the Alexnet architecture is completely over-fitting. The training loss was constantly decreasing while the validation loss doesn't, which implies that the system is complex enough to memorize the training data patterns. In such situations, it is crucial to regularize the model, such as reduce the number of neural network layers, minimize the number of parameters by reducing the number of neurons in each layer, and add more data for training if possible.

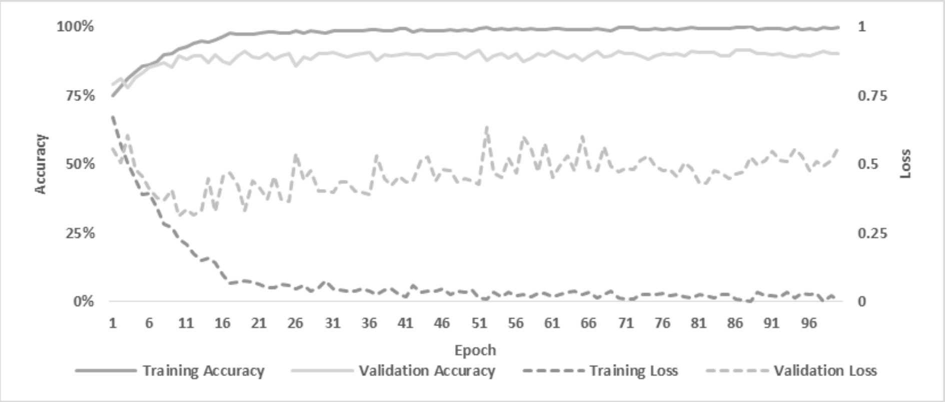

Fig. 5. The Accuracy and Loss for Training and Validation Phase in Alexnet Architecture

The accuracy, recall, and f1-score values can be identified in the confusion matrix.

Table 6. The Confusion Matrix of AlexNet Architecture

		Predicted							
		Forest	***Highway***	***Industrial***	***Pasture***	***Residential***	***River***	***Sea-Lake***	***Vegetation***
Actual	Forest	725	0	0	3	0	3	1	7
	Highway	0	552	21	12	17	38	0	32
	Industrial	0	7	551	0	36	3	0	12
	Pasture	33	9	0	391	3	16	1	37
	Residential	0	2	6	0	732	0	0	1
	River	3	66	4	10	0	540	0	9
	Sea-Lake	40	2	0	2	0	6	718	1
	Vegetation	11	16	9	13	22	10	1	641

Using equations 1 through 8, the overall value for precision, recall, and f1-score is 0.84.

Table 7. The Precision, Recall, and f1-score of AlexNet Architecture

	Precision	***Recall***	***f1-score***
Forest	0.89	0.93	0.91
Highway	0.99	0.89	0.94
Industrial	0.81	0.83	0.82
Pasture	0.73	0.63	0.68
Residential	0.80	0.81	0.80
River	0.85	0.94	0.89
Sea-Lake	0.79	0.81	0.80
Vegetation	0.91	0.88	0.89

Based on the results of Alexnet, the study was continued using CNN, but with a reduced number of layers, and a simpler CNN architecture with four hidden layers was proposed. The best results were obtained when the input size was 64 × 64, epoch 100, batch size 32, and the learning rate 0.001. In this condition, the system accuracy is 95.2%, with a loss of 0.17.

The model performance is not over-fitting, as shown by the graph in Fig. 6.

Overfitting occurs when a model learns too much information and noise from the training data. This means that the model picks up noise or random variations in the training data and learns them as concepts. The issue is that these principles do not extend to new data, limiting the models' ability to generalize.

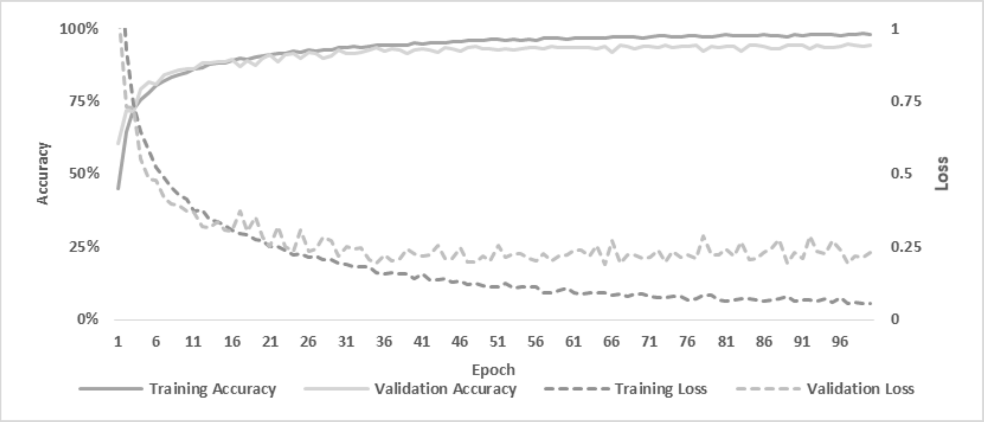

Fig 6. The Accuracy and Loss for Training and Validation Phase in 4 Hidden Layer CNN Architecture

We might use the confusion matrix to evaluate the degree of precision, recall, and f1-score for each class.

Table.8. The Confusion Matrix of 4 Hidden Layer CNN Architecture

		Predicted							
		Forest	***Highway***	***Industrial***	***Pasture***	***Residential***	***River***	***Sea-Lake***	***Vegetation***
Actual	Forest	715	1	0	20	0	2	0	6
	Highway	0	567	9	6	5	29	1	19
	Industrial	0	15	580	0	14	1	0	6
	Pasture	5	3	0	488	0	7	3	18
	Residential	0	4	7	1	731	0	0	1
	River	0	38	1	9	2	564	4	5
	Sea-Lake	5	0	0	1	0	3	730	0
	Vegetation	2	5	5	14	4	4	1	714

The precision, recall, and f1 score for each class can be calculated using the confusion matrix.

Table 9. The Precision, Recall, and f1-score of 4 Hidden Layer CNN Architecture

	Precision	**Recall**	**f1-score**
Forest	0.98	0.96	0.97
Highway	0.90	0.89	0.89
Industrial	0.96	0.94	0.95
Pasture	0.91	0.93	0.92
Residential	0.97	0.98	0.97
River	0.92	0.91	0.91
Sea-Lake	0.99	0.99	0.99
Vegetation	0.93	0.95	0.94

It's crucial to think for both precisions and recall while evaluate a model's effectiveness and to measure the quality of predictions. The forest class has the highest precision value, while the sea-lake has the highest recall value. The f1-score is the harmonic mean of the precision and recall of the model.

IV. DISCUSSION

We have gathered several discussions from this research based on the findings outlined in the previous section. The validation accuracy rate of the AlexNet architecture is 86.6%, which is very good, but it also turns out that this architecture has a very high loss rate of 0.673. Throughout this context, loss refers to the CNN neuron's inability to make accurate predictions. A loss function is used to measure the magnitude of this loss.

For the second model, a basic CNN with four hidden layers, we focused on four scenarios. The four scenarios are evaluated to see how system performance influenzed by input size, epoch, batch size, and learning rate. When the system is operated with an input size of 64 × 64, the epoch of 50, batch size 16, and the learning rate of 0.001, the system performs the best accuracy. Input size determined the volume of the input information system. The test results also indicate when the input size is 64 × 64, the system generates a higher accuracy and lower loss compare with input size 32 × 32. The less information entered, the more likely the system will lose critical data.

On the other hand, if a system receives too much information, it will have to be compensated by increasing the complexity. Large images take up more memory, but they also require a larger neural network. As a consequence, both the spatial and temporal complexity of the system is increased [18]. Modifications in image resolution can affect the visual information contained in the image. When the resolution of an image with simple visual information is decreased, there are no significant differences; however, when the visual information is complex, the disparity changes significantly [19].

The epoch implies that the model has completed its traversal of the entire dataset [20]. Under-fitting and over-fitting are two significant problems that epoch optimization must avoid. When the epoch value is small, the weight updating process fails to reach its optimal point. Excessive epoch, on the other hand, would only trigger weight to memorize training data. As a result, they can fail to recognize the test dataset's characteristics. As shown in Table 4, the greater epoch value, resulting in greater accuracy, but when the epoch 100 and accuracy, the value of loss also increased.

The number of images used to train a single forward and backward pass is referred to as batch size [21]. The smaller batch size cause the faster it converges since it doesn't have to go through any of the training data to update the weights. Fig.8 shows the study's results, demonstrating that the smaller batch size will produce a more significant system's accuracy.

Last test scenarios were executed to see how modifications in learning rate affected system performance. Learning rate is a configurable hyperparameter in neural network training that indicates how rapidly the model evolves to the problem. A small learning rate could even slow down convergence, whereas a high learning rate can avoid convergence by causing the loss function to fluctuate, become stuck in a local minimum, or even diverge [22]. It can be seen from the research results shown in Fig.9; the best accuracy is achieved when using a learning rate of 0.001, better than a lower learning rate of 0.0001 or a higher learning rate of 0.01.

V. CONCLUSSION

This study has developed a CNN model that is applied to classify the eight classes of satellite images. The study considered two training models, a simple four hidden layer CNN architecture compared with AlexNet. The Alexnet architecture generates the best performance while using 128 x 128 input size, with the epoch of 100, batch size of 128, and learning rate of 0.0001. The validation accuracy is 90.23%, with a loss of 0.56. The validation loss value increases relatively when compared to the training loss, indicating that the system is over-fitting. We also investigate the training variables of a simple four hidden layer CNN architecture. The performance of the training model with optimized parameters got 95.2% accuracy and a loss of 0.17. This performance is achieved when the model is designed with an input size of 64 × 64, epoch 100, batch size 32, and a learning rate of 0.001. This second architecture demonstrates that the model does not suffer from overfitting. So that the system can recognize new images, then be classified. It is expected that this land cover identification system can assist relevant authorities in the early detection of deforestation.

REFERENCES

[1] M. Bologna and G. Aquino, “Deforestation and world population sustainability: a quantitative analysis,” *Scientific Reports*, vol. 10, no. 1, pp. 1–9, May. 2020.

[2] D. Fuller, J. Miettinen, and E. Meijaard, “Deforestation, Degradation, and Forest Regrowth in Indonesia’s Protected Areas From 2000-2010,” *Indonesian Journal of Conservation*, vol. 2, no. 1, pp. 1–13, June. 2013.

[3] K. G. Austin, A. Schwantes, Y. Gu, and P. S. Kasibhatla, “What causes deforestation in Indonesia?,” *Environmental Research Letters*, vol. 14, no. 2, Feb. 2019.

[4] S. Wegscheider *et al.*, “Current achievements to reduce deforestation in Kalimantan,” *Indonesian Journal of Geography*, vol. 50, no. 2, pp. 109–120, Dec. 2019.

[5] P. V. G. Reddy and L. R. Parvathy, “Satellite Image Classification for Environmental Analysis using Image Processing,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 3, pp. 1513–1517, Jan. 2020.

[6] M. S. Devi and S. Chib, “Classification of Satellite Images Using Perceptron Neural Network,” *International Journal of Computational Intelligence Research*, vol. 15, no. 1, pp. 1–10, 2019.

[7] V. A. Fedoseev, “Hyperspectral satellite image classification using small training data from its samples,” *Journal of Physics: Conference Series*, vol. 1096, no. 1, April. 2018.

[8] A. H. T. Al-Ghrairi, Z. H. Abed, F. H. Fadhil, and F. K. Naser, “Classification of Satellite Images Based on Color Features Using Remote Sensing,” *International Journal of Computer*, vol. 31, no. 1, pp. 42–52, Sept. 2018.

[9] A. K. Rai, N. Mandal, A. Singh, and K. K. Singh, “Landsat 8 OLI Satellite Image Classification using Convolutional Neural Network,” *Procedia Computer Science*, vol. 167, pp. 987–993, 2020.

[10] M. A. Kadhim and M. H. Abed, "Convolutional neural network for satellite image classification", in *Studies in Computational Intelligence,* vol. 830, Springer International Publishing, 2020, pp. 165-178.

[11] S. Watanabe, K. Sumi, and T. Ise, “Identifying the vegetation type in Google Earth images using a convolutional neural network: a case study for Japanese bamboo forests,” *BMC Ecology*, vol. 20, no. 1, pp. 1–14, Nov. 2020.

[12] E. Miranda, A. B. Mutiara, Ernastuti, and W. C. Wibowo, “Forest classification method based on convolutional neural networks and sentinel-2 satellite imagery,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 19, no. 4, pp. 272–282, Dec. 2019.

[13] P. Helber, B. Bischke, A. Dengel, and D. Borth,

"Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, Feb. 2019.

[14] M. Xin and Y. Wang, "Research on image classification model based on deep convolution neural network," *Eurasip Journal on Image and Video Processing*, vol. 2019, no. 1, Feb. 2019.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *ACM International Conference Proceeding Series*, 2012, pp. 145–151.

[16] M. Z. Alom *et al.*, "The history began from AlexNet: A comprehensive survey on deep learning approaches," *arXiv*, 2018.

[17] C. Manliguez, "Generalized Confusion Matrix for Multiple Classes," no. November, pp. 2–4, 2016.

[18] M. Hashemi, "Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation," *Journal of Big Data*, vol. 6, no. 1, Nov. 2019.

[19] S. P. Kannojia and G. Jaiswal, "Effects of Varying Resolution on Performance of CNN based Image Classification An Experimental Study," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 9, pp. 451–456, Sept. 2018.

[20] S. Afaq and S. Rao, "Significance Of Epochs On Training A Neural Network," *International Journal of Scientific and Technology Research*, vol. 19, no. 6, pp. 485–488, June. 2020.

[21] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020.

[22] Y. Wu *et al.*, "Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks," in *2019 IEEE International Conference on Big Data, Big Data 2019*, 2019, pp. 1971–1980.