

COMPARISON OF SGD, RMSProp, AND ADAM OPTIMIZATION IN ANIMAL CLASSIFICATION USING CNNs

Desi Irfan¹, Teddy Surya Gunawan², Wanayumini³

Universitas Potensi Utama, Medan
Indonesia

¹desiirfan@gmail.com

²tsgunawan@iium.edu.my

³wanayumini@gmail.com

Abstract. Many measures have been taken to protect endangered species by using "camera trap" technology which is widespread in the field of technology-based nature protection field research. In this study, a machine learning-based approach is presented to identify endangered wildlife images with a data set containing 5000 images taken from Kaggle and some other sources. The Gradient Descent optimization method is often used for Artificial Neural Network (ANN) training. This method plays a role in finding the weight values that give the best output value. Three optimization methods have been implemented, namely Stochastic Gradient Descent (SGD), ADADELTA, and Adam on the Artificial Neural Network system for animal data classification. In some of the studies reviewed there are differences in the results of SGD and ADAM, which on the one hand SGD is superior, and on the one hand ADAM is superior with the appropriate learning rate. The results of this study show that the CNN method with the Adam optimization function produces the highest accuracy compared to the SGD and RMSprop optimization methods. The model trained using Adam's optimization function achieved an accuracy of 89.81% on the test, showing the feasibility of the approach.

Keywords—Optimization Function, SGD, Adam, RMSProp

I. INTRODUCTION

As many as one million species of animals and plants on land, sea, and air are threatened with extinction, due to human actions, according to a 1,800-page UN report[1]. One of the efforts in making it easier for researchers to find out the number of endangered animals is to implement an automation system using digital image processing. Because of this, many measures have been taken to protect endangered species[2], and "camera trap" technology[3] is widespread in the field of technology-based nature protection field research. Rich and Knight (1991) mentioned that Artificial Intelligence (AI) is a study of how to make computers do things that can currently be done better by humans [4]. Computer recognition starts from the process of classifying objects/images and is a fairly easy task for humans, but for computers/machines classifying objects/images is a very complex task, so it would be very useful if we can automate this whole process using Computer Vision. Deep Learning is a technology used in image

II. THEORETICAL STUDY

a. Animal

Satwa in the large Indonesian dictionary states satwa is a synonym for animal or animal [5] Reporting to the World Wildlife Fund, many endangered animals such as Sumatran elephants, Asian elephants, African elephants, blue whales, hawksbill turtles, orangutans, Javan rhinos, dugongs, hippos, turtles, polar bears, penguins, and many others [6].



Fig.1 Animal Image

b. Convolutional Neural Network

A convolutional neural network (or CNN) is a special type of multilayer neural network or deep learning architecture inspired by the visual system of living things[7]. A convolutional neural network (CNN) is a special type of neural network for processing data that has a grid-like topology[8].

An example of such data is an image. An image can be considered as a 2-dimensional grid of pixels. The use of pixel optimization is useful for object detection, and the segmentation of pixel values is considered a significant factor[9]. The name "convolutional neural network" indicates that it uses a mathematical operation called convolution. CNN has three main types of layers viz: convolutional layer, pooling layer, and fully-connected (FC) layer [10]. The basic unit of computation in a neural network is the neuron, often also called a node or unit. Nodes receive input from several other nodes or from external sources, after which the node processes the input and produces output. Each input has associated weights (w). Nodes apply the function f to the weighted sum as shown in Fig.3 [11].

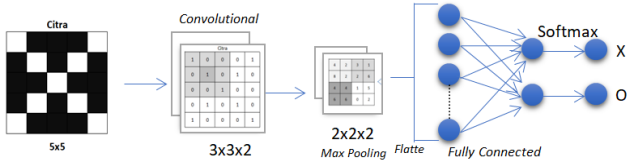


Fig.2 Example of CNN Architecture

c. Weight Optimization

Fig.3 is the Back-propagation process used to update the weights in a neural network [13].

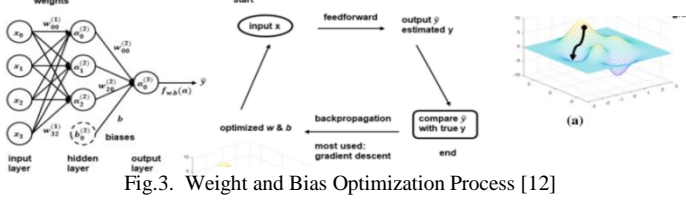


Fig.3. Weight and Bias Optimization Process [12]

The gradient of the parameter model is sampled iteratively, behind the direction of the network weights, to find new weights that minimize the error value in terms of classification [14].

d. Optimizer

The first proposed optimizer is SGD. SGD[15] follows the gradient of a randomly selected minibatch downhill. To train a neural network using SGD, first, the estimated gradient is calculated using a loss function. Then, the update at iteration k applied with parameter θ . Calculation for each minibatch m instances of the training set $\{x^{(1)}, \dots, x^{(m)}\}$ with appropriate targets $y^{(i)}$, equation as follows:

$$\hat{g} \leftarrow \frac{1}{m} \nabla_0 \sum L(f(x^{(i)}; \theta), y^{(i)}) \quad (1)$$

$$\theta \leftarrow \theta - \epsilon_k \hat{g} \quad (2)$$

Here, the learning rate ϵ_k is a very important hyperparameter. The magnitude of the update depends on the learning rate. If it is too large, the update depends too much on the recent case. If it is small, many updates may be required for convergence [16]. This hyperparameter can be chosen by trial and error. One way is to choose one of several learning rates that produce the smallest loss function value. This is called line search. Another way is to monitor the first few epochs and use a learning rate that is higher than the best learning rate. In Equation 2, the learning rate is denoted as k at iteration k because, in practice, it is necessary to decrease the learning rate gradually [17].

The second proposed optimizer RMSProp[18] is an optimization algorithm that calculates the learning rate by an exponential average of the squared gradient. To implement RMSProp, the squared gradient is accumulated after calculating the gradient:

$$r \leftarrow \rho r + (1 - \rho) g \odot g \quad (3)$$

where ρ is the decay rate. Then the parameter update is calculated and applied as follows:

$$\Delta \theta = -\frac{\epsilon}{\sqrt{\delta + r}} \odot g \quad (4)$$

$$\theta \leftarrow \theta + \Delta \theta \quad (5)$$

The last proposed optimizer is adam. The third optimizer ADAM [19] is one of the most and most efficient optimizer algorithms calculating the learning rate for each parameter. The algorithm updates exponential moving averages of the gradient m_t and the square of the gradient u_t where the hyperparameters ρ_1, ρ_2 control the decay rate of these exponential moving averages. The exponential moving averages themselves are estimates of the first moment (mean) and second raw moment (uncentered variance) of the gradient. Adam's algorithm requires the first and second-moment variables m and u . After computing the gradient, the biased estimates of the first and second moments are updated every time step t :

$$m_t \leftarrow \rho_1 m_{t-1} + (1 - \rho_1) g_1 \quad (6)$$

$$u_t \leftarrow \rho_2 u_{t-1} + (1 - \rho_2) g \odot g \quad (7)$$

Next, the bias is corrected in the first and second moments. Using the adjusted moments, the updated prediction parameters are calculating and applied:

$$\hat{m}_t \leftarrow \frac{m_t}{1 - \rho_1^t} \quad (8)$$

$$\hat{u}_t \leftarrow \frac{u_t}{1 - \rho_2^t} \quad (9)$$

$$\Delta \theta = -\epsilon \frac{\hat{m}_t}{\sqrt{\hat{u}_t + \delta}} \quad (10)$$

$$\theta_t \leftarrow \theta_{t-1} + \Delta \theta \quad (11)$$

Adam has many advantages. First of all, it requires little tuning for the learning rate. Also, it is a gradient diagonal scaling method that is easy to implement and immutable. It is computationally efficient and also has little memory requirements. Moreover, Adam is suitable for non-stationary purposes and problems with very noisy and sparse gradients[19].

III. METHODS

In this study, a classification system of animal species in the wild was designed to determine accuracy using digital image processing methods. Fig.4 shows the system block diagram designed in this study.

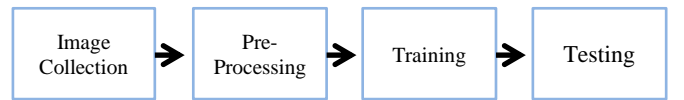


Fig.4 General System Block Diagram

In general, the systematic system block diagram shown in Figure 4 is Animal image collection, preprocessing with resize and data augmentation stages, Training, and Testing.

A. Image collection (3)

The data used in this research is secondary data. The data is sourced from Kaggle. The reason the author takes data from Kaggle is because of the reliability of the dataset that has been

tested. Keep in mind, using high-resolution training images can also be used to get better accuracy[21].

B. Preliminary Observation

Results To have a foundation that we can build upon comparison of how good our model is forbidden to do things naturally, us using pre-trained VGG-19 with following structure:

TABLE.I
Hyperparameter Model CNN

No	Layer	Output Shape
1	Batch Size	128
2	Crop size	64
3	Input Layer	3 x 64 x 64
4	nn.model	64 x 4 x 4
5	Global Average pooling 2d layer	3, 8, 3, 1, 1
6	Dropout	10 %

In all the experiments we have done do, the only change we do replace the Optimization Function with Adam, RMSProp, and SGD with the appropriate Learning Rate.

C. Preprocessing.

After the image collection process, pre-processing is done to optimize the quality of the image, and to facilitate and boost the system's ability to identify objects. Pre-processing augmentation is done by resizing and data augmentation.

D. Training

At the training stage, the learning process is carried out on the image, which then outputs a model that will be stored for use in the testing process. Model building is the process of training and training image data in identifying objects and categorizing them according to their class.

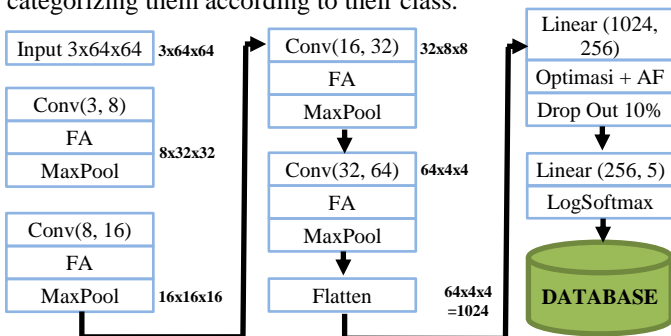


Fig.5 Flowchart of Training System Stages referring to LeNet-5

In this study, the method refers to the LeNet-5 architecture which is very popular and has been tested using 2 layers shown in Figure 5. Input size 64 x 64 x 3. In the first convolution using the number of kernels as much as 10 with a 3x3 matrix with a padding value = valid, ReLu activation is used in this convolution process as Non-Linearity. The pooling process, especially max pooling, uses a 2x2 size and

at the second convolution stage uses a total of 20 kernels with a 5x5 matrix, using ReLU with a padding value = valid. Furthermore, flatten is changing the output of the convolution process in the form of a matrix into a vector that will be forwarded to the classification process using MLP (Multi-Layer Perceptron) with a predetermined number of neurons in the hidden layer. In SGD Optimization, RMSProp and Adam will be applied to the node for weight and bias optimization with the default Learning rate using the softmax activation function according to the number of classes, in this study, there are 5 classes of neurons. The class of the image is then classified based on the value of the neurons in the hidden layer using the softmax activation function.

E. Testing

Fig.6 shows the flowchart of the system testing stage. The testing stage is the process of classifying animal species by testing test image data and comparing it with the training model results of training image data stored in the database. Image data taken as much as 1000 for the original data then 3000 image data augmentation results. The image that has been taken will be processed by the CNN algorithm until it produces system output in the form of Animal Class information.

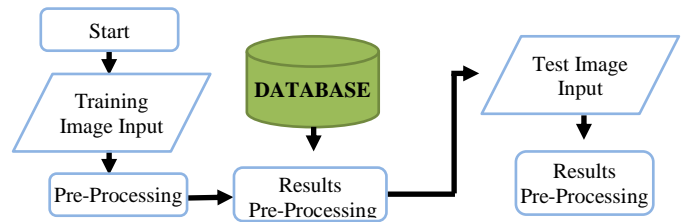


Fig.6 Flowchart of System Testing Stage

IV. RESULTS AND DISCUSSION

Tests on the system that has been designed using the CNN method with architecture referring to LeNet-5 to determine the type of animal from a dataset that is divided into five classes namely Bear, Elephant, Orang Utan, Tiger, and Zebra with consideration. The test system is formed by utilizing hyperparameter changes in the data before augmentation and after augmentation. The hyperparameters used are changes in the Optimizer type, namely Adam, SGD, and RMSprop, changes in batch size, namely 16, 32, 64, and 128 changes in learning rate values, namely 0.1, 0.01, and 0.001 and the number of training iterations (epochs) in this case using early stop.

A. Data Testing and Analysis

The first data to be tested is the original data totaling 5000 images. In the training process, the data used amounted to 0,8 of the total data at 4000 data. While in the testing process the data used amounted to 20% of the total data or 1000 data. This test uses three Optimizers, namely Adam, SGD, and RMSprop with parameters, namely batch size of 16, for Epoch here

using early stop, where when the model cannot get accuracy and loss again then the process automatically stops. To determine the Learning Rate in accordance with the Optimizer, several Learning Rate values will be tried, namely 0.1, 0.01, and 0.001.

1) *Learning Rate 0,1*

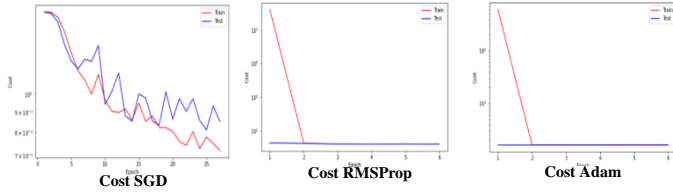


Fig.7 Plot of Adam's cost, SGD and RMSProp

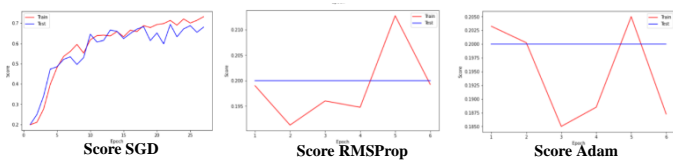


Fig.8 Plot of Adam's Score, SGD and RMSProp

To find the right learning rate for the model created, three learning rate values were tried. It can be seen in Fig.7 and Fig.8, from the three optimization algorithms above SGD, outperforms Adam's algorithm both from the Cost and Score values. Adam and RMSProp optimizers are very bad at a learning value of 0.1, this means we will change the learning rate value to find a better score and cost.

TABLE.II

COMPARISON OF OPTIMIZER WITH LEARNING RATE 0.1

Optimizer	Test Score	Test Cost	Best Epoch
SGD	0.7480	0.7255	25
RMSProp	0.2000	1.6140	3
Adam	0.2000	1.6122	2

From the table.II above SGD is superior to Adam and RMSProp with the acquisition of a Test Score of 74.80% and Test Cost of 72.55%. And it can be seen that the Test Score on SGD and RMSProp cannot exceed 2% and the Cost is still very high, this means that at learning 0.1 there is a problem with the RMSPop and Adam optimizers. Therefore, researchers changed the Learning rate value to 0.01 in the second experiment.

2) *Learning Rate 0.01*

In the first experiment with a Learning Rate value of 0.1, it turns out that SGD and RMSProp produce costs and scores that are still bad. It can be seen in Fig.9 and Fig.10 of the three optimization algorithms used RMSProp outperforms the SGD and Adam optimization algorithms both from the Cost and Score values.

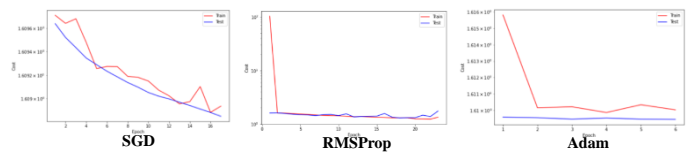


Fig.9 Plot of Cost Adam, SGD and RMSProp

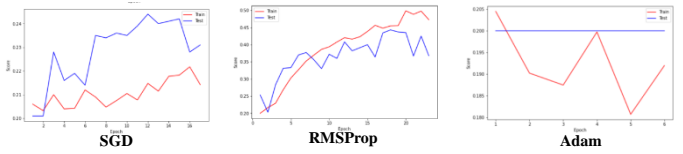


Fig.10 Plot of Adam's Score, SGD and RMSProp

But the results obtained from the three models are still very bad considering the dataset used is good and the model also uses an activation function that performs very well. This means that there is still a problem with the Learning rate value used in the optimizer algorithm in this model.

TABLE.III

OPTIMIZER COMPARISON WITH LEARNING RATE 0.01

Optimizer	Test Score	Test Cost	Best Epoch
SGD	0.2440	1.6090	12
RMSProp	0.4430	1.3005	8
Adam	0.2000	1.6096	1

From table III, the highest Score only reaches 0.4430 with the lowest Cost of 1.3005 obtained by RMSProp at epoch 12. This is very far from the expected model performance. Therefore, the researcher changed to the next Learning rate value of 0.001 in the third experiment.

3) *Learning Rate 0,001*

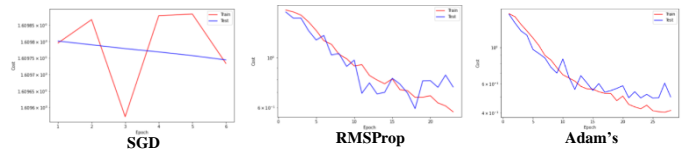


Fig.11 Plot of Cost Adam, SGD and RMSProp

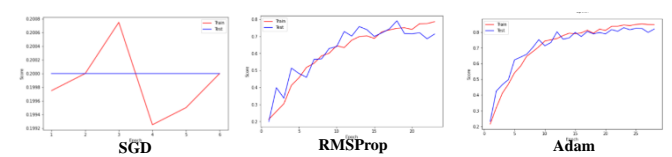


Fig.12 Plot of Adam's Score, SGD and RMSProp

The results of the SGD Optimizer Plot in Fig.11 and Fig.12 are still very bad, which is shown in the cost and score values at the time of testing the graph is very monotonous and during the training process, the graph moves otherwise very unstable. This indicates that SGD is not very suitable for use at this learning rate value. But on the contrary for Adam and RMSProp the cost and score results are very good even though RMSProp is not as good as the Adam optimizer. From

several experiments by changing the value of the learning rate, it is concluded that the SGD, RMSProp, and Adam optimizers have their respective ideal learning values.

TABLE.IV
OPTIMIZER COMPARISON WITH LEARNING RATE 0.001

Optimizer	SGD	RMSProp	Adam
Test Score	0.2660	0.7920	0.8280
Test Cost	1.6090	0.5925	0.4926
Best Epoch	2	18	23
Precision	0.2	0.85	0.82
Recall	0.08	0.82	0.78
F1 Score	0.04	0.84	0.80

Table.IV shows that SGD at learning 0.001 is not optimal, but we have got the right Learning rate value at 0.1. It can be seen that the RMSProp and Adam optimizers produce quite good Score, Cost, Precision, Recall, and F1 Score values. But to see which optimizer is actually superior to the three optimizers used, we try several experiments to see the consistency of the optimizer's performance with the suitability of the Learning Rate described above.

B. Optimizer Performance Consistency

In this research, the drawing is done randomly which results in different accuracy results from the model built. This is where consistent performance results are needed from the model, especially related to optimizer algorithms such as SGD, Adam, and RMSProp to determine the reliability of the optimizer consistently so that it can be concluded which optimizer performance is the best. From the above experiments, it can be seen that the best performance of SGD uses a Learning Rate value of 0.1 with a momentum of 0.9 and RMSProp and Adam with a Learning rate value of 0.001. The performance of the three optimizers will be tested with 6 trials with the best learning rate value of each optimizer algorithm.

TABLE.V
COMPARISON OF OPTIMIZER WITH LEARNING RATE 0.1 AND 0.01

Test	SGD (Lr=0.1)		RMSProp (Lr=0.001)		Adam (Lr=0.001)	
	Score	Cost	Score	Cost	Score	Cost
1	0.7480	0.7255	0.7920	0.5925	0.8280	0.4926
2	0.5470	1.1258	0.7810	0.5898	0.8410	0.4661
3	0.6310	0.9648	0.6980	0.7726	0.8500	0.4278
4	0.5840	1.0965	0.7940	0.5369	0.8320	0.4594
5	0.6410	0.9316	0.7010	0.8466	0.8220	0.5095
6	0.5900	1.0634	0.8060	0.5602	0.8470	0.4614

TABLE.VI
COMPARISON OF PRECISION, RECALL AND F1-SCORE

Result		Test					
		1	2	3	4	5	6
SGD	Precision	0.73	0.46	0.47	0.38	0.58	0.50
	Recall	0.82	0.30	0.50	0.19	0.47	0.34
	F1-Score	0.76	0.36	0.48	0.26	0.49	0.40
RMSProp	Precision	0.85	0.72	0.54	0.57	0.58	0.71
	Recall	0.82	0.74	0.46	0.63	0.64	0.75
	F1-Score	0.84	0.73	0.47	0.58	0.57	0.76
Adam	Precision	0.82	0.74	0.87	0.75	0.70	0.78
	Recall	0.78	0.74	0.82	0.76	0.78	0.84
	F1-Score	0.80	0.74	0.84	0.75	0.74	0.80

After 6 experiments, the results were obtained as shown in Table.V that Adam's Score is the highest with the lowest Loss Cost. Likewise Table.VI the results of Precision, Recall, and F1-Score Optimizer adam are very stable which shows that adam is the best model that can be applied in this study. Followed by RMSProp and SGD optimizers.

C. Confusion Matrix Comparison

Fig.13 shows the plot of the confusion matrix results from the training process using the three optimizer algorithms. On the left side of the plot, there are True Labels of 5 animal classes where this is the actualization of the real animal class and at the bottom, there are Predicted labels where this is the prediction of the training process.

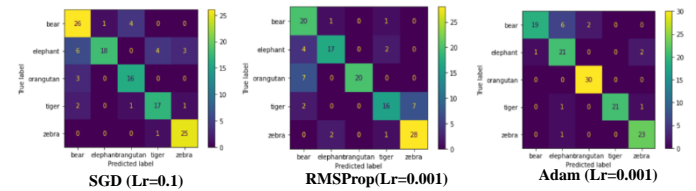


Fig.13 Confusion Matrix of Adam, SGD and RMSProp

D. Manual Calculation of Confusion Matrix

Because in this study the Adam Optimizer is the best algorithm, the researcher only calculates the performance of the optimizer system. Fig.14 is the result of retesting and generating a Confusion matrix using Adam with 5 classes namely bear, elephant, orangutan, tiger, and zebra.

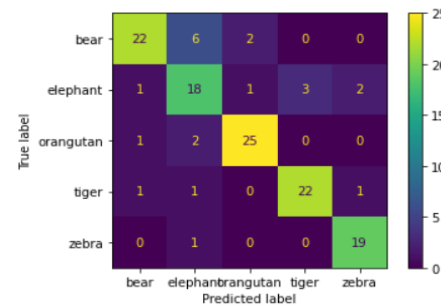


Fig.14 Confusion Matrix using Adam's

At this stage, researchers will try to calculate the class manually by taking an example of one bear class as a representative of other classes.

True Positive = Actual Bear(22) predicted Bear.
 False Negative = Actual Bear(6) predicted elephant + Actual Bear(2) predicted orangutan.
 False Positive = Actual Elephant(1) predicted Bear + Actual orangutan(1) predicted Bear + Actual tiger(1) predicted bear.
 True Negative = Actual Elephant(18) predicted Elephant + Actual Elephant(1) predicted orangutan + actual elephant(3) predicted tiger + actual elephant(2) predicted zebra + actual orangutan(25) predicted orangutan + actual orangutan(2) predicted elephant + actual tiger(22) predicted tiger + actual tiger(1) predicted elephant + actual tiger(1) predicted zebra + actual zebra(19) predicted zebra + actual zebra(1) predicted elephant.

TP = 22, FN = 8, FP = 3, TN = 95.

$$Akurasi (\%) = \frac{TP+TN}{TP+FP+FN+TN} = \frac{22+95}{22+3+8+95} = \mathbf{0,914063}$$

$$Presisi (\%) = \frac{TP}{TP+FP} = \frac{22}{22+3} = \mathbf{0,88}$$

$$Loss (\%) = \frac{FP+FN}{TP+TN+FP+FN} = \frac{3+8}{22+95+3+8} = \mathbf{0,085938}$$

While on the right there are color parameters that indicate the number of images or datasets that are drawn so that they become parts of True Positive, False Negative, false positive, and true negative.

After the precision, accuracy, and loss of each class have been searched, the average will be calculated to determine the precision, accuracy, and loss of the model with the Adam optimizer.

TABLE.VII

MANUAL CALCULATION ACCURACY, PRECISION AND LOSS

Class	Bear	Elephant	Orangutan	Tiger	Zebra	Mean
TP	22	18	25	22	19	
FN	8	7	3	3	1	
FP	3	10	3	3	3	
TN	95	93	97	100	105	
Bitch Size	128	128	128	128	128	
Precision	0,880	0,642	0,892	0,953	0,968	0,867
Accuracy	0,914	0,867	0,953	0,892	0,863	0,898
Loss	0,085	0,132	0,046	0,046	0,03	0,068

From Table.VII above, we can conclude that the best Accuracy is owned by the Orangutan class followed by Bear,

Tiger, Elephant, and Zebra. Meanwhile, the lowest Los is obtained by the Zebra class followed by Tiger/zebra, Bear, and Elephant. And the acquisition of the entire class resulted in a precision of 86.75%, Accuracy of 89.81%, and Loss of 6%. This is a good result considering the data processed in the form of images in a number of varied classes.

E. Data Visualization

To see how the prediction visualization results of the model built will be displayed in the form of images. As explained in the previous explanation that the withdrawal of image data is 128 images. To adjust the display, dimensions of 5 rows and 8 columns are used which only display 40 images.

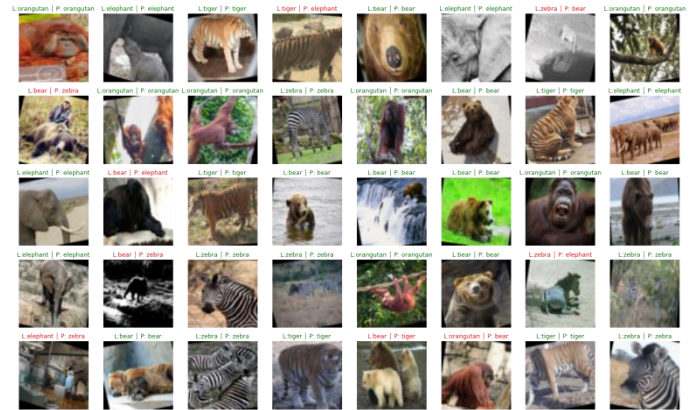


Fig.15 Visualization of Prediction result image using CNN with Adam's Optimizer

From fig.15 It can be seen that the actual label will be compared with the prediction label where if the actual label and prediction label display the same class it will be green, in other words, True Positive. And if the actual label and the prediction label display a different class, it will be red False Positive. Prediction errors can occur because image factors can be caused by several factors such as testing data in this case in the form of unclear images, background influences such as nature, similarities in color and shape, and so on. As in the example of the fourth column of the first row, there is an image of a tiger with the head cut off, a larger machine presented to the elephant image. This can be overcome by adding a tiger image without the head part to the training data.

V. CONCLUSION

In this study, we provide a solution to help scientists identify and monitor protected animal species more accurately, with 89.81% accuracy for our best model. The solution offered is that it can help in monitoring animal species, especially protected ones, cheaper, faster, and more reliably. This research also proves that with the appropriate learning rate in each estimation function, Adam is superior followed by RMSProp and SGD. Suggestions for further research are to compare the effect of activation functions with the same dataset and optimizer in this study to further improve the performance of the previous model.

REFERENCES

- [1] PBB: *One million species of animals and plants are threatened with extinction due to human activity* (2019) *BBC News Indonesia*. BBC. Available at: <https://www.bbc.com/indonesia/magazine-48189137> (Accessed: December 15, 2022).
- [2] J. A. Veech, "A comparison of landscapes occupied by increasing and decreasing populations of grassland birds," *Conserv. Biol.*, vol. 20, no. 5, pp. 1422–1432, 2006, doi: 10.1111/j.1523-1739.2006.00487.x.
- [3] P. D. Meek, G. A. Ballard, P. J. S. Fleming, M. Schaefer, W. Williams, and G. Falzon, "Camera traps can be heard and seen by animals," *PLoS One*, vol. 9, no. 10, 2014, doi: 10.1371/journal.pone.0110832.
- [4] Singh, S.P. (2019) *Fully connected layer: The brute force layer of a machine learning model*, *OpenGenus IQ: Computing Expertise & Legacy*. OpenGenus IQ: Computing Expertise & Legacy. Available at: <https://iq.opengenus.org/fully-connected-layer/> (Accessed: December 15, 2022).
- [5] Setiawan, E. (no date) *Indonesia Dictionary (ID), Meaning of the word animal - Indonesia Dictionary (ID) Online*. Available at: <https://kbbi.web.id/satwa> (Accessed: December 15, 2022).
- [6] Sartika, R.E.A. (2019) *As a result of human life, one million species are threatened with extinction from the earth page all*, *KOMPAS.com*. Kompas.com. Available at: <https://sains.kompas.com/read/2019/05/09/163500923/akibat-kehidupan-manusia-satu-juta-spesies-terancam-punah-dari-bumi?page=all> (Accessed: December 15, 2022).
- [7] D. Saravanan, D. Joseph, and S. Vaithyasubramanian, *Effective utilization of image information using data mining technique*, vol. 172. 2019. doi: 10.1007/978-3-030-32644-9_22.
- [8] Hidayatullah, P., Wang, X., Yamasaki, T., Mengko, T. L. E. R., Munir, R., Barlian, A., Sukmawati, E., & Suprpto, S. (2021). DeepSperm: A robust and real-time bull sperm-cell detection in densely populated semen videos. *Computer Methods and Programs in Biomedicine*, 209, 106302. <https://doi.org/10.1016/j.cmpb.2021.106302>
- [9] Wanayumini, Sitompul, O. S., Suwilo, S., & Zarlis, M. (2020). Supervised image classification of chaos phenomenon in cumulonimbus cloud using spectral angle mapper. *International Journal on Advanced Science, Engineering and Information Technology*, 10(3), 987–992. <https://doi.org/10.18517/ijaseit.10.3.11493>
- [10] Courville, I. G. and Y. B. and A. (2016). Deep learning 简介一、什么是 Deep Learning ? . *Nature*, 29(7553), 1–73. <http://deeplearning.net/>
- [11] Ujjwalkarn (2016) *A quick introduction to Neural Networks*, *Ujjwalkarn*. Available at: <https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks> (Accessed: December 15, 2022).
- [12] Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.
- [13] Madduri, A., Adusumalli, S. S., Katragadda, H. S., Dontireddy, M. K. R., & Suhasini, P. S. (2021). Classification of Breast Cancer Histopathological Images using Convolutional Neural Networks. *Proceedings of the 8th International Conference on Signal Processing and Integrated Networks*, SPIN 2021, 755–759. <https://doi.org/10.1109/SPIN52536.2021.9566015>
- [14] Li, J., Cheng, J. H., Shi, J. Y., & Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. *Advances in Intelligent and Soft Computing*, 169 AISC(VOL. 2), 553–558. https://doi.org/10.1007/978-3-642-30223-7_87
- [15] Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3), 400–407. <https://doi.org/10.1214/aoms/1177729586>
- [16] Ethem Alpaydm. (2019). Introduction to Machine Learning. In *Library of Congress Cataloging-in-Publication Information Alpaydm*, (Third Edit, Vol. 4, Issue 1).
- [17] Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. *Genetic Programming and Evolvable Machines*, 19(1–2), 305–307. <https://doi.org/10.1007/s10710-017-9314-z>
- [18] Mukkamala, M. C., & Hein, M. (2017). Variants of RMSProp and adagrad with logarithmic regret bounds. *34th International Conference on Machine Learning, ICML 2017*, 5, 3917–3932.
- [19] Zhang, Z. (2019). Improved Adam Optimizer for Deep Neural Networks. *2018 IEEE/ACM 26th International Symposium on Quality of Service, IWQoS 2018*, 1, 1–2. <https://doi.org/10.1109/IWQoS.2018.8624183>
- [20] Lan, K., Liu, L., Li, T., Chen, Y., Fong, S., Marques, J. A. L., Wong, R. K., & Tang, R. (2020). Multi-view convolutional neural network with leader and long-tail particle swarm optimizer for enhancing heart disease and breast cancer detection. *Neural Computing and Applications*, 32(19), 15469–15488. <https://doi.org/10.1007/s00521-020-04769-y>
- [21] Gunawan, T. S., Gani, M. H. H., Rahman, F. D. A., & Kartiwi, M. (2017). Development of face recognition on raspberry pi for security enhancement of smart home system. *Indonesian Journal of Electrical Engineering and Informatics*, 5(4), 317–325. <https://doi.org/10.11591/ijeei.v5i4.361>