

Improvement of Convolutional Neural Network Accuracy on Salak Classification Based Quality on Digital Image

Muhammad Faqih Dzulqarnain*¹, Suprpto², Faizal Makhrus³

¹Master Program in Computer Science, FMIPA UGM, Yogyakarta, Indonesia

^{2,3}Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia

e-mail: *m.faqih.d@mail.ugm.ac.id, sprpto@ugm.ac.id, faizal_makhrus@ugm.ac.id

Abstrak

Salak merupakan buah musiman dan menjadi buah dengan nilai ekspor tinggi. Keberhasilan ekspor buah salak dipengaruhi oleh proses pemilihan atau sortasi buah salak. Namun, terdapat permasalahan pada pemilihan buah salak yang siap ekspor tersebut. Pemilihan buah salak yang siap ekspor masih dilakukan secara manual dan berpotensi mengalami kesalahan sortasi. Penelitian tentang otomatisasi sortir atau pemilihan buah salak siap ekspor sudah pernah dilakukan sebelumnya. menggunakan convolutional neural network (CNN) berdasarkan citra buah salak. Nilai akurasi yang dihasilkan pada penelitian sebelumnya mampu mencapai 70,7% untuk model klasifikasi empat kelas dan 81,45% untuk model klasifikasi dua kelas. Penelitian ini dilakukan untuk meningkatkan nilai akurasi klasifikasi buah salak berdasarkan penelitian sebelumnya dengan cara mengubah proses penghilangan noise saat preprocessing citra untuk menghasilkan citra yang lebih baik. Perubahan juga dilakukan dengan menambahkan parameter pada CNN seperti Stride, Zero Padding, dan Adam optimizer. Lapisan konvolusi pada arsitektur CNN dibuat lebih dalam dengan harapan perubahan tersebut dapat meningkatkan akurasi. Hasil penelitian ini menunjukkan peningkatan akurasi sebesar 22,72% dari 70,70% menjadi 93,42% untuk model CNN dengan output empat kelas dan kenaikan akurasi sebesar 13,29% dari 81,45% menjadi 94,74% untuk model CNN dengan output dua kelas.

Kata kunci— Pemilihan buah salak, Convolutional Neural Network, citra digital, peningkatan akurasi, parameter

Abstract

Salak is a seasonal fruit that has high export value. The success of salak fruit exported is influence by selection process, but there is still a problem in it. The selection of salak still done manually and potentially misclassified. Research to automate the selection of salak fruit has been done before. The process of selection this salak fruits used convolutional neural network (CNN) based on image of salak fruits. The resulting of accuracy value from previous research is 70.7% for four class classification model and 81.45% for two class classification model. This research was conducted to increase accuracy value the classification of salak exported based on previous research. Accuracy improvement by changing the noise removal process to produce a better image. The changing also occur in the CNN architecture that layer convolution is more deep and with additional parameters such as Stride, Zero Padding, and Adam Optimizer. This change hopefully can increase the accuracy value of the salak classification. The results showed an accuracy value increased 22.72% from 70.70% to 93.42% for the category of four classes CNN models and increased 13.29% from 81.45% to 94.74% for category two classes.

Keywords— sorting salak fruit, Convolutional Neural Network, digital image, increased accuracy, parameter

1. INTRODUCTION

Salak is a productive seasonal fruit that can produce throughout the year. Apart from being one of the biggest fruits production and becoming a high export commodity [1]. The export success is influenced by several factors including the selection process of salak. However, from the selecting salak fruit, there are obstacles that have potential make mistake for export because they are still done manually using human power. The problem of choosing salak fruit can be prevented by automating the selection for export based on the image of salak.

Automation of selection process salak fruit based on digital image has been done before by paying attention to the features extraction from the fruit. This feature extraction process can be done with Convolutional Neural Network (CNN). Previous research was able to extract the classification features of salak fruit with accuracy value is 70.7% for four class classification model and 81.45% for two class classification model [1]. Previous researchers stated that the accuracy value can increased.

Increasing the accuracy of CNN classification can be done by preprocessing to produces a better image and adds some of parameters to the CNN architecture. Some parameters of Convolutional Neural Network (CNN) that can be applied are Stride and Padding Layer parameters (Zero Padding). The usefulness of these two CNN parameters is for determine pixel shift to get more detailed information from the input image and increase the accuracy value of a convolution model because the convoluted filter will be focus on finding the information and eliminating unnecessary information. In addition to adding some of parameters, architectural of CNN are giving deeper convolution layers that can be useful to improve classification accuracy because more image features are extracted by CNN and provide more information for the classification of salak.

2. METHODS

2.1 Salak

Salak is a tropical fruit that is not only in Indonesian territory. This fruit spreads through traders to the Philippines, Malaysia, Brunei and Thailand. In Indonesia, the development of salak has become more widespread and has produced several general commodities. Salak, which has the Latin name *Zalacca edulis* Reinw., is divided into several types of group name: Javanese zalacca (*Salacca zalacca* (Gaertner) Voss) with seeds of 2-3 grains and white flesh of yellowish bone, Balinese salak (*Salacca amboinensis* (Becc) Mogeia) with seeds of 1-2 grains and white flesh of yellowish bones, and salak of Padang Sidempuan (*Salacca sumatrana* (Becc)) which has a rather reddish flesh [2].

2.2 Image

Image is a spatial representation of an actual object in two-dimensional field that is usually written in (x,y) cartesian coordinates, and each coordinate represents the smallest signal of the object [3].

Digital image is two-dimensional function $f(x,y)$, which is a function of light intensity where the values of x and y are spatial coordinates and function values at each point, and (x, y) is the level of grayscale image at that point. Digital images are expressed by a matrix where rows and columns express a point in the image and the matrix element (called an image element or pixel) states the gray level at that point. The matrix of digital images measuring $N \times M$ (row x column), with:

N = number of rows $0 \leq y \leq N - 1$
 M = number of columns $0 \leq x \leq M - 1$
 L = max value of gray level image $0 \leq f(x,y) \leq L - 1$

Degree of gray level can be expressed in matrix form as follows:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N - 1) \\ f(1,0) & f(1,1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M - 1,0) & f(M - 1,1) & \cdots & f(M - 1, N - 1) \end{bmatrix} \quad (1)$$

2.3 Image Preprocessing

2.3.1 Noise Removing with Gaussian Blur

Gaussian blur is a method that uses a Gaussian function and to noise removing an image. Gaussian blur obtained from a convolution operation. The convolution operation starts from calculating of weight value for matrix of Gaussian kernel. Calculation of matrix weight Gaussian kernel is obtained from the Gaussian distribution function, as in the following equation [1]:

$$g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{(x)^2+(y)^2}{2\sigma^2}} \quad (2)$$

Where:

σ = standard deviation of Gaussian distribution that is a constant value.
 $g(x,y)$ = the element of the weight distribution matrix based on the position (x, y) with x is distance from starting point in horizontal axis and y is distance from starting point in vertical axis.

After getting Gaussian matrix value $G(x,y)$, convolution is done to get a new pixel value that makes blur image so that noise in the image can be reduced or eliminate. The calculation of Gaussian matrix convolution with the original image is shown in the following equation:

$$B(k, l) = \sum_{x=-1}^1 \sum_{y=-1}^1 G(x, y) f(m + x, n + y) \quad (3)$$

Where:

$B(k,l)$ = pixel value of blur image in position (k,l)
 $f(m,n)$ = pixel value of original image in position (m,n)
 $G(x,y)$ = Gaussian matrix value in position (x,y)

2.4 Segmentasi Otsu Thresholding

Otsu method is intended to find the optimal threshold value from a global threshold. This method works by finding the maximum limit value of between-class variance. Basic idea of a class with an optimal boundary value is that both classes must have the highest pixel intensity value. In addition that has to get an optimal threshold, Otsu method also has important properties, namely the calculation to get the boundary value based on histograms of easily calculated images [4].

Getting the optimal *threshold* k which maximizes function η or equivalent maximizing value of σ_B^2 can use the following equation [6]:

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (4)$$

σ_B^2 is a between-class variance with μ_T average of level from image histogram from $\omega(k)$ and $\mu(k)$ which is zeroth-order and first-order cumulative in k level of global threshold. Otsu

thresholding intended to find the optimal threshold value of an image by finding the maximum value using the following equation [6]:

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \quad (5)$$

2.5 Morphology

Morphological is a generally known as branches in biological sciences that discuss of shape or structure of animals and plants. In the context of digital imagery, morphology is a way to extract image components that are useful in form representations and descriptions such as boundaries, skeletons and convex hulls [5]. Morphology in digital images processing is divided into several types. Basic of morphology in digital images is dilation, erosion, opening and closing [5].

Dilation aims to increase pixels of an image on boundary between object and background and cause object become larger than its origin. Dilation has an opponent whose function is to reduce the pixel image on boundary between object and background called erosion. Erosion is the opposite of the dilation that erodes the image object.

Opening is combination of erosion process followed by dilation on a digital image. Opening operations on images have the effect of smoothing the boundaries of objects, separating previously held objects, and removing objects that are smaller than the size of the structure or noise. Just like dilation and erosion, opening has an opponent that is closing. Closing is the opposite of opening operation where the image is first dilated and then followed by erosion. Closing operation will refine object in the image, but by connecting fragments (fuses narrow breaks and thin gulf) and removing small holes in the object.

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the algorithms from deep learning which is the development of Multilayer Perceptron (MLP) which is designed to process data in two-dimensional form, such as images or sounds. CNN is used to classify labeled data using supervised learning methods. CNN is often used to recognize objects or sights, and to detect and segment objects [2] which have the architecture shown in Figure 1.

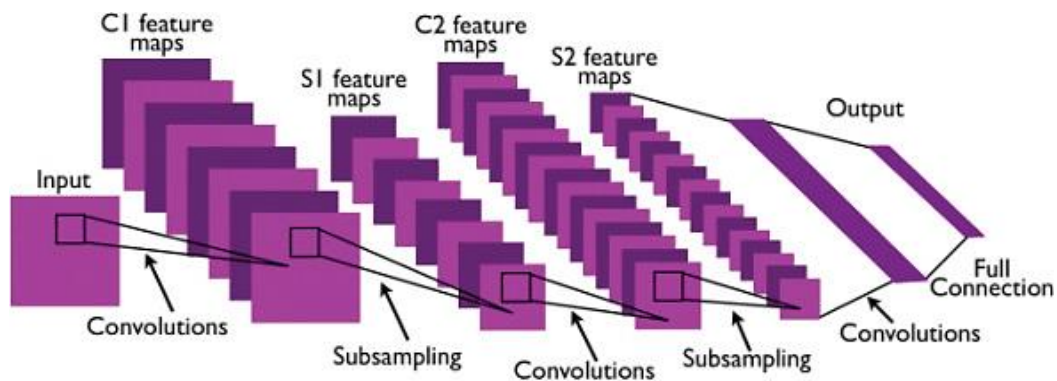


Figure 1 Architecture Convolutional Neural Network

Figure 1 shows the architecture of CNN that consists of several stages of operation. The operating stages are convolution operations, pooling operations and activation functions.

2. 6.1 Convolution Operation

Basic operations in CNN is convolution operation or $h(x)$. Convolution has two functions $f(x)$ that are functions of the original object and $g(x)$ as a convolution kernel function that is defined as an equation (6) [3].

$$h(x) = f(x) * g(x) = \sum_{i=-\infty}^{i=+\infty} f(i)g(x - i) \quad (6)$$

Convolution operations are also imposed on function $s(t)$ of multi-dimensional arrays that can be formulated as equation (7).

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a) * w(t - a) \quad (7)$$

Where:

- $s(t)$ = result function of convolution operation
- x = multi-dimensional of array data
- w = weight or kernel
- t = variable from function
- a = dummy variable

In machine learning applications, weights (w) are multi-dimensional arrays which are parameters that can be learned.

2. 6.2 Activation Function

The activation function is calculated after convolution operation. Activation functions that are often used in convolutional neural networks include $\tanh()$, ReLu (Rectified Linear Unit), sigmoid, and softmax [6]. This research will be use ReLu and SoftMax activation functions.

1. Relu

The ReLu function is a function that output value of a neuron can be expressed as 0 if input value is negative. If input value is positive, the output of neuron is the activation input value itself. This function equation can be shown in equation (8).

$$f(x) = \max(0, x) \quad (8)$$

2. Softmax

Softmax activation is applied in the last layer on neural network. Softmax is more commonly used than ReLU, sigmoid or $\tanh()$. Softmax is useful for changing output in neural network into a basic distribution probability. The softmax equation is shown as follows [7]:

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \quad \text{for } i = 1, 2, 3, \dots, k \quad (9)$$

2. 6.3 Pooling Operation

After calculating activation function, pooling operations are carried out by reducing size of matrix by means of max-pooling or average-pooling. Output from pooling operation is a matrix with smaller dimensions compared to the initial image. Convolution and pooling process is carried out to obtain the desired feature map to be input into fully connected layer [3]. The pooling illustration is shown in Figure 2, namely pooling by max-pooling.

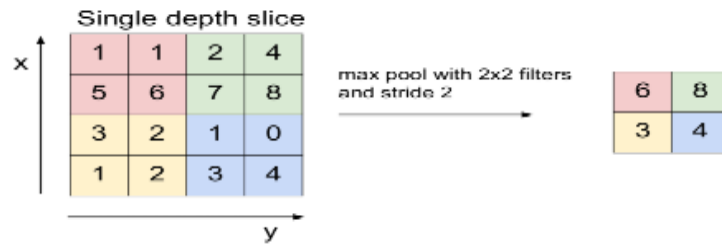


Figure 2 Pooling with max-pooling

2.7 Stride

Stride is a parameter that determines number of filter shifts in an image pixel. If the stride value is 1, the convolution filter will shift by 1 pixel horizontally and vertically. More smaller stride value, model will capture more detailed information from an input image, but it requires more computation when compared to a large stride [8].

A small stride value does not always produce better pixel information details, but with a small stride value prevents stacking of unused pixel information.

2.8 Padding

Padding or Zero Padding is a parameter that determines number of pixels (containing a value of 0) to be added to each side of the input. This is used in order to manipulate the output dimensions of the convolution layer (Feature Map) [9].

Purpose using padding layer is output dimensions of the convolution layer will always be smaller than the input (except the use of a 1x1 filter with stride 1) so that more information is wasted which is not needed when the convolution process is running. In addition, zero padding will set output layer's dimensions to remain the same as the input dimension or at least not drastically reduced.

If in a dimension actually input is 5x5, then convolution is done with a 3x3 and stride filter of 2, then a 2x2 feature map will be obtained. But if you add zero padding with a value of 1x1, then the resulting map feature is 3x3 (more information is generated). Calculating the dimensions of a feature map can be used the following equation [9]:

$$\text{output} = \frac{V - F + 2P}{S} + 1 \quad (10)$$

Where:

V = Volume Size

F = Filter height

P = Zero Padding

S = Stride

2.9 Adam Optimizer

Adam's optimization was introduced by Diederik Kingma from OpenAI and Jimmy Ba from the University of Toronto in 2015 ICLR paper entitled "Adam: A Method for Stochastic Optimization". Adam stands for Adaptive Moment Estimation [10]. Adam optimizer is an optimization algorithm that is used as a substitute for classic gradient stochastic procedures that will update network weights based on iteratives without changing the learnign rate. The algorithm from Adam Optimizer is shown in Figure 3.

```

Require:  $\alpha$ : Stepsize
Require:  $\beta_1, \beta_2 \in [0, 1)$ : Exponential decay rates for the moment estimates
Require:  $f(\theta)$ : Stochastic objective function with parameters  $\theta$ 
Require:  $\theta_0$ : Initial parameter vector
 $m_0 \leftarrow 0$  (Initialize 1st moment vector)
 $v_0 \leftarrow 0$  (Initialize 2nd moment vector)
 $t \leftarrow 0$  (Initialize timestep)
while  $\theta_t$  not converged do
   $t \leftarrow t + 1$ 
   $g_t \leftarrow \nabla_g f_t(\theta_{t-1})$  (Get gradients w.r.t. stochastic objective at timestep  $t$ )
   $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$  (Update biased first moment estimate)
   $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$  (Update biased second raw moment estimate)
   $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$  (Compute bias-corrected first moment estimate)
   $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$  (Compute bias-corrected second raw moment estimate)
   $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$  (Update parameters)
end while
return  $\theta_t$  (Resulting parameters)

```

Figure 3 Adam Optimizer algorithm

3. RESULTS AND DISCUSSION

3.1 Research Condition

Research was conducted by giving a deeper convolution layer than previous research and making changes to preprocessing method and adding Stride and Zero Padding parameters. Preprocessing that is used is Gaussian blur with 5x5 kernel and performs morphology with opening operations after the segmentation is executed. Stride and Zero Padding are included in the CNN architecture which has a number of 5 convolution layers and 2 hidden layers. The CNN architecture in this research can be shown in Figure 4.

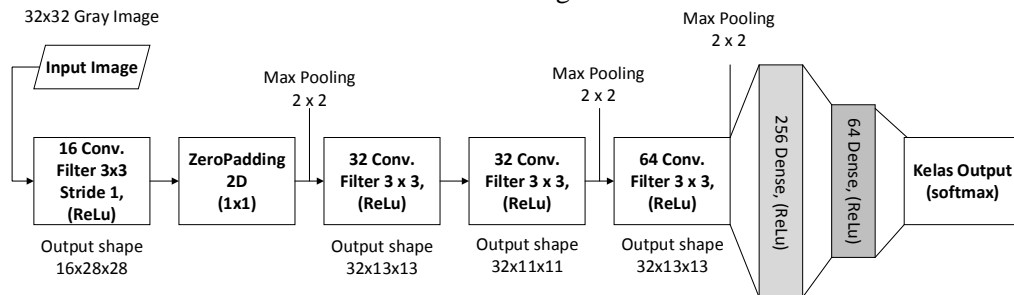


Figure 4 CNN architecture

3.2 Image Resource

The image data used amounts to 756 images and this study provides additional conditions namely changing the orientation of the image vertically and horizontally. Image used from the previous research which is divided into 4 classes based on the SNI document [2] and grouped in 2 classes as previous research [2]. Details of image data can be shown in Table 1 and Table 2.

Table 1 Detail of 4 class image data

Class	Total
Sangat bagus or grade AA	384 images
Bagus or grade AB	130 images
Kurang bagus or grade C	102 images
Tidak bagus or grade TB	140 images

Table 2 Detail of 2 class image data

Class	Total
Layak ekspor	384 images
Tidak layak ekspor	372 images

3.3 Preprocessing Analysis

Research on increasing CNN accuracy on salak based on digital imagery used preprocessing to extract the characteristics of fruit skin color. Feature extraction of skin color is represented in binary image. Binary image used to output representation of the segmentation process. The segmentation results will ensure that the color traits captured can be well represented for each category of salak images. Preprocessing results will be a dataset for the CNN model. The results of image preprocessing are shown in Figure 5 (a) and (b).

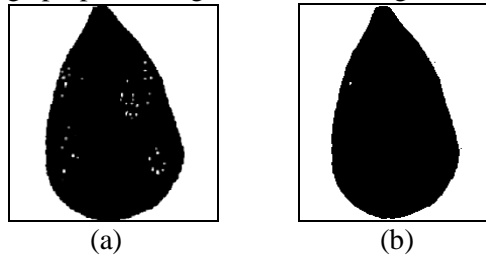
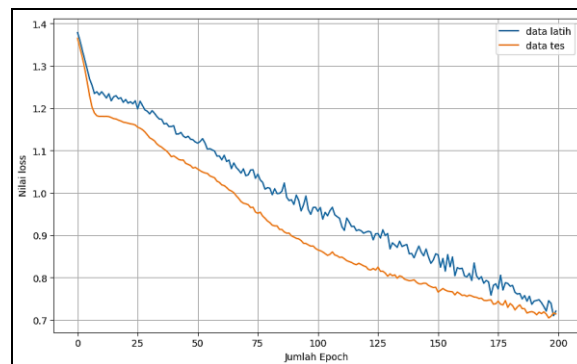


Figure5 Image preprocessing result

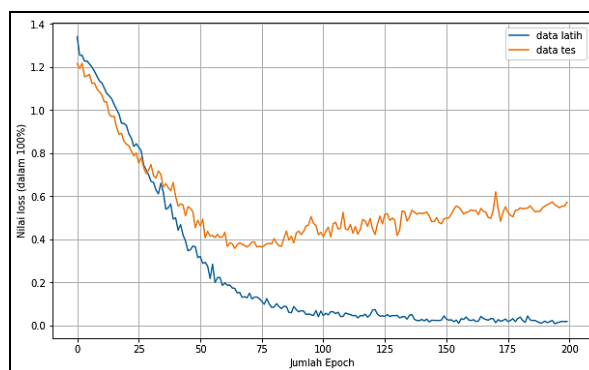
Figures 4 (a) and (b) show each preprocessing image of the previous research and this research. Preprocessing previous research uses Averaging but still has noise detected while this research uses Gaussian blur and opening operation morphology so that the resulting image has no noise.

3.4 Learning Rate Implementation

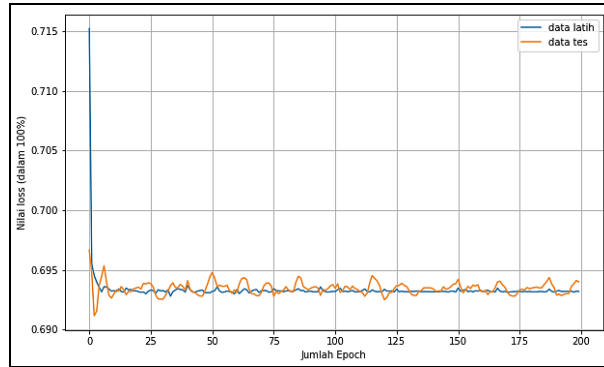
Determination of learning rate value is done to produce a model with a stable training loss value and minimum by using Adam optimizer. The learning rate values given are 0.0001, 0.001 and 0.01. The results of learning rate implementation can be seen from Figures 6 (a), (b), and (c).



(a)



(b)



(c)

Figure 6 Result of learning rate implementation

Based on Figure 5, it can be seen that the learning rate is able to achieve minimum value and most stable training loss is the value of 0.001, namely Figure 5 (b). Other learning rate values are not able to produce small or stable training loss values. Minimum value and stable training loss values will affect the accuracy of the model during training.

3.5 Classification

The classification process is done after the image becomes a dataset that is ready to be trained. The dataset is divided using cross validation with a ratio of 80%: 20% or 0.2. The training process takes place with 200 epochs. The CNN classification architecture as described in section 3.1 has 5 convolution layers, 2 hidden layers plus Stride and Zero Padding parameters. The results of classification accuracy can be shown in Table 3.

Table 3 Accuracy value from classification

Ouput category	Accuracy
4 classes of classification model	93,42 %
2 classes of classification model	94,74 %
Vertical flip dataset	93,86%
Horizontal flip dataset	94,76%

Table 3 shows the accuracy value of CNN classification with output of four classes, two classes and datasets given orientation vertically and horizontally. The resulting of accuracy value has increased from previous research. Comparison of increasing accuracy values from previous studies is shown in Table 4.

Table 4 Comparison of accuracy value

Model	Output category	Accuracy
CNN current model (Rismiyati, 2016)	4 classes	70,70 %
	2 classes	81,45 %
CNN new model	4 classes	93,42 %
	2 classes	94,74 %

4. CONCLUSIONS

Changes in preprocessing method from Averaging becomes Gaussian blur and Morphological Opening, is able to produce a better image and has no noise.

Adam optimizer with learning rate value 0.001 is able to provide training score value which is very minimal so the process of training classification model will produce a high accuracy value.

Images that orientation is changed vertically and horizontally can still be classified properly using CNN. The accuracy results in the test are 93.86% for vertically orientated image data and 94.76% for horizontally oriented image data.

Convolutional Neural Network model from this research was able to increase the classification accuracy value by 22.72% from 70.70% to 93.42% for category four-class and increased 13.29% from 81.45% to 94.74% for category two-class. This accuracy increase due to the depth of convolution layer and use of Stride and Zero Padding parameters.

REFERENCES

- [1] Rismiyati and SN. Azhari, "Convolutional Neural Network implementation for image-based Salak sortation," ICST (International Conference on Science and Technology-Computer, 27-28 Oct. 2016 [Online]. Available:<https://ieeexplore.ieee.org/document/7877351>. [Accessed: 28-Aug-2018]
- [2] P. Rianto and A. Harjoko, "Penentuan Kematangan Buah Salak Pondoh di Pohon Berbasis Pengolahan Citra Digital" IJCCS (Indonesian J. Comput. Cybern. Syst., vol.11, no. 2, 2017 [Online]. Available:<https://jurnal.ugm.ac.id/ijccs/article/view/17416>. [Accessed: 10-Dec-2018]
- [3] R. C. Gonzalez, and R. E. Woods, *Digital Image Processing*, Ed.3, New Jersey: Prentice Hall, 2008.
- [4] N. Otsu, "A Threshold Selection Method from Gray-level Histogram," IEEE Transactions on System, Man, and Cybernetics, vol.9, no.1, 1979 [Online]. Available: <https://ieeexplore.ieee.org/document/4310076>. [Accessed: 28-Aug-2018]
- [5] C.S. Nandi, B. Tudu. and C. Koley, "A Machine Vision-Based Maturity Prediction System for Sorting of Harvested Mangoes," IEEE Transactions on Instrumentation and Measurement., vol.63, no.7, 2014 [Online]. Available:<https://ieeexplore.ieee.org/document/6730653>. [Accessed: 28-Aug-2018]
- [6] S.R. Rupanagudi, G.B. Varsa, dan B.S. Ranjani, "A cost effective tomato maturity grading system using image processing for farmers," IC3I (International Conference on Contemporary Computing and Informatics, 27-29 Nov. 2014 [Online]. Available:<https://ieeexplore.ieee.org/document/7019591>. [Accessed: 28-Aug-2018]
- [7] B. Xin, T. Wang, and T. Tang, "A deep learning and softmax regression fault diagnosis method for multi-level converter," SDEMPED (International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, 29 Aug.-1 Sept. 2017 [Online]. Available:<https://ieeexplore.ieee.org/document/8062370>. [Accessed: 5-Sep-2018]
- [8] Y. Luo, H. Cheng, and L. Yang, "Size-Invariant Fully Convolutional Neural Network for vessel segmentation of digital retinal images," APSIPA (Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 13-16 Dec. 2016 [Online]. Available:<https://ieeexplore.ieee.org/document/7820677>. [Accessed: 28-Aug-2018]
- [9] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in Convolution for Network in Network," IEEE Transactions on Neural Networks and Learning Systems, vol.29, no. 5, 2018 [Online]. Available:<https://ieeexplore.ieee.org/document/7879808>. [Accessed: 28-Aug-2018]
- [10] D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," Cornell University Library, Jan. 2017 [Online]. Available:<https://arxiv.org/abs/1412.6980> [Accessed: 28-Aug-2018]