JURNAL INFOTEL
VOL. 16, NO. 1, FEBRUARY 2024, PP. 82–95. DOI:10.20895/INFOTEL.V16I1.1004

RESEARCH ARTICLE

Butterfly Image Classification using Convolution Neural Network with AlexNet Architecture

Ainin Maftukhah[1,*], Abdul Fadlil[2], and Sunardi[3]

[1]Master of Informatics Study Program, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia
[2,3]Electrical Engineering Study Program, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia

*Corresponding email: 2207048003@webmail.uad.ac.id

Received: July 19, 2023; Revised: November 17, 2023; Accepted: January 22, 2024.

Abstract: A lack of knowledge about butterflies can cause problems because butterflies play an essential role in the ecosystem. The urgency of this research was related to the field of biology, namely, the classification of butterfly images, which can help in understanding migration patterns, mating patterns, and behavior patterns of butterflies in interactions with the surrounding area. The purpose of this research was to classify butterfly species. The dataset used was a butterfly image dataset of 5,499 with 50 species. The method applied was a convolution neural network (CNN) with AlexNet architecture. The training process using the AlexNet architecture began with the input of the image dataset; the dataset would be preprocessed, such as resizing and RGB to grayscale. Then, do the filter or kernel. The output from the kernel was used to perform pooled convolution. Convolution and pooling were performed five times. Each of the last max pooling results was flattened three times to convert the matrix-shaped image into three dimensions. After that, it was fully connected. The last stage was that the image can be classified. The testing process using the AlexNet architecture began with the input of the image dataset; the dataset was preprocessed, such as resizing and RGB to grayscale. Then, the dataset was classified with the AlexNet architecture CNN. After that, an evaluation model was carried out, and finally, the results of the butterfly image classification. The classification results obtained an accuracy of 80 % with a 100 $\times$ 100 resize, 82 % with a 150 $\times$ 150 resize, and 82 % with a 200 $\times$ 200 resize.

Keywords: butterfly, classification, CNN, AlexNet

 ISSN: 2085-3688; e-ISSN: 2460-0997

1 Introduction

Butterflies are brightly colored insects with triangular wings, that belong to a large group of insects called Lepidoptera. Lack of knowledge about butterflies raises concern since they also play an essential role in our ecosystem. Due to the lack of general knowledge about butterfly species, there still needs to be more interest in butterfly conversion [1]. Butterfly image classification is urgent in biological research because it can provide insight into butterfly behavior. The main problem is the wide variation of butterflies' shape and color, makes species identification difficult. Image classification can be an invaluable tool for butterfly species identification by minimizing errors in grouping butterfly species. Information on butterfly species found in an area can provide insight into the ecology, environmental sustainability, and changes in butterfly populations. The urgency of butterfly image classification can help aid interactions with their ecosystems, which has an essential involvement in understanding and protecting biodiversity [2]. Butterflies can be distinguished by color, wing shape, and other features, which are extracted according to the included image. Then, the appropriate classification results are returned according to their similarity [3].

The development of artificial intelligence technology in image processing is proliferating. One of the available techniques is deep learning, which can detect and recognize an object in a digital image [4]. Finding patterns and classifying data becomes an interesting discussion from time to time. Pattern recognition solves problems related to recognition or classification, such as speech, document text, handwritten character classification, and batik pattern recognition.

Convolution neural network (CNN) have been widely used in visual pattern recognition systems. Research by Wu *et al.* [5] uses CNN to detect faces and facial expressions. Research by Grossberg [6] uses CNN surprise shut to detect images of eyes and faces. Although many image recognition algorithms are based on CNN, their recognition effect is perfect. However, many detection algorithms now use specialized databases to design network depths and layers. Through continuous exploration, the best parameters and optimization algorithms are found [7]. CNN has several VGG-16, VGG-19, GoogLeNet, ResNet, and AlexNet architectures that have been applied to image classification without involving the segmentation process or using the segmentation process [8].

The deep learning method that can recognize objects in an image is CNN. CNN capability is considered the best method for object detection and recognition. CNN has the same method as neural networks, consisting of neurons with weights, biases, and activation functions. The recognition technique using CNN can replace the human eye because of the accuracy, level of sharpness, and contrast in the image for precise results [9, 10]. CNN has a multi-layer arrangement consisting of a pooling layer and a fully connected layer. The CNN layer arranges neurons so that they have three dimensions [11, 12].

Several related studies have conducted research using the butterfly object and the CNN method but obtaining different accuracy values. Prayogi *et al.* [13] classified various clove images using deep learning with the CNN algorithm. The model results give a reading accuracy of 65.25 %. The CNN hyperparameter increases the accuracy of the test data reader to 87.75 %. Zhao *et al.* [3] performed butterfly recognition based on faster R-CNN. This study uses the object image of a butterfly. The average classification accuracy can reach 70.4 % when applying the R-CNN algorithm. Tang *et al.* [14] use deep learning for automated butterfly segmentation in the Leeds Butterfly Public dataset, demonstrating that this method outperforms sophisticated deep learning-based image segmentation approaches.

Annesa *et al.* [15] identified reptile species using CNN. This study uses the image of a reptile. This study achieved 93 % accuracy in identifying 14 different types of reptiles. Kato *et al.* [16] performed preprocessing using multiple steganography for intentional image downsampling on CNN-based steganalysis. Ghazal *et al.* [17] accurately detected non-proliferative diabetic retinopathy on optical coherence tomography images using CNN. Sunardi *et al.* [18] researched facial recognition systems in CNN-based room security. The object used is a facial image. The results of predictions with 100 % accuracy mean that all data can be identified correctly. The CAD learning-based transfer system achieves a promising accuracy of 94 %. The difference from related research is in the research object. The similarities from related research are the object of research using butterflies and the CNN method.

While classification models have evolved, improving accuracy in recognizing similar butterfly species remains essential. A deeper understanding of butterfly behavior and ecology through CNN methods is still a capability that has yet to be fully utilized. Increasing the classification accuracy of more sophisticated CNN models and more careful preprocessing methods will improve the accuracy in identifying butterfly species, especially those with significant visual similarity. This research focuses more on using image data to classify butterfly images with CNN AlexNet classification to determine the accuracy of accuracy values. The main contribution of this research is to improve classification accuracy, supporting the understanding of butterfly ecology.

This study uses classification to classify butterfly species. The image dataset consists of 5,499 RGB color images of butterflies with 50 species consisting of 10% testing and 90% training data. This study aims to classify butterfly species using the CNN method with the AlexNet architecture. The use of Alexnet architecture is because it has been tested to provide excellent performance in image processing.

2 Research Method

The research method is an overview of research conducted from start to finish, with the research flow shown in Figure 1.

The stages of the research can be seen in Figure 1a. First, the input dataset was tested, and then the dataset was preprocessed, such as resizing and converting RGB to grayscale. The output of the preprocessing dataset test was classified using CNN with the AlexNet architecture. The AlexNet architecture classification stage results were then used to carry out the data using an evaluation model. The final results of the data were classified according to the species.

The research stages in Figure 1b can be seen first when the input dataset is trained and then preprocessed, such as resizing and converting RGB to grayscale. The output from the preprocessing, the training dataset, was classified using the AlexNet architecture. The results of the preprocessing were done by convolution using filter 96. The output convolution was done by batch normalization or ReLU; after that, the results of the batch normalization were done by max-pooling. The convolution, batch normalization, and max-pooling stages were performed five times. Then, the last result of max-pooling was done flattening, and the result of flattening was done densely two times. The results of the dense evaluation model were carried out to get the best results. The output of the last stage of classification resulted.

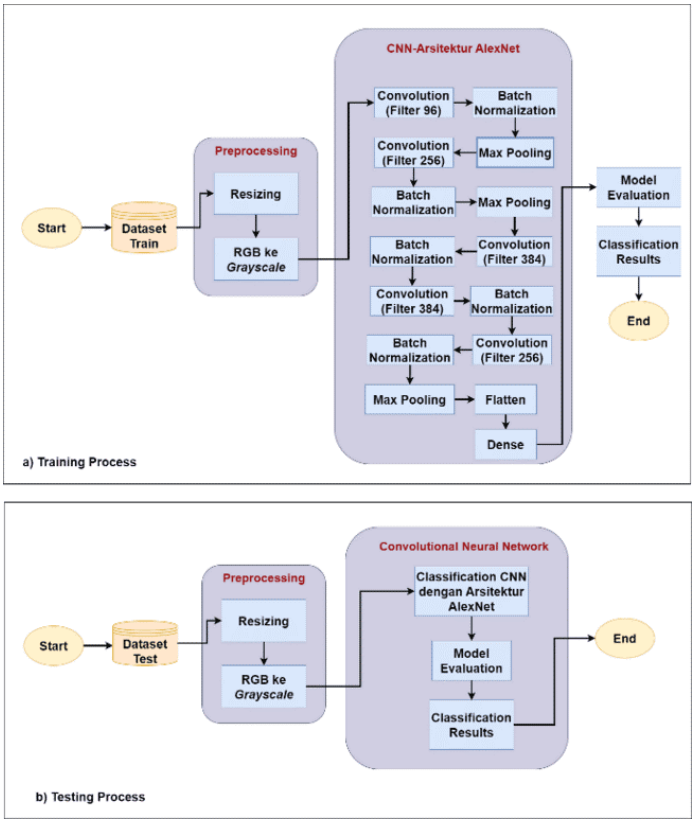


Figure 1: Research flow a) testing and b) training.

2.1 Datasets

The dataset was obtained from the Kaggle Website (https://www.kaggle.com/datasets/gposenka/butterfly-images40-species). The dataset used was 5,499 RGB color butterfly images with 50 species consisting of 90 % training data and 10 % testing data. Names of butterfly species used in classification tests are Adonis, African Giant Swallowtail, American Snoot, ANN 88, and Gray Hairstreaks. Figure 2 is an example of butterfly image data.

2.2 Preprocessing

Preprocessing was the stage used to prepare image data before further processing [19]. Preprocessing techniques to reduce these differences were based on increasing contrast and centering data [20, 21]. Preprocessing stage where all unequal image sizes were changed to 150 $\times$ 150 pixels because CNN received the same size. After resizing, the RGB image was converted to grayscale for processing and model training [22]. This preprocessing was used to normalize images in assessing their impact on the classification results of each CNN model. The data preprocessing stage used the tensorFlow preprocessing library [23].

Figure 2: Image of a butterfly.

2.2.1 Resizing

Resizing was the process of resizing an image by changing its horizontal and vertical resolution [24]. Resizing was aimed at the efficiency and effectiveness of the CNN input system [25]. Images were resized to 100 × 100, 150 × 150, and 200 × 200 pixels. The goal was to equalize the size of the input image for the butterfly image and look for different levels of accuracy for the different input images [26].

2.2.2 Grayscale image

Grayscale image processes a gradient of black and white, which produced a gray effect. In images of this type, color was represented by an intensity between 0 and 255. A value of 0 meant black and a value of 255 meant white [27].

2.3 CNN-AlexNet Architecture

CNN is a multi-layered deep learning technique that extends an artificial neural network (ANN). CNN is processed by the network layer and produces output of a certain class. Each level conducts training and the output from each level is used as input for the next level. Initially, CNN generates simple features such as color, brightness, and edges, while later levels generate more complex features.

In 2012, as in [28], creator AlexNet won the ILSVRC (2012) by a significant margin. Five convolutional layers and three fully connected layers comprise the CNN. With width, height, and depth (red, green, and blue) measurements of 227 × 227 × 3, the first AlexNet layer was utilized as the input filtered image. The remaining layers were feature extractors when the final connected layer joined the other 1,000 connected layers. AlexNet created a 4,096-dimensional feature vector for each input image, with a hidden layer instantly activated before the output layer. The enormous AlexNet system had 650,000 neurons and 60 million parameters. The model was tested on the ImageNet dataset of 150,000 test images after training on roughly 1.2 million training images. This model reduced the problem of overfitting very effectively with the help of maintaining dropouts and data augmentation [29].

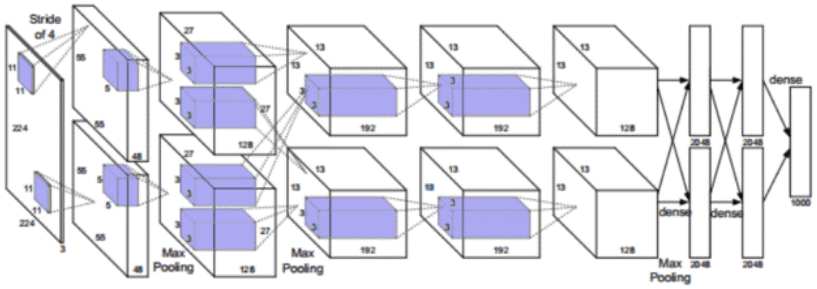


Figure 3: AlexNet architecture.

Figure 3 shows the AlexNet architecture starting with preprocessed image input with a result size of 150 × 150, and then filtering or kernel is performed. The output from the kernel was used by pooled convolution. Convolution and max pooling was done five times. The last max-pooling was flattened for every result to change the matrix image into one dimension. After that, I fully connected three times. The last stage of the image could be classified.

2.3.1 Convolution layer

The convolution layer was used to filter the inputted image using the kernel by applying convolution. The filter came through the input tensor and generated an output feature. Convolution was capable of extracting different features from images, such as edges, textures, objects, and scenes [30].

2.3.2 Pooling layer

Pooling layers were used to reduce object size and a number of parameters. The pooling layer was done after the convolution layer stage. This function calculated the average of each filter from the convolution layer [31]. The pooling layer was also for adjusting data overfitting with nonlinear down sampling via operators [32].

2.3.3 ReLU activation

ReLU (Rectified Linear Unit) activation was an activation layer in the CNN model that applied the function $f(x) = \max(0, x)$, which meant that this function performs zero value thresholding on the pixel value in the input image. This activation caused all pixel values that were less than zero in the image to be 0.

2.3.4 Fully connected layer

The fully connected layer was similar to the fully connected layer in a general neural network, having complete connections to all the neurons in the previous layer [33]. After the convolution and pooling layer stages had been carried out, the entire feature matrix was converted into one vector by applying a procedure called flatten [31].

2.4 Model Evaluation

Evaluation of the AlexNet architectural model was used to determine the effect. This evaluation used an accuracy matrix that could be calculated using (1).

$$\text{Accuracy (\%)} = \frac{\text{The number of correct classification}}{\text{Total testing data}} \times 100\% \quad (1)$$

3 Result

The research discussed the analysis of research that had been done on butterfly image classification using the AlexNet architecture. This research used Python tools and an i5 processor with 8 GB of RAM. The dataset was preprocessed before entering the AlexNet architecture stage.

3.1 Model testing results

Figure 4 shows the preprocessing results of the reesizing process. The initial image size was 200 × 200, and the images were resized to 100 × 100, 150 × 150, and 200 × 200. Image results that had been done resizing would be changed to grayscale.

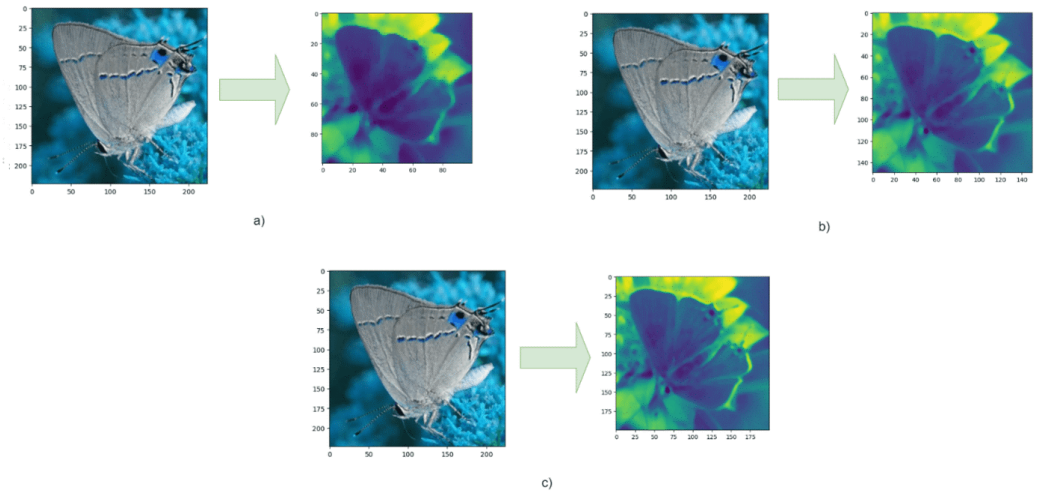


Figure 4: Image resizing a) 100 × 100, b) 150 × 150, and c) 200 × 200.

Figure 5 shows the results of the preprocessing of changing the RGB image to a grayscale image. After preprocessing, the next step was to create graphics using a sequential model. We first added the two convolutional layers on top of each other, then applied the max pooling operation to the output of the second convolutional layer. Next, implemented the dropout. Before entering the resulting output on the dense layer, it was necessary to align the variables to match the shape of entering the dense layer. The output

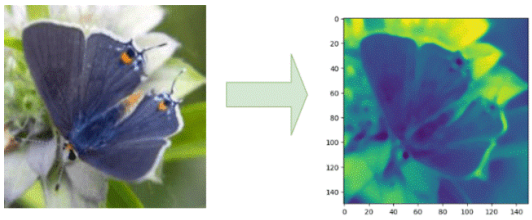


Figure 5: RGB image results to grayscale.

produced by this dense layer was arranged according to the dropout; the resulting output was then used as input to the last dense layer for classification. Softmax functioned to convert results into something interpreted in the form of probabilities. Figure 6 is a sequential model. Starting with a convolutional layer with 96 of 11 $\times$ 11 filters total and an activation function represented by a rectified linear unit (ReLU). The input format was similar to any image size, or 150 $\times$ 150 $\times$ 3 (color images contain three. channels – RGB). Afterwards, a max pooling layer and a further convolutional layer were applied. Due to the 50 classes available, it had a dense, fully connected layer. A softmax activity was present in the final output layer.

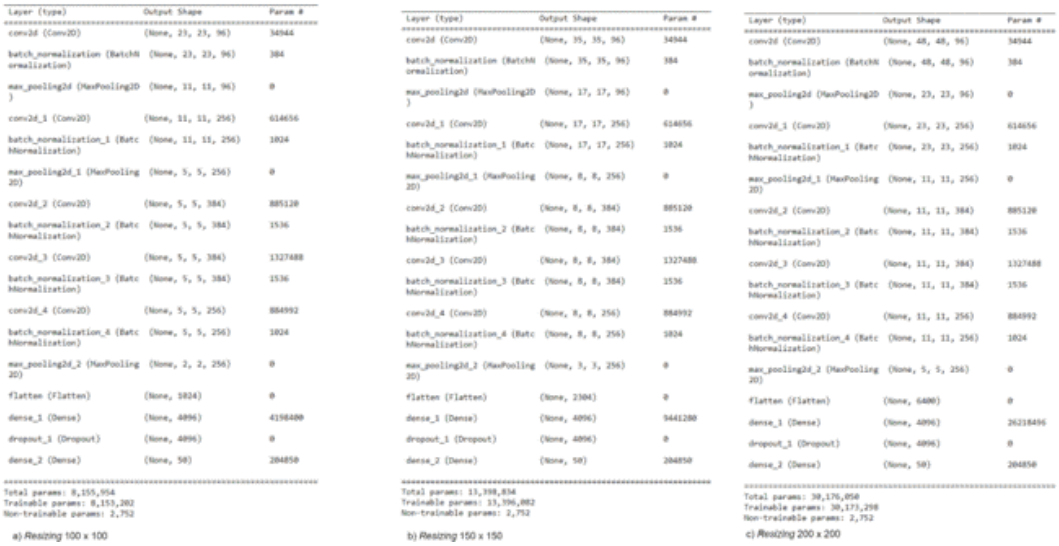


Figure 6: Sequential models.

After 20 epochs, the models with different resize achieved 98 % accuracy from 100 $\times$ 100 resize, 150 $\times$ 150 resize, and 95 % from 200 $\times$ 200 resize in the data train set, satisfactory results for simple models. Table 1, which consists of accuracy and loss columns, is the result of an epoch that has been carried out. The results of the epochs were seen from the value of the loss function and accuracy to assist in evaluating the model's performance in classifying the training data and helping to improve each epoch during the training process. Several

epochs that were run from different resizes could be seen from the speed performance of the displayed epoch results.

Table 1: Process epoch

Epoch	Resize 100 × 100		Resize 150 × 150		Resize 200 × 200	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
1	0.11	4.67	0.14	5.43	0.09	11.36
2	0.28	2.62	0.34	2.37	0.26	2.95
3	0.41	2.07	0.46	1.87	0.38	2.31
4	0.50	1.70	0.56	1.52	0.47	1.93
5	0.57	1.44	0.64	1.22	0.53	1.66
6	0.63	1.24	0.67	1.07	0.57	1.50
7	0.66	1.12	0.82	0.57	0.59	1.40
8	0.70	0.98	0.87	0.38	0.63	1.28
9	0.75	0.84	0.89	0.38	0.67	1.14
10	0.77	0.75	0.92	0.24	0.69	1.09
11	0.86	0.39	0.95	0.15	0.71	0.98
12	0.91	0.26	0.96	0.12	0.74	0.90
13	0.93	0.19	0.97	0.09	0.83	0.53
14	0.93	0.17	0.97	0.07	0.88	0.35
15	0.95	0.13	0.98	0.06	0.90	0.29
16	0.95	0.12	0.98	0.05	0.91	0.24
17	0.96	0.10	0.98	0.05	0.92	0.21
18	0.96	0.09	0.98	0.05	0.92	0.21
19	0.97	0.08	0.99	0.04	0.95	0.15
20	0.98	0.05	0.98	0.04	0.95	0.13

Figure 7 is an accuracy graph and Figure 8 is a loss graph at resize 100 × 100, resize 150 × 150, resize 200 × 200. Line blue is data training while the red line is data validation. The accuracy graph is a comparison between data that is correctly predicted according to the target class and all data from the AlexNet architecture model. The accuracy graph in Figure 7 shows an increase in finding the best match with training data from different resizing trials. The loss graph in Figure 8 is the error value between the training results and the target from different resizing trials.

Figure 9 is a classification report from resizing 100 × 100 results in an accuracy of 80 % and classification of resizing 150 × 150 and 200 × 200 results in an accuracy of 82 %. It can be seen from Figure 9 that the larger the image size, the higher the accuracy produced by the model. However, keep in mind that the larger the image size used, the longer it will take to train and test the model.

3.2 Discussion

Table 2 is the result of confusion matrix which provides important information when evaluating the performance of a classification model. Confusion The matrix provides information such as the number of true positives, true negatives, false positives, and false negatives for each class in the classification problem. In the results of this study the confusion matrix was made using tables, from Table 2 it can analyze the strengths and weaknesses of the

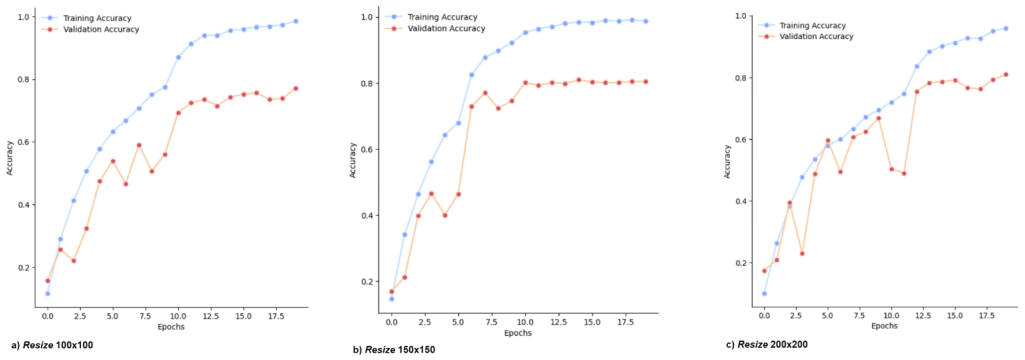


Figure 7: Accuracy graph with different resizing.

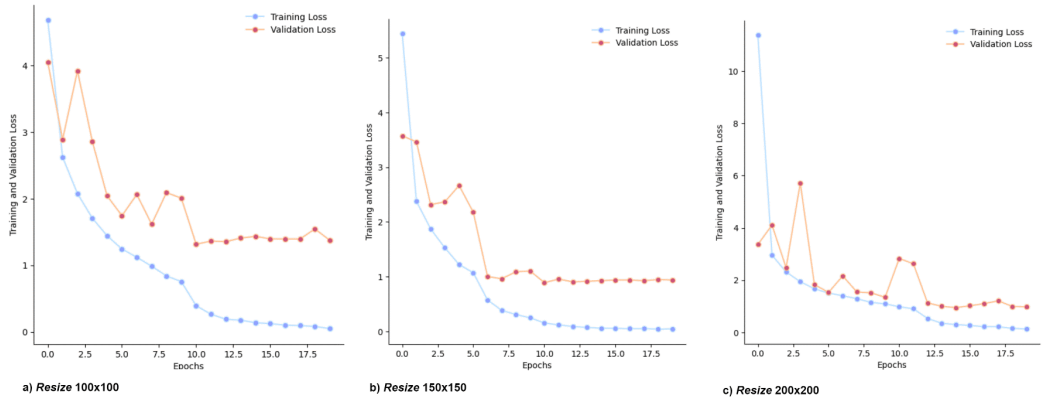
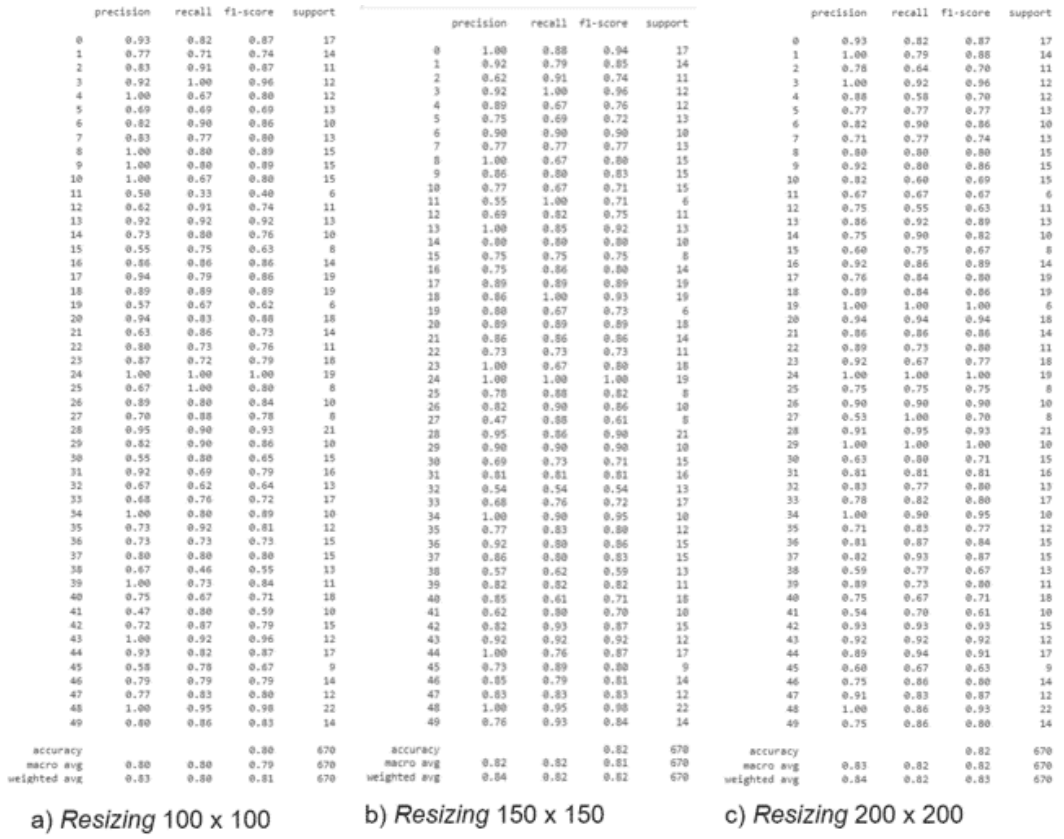


Figure 8: Loss graph with different resizing.

classification model properly. Namely understanding how well it can classify data correctly and identify errors that occur.

Table 2 has the same accuracy values as Figure 9 regarding the classification results. Table 2 shows the classification results for 50 different species or classes using AlexNet architecture with 100 $\times$ 100, 150 $\times$ 150, and 200 $\times$ 200 images. When the image is resized to 100 $\times$ 100, the accuracy obtained is 80 %. With 150 $\times$ 150, the accuracy increased to 82 %, and 200 $\times$ 200 also gave 82 % accuracy. This means that the model could recognize and classify the data well after resizing the image. It should be noted that the validation process for the three image sizes proceeded at different speeds over 20 epochs. These accuracy results provided the ability of the model to classify data using images that had been resized.



	precision	recall	f1-score	support
0	0.93	0.82	0.87	17
1	0.77	0.71	0.74	14
2	0.83	0.91	0.87	11
3	0.92	1.00	0.96	12
4	1.00	0.67	0.80	12
5	0.69	0.69	0.69	13
6	0.82	0.90	0.86	10
7	0.83	0.77	0.80	13
8	1.00	0.80	0.89	15
9	1.00	0.80	0.89	15
10	1.00	0.67	0.80	15
11	0.50	0.33	0.40	6
12	0.62	0.91	0.74	11
13	0.92	0.92	0.92	13
14	0.73	0.80	0.76	10
15	0.55	0.75	0.63	8
16	0.86	0.86	0.86	14
17	0.94	0.79	0.86	19
18	0.89	0.89	0.89	19
19	0.57	0.67	0.62	6
20	0.94	0.83	0.88	18
21	0.63	0.86	0.73	14
22	0.80	0.73	0.76	11
23	0.87	0.72	0.79	18
24	1.00	1.00	1.00	19
25	0.67	1.00	0.80	8
26	0.89	0.80	0.84	10
27	0.70	0.88	0.78	8
28	0.95	0.90	0.93	21
29	0.82	0.90	0.86	10
30	0.55	0.80	0.65	15
31	0.92	0.69	0.79	16
32	0.67	0.62	0.64	13
33	0.68	0.76	0.72	17
34	1.00	0.80	0.89	10
35	0.73	0.92	0.81	12
36	0.73	0.73	0.73	15
37	0.80	0.80	0.80	15
38	0.67	0.46	0.55	13
39	1.00	0.73	0.84	11
40	0.75	0.67	0.71	18
41	0.47	0.80	0.59	10
42	0.72	0.87	0.79	15
43	1.00	0.92	0.96	12
44	0.93	0.82	0.87	17
45	0.58	0.78	0.67	9
46	0.79	0.79	0.79	14
47	0.77	0.83	0.80	12
48	1.00	0.95	0.98	22
49	0.80	0.86	0.83	14
accuracy			0.80	670
macro avg	0.80	0.80	0.79	670
weighted avg	0.83	0.80	0.81	670

a) *Resizing* 100 x 100

	precision	recall	f1-score	support
0	1.00	0.88	0.94	17
1	0.92	0.79	0.85	14
2	0.62	0.91	0.74	11
3	0.92	1.00	0.96	12
4	0.89	0.67	0.76	12
5	0.75	0.69	0.72	13
6	0.90	0.90	0.90	10
7	0.77	0.77	0.77	13
8	1.00	0.67	0.80	15
9	0.86	0.80	0.83	15
10	0.77	0.67	0.71	15
11	0.55	1.00	0.71	6
12	0.69	0.82	0.75	11
13	1.00	0.85	0.92	13
14	0.80	0.80	0.80	10
15	0.75	0.75	0.75	8
16	0.75	0.86	0.80	14
17	0.89	0.89	0.89	19
18	0.86	1.00	0.93	19
19	0.80	0.67	0.73	6
20	0.89	0.89	0.89	18
21	0.86	0.86	0.86	14
22	0.73	0.73	0.73	11
23	1.00	0.67	0.80	18
24	1.00	1.00	1.00	19
25	0.78	0.88	0.82	8
26	0.82	0.90	0.86	10
27	0.47	0.88	0.61	8
28	0.95	0.86	0.90	21
29	0.90	0.90	0.90	10
30	0.69	0.73	0.71	15
31	0.81	0.81	0.81	16
32	0.54	0.54	0.54	13
33	0.68	0.76	0.72	17
34	1.00	0.90	0.95	10
35	0.77	0.83	0.80	12
36	0.92	0.80	0.86	15
37	0.86	0.80	0.83	15
38	0.57	0.62	0.59	13
39	0.82	0.82	0.82	11
40	0.85	0.61	0.71	18
41	0.62	0.80	0.70	10
42	0.82	0.93	0.87	15
43	0.92	0.92	0.92	12
44	1.00	0.76	0.87	17
45	0.73	0.89	0.80	9
46	0.85	0.79	0.81	14
47	0.83	0.83	0.83	12
48	1.00	0.95	0.98	22
49	0.76	0.93	0.84	14
accuracy			0.82	670
macro avg	0.82	0.82	0.81	670
weighted avg	0.84	0.82	0.82	670

b) *Resizing* 150 x 150

	precision	recall	f1-score	support
0	0.93	0.82	0.87	17
1	1.00	0.79	0.88	14
2	0.78	0.64	0.70	11
3	1.00	0.92	0.96	12
4	0.88	0.58	0.70	12
5	0.77	0.77	0.77	13
6	0.82	0.90	0.86	10
7	0.71	0.77	0.74	13
8	0.80	0.80	0.80	15
9	0.92	0.80	0.86	15
10	0.82	0.60	0.69	15
11	0.67	0.67	0.67	6
12	0.75	0.55	0.63	11
13	0.86	0.92	0.89	13
14	0.75	0.90	0.82	10
15	0.60	0.75	0.67	8
16	0.92	0.86	0.89	14
17	0.76	0.84	0.80	19
18	0.89	0.84	0.86	19
19	1.00	1.00	1.00	6
20	0.94	0.94	0.94	18
21	0.86	0.86	0.86	14
22	0.89	0.73	0.80	11
23	0.92	0.67	0.77	18
24	1.00	1.00	1.00	19
25	0.75	0.75	0.75	8
26	0.90	0.90	0.90	10
27	0.53	1.00	0.70	8
28	0.91	0.95	0.93	21
29	1.00	1.00	1.00	10
30	0.63	0.80	0.71	15
31	0.81	0.81	0.81	16
32	0.83	0.77	0.80	13
33	0.78	0.82	0.80	17
34	1.00	0.90	0.95	10
35	0.71	0.83	0.77	12
36	0.81	0.87	0.84	15
37	0.82	0.93	0.87	15
38	0.59	0.77	0.67	13
39	0.89	0.73	0.80	11
40	0.75	0.67	0.71	18
41	0.54	0.70	0.61	10
42	0.93	0.93	0.93	15
43	0.92	0.92	0.92	12
44	0.89	0.94	0.91	17
45	0.60	0.67	0.63	9
46	0.75	0.86	0.80	14
47	0.91	0.83	0.87	12
48	1.00	0.86	0.93	22
49	0.75	0.86	0.80	14
accuracy			0.82	670
macro avg	0.83	0.82	0.82	670
weighted avg	0.84	0.82	0.83	670

c) *Resizing* 200 x 200

Figure 9: Classification results with different resizing.

4 Conclusion

This study used the CNN classification by applying the AlexNet architecture to produce an accuracy of 80 % from 100 × 100 resizing, 82 % from 150 × 150 resizing results, and 82 % from resizing results. 200 × 200 corrected data. Each class had high accuracy and low accuracy. This study could not achieve classification accuracy of up to 100 %, so additional datasets to be tested, epochs, and feature extraction could be carried out to achieve higher accuracy. Suggestions for further research on developing more complex CNN models by considering architecture and effective transfer learning techniques.

References

[1] A. Rajab and D. Asriady, *Keanekaragaman Jenis Kupu-Kupu Papilionidae*. Tn-Babul, 2015.

[2] H. Harmonis, "Studi baseline keragaman kupu-kupu untuk kawasan pelestarian plasma nutfah pt sylva rimba lestari, kalimantan timur," *Jurnal Hutan Tropis*, vol. 3,

Table 2: Results of butterfly image classification

Class	Image Size						
	Amount of data	True	False	True	False	True	False
Adonis	17	14	3	15	2	14	3
African Giant Swallowtail	14	10	4	11	3	11	3
American Snoot	11	10	1	10	1	7	4
An 88	12	12	0	12	0	11	1
Appollo	12	8	4	8	4	7	5
Arcigera Flower Moth	13	9	4	9	4	10	3
Atala	10	9	1	9	1	9	1
....	...	...	...	...	...	...	...
Great Eggfly	9	7	2	8	1	6	3
Great Jay	14	11	3	11	3	12	2
Green Celled Cattleheart	12	10	2	10	2	10	2
Green Hairstreak	22	21	1	21	1	19	3
Grey Hairstreak	14	12	2	13	1	12	2
Total	**670**	**539**	**131**	**549**	**121**	**552**	**118**
Accuracy (%)		**80%**		**82%**		**82%**	

no. 1, pp. 25–31, 2015.

[3] R. Zhao, C. Li, S. Ye, and X. Fang, "Butterfly recognition based on faster r-cnn," *Journal of Physics: Conference Series*, vol. 1176, p. 032048, Mar. 2019.

[4] P. Nyoman and Putu Kusuma Negara, "Deteksi masker pencegahan covid19 menggunakan convolutional neural network berbasis android," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, pp. 576–583, June 2021.

[5] C. Wu, L. Chai, J. Yang, and Y. Sheng, "Facial expression recognition using convolutional neural network on graphs," in *2019 Chinese Control Conference (CCC)*, (Guangzhou, China), pp. 7572–7576, IEEE, July 2019.

[6] S. Grossberg, "The resonant brain: How attentive conscious seeing regulates action sequences that interact with attentive cognitive learning, recognition, and prediction," *Attention, Perception, & Psychophysics*, vol. 81, pp. 2237–2264, Oct. 2019.

[7] Y. Tian, "Artificial intelligence image recognition method based on convolutional neural network algorithm," *IEEE Access*, vol. 8, pp. 125731–125744, 2020.

[8] H. Akbar and S. Sandfreni, "Klasifikasi kanker serviks menggunakan model convolutional neural network alexnet," *JIKO (Jurnal Informatika dan Komputer)*, vol. 4, pp. 44–51, Apr. 2021.

[9] H. Hendriyana and Yazid Hilman Maulana, "Identification of types of wood using convolutional neural network with mobilenet architecture," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, pp. 70–76, Feb. 2020.

[10] Y. S. Hariyani, S. Hadiyoso, and T. S. Siadari, "Deteksi penyakit covid-19 berdasarkan citra x-ray menggunakan deep residual network," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, p. 443, May 2020.

[11] Melly Damara Chaniago, Amellia Amanullah Sugiharto, Qhistina Dyah Khatulistiwa, Zamah Sari, and Agus Eko, "Covid-19 detection using convolutional neural networks (Cnn) classification algorithm," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, pp. 190–197, Apr. 2022.

[12] F. Han, J. Yao, H. Zhu, and C. Wang, "Marine organism detection and classification from underwater vision based on the deep cnn method," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–11, Feb. 2020.

[13] I. Y. Prayogi, Sandra, and Y. Hendrawan, "Image classification of different clove (Syzygium aromaticum) quality using deep learning method with convolutional neural network algorithm," *IOP Conference Series: Earth and Environmental Science*, vol. 905, p. 012018, Nov. 2021.

[14] H. Tang, B. Wang, and X. Chen, "Deep learning techniques for automatic butterfly segmentation in ecological images," *Computers and Electronics in Agriculture*, vol. 178, p. 105739, Nov. 2020.

[15] O. D. Annesa, Condro Kartiko, and Agi Prasetiadi, "Identification of reptile species using convolutional neural networks(Cnn)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, pp. 899–906, Oct. 2020.

[16] H. Kato, K. Osuge, S. Haruta, and I. Sasase, "A preprocessing by using multiple steganography for intentional image downsampling on cnn-based steganalysis," *IEEE Access*, vol. 8, pp. 195578–195593, 2020.

[17] M. Ghazal, S. S. Ali, A. H. Mahmoud, A. M. Shalaby, and A. El-Baz, "Accurate detection of non-proliferative diabetic retinopathy in optical coherence tomography images using convolutional neural networks," *IEEE Access*, vol. 8, pp. 34387–34397, 2020.

[18] S. Sunardi, A. Fadlil, and D. Prayogi, "Sistem pengenalan wajah pada keamanan ruangan berbasis convolutional neural network," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 6, no. 2, pp. 636–647, 2022.

[19] L. Hakim, H. R. Rahmanto, S. P. Kristanto, and D. Yusuf, "Klasifikasi citra motif batik banyuwangi menggunakan convolutional neural network," *Jurnal Teknoinfo*, vol. 17, p. 203, Jan. 2023.

[20] L. F. Rodrigues, M. C. Naldi, and J. F. Mari, "Comparing convolutional neural networks and preprocessing techniques for HEp-2 cell classification in immunofluorescence images," *Computers in Biology and Medicine*, vol. 116, p. 103542, Jan. 2020.

[21] L. Faria, L. Rodrigues Moreira, and J. Mari, "Cell classification using handcrafted features and bag of visual words," in *Workshop de Visao Computacional*, (Ilheus, BA), IEEE, 11 2018.

[22] M. M. Khodier, S. M. Ahmed, and M. S. Sayed, "Complex pattern jacquard fabrics defect detection using convolutional neural networks and multispectral imaging," *IEEE Access*, vol. 10, pp. 10653–10660, 2022.

[23] D. Iswantoro and D. Handayani Un, "Klasifikasi penyakit tanaman jagung menggunakan metode convolutional neural network(Cnn)," *Jurnal Ilmiah Universitas Batanghari Jambi*, vol. 22, p. 900, July 2022.

[24] F. R. Lestari, I. P. N. Purnama, A. M. Sajiah, and L. M. B. Aksara, "Identifikasi penyakit tanaman jeruk siam menggunakan metode m-svm," *SEMINAR NASIONAL APTIKOM (SEMNASTIK) 2019*, pp. 441–448, Nov. 2019.

[25] P. Winardi and E. Setyati, "Identifikasi jenis daging dengan menggunakan algoritma convolution neural network," *Journal of Information System,Graphics, Hospitality and Technology*, vol. 3, pp. 82–88, Dec. 2021.

[26] A. Antoni, T. Rohana, and A. R. Pratama, "Implementasi algoritma convolutional neural network untuk klasifikasi citra kemasan kardus defect dan no defect," *Building of Informatics, Technology and Science (BITS)*, vol. 4, Mar. 2023.

[27] F. D. Marleny, *Mengenal Pengolahan Citra Digital menggunakan Python*. CV. Pena Persada, 2021.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[29] F. R. Mashrur, A. Dutta Roy, and D. K. Saha, "Automatic identification of arrhythmia from ecg using alexnet convolutional neural network," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, (Khulna, Bangladesh), pp. 1–5, IEEE, Dec. 2019.

[30] X. Liu, W. Lu, W. Liu, S. Luo, Y. Liang, and M. Li, "Image deblocking detection based on a convolutional neural network," *IEEE Access*, vol. 7, pp. 26432–26439, 2019.

[31] K. Davoudi and P. Thulasiraman, "Evolving convolutional neural network parameters through the genetic algorithm for the breast cancer classification problem," *SIMULATION*, vol. 97, pp. 511–527, Aug. 2021.

[32] S. Chatterjee, S. S. Roy, K. Samanta, and S. Modak, "Sensing wettability condition of insulation surface employing convolutional neural network," *IEEE Sensors Letters*, vol. 4, pp. 1–4, July 2020.

[33] D. Sarkar, R. Bali, and T. Sharma, *Practical machine learning with python*. Berkeley, CA: Apress, 2018.