INFORM : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi
Vol. 5 No. 2, July 2020, P-ISSN: 2502-3470, E-ISSN: 2581 – 0367

Modeling of Convolutional Neural Network Architecture for Recognizing The Pandava Mask

Andi Sanjaya[1], Endang Setyati[2], Herman Budianto[3]

[1,2,3]*Information System Technology Department, Institut Sains dan Teknologi Terpadu Surabaya, Indonesia*

[1]4ndisanjaya@gmail.com*

[2]endang@stts.edu, [3]herman.budianto@gmail.com

Abstract— This research was conducted to observe the use of architectural model Convolutional Neural Networks (CNN) LeNEt, which was suitable to use for Pandava mask objects. The Data processing in the research was 200 data for each class or similar with 1000 trial data. Architectural model CNN LeNET used input layer 32x32, 64x64, 128x128, 224x224 and 256x256. The trial result with the input layer 32x32 succeeded, showing a faster time compared to the other layer. The result of accuracy value and validation was not under fitted or overfit. However, when the activation of the second dense process as changed from the relu to sigmoid, the result was better in sigmoid, in the tem of time, and the possibility of overfitting was less. The research result had a mean accuracy value of 0.96.

Keywords— classification; CNN; lenet; relu; activation; sigmoid activation

I. Introduction

Face recognition becomes one of the research focus lately, just like research about human face expression [1] and the process of face recognition like face leveling out and the extraction of its feature [2]. The research of heterogeneous face recognition is to match the picture of the face obtained from sensing modality, which was different from mission-critical application in forensic, security, and commercial [3]. The research about the comprehensive work scheme to deal with the challenge in face recognition was video-based [4] and the research about face recognition based on face features [5].

Many algorithms used to classify the pictures before CNN becomes very popular. The researchers usually made the feature from the pictures and then inserted the features into some classified algorithms like SVM. Some algorithms used picture pixel value as a vector feature. For example, it is for training SVM with 784 features in which every feature is the pixel value for 28 x 28 pictures.

CNN now follows the hierarchy model, which has the function to built networks like the tunnel and, in the and gives layer that is fully connected to every neurons, connected each other, and output process. CNN has developed rapidly in the computer vision field [6]. In the computer vision field, CNN was usually used as an introduction method dan object classification [7]. In implementing, CNN needs to be determined by the architectural model. The architectural model on CNN has many parameters [8]. The parameter will be changed as needed. The determination of the architectural model was done by doing a test to know which good architectural model is suitable for an object. [9].

The aim of this paper is to use the Architectural model LeNet [9][10] implemented the Pandava mask object. The researcher used some input data as a training data test with the form of an RGB color. Next, it compared the result of input data with another form of color like grayscale and Black and White to the same architectural model or even by changing some parameters. Furthermore, it was found a CNN architectural model with a suitable parameter for the Pandava mask. By using some parameters, hopefully, this research was able to give a maximum accurate result.

II. Research Methodology

The description of the activity in this paper is divided into two processes, the collection of the data and conducting parameter design. The data collection discussed how we prepared the data used as the data set for training for CNN LeNet architectural model. The conducting parameter design discussed what parameter I used to do the trial.

A. *Dataset test*

A dataset trial is my personal collection, which I got from Ganesya Museum in Malang, East Java, Indonesia. There is five class of the dataset such as Yudhishthira, Bhima, Arjuna, Nakula, and Sahadewa. Each class has 200 sets of data. So, the total data used in this research was the 1000 data set, as seen in Figure 1.

Arjuna

Bima

DOI: http://dx.doi.org/10.25139/inform.v0i1.2740

Nakula

Sadewa

Yudistira

Figure 1. The dataset of each class

B. Data Processing (Pre-processing)

This data processing was needed to be able to get the maximum accurate result [11][12]. The technique in the picture data processing or image processing transformed image to another image; hopefully, the quality of the image was getting better. There were two processes conducted in processing the data. They are :

- *Cropping* was a process used to change the image, becoming more focused and similar in shape and size. Cropping was a way to erase the out test area of an undesirable area of the image. *Cropping* also means erasing the area of the angle of the image to cut out/to take/to take out some of the content of the image to get the desired result. This process usually consists of the erasing of some image peripherals to dismiss the unneeded area, to increase the framing, and to change the size of the ratio aspect. The process was done by using software to process the image and conducted manually.
- *Clean The Noise.* Noise is a term in photography. It is a colored spot which is usually disturbing the photo result, so it made the image not smooth enough. The noise had ordinarily resulted when the picture was taken luck of lighting and when in the picture contained many disturbing spots. Clean the noise was usually conducted to increase the picture quality, so in the end, it will produce high accuracy value. This process was also done manually by the aid of the image processing application.

The example of the picture which had been through preprocessing can be seen in Figures 2.

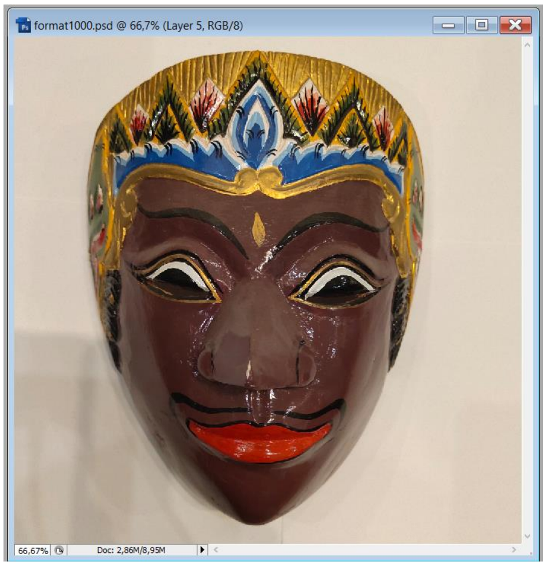


Figure 2. Image after preprocessing

C. Convolutional Neural Network

A convolutional neural network (CNN) is a development from multilayer Perceptron (MPL), which has the function to process two-dimension data. CNN was firstly developed by using the name NeoCognitron by Kunihiko Fukushima, a researcher from NHK Broadcasting Science Research laboratories, Kinuta, Setagaya, Tokyo, Japan [13]. Convolutional neural network (CNN) is a kind of artificial neural network used in the recognition and procession of an image. CNN imitate the way our neural cells to communicate with the connected neuron. CNN has the architectural. This artificial neural network has its own uniqueness. It is the convolutional applying filter for each input section before conducting the extraction pattern and feature maps.

One of the popular research in this field is the development of LeNet by Lecunn [14]. This is one of the CNN which was used for the first time in the bank to read the cheque by real-team. Though there was Another algorithm like SVM, which was the accuracy almost like LeNet, many opinions said that the speed calculation of CNN expotionally faster than others.

D. Parameter Design

The architectural model used in this discussion paper applied model CNN LeNet which was developed for the first time by LeCun [14][15][16]. In the stage of the making, the CNN model in the stage of training was a parameter design that will be used to classify. By determining the model, will get a good level of accuracy.

The matrix input used is 32 x 32 x 1 with the filter 5 x 5,1 strides and 0 pool. The size of matrix 32 x 32 by forming the convolution matrix by the size 28 x 28 based on equation(1) and equation (2). The discretization process was done based on the sample using max-pooling 2D, and Filter 2x2, just as shown in Figures 3.

DOI: http://dx.doi.org/10.25139/inform.v0i1.2740

The process of Figures 3 was explained that the process by inputting the use of parameter design. The image will be proceeded until getting an accuracy result, which was considered maximum, and this was conducted until 100 epoch. The process can be seen in Figure 4.

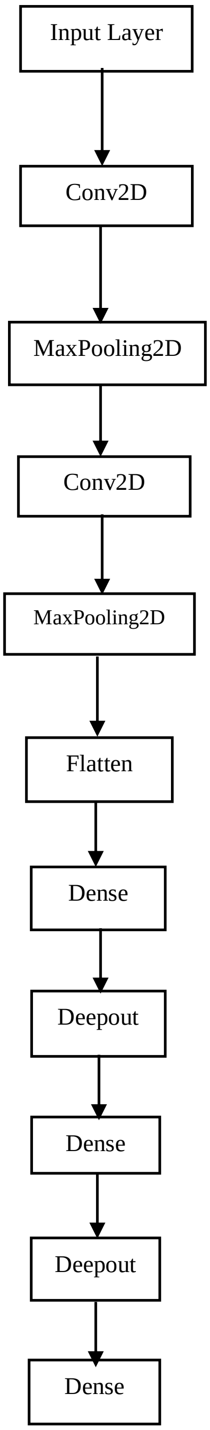
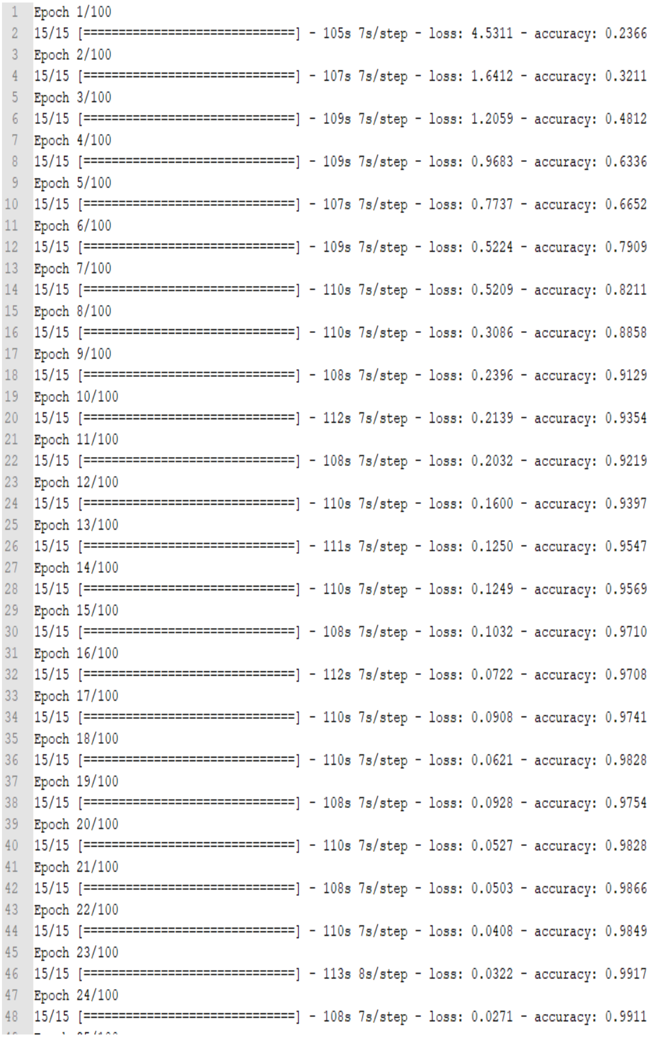


Figure 3. LeNet model architecture



```
Epoch 1/100
15/15 [==============================] - 105s 7s/step - loss: 4.5311 - accuracy: 0.2366
Epoch 2/100
15/15 [==============================] - 107s 7s/step - loss: 1.6412 - accuracy: 0.3211
Epoch 3/100
15/15 [==============================] - 109s 7s/step - loss: 1.2059 - accuracy: 0.4812
Epoch 4/100
15/15 [==============================] - 109s 7s/step - loss: 0.9683 - accuracy: 0.6336
Epoch 5/100
15/15 [==============================] - 107s 7s/step - loss: 0.7737 - accuracy: 0.6652
Epoch 6/100
15/15 [==============================] - 109s 7s/step - loss: 0.5224 - accuracy: 0.7909
Epoch 7/100
15/15 [==============================] - 110s 7s/step - loss: 0.5209 - accuracy: 0.8211
Epoch 8/100
15/15 [==============================] - 110s 7s/step - loss: 0.3086 - accuracy: 0.8858
Epoch 9/100
15/15 [==============================] - 108s 7s/step - loss: 0.2396 - accuracy: 0.9129
Epoch 10/100
15/15 [==============================] - 112s 7s/step - loss: 0.2139 - accuracy: 0.9354
Epoch 11/100
15/15 [==============================] - 108s 7s/step - loss: 0.2032 - accuracy: 0.9219
Epoch 12/100
15/15 [==============================] - 110s 7s/step - loss: 0.1600 - accuracy: 0.9397
Epoch 13/100
15/15 [==============================] - 111s 7s/step - loss: 0.1250 - accuracy: 0.9547
Epoch 14/100
15/15 [==============================] - 110s 7s/step - loss: 0.1249 - accuracy: 0.9569
Epoch 15/100
15/15 [==============================] - 108s 7s/step - loss: 0.1032 - accuracy: 0.9710
Epoch 16/100
15/15 [==============================] - 112s 7s/step - loss: 0.0722 - accuracy: 0.9708
Epoch 17/100
15/15 [==============================] - 110s 7s/step - loss: 0.0908 - accuracy: 0.9741
Epoch 18/100
15/15 [==============================] - 110s 7s/step - loss: 0.0621 - accuracy: 0.9828
Epoch 19/100
15/15 [==============================] - 108s 7s/step - loss: 0.0928 - accuracy: 0.9754
Epoch 20/100
15/15 [==============================] - 110s 7s/step - loss: 0.0527 - accuracy: 0.9828
Epoch 21/100
15/15 [==============================] - 108s 7s/step - loss: 0.0503 - accuracy: 0.9866
Epoch 22/100
15/15 [==============================] - 110s 7s/step - loss: 0.0408 - accuracy: 0.9849
Epoch 23/100
15/15 [==============================] - 113s 8s/step - loss: 0.0322 - accuracy: 0.9917
Epoch 24/100
15/15 [==============================] - 108s 7s/step - loss: 0.0271 - accuracy: 0.9911
```

Figure 4. Training process

The Scheme of the first trial based on the flow in Figure 5 was conducted using five design parameters. It was prepared with the design parameter input layer using RBG colored form. The first trial was:

- Using the design parameter using another input layer 32x32.
- They are continuing the trial with the second design parameter with input layer 64x64.
- The third design parameter using input layer 128x128.
- The fourth design parameter using input layer 224x224.
- The fifth design parameter using input layer 256x256.

The result of the trial with the first parameter will be concluded by the input layer, which produced good results. The indicator to state the good result are :

- Processing timing.
- Accuracy and validation accuracy was not overfit or underfit.
- To choose input layer to do the second test based on the Scheme in Figure 6.

DOI: http://dx.doi.org/10.25139/inform.v0i1.2740

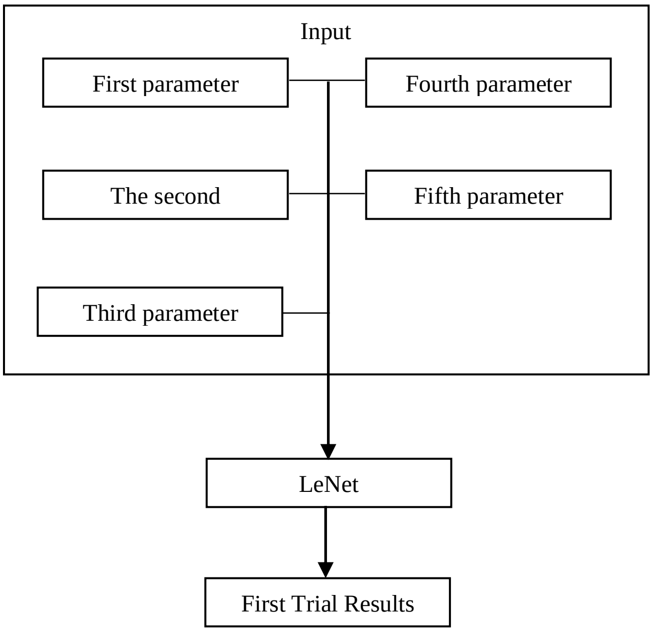


Figure 5. Systematic first try

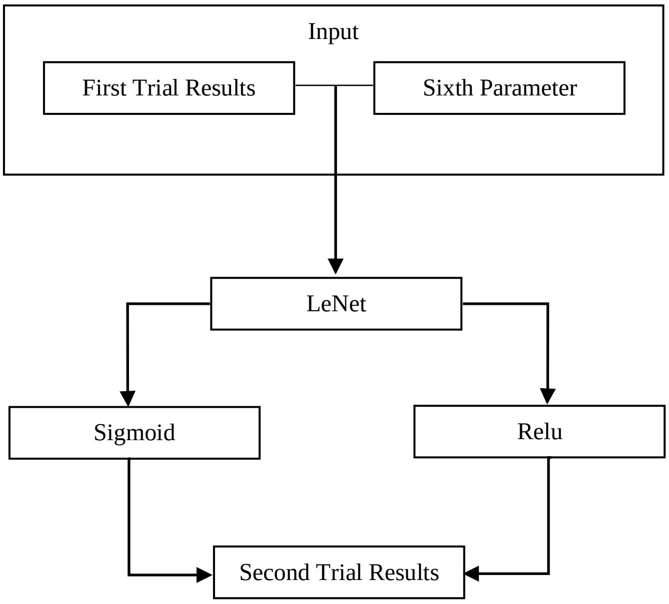


Figure 6. The second systematic test

The second trial used parameter six. Parameter six used an architectural model like parameter one, but the dense process two used sigmoid as the activation. The result of the second trial can be noticed how the performance of the architectural model by changing the activation to the last dense process. The parameter design will be trialed to the dataset, as seen in Table I.

TABEL I
FIRST TRIAL PARAMETER DESIGN

Epoch	***Parameter design***				
	1	**2**	**3**	**4**	**5**
1	0,4104	0,2958	0,4417	0,2125	0,4569
2	0,4188	0,6432	0,4145	0,3889	0,5108
3	0,5299	0,5385	0,4979	0,406	0,7478
4	0,5876	0,7137	0,7415	0,4658	0,7284
5	0,5406	0,6859	0,7714	0,7094	0,9267
6	0,6603	0,7564	0,8675	0,7799	0,9246
7	0,6752	0,8034	0,7714	0,8056	0,9263
8	0,7009	0,8056	0,9509	0,8376	0,9784
9	0,7051	0,8568	0,8953	0,8846	0,9914
10	0,7543	0,7799	0,9701	0,9487	0,9892
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
100	1	1	1	1	1

III. RESULT AND DISCUSSION

In the first trial on the Table I succeeded in getting the data of the design parameter in the first trial. It was recognized that validation accuracy increased in every input layer, even it was fluctuating. Table II describes the mean value of accuracy (MA), the mean value of Validation Acucraccy (MVA), NA value=1 was the sum of the accuracy which reached 100%, Non-vegetated Vertical Accuracy value (NVA) was the sum of validation accuracy which reached 100%.

The result of Table II had a very good accuracy value because it was more than 0.8 or 80 %. Yet, the necessarily observed material was NA and NVA, the prediction can be 100 %. If the prediction often appeared, so it was called overfit [9]. Therefore, the first parameter chose *input layer* 32x32 as a good parameter.

DOI: http://dx.doi.org/10.25139/inform.v0i1.2740

TABEL II
COMPARATIVE RESULTS OF THE FIRST TRIAL

Input Layer	RA	RVA	NA = 1	NVA = 1	Processing time (seconds)
32x32	0,9	0,94	0	40	1022
64x64	0,89	0,95	0	47	4392
128x128	0,94	0,97	4	75	18410
224x224	0,93	0,96	1	51	57450
256x256	0,96	0,98	18	84	74228

The second experiment compared the *activation function* sigmoid with relu. The task of the activation was to make the neural network become no-linear. Sigmoid had a value which ranged from 0 to 1, it can be seen at Figure 7.

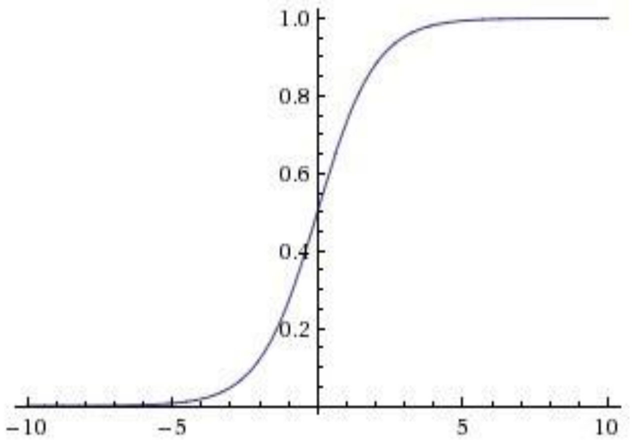


Figure 7. Activation Function Sigmoid

Meanwhile relu, in the point, only made restriction to number 0, it means if x ≤ 0 so x = 0 and if x > 0 so x = x, can be seen in Figure 8.

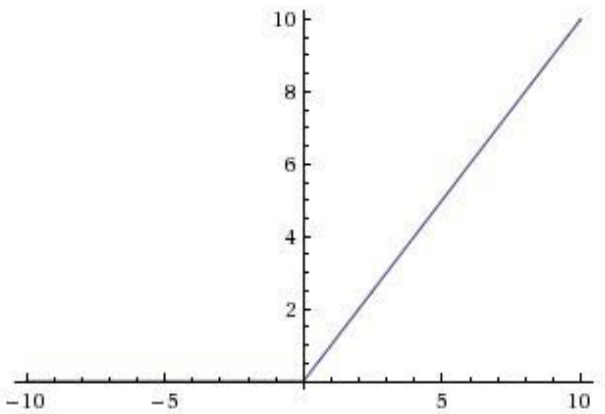


Figure 8. Activation Function Relu

The second trial can be noticed that the comparison results in Table III. We can conclude that sigmoid was faster than relu by seeing the result of the trial. The performance from LeNET indicated the result, as in Table III. In performance MA and MVA of sigmoid lower than relu, but for the NVA relu was more than sigmoid.

TABEL III
COMPARATIVE RESULTS OF THE SECOND TRIAL

Aktivasi	RA	RVA	NA = 1	NVA = 1	Processing time (seconds)
relu	0,9	0,94	0	40	1022
sigmoid	0,86	0,9	0	11	931

IV. CONCLUSION

The difference between the object made a difference in the architectural model used. This research had conducted any kind of input layer with a similar architectural model. But the result differed very much, like in Table I. It can be inferred that the accuracy value in the first experiment was 0.75, the second was 0.77, the third was 0.97 the fourth was 0.94 and the fifth was 0.98. Every parameter in the architectural model used CNN will provide different results. The architectural model was considered good if it was not under fitted or overfitted.

REFERENCE

[1] D. Qu, Z. Huang, Z. Gao, Y. Zhao, X. Zhao, and G. Song, "An Automatic System for Smile Recognition Based on CNN and Face Detection," *2018 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2018*, pp. 243–247, 2018, doi: 10.1109/ROBIO.2018.8665310.

[2] S. Sharma, K. Shanmugasundaram, and S. K. Ramasamy, "FAREC - CNN based efficient face recognition technique using Dlib," *Proc. 2016 Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2016*, no. 978, pp. 192–195, 2017, doi: 10.1109/ICACCCT.2016.7831628.

[3] R. He, S. Member, X. Wu, Z. Sun, and T. Tan, "Wasserstein CNN : Learning Invariant Features for NIR-VIS Face Recognition," vol. 14, no. 8, pp. 1–11, 2017.

[4] C. Ding and D. Tao, "Trunk-Branch Ensemble Convolutional Neural Networks for Video-based Face Recognition," no. X, pp. 1–14, 2017.

[5] H. Zhang, Z. Qu, and L. Yuan, "A Face Recognition Method Based on LBP Feature for CNN," pp. 544–547, 2017.

[6] M. Y. W. Teow, "Understanding Convolutional Neural Networks Using A Minimal Model for Handwritten Digit Recognition," no. October, pp. 167–172, 2017.

[7] S. Albawi and T. A. Mohammed, "Understanding of a Convolutional Neural Network," no. April 2018, 2017, doi: 10.1109/ICEngTechnol.2017.8308186.

[8] F. Ertam, "Data Classification with Deep Learning using Tensorflow," no. October, 2017, doi: 10.1109/UBMK.2017.8093521.

[9] T. Septianto, E. Setyati, and J. Santoso, "Model CNN LeNet dalam Rekognisi Angka Tahun pada Prasasti Peninggalan Kerajaan Majapahit," vol. 6, no. April, pp. 106–109, 2018, doi: 10.14710/jtsiskom.6.3.2018.106-109.

[10] S. S. Ahranjany, F. Razzazi, and M. H. Ghassemian, "A very high accuracy handwritten character recognition system for Farsi/Arabic

DOI: http://dx.doi.org/10.25139/inform.v0i1.2740

digits using convolutional neural networks," *Proc. 2010 IEEE 5th Int. Conf. Bio-Inspired Comput. Theor. Appl. BIC-TA 2010*, pp. 1585–1592, 2010, doi: 10.1109/BICTA.2010.5645265.

[11] T. I. Saputra, F. Fauziah, and N. Hayati, "Implementasi Discrete Wavelet Transform Pada Aplikasi Kompresi Citra Medis," *J. InfomediaTeknik Inform. Multimed. Jar.*, vol. 4, no. 2, pp. 101–107, 2020, doi: 10.30811/JIM.V4I2.1574.

[12] I. Riadi, R. Umar, and F. D. Aini, "Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (Svm)," *Ilk. J. Ilm.*, vol. 11, no. 1, pp. 17–24, 2019, doi: 10.33096/ilkom.v11i1.361.17-24.

[13] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.

[14] A. Carruthers and J. Carruthers, "Handwritten Digit Recognition with a Back-Propagation Network," *Dermatol. Surg.*, vol. 39, no. 1 Pt 2, p. 149, 2013, doi: 10.1111/dsu.12130.

[15] I. W. S. E. Putra, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101 Image Classfication Using Convolution Neural Network (Cnn) on Caltech 101," *Inst. Teknol. Sepuluh Novemb.*, vol. 5, no. 1, pp. 1–76, 2016.

[16] H. Darmanto, "Pengenalan Spesies Ikan Berdasarkan Kontur Otolith," *Joined J.*, vol. 2, 2019.

DOI: http://dx.doi.org/10.25139/inform.v0i1.2740