2020 Information Technology International Seminar (ITIS)
Surabaya, Indonesia, October 14-16, 2020

A Modification of Convolutional Neural Network Layer to Increase Images Classification Accuracy

Wahyudi Agustiono
Information System Department
Engineering Faculty
Trunojoyo University
Madura, Indonesia
wahyudi.agustiono@trunojoyo.ac.id

Mohammad Imam Utoyo
Applied Mathematics Department
Sains and Technology Faculty
Airlangga University
Surabaya, Indonesia
m.i.utoyo@fst.unair.ac.id
https://orcid.org/0000-0003-2292-8443

Riries Rulaningtyas
Physics Department
Sains and Technology Faculty
Airlangga University
Surabaya, Indonesia
riries-r@fst.unair.ac.id

Budi Dwi Satoto*
Information System Department
Engineering Faculty
Trunojoyo University
Madura, Indonesia
* Corresponding author:
budids@trunojoyo.ac.id
https://orcid.org/0000-0002-1919-0540

2020 6th Information Technology International Seminar (ITIS) | 978-1-7281-7726-7/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITIS50118.2020.9321011

***Abstract*— Image classification is one of the fundamental steps in digital image processing. Research in this area has received considerable attention, with photos shared on social media which are sometimes similar but have different identities. There are various classification methods proposed in the literature to improve accuracy. One important strategy is Convolutional Neural Networks (CNN). Although CNN is superior in pattern recognition, it has limitations inaccuracy. It requires additional training time, especially when dealing with variants in data generated from a large number of images but of similar properties. Therefore, this study aims to overcome this problem by proposing a modification of the CNN layer to increase the accuracy of the multi-class image classification. This research used four different flower species with similar patterns added from a public database. Each category consists of 400 colour images with different angles, backgrounds, and lighting conditions that provide different variations to the training process. Through experiments using 1,600 of the four flower species, this study shows that the 18-34 layer modification produces the most optimal accuracy in the training process ranging from 99.3% with misclassification of MSE= 0.0025, RMSE= 0.1606, and MAE= 0. 0133. Meanwhile, the computation time required to compile the data set is 3 minutes, 18 seconds. This result is 50% faster when compared to computation time using existing architecture such as Alexnet model with a similar number of layers.**

Keywords—Multi-class image, Convolutional neural network, data augmentation, Flower classification.

I. INTRODUCTION

The purpose of flower classification is to make it easier to identify, compare and research living things. Comparing means looking for similarities and differences in the properties or characteristics of these flowers [1]. The benefit is to know the relationship between living things with one another. Because there are many variants of interest, the classification is carried out on multiple interest classes. The multi-class itself is a classification carried out for the number of courses that are more than two (binary classes). In a class binary, there are only two outputs, but in multi-class, there can be more than two outputs [2].

Siyuan Lu researched flower classification in 2017. The step was to extract the colour features and wavelet entropy from the image of flower petals. The experimental results show that Weighted K-nearest Neighbors performs best among the four classifiers, with an overall accuracy of 99.4% [3] . Gavai, 2017 also conducted experimental performance research using the combined method of Googlenet architecture with inception3 is used. In experimental trials with a mobile-net, it was found that accuracy was 91%, 4% better than Squeezenet, 9.4 times more efficient than alexnet in terms of computation time [4].

Hairy, 2017 researched flower classification by Interest segmentation approach. Binary classification is proposed to get the model. Next, a robust convolutional neural network classifier is built to differentiate between different types of flowers. The architecture used is VGG-16. The classification results show the average accuracy exceeding 97% in all data sets [5].

Flower classification research was also conducted by Busra, 2018, which was conducted by comparing the learning performance of classifiers such as SVM, Random Forest, KNN, and Multi-Layer. The results of the training show that the average accuracy for the SVM Classifier is 98.5% using Oxford-102. On the Oxford 17-Flowers Dataset, the best accuracy is 99.8% using the Multilayer Perceptron Classifier (MLP) [6].

This research proposes a neural network method to solve interest classification problems. GAP with previous research is that this research considers the use of Convolutional Neural Network (CNN) as a new method of using machine learning. In this research described above, previous researchers still used supervised machine learning. Besides, the previous CNN research even used the existing architecture. This research also considers the computation time of the training process, which was not discussed in previous studies. The weakness of CNN is that it uses a lot of hidden layers to get high accuracy. The novelty offered is the use of a custom 18-34-layer to save computing time in the training process. Besides, there is augmentation data to increase data variation when the amount

of training data is not large. The research feature that can be developed is the use of optimization to obtain training process parameters.

II. Related Work

A. Image multi-class classification of Flower

Multi-class is a classification of a feature into one target variable, with more than two class choices. Meanwhile, the multi-label category is a generalization of the multi-class type [7]. A flower will be classified shown in Fig. 1.

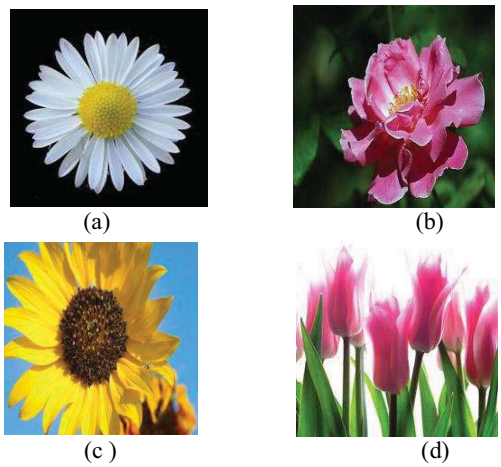


Fig. 1. Flower (a) Daisy (b) Rose (c) Sunflower (d) Tulip

In multi-label classification, there are different classes while in multi-class it is a problem with single labelling but is categorized in one of more than two categories correctly. In a multi-label problem, there is no limit to how many classes are used in the problem [8].

B. Convolution Neural Network

The CNN contribution lies in the convolution layer and the pooling layer. Convolution works on the principle of sliding windows and weight sharing to reduce the complexity of calculations . The axon output can be written in (1).

$$y_i = \sum_i w_i x_i + b \tag{1}$$

with w_i = weight / synapse, x_i=axon from a neuron, and b=biases. While the pooling layer is useful for summarizing the information generated by a convolution by reducing dimensions [9]. In general, the convolution layer receives image data with size $W_1 x H_1 x D_1$ where there are four hyperparameter components, namely Number of filter K, spatial extend F, stride S dan amount of zero padding P [10]. So that the output at the second layer can be defined in (2).

$$W_2 = \frac{(W_1 - F + 2P)}{S+1}, H_2 = \frac{(H_1 - F + 2P)}{S+1} \text{ and } D_2 = K \tag{2}$$

W= Width, H= height, and D=depth. With the number of parameters shared, the value is $F.F.D_1$ Weight per filter. The total weight value is $(F.F.D_1).K$. The standard setting from hyperparameter CNN is F=3, S=1 and P=1. CNN has several existing architectures including alexnet, google net, vgg16, vgg19, resnet18, resnet50, resnet101, densenet, and squeeze net. This architecture can be used as a comparison in determining the results and computation time [11].

C. Stride and Padding

Stride is a parameter that determines the number of filter shifts. If the value of stride is 1, then Conv. The filter moves expressed in (3).

$$Out_{shape} = (-k_h + p_h + 1)\, x\, (n_w - k_w + p_w + 1) \tag{3}$$

When $n_h x n_w$=input matrix, $k_h x k_w$ =kernel size, p_h= Rows of padding, and p_w = Column of padding. The filter will shift by 1 pixel horizontally and then vertically shown in Fig. 2.

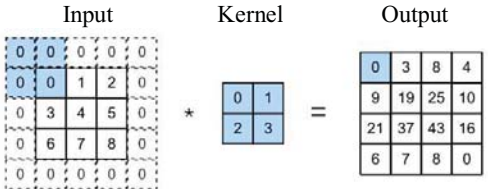


Fig. 2. Two-dimensional cross-correlation with padding

Padding is a parameter that determines the number of pixels (containing the value 0) to be added on each side of the input [12].

D. Batch Normalization

Batch normalization is the input or output normalization of the activation function, which is in the hidden layer. With information from x via minibatch, $B = \{x_{1...m}\}$ and parameter learned γ, β and output $\{y_i = BN_{\gamma,\beta}(x_i)\}$. It can be expressed in (4).

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i\,, \sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{4}$$

The batch normalization value is obtained in (5).

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \text{ and } y_i \leftarrow \gamma \widehat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{5}$$

With μ_B= mini-batch mean, σ_B^2= minibatch variance, $\widehat{x_i}$= normalization, y_i =scale and shift. Batch Normalization provides the benefits of Making the neural network more stable by protecting it against outlier weight, Allows higher learning speeds and Reduces overfitting [13] .

E. Data augmentation

Data augmentation is a process in image data processing, and boost is the process of changing or modifying an image in such a way that the computer will detect that the transformed image is a different image [14]. Scaling is x, y, with a direction shown in (6).

$$A = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \tag{6}$$

Rotation methods are written in (7) [15].

$$A = \begin{pmatrix} \cos\theta & -sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{7}$$

F. Confusion Matrix

The Confusion Matrix is also often called the misclassification matrix. Confusion matrix in the form of a matrix table that describes the performance of the classification model on a series of test data for which the actual value is previously known. Confusion matrices provide information on the comparison between the classification results carried out by the system (model) and the actual classification results. [16].

G. Error Classification

There are three indicators of error that are used as a reference in this study, namely Mean Square Error (MSE), Root Mean Square (RMSE) dan Mean Absolute Error (MAE). It calculated based on the predicted label and the actual label as expressed in (8)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \quad (8)$$

MAE is the absolute average error value of the prediction process as expressed in (9)

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \quad (9)$$

MSE is minimized by the conditional mean, while the conditional median calculates MAE [17].

III. Research Methods

The research chart can be seen in Fig. 3.

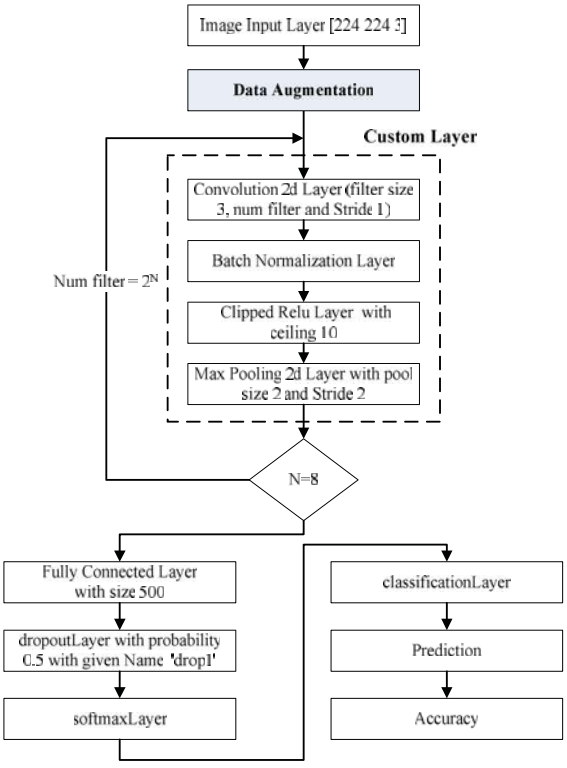


Fig. 3. Research Methods

A. Flow of Framework

The research method begins with pre-processing equalizing the dimensions of file width, height and bit depth. After the data is ready, and there are no errors, the image auto contrast is performed. Furthermore, the system will provide a convolutional layer to obtain a feature map after the filtering process is normalized to stabilize the network. The requirement should be met an option for training has the same setting. The one that can be a different just number of layer and optimization training. Training process time not more than three minutes. After full five retraining, the result has average accuracy not less than 98% and The training process should meet the stability.

B. Hardware dan Software

The hardware used in this research is Intel core i-7 RAM, 8 GB with a 4GB NVIDIA GTX 1050 GPU. The software used is MATLAB2019a

C. Dataset

Secondary data are available on the internet that can be used. The various of Flower dataset address used is https://www.kaggle.com/alxmamaev/flowers-recognition. Total File Size of 450MB contains 4242 files of a flower. The data should conversion first before used because there are several files has different dimension or depth of bit. There are four class used to train contain 400 file daisy, 400 file rose, 400 file sunflower, and 400 file tulip. Total used after size pre-processing is 1600 file.

IV. Results and Discussion

A. Pre-processing

The initial pre-processing that is carried out is to ensure that the entire dataset to be used is following the requirements of the CNN input layer, namely [224 224 3].

(a) (b)

Fig. 4. Preprocessing (a) daisy (b) rose

Fig. 4 shows that all image in each folder already has the same size. If there several file miss, the process could not be done, or the program will be halt.

B. Feature Maps Layer

The feature contains Convolutional layer and pooling layer. Input layer that includes formula in (10)

$$n_{out} = \frac{n_{in}+2P-F}{S} + 1 \quad (10)$$

With n_{in} = neuron length or High of Input, F=Length or Height of Filter, P=Zero Padding, S=Stride

(a) (b)

Fig. 5. Feature Maps (a) layer conv1-7x7_s2 (b) layer conv2-3x3

Fig. 5 shows the convolution process in Googlenet architecture carried out on the feature maps layer. The deeper the learning, the more detailed the image dimensions will be. The use of more convolutional layers can increase accuracy.

C. Data Augmentation

The data augmentation stage was carried out to increase variation due to the limited amount of data. Apart from scaling and rotation, what can be done is ShearsX and ShearY, which are expressed in (11) and Fig. 6.

$$A = \begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, x' = x + s_v y \; and \; y' = y \qquad (11)$$

(a) (b)

Fig. 6. Data Augmentation (a) Scaling&Rotation (b) ShearX&ShearY

D. Training Progress

The training process is carried out using a dataset with a ratio of 80% data validation and 20% data testing of the total training data. Training options pay attention to minibatch, Initial Learning Rate 3e-4, MaxEpochs 30, Validation Frequency 30, Verbose true. Training has shown in Fig. 7.

E. Confusion matrix

The confusion matrix works based on the difference between the actual label and the prediction label expressed in (12).

$$Accuracy = \frac{TP+T}{TP+TN+FP+F} \qquad (12)$$

With TP=True Positive, TN= True Negative, FP=False positive, FP=False positive.

Fig. 8. Confusion Matrix

From Fig. 8, the Confusion matrix can be explained that the average classification accuracy result is 99.3%. From 80% of the 400-training data or about 320 data, the classification of Rose and sunflower can be classified correctly. In the daisy class, there is one error entering rose class. Class sunflower enters daisy two files. Tulip class enter rose six.

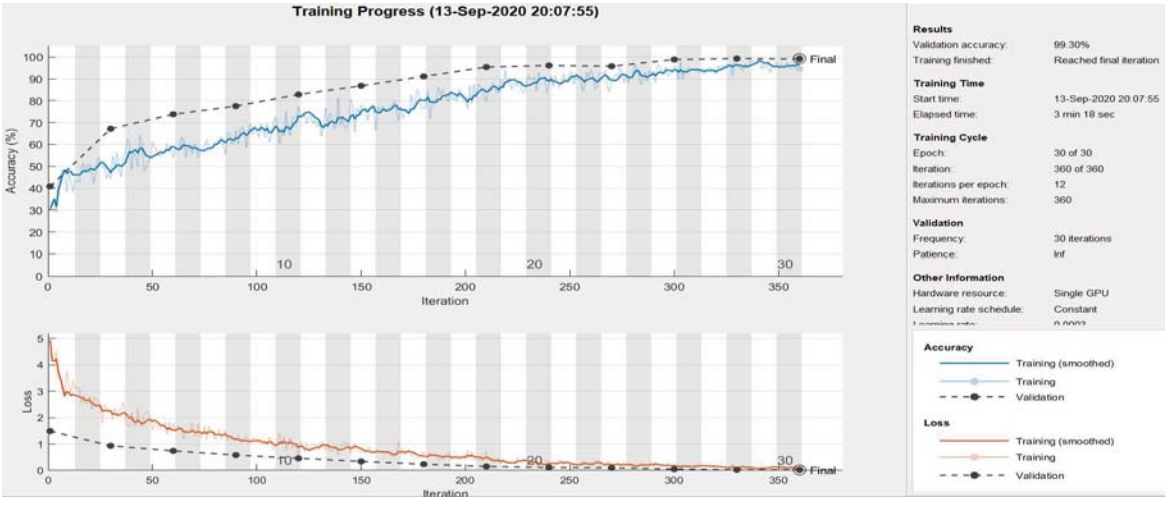


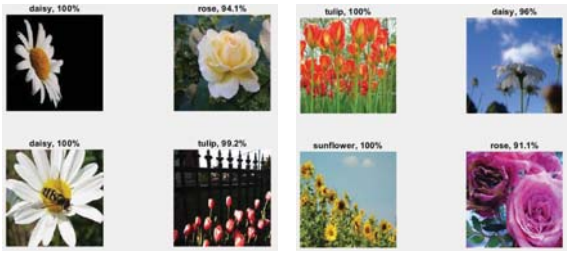
Fig 7. Training CNN using 26 Layer

F. Prediction

The predictions on CNN aim to gain a level of confidence in the training process expressed in (13).

$$Acc = \frac{1}{num_{val}} \sum_i Y_{pred} \equiv Y_{val} \quad (13)$$

The prediction results are shown as follows:



(a) (b)

Fig. 9. Flower Prediction Result (a) Prediction 1 (b) Prediction 2

Fig. 9 shows the predicted results. Fig. 9 (a) shows that the predictive confidence level for daisy is 100%, rose 94% and tulip 99.2%. Fig. 9 (b) shows the predictive confidence level of tulips and sunflowers 100%, daisy 96% and a rose 91.1%. The overall results show that the system can perform the prediction process well.

G. Comparison Another dataset

Layer comparison is the ratio layer 18-34 in making predictions. It is done to get the average value of the accuracy of the purposed method. In contrast, the trials were conducted using the DIBAS, BIRD, Fruit, Animal, and Food101 databases as a comparison. The comparison is shown in TABLE I. The average accuracy result is still quite good with the computation time of the training process, which does not take quite a long time, around 3 minutes. Longest time on FOOD101 because the number of images used is 4000 images. When using 1600 data in the Fruit, Animal and Flower databases, the average time needed is 3 minutes. It shows that when the amount of data is small, the computation time is reduced. It also applies to the use of the number of colour channels, and the computation time is also reduced

H. Comparison Other Architecture

The architectural comparison in the convolutional layer aims to compare the use of time-consuming with the number of different layers. In this comparison, the alexnet, google net, vgg, resnet, densenet and squeeze net architectures are used.

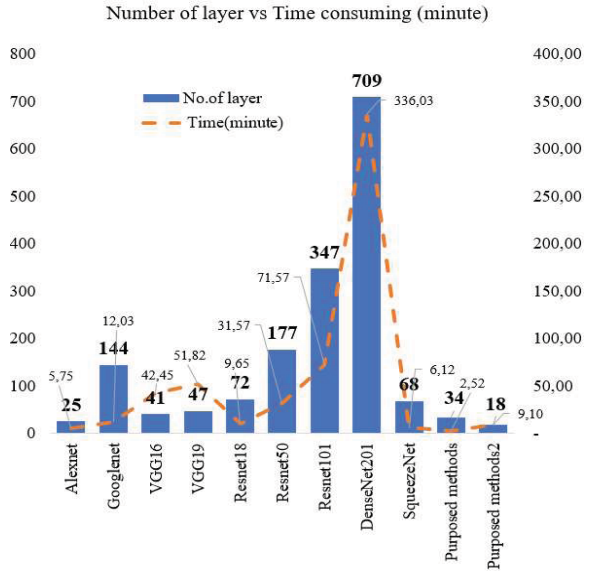


Fig. 10. Comparison using other architecture of CNN

In Fig. 10, the comparison above shows that densenet has the highest number of layers, namely 703, with a computation time of 336 minutes or about 5 hours. From the existing architecture, the one with the smallest layer is Alexnet with 25 layers, and the computation time is about 5 minutes 45 seconds. In the use of purposed layers with 18-34 layers, the average computation time is about 2 minutes 45 seconds. Other architectures with a single GPU, squeeze net takes 6 minutes, google net 12 minutes, vgg16 and vgg19 42 minutes, average resnet time is 70 minutes or around 1 hour. From the

TABLE I. COMPARISON ANOTHER DATASET

No	Name of database	No.of image	Layer			Training		Accuracy Calculation			Error Classification			Name of Class
			No.of layer	Size of image	Class	Training Accuracy	Time (minute)	Precision	Recall	F-1 Score	MSE	RMSE	MAE	
1	DIBAS Bacteria	1000	34	[224 224 3]	4	99,88%	01:49	0,9988	0,9988	0,9988	0,0013	0,0354	0,0013	Acinetobacter, Escherichia_coli,
			26	[224 224 3]	4	99,63%	01:50	0,9963	0,9963	0,9963	0,0075	0,0866	0,0050	N_gonorrhoeae, P_aeruginosa
2	BIRD	349	34	[224 224 3]	4	99,61%	00:32	0,9961	0,9962	0,9961	0,0352	0,1875	0,0117	Albatross, Bald eagle,
			26	[224 224 3]	4	92,31%	00:30	0,9531	0,9578	0,9548	0,3398	0,5830	0,1211	Peacock, Pelican
3	Fruit	1600	34	[224 224 3]	4	100,00%	05:56	1,0000	1,0000	1,0000	-	-	-	Apple Red 1, Guava,
			26	[224 224 3]	4	100,00%	03:10	1,0000	1,0000	1,0000	-	-	-	Mango, Tomato
4	Animal	1600	34	[224 224 3]	4	99,70%	03:13	0,9977	0,9977	0,9977	0,0047	0,0685	0,0031	Butterfly, Dog, Elepant,
			26	[224 224 3]	4	100,00%	03:13	1,0000	1,0000	1,0000	-	-	-	Horse
5	Flower	1600	34	[224 224 3]	4	99,30%	03:18	0,9930	0,9931	0,9930	0,0258	0,1606	0,0133	daisy, rose, sunflower,
			26	[224 224 3]	4	99,92%	03:34	0,9992	0,9992	0,9992	0,0070	0,0836	0,0023	tulip
6	Food101	4000	34	[224 224 3]	4	99,28%	10:34	0,9928	0,9928	0,9928	0,0219	0,1479	0,0119	club_sandwich, donuts,
			26	[224 224 3]	4	97,06%	09:41	0,9706	0,9720	0,9707	0,0994	0,3152	0,0519	french fries, ice cream

(a) (b) (c) (d)

Fig 11. Comparison another dataset (a) (b) Bird Prediction (c)(d) Food Prediction

results of this comparison, it appears that the use of a custom layer significantly saves the computation time of the CNN training process. The prediction shows in Fig. 11.

I. Comparison Another methods

A comparison of the methods for interest classification research is shown below.

TABLE II. COMPARISON OF METHODS

No	Researchers	Pre-Processing	Image per class	Methods	No of layer	Augmentation	Avg Acc	Training time
1	Siyuan Lu, 2017	Petal Image cropped image	157	K-nearest Neighbors	#N/A	#N/A	99,40%	#N/A
2	Gavai, 2017	labelling	258	Mobilenets	28	#N/A	91,00%	#N/A
3	Hiary, 2017	connected, bounding	#N/A	FCN-VGG	58	Available	98.5%	5-8 hours
4	Busra, 2018	#N/A	200	Googlenet InceptionV-3	48	Available	99,00%	2 minute
5	Purposed Method	resize	400	Custom Layer	18-34	Available	99,30%	3.5 minute

From TABLE II, what needs to be considered is to get optimum accuracy with minimal computation time, apart from being determined the method selection is also determined by other factors, namely the number of layers and the amount of data per class.

V. CONCLUSION

This study builds an interest classification using a Convolutional Neural Network by considering the computational time of the training process. The GAP compared to previous research is the use of a custom layer to save computation time in the training process. The results showed that the use of 18-34 layers in 1600 image dataset Flower obtained an average accuracy of the training process of 99.3% in 3.5 minutes. Classification error MSE 0.0025, RMSE 0.1606 and MAE 0.0133. When compared with the existing architecture purposed method 18-34 layers with almost the same number of layers, namely Alexnet with 25 layers, the computation time used is 50% better.

ACKNOWLEDGEMENT

The author, thanks to Engineering Faculty, Trunojoyo University that help publish this current research.

REFERENCES

[1] M. Seeland, M. Rzanny, N. Alaqraa, "Plant species classification using flower images—A comparative study of local feature representations," PLOS ONE. Vol. 12, pp e0170629, 2017.
[2] T. Alam, C.F. Ahmed, S.A. Zahin, "An Effective Recursive Technique for Multi-Class Classification and Regression for Imbalanced Data," IEEE Access. Vol. 7, pp 127615-127630, 2019.
[3] S. Lu, Z. Lu, X. Chen, "Flower classification based on single petal image and machine learning methods." in 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Vol. 826-831.2017.
[4] N.R. Gavai, Y.A. Jakhade, S.A. Tribhuvan, "MobileNets for flower classification using TensorFlow." in 2017 International Conference on Big Data, IoT and Data Science (BID), Vol. 154-158.2017.
[5] H. Hiary, H. Saadeh, M. Saadeh, "Flower classification using deep convolutional neural networks," IET Computer Vision. Vol. 12, pp 855-862, 2018.
[6] B.R. Mete, and T. Ensari, "Flower Classification with Deep CNN and Machine Learning Algorithms." in 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Vol. 1-5.2019.
[7] P.A. Traganitis, A. Pagès-Zamora, and G.B. Giannakis, "Blind Multiclass Ensemble Classification," IEEE Transactions on Signal Processing. Vol. 66, pp 4737-4752, 2018.
[8] W. Weng, C. Chen, S. Wu, "An Efficient Stacking Model of Multi-Label Classification Based on Pareto Optimum," IEEE Access. Vol. 7, pp 127427-127437, 2019.
[9] J.P. Vigueras-Guillén, B. Sari, S.F. Goes, "Fully convolutional architecture vs sliding-window CNN for corneal endothelium cell segmentation," BMC Biomedical Engineering. Vol. 1, pp 4, 2019.
[10] R. Yamashita, M. Nishio, R.K.G. Do, "Convolutional neural networks: an overview and application in radiology," Insights into Imaging. Vol. 9, pp 611-629, 2018.
[11] M.F. Aydogdu, V. Celik, and M.F. Demirci, "Comparison of Three Different CNN Architectures for Age Classification." in 2017 IEEE 11th International Conference on Semantic Computing (ICSC), Vol. 372-377.2017.
[12] F. Chou, Y. Tsai, Y. Chen, "Optimizing Parameters of Multi-Layer Convolutional Neural Network by Modeling and Optimization Method," IEEE Access. Vol. 7, pp 68316-68330, 2019.
[13] A.N. Abbasi, and M. He, "Convolutional Neural Network with PCA and Batch Normalization for Hyperspectral Image Classification." in IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Vol. 959-962.2019.
[14] A. Mikołajczyk, and M. Grochowski, "Data augmentation for improving deep learning in image classification problem." in 2018 International Interdisciplinary PhD Workshop (IIPhDW), Vol. 117-122.2018.
[15] S.C. Wong, A. Gatt, V. Stamatescu, "Understanding Data Augmentation for Classification: When to Warp? ." in 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Vol. 1-6.2016.
[16] X. Zhou, and A.D. Valle, "Range Based Confusion Matrix for Imbalanced Time Series Classification." in 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Vol. 1-6.2020.
[17] J. Qi, J. Du, S.M. Siniscalchi, "On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression," IEEE Signal Processing Letters. Vol. 27, pp 1485-1489, 2020.