I.J. Intelligent Systems and Applications, **2021, 2, 52-61**
Published Online April 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2021.02.04

Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection

Jasman Pardede
Department of Informatics Engineering, Institut Teknologi Nasional Bandung, Bandung, Indonesia
E-mail: jasman@itenas.ac.id

Benhard Sitohang, Saiful Akbar, and Masayu Leylia Khodra
School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Bandung, Indonesia
E-mail: {benhard, saiful, masayu}@informatika.org

Received: 12 June 2020; Accepted: 06 December 2020; Published: 08 April 2021

Abstract: In previous studies, researchers have determined the classification of fruit ripeness using the feature descriptor using color features (RGB, GSL, HSV, and L * a * b *). However, the performance from the experimental results obtained still yields results that are less than the maximum, viz the maximal accuracy is only 76%. Today, transfer learning techniques have been applied successfully in many real-world applications. For this reason, researchers propose transfer learning techniques using the VGG16 model. The proposed architecture uses VGG16 without the top layer. The top layer of the VGG16 replaced by adding a Multilayer Perceptron (MLP) block. The MLP block contains Flatten layer, a Dense layer, and Regularizes. The output of the MLP block uses the softmax activation function. There are three Regularizes that considered in the MLP block namely Dropout, Batch Normalization, and Regularizes kernels. The Regularizes selected are intended to reduce overfitting. The proposed architecture conducted on a fruit ripeness dataset that was created by researchers. Based on the experimental results found that the performance of the proposed architecture has better performance. Determination of the type of Regularizes is very influential on system performance. The best performance obtained on the MLP block that has Dropout 0.5 with increased accuracy reaching 18.42%. The Batch Normalization and the Regularizes kernels performance increased the accuracy amount of 10.52% and 2.63%, respectively. This study shows that the performance of deep learning using transfer learning always gets better performance than using machine learning with traditional feature extraction to determines fruit ripeness detection. This study gives also declaring that Dropout is the best technique to reduce overfitting in transfer learning.

Index Terms: Fruit ripeness, transfer learning, MLP, overfitting, accuracy.

1. Introduction

Indonesia, as an agricultural country, has several agricultural products included mango, apple, orange, banana, durian, tomato, and others. Good quality fruit will have good selling points [1]. It is necessary to maintain the quality of fruit consistently to increase the selling value of the fruit. The fruit quality determined by the level of fruit maturity [2], so we need a system that can be determining fruit quality well. Determination of fruit quality by using the human senses often does not produce consistent quality and requires a long time. Besides that, to delegate one's expertise to others in determining the maturity of the fruit requires a long learning process. Therefore, we need tools that can help determine effectively and efficiently fruit maturity.

Today, fruit recognition systems have become a challenging topic in the field of computer vision [3,4]. Various image analysis techniques have been developed to help and facilitate work, including automatic fruit harvest detection systems [5], automatic fruit recognition [6,7], fruit classification system [8], ripeness fruit system [9], fruit diseases detection system [10], and others. Generally, fruit recognition systems use the feature descriptor, namely the shape features [11], texture features [12], and color features [2,11-13]. The feature descriptor algorithm used to extract mainly information from the image stored in a feature vector [3,4,9]. But the performance of the feature descriptor, in the fruit recognition system, results in suboptimal performance [13]. To improve image classification performance requires an effective/robust feature descriptor.

One of the best image feature descriptors in the field of machine learning and computer vision that has had great success to date is convolutional neural networks. Several researchers have utilized CNN as feature extraction in various applications. Feature extraction with CNN has proven to be very robust. CNN method has a modification in the form of Deep Convolutional Neural Network (Deep CNN) that is the beginning of deep learning. Digital image processing methods have been applied in the deep learning classification method also. Deep Learning has become one of the hot topics in the world of Machine Learning because of its significant capabilities in modeling complex data such as imagery and sound. The Deep Learning method that currently has the most significant performance results in image recognition is Deep CNN [14-16].

However, Deep CNN, like other Deep Learning methods, has a weakness that is the training process that takes a long time and needs the availability of large-scale databases. Several studies have proposed transfer learning to overcome these problems [17-19]. Transfer learning techniques have been applied successfully in many real-world applications, including solving NLP problems [20], image classification [19], CAD [21], name-entity recognition problems [22], and others. Like machine learning, overfitting problems are also problems that often arise. In the process of forming the Deep CNN model with transfer learning, the overfitting problems must be avoided [23-26]. To reduce overfitting in machine learning and deep learning can be done using Data Augmentation, Dropout [27], regularizes [16,26], early stopping [16,23], and Batch Normalization [28].

In previous studies, researchers have determined the classification of fruit ripeness using the feature descriptor with various colors features using the SVM technique. But from the experimental results obtained, the performance still results in less than maximum performance, namely the maximal accuracy obtained is only 76% [13]. Based on the advantages that have been achieved by the Deep CNN method on image classification and reducing overfitting techniques, then in this study, proposed transfer learning techniques to form a model that can determine the maturity of the fruit automatically. Deep CNN with transfer learning used in this study is to use VGG16 without the top layer. We use VGG16 because VGG16 has a smaller network architecture and easy to implement. The top layer of VGG16 was replaced by adding a Multi-layers Perceptron (MLP) block. In this study, the study used Data Augmentation, Dropout, regularizes, early stopping, and Batch Normalization to reduce overfitting. The focus of attention in this study is the effect of overfitting reduction techniques on transfer learning using VGG16 with the upper layer replaced with MLP blocks in determining fruit ripeness. Also, we have analyzed the outcome selecting of reducing overfitting techniques with the extracted features from VGG16 architectures. This study shows that the performance of fruit ripeness detection using transfer learning is better than machine learning with traditional feature extraction and achieved when obtaining the best overfitting reduction technique.

This paper is organized as follows. In section 2 gives related work about deep learning, transfer learning, and overfitting. In section 3 discusses the fruit ripeness dataset that was conducted in the experiment. In section 4 describes the proposed architecture transfer learning. Testing results based on numerical experimental introduce in section 5. Discussion about experimental results explained in section 6 and finally, we conclude our work in section 7.

2. Related Work

2.1. Deep Learning

Deep learning is a new scientific field in the field of machine learning and has excellent skills in computer vision [14]. Deep Learning has various sets of methods for training multi-layer artificial neural networks [15]. Deep learning systems and models are layered architectures that study different features at different layers [16]. The series of methods become very more effectively and better easier to identify patterns from the data entered. The Deep learning system is representational learning, where each layer gets input from the previous layers and distributed to the next layers [14,16,29]. The starting layer is associated with more general and coarse features. The deeper layers of the network learn details that are better and more specific than the dataset that ultimately provides output with certain belief factors.

Deep Learning impacts the development progress that AI has achieved in stages. Not only for software, but the implementation of deep learning has also penetrated various industrial fields. Deep Learning has become one of the hot topics in the world of Machine Learning because of its significant capabilities in modeling complex data such as imagery and sound [14,15,30]. Deep learning approaches categorized into four categories [15], namely: Deep Supervised learning, Deep semi-supervised learning, deep unsupervised learning, and deep reinforcement learning. Deep supervised learning is a learning technique that uses data labeled. There are several different deep supervised learning approaches, namely: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) [31].

The Deep Learning method that currently has the most significant results in image recognition is the Convolutional Neural Network (CNN) [21,29]. CNN has a large number of parameters and hyper-parameters such as weights, biases, number of processing units (number of neurons), number of layers, filter size, stride, learning rate, activation function, etc. [16]. Various CNN architectures discussed in several surveys. CNN architecture grouped into 7 (seven) categories [29], namely: spatial exploitation, depth, multi-path, width, feature map exploitation, channel boosting, and attention-based CNNs. The architecture of CNN consists of two main parts [14,32], namely: Feature extractors and classifiers. The CNN extractor feature usually consists of a stack of several convolutional layers and a max-pooling layer. The classifier of CNN is a fully connected and softmax layer on the last screen. Some architectural models of CNN's spatial exploration include LeNet, Alexnet, ZefNet, VGG, and GoogleNet. Some architectures from CNN are specifically designed for large-scale data analysis, (such as GoogleNet, ResNet), while VGG is considered an architecture general [29]. Even so, to get the best performance, deep learning generally requires an amount of data and training time more than traditional machine learning systems.

2.2. Transfer Learning

Traditional machine learning techniques try to learn each task from the beginning. But the fact is, in cases certainly, people can intelligently apply knowledge learned beforehand to solve new problems faster or with better solutions [18-20]. For example, we might find that learning to recognize a motorcycle quickly if we already know to recognize a bicycle well. Likewise, learning to play the keyboard will be able to help facilitate piano learning. Based on these facts, then to solve new problems faster and better can use the transfer of knowledge previously owned. This knowledge transfer technique is called transfer learning or self-taught learning [33]. Transfer learning is a technique or method that utilizes a model that has trained on a particular dataset. The model used again to solve other similar problems by using it as a starting point, modifying and updating its parameters so that it matches the newly given dataset [17,18]. Transfer learning aims to find a model that is suitable for the training dataset by using learning results from other models that have been conducted before [34-35].

Three main issues have to consider in conducting transfer learning [36], namely: (1) What to transfer; (2) How to transfer, (3) When to transfer. "What to transfer" asks which piece of knowledge can transfer between domains or tasks (across domains). "How to transfer" issue after discovering which knowledge can transfer. The learning algorithm needs to develop knowledge transfer because some knowledge is specific to certain individual domains or tasks. While "when to transfer" asks in which situations the process of knowledge transfer must be carried out. Based on different situations between source and target domains and tasks, transfer learning can be categorized into 3 (three) categories [18], namely: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. Various transfer learning approaches can be read in [17,18], [34]. Transfer learning techniques have been implemented in a variety of real-world applications, included cross-class classification problems [37], name-entity recognition problems [22,38], NLP problems [20], image classification problems [19,39], and so on. Transfer learning methods with deep vision models based on computer vision used to overcome various complex problems widely [36].

2.3. Overfitting

Overfitting problems are problems often faced in machine learning and deep learning. Overfitting occurs when finding a good model (good fit) of training data, but the model properly cannot be generalized to new data (data not previously known) [23-26]. In other words, the model can only study patterns of trained data before, but the model is not relevant to other data. Overfitting identified by paying attention to the validation matrix, such as loss or accuracy. Usually, overfitting can occur when the validation metric stops increasing after a certain number of epochs and then will start to decrease. While training metrics will continue to improve because the model looks for the best fit for training data.

There are several ways you can do to reduce overfitting in the deep learning model. The best option is to prepare more training data [23]. Unfortunately, in real-world situations, we are often unable to set up more training data because of the need for time, cost, and technical constraints. Another way to reduce overfitting is to reduce the capacity of the model to remember training data [24]. Like, the model will need to focus on relevant patterns in training data, which results in a more general model. Several options can do to deal with overfitting, namely:

i. Remove the hidden layer. Reducing network capacity can be done by removing layers or reducing the number of elements in hidden layers. Removing one hidden layer so reducing the number of layer elements [24].
ii. Regularizer kernel. Apply regularization, by adding a loss function for large weights. To solve overfitting can be done by applying weight regularization to the model. This will add a network loss function for large weights. As a result, we have the simplest model which can force us to only study relevant patterns in training data. There are two regularization, namely L1 regularization and L2 regularization. L1 regularization will add a cost with the absolute value of the parameter. L2 regularization will add a cost with the quadrant value of the parameter.
iii. Dropout. Using Dropout, randomly delete certain features by setting these features to zero. [27] states that the most widespread regularization technique used to reduce overfitting in deep learning is Dropout.
iv. Early Stopping, another way to prevent happened overfitting is to stop the training process early. Instead of doing a training on several specific epochs, we can stop it from immediately generating a validation loss, because after that the model will get worse with increasing training [16].
v. Batch Normalization. Batch Normalization used to solve problems related to the internal covariance shift of feature maps. In [28] stated that Batch Normalization does as regularization, which in some cases eliminates the need for Dropout.

3. Fruit Ripeness Dataset

In this study, the study created a fruit maturity dataset from the results of taking photographs with the camera. First of all, researchers collected ripe apples, ripe apples, ripe mango, ripe mango, ripe orange, ripe orange, ripe tomatoes, and not ripe tomatoes. Fruit image capturing is done with a camera at a distance of 1 (one) meter from the fruit object and with the help of lighting lamps. During the process of capturing the fruit image, we set one white cardboard as the background. Each one fruit is taken 4 (four) fruit images by rotating 90 degrees from the side. Each captured images stored in the jpg extension, with image sizes between 2621x2166 pixels and 2043x1772 pixels. The total number of fruit images created is 1120 images. Each fruit images contain 140 categories. Some image categories of fruit are as stated in Fig. 1.

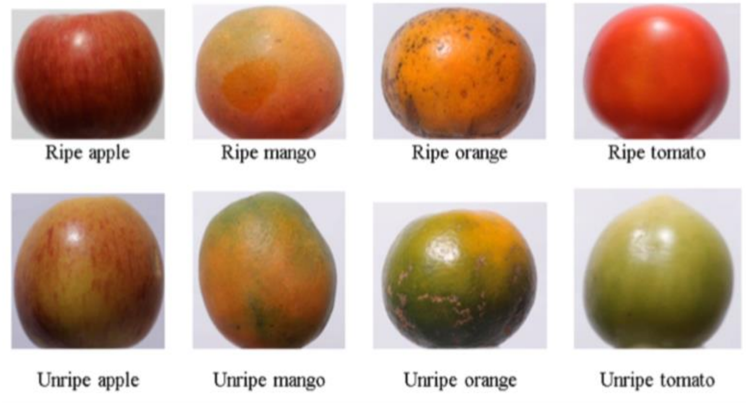


Fig.1. Represented of each the categories of fruit ripeness dataset.

4. Proposed Transfer Learning Architecture

As stated in the deep learning sub-chapter, the VGG model is considered the most common deep learning architecture. So, this research proposes deep CNN with transfer learning using VGG16 without the upper layer. The top layer of VGG16 was replaced by adding a Multi-layers Perceptron (MLP) block. This study uses a fruit maturity dataset labeled. Therefore, the category of learning transfer used in this study is inductive transfer learning using the ImageNet weighted VGG16 model. The knowledge transferred from the VGG16 model is the weight held by the VGG16 model from generating the model to the classification of 1000 on 14 million images. The weight of the VGG16 model used to perform feature extraction of the fruit maturity dataset.

The feature extraction results from the VGG16 model fed to the MLP block. Thus, the feature extractor used in transfer leaning uses the feature extractor from VGG16, while the classifier of the proposed architecture uses the MLP block with the softmax activation classifier. MLP blocks contain Flatten layer, a Dense layer, and Regularizes. Regularizes used to get the best classification performance of the system (reducing overfitting) is Dropout, Batch Normalization, and Regularizes Kernel. Because the number of datasets in each image category only has 140 images, to reduce overfitting, besides using Regularizes, the Data Augmentation process done using ImageDataGenerator from Keras libraries. The activation function used in the MLP block is ReLu activation. The MLP block output layer uses the softmax activation function with 8 (eight) classes. The system architecture used for the classification of fruit maturity is as shown in Fig. 2. The stages of the model formation are as follows:

(i) Generate training image data using the Data Augmentation process. While the validation and testing of image data do not use the Data Augmentation process.
(ii) Feature extraction of each image (training image, validation image, and testing image) using feature extraction of the VGG16 model with ImageNet weight.
(iii) Fed result of extraction features from the VGG16 model to Regularizer (Dropout, Batch Normalization, Regularizer Kernel).
(iv) Fed result of Regularizer to flatten layer.
(v) Fed result of Flatten layer to the Dense layer and using ReLu activation.
(vi) Fed result of Dense layer to the Dense layer using the softmax activation function with 8 (eight) classes.

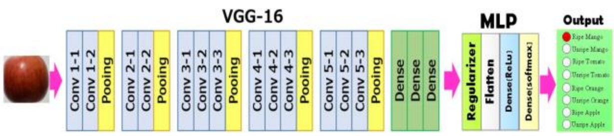


Fig.2. Proposed transfer learning architecture using VGG16.

This study considers the influence of regularizers in reducing overfitting. In stage (iii) the formation of the model of transfer learning is carried out to replace the type of regularizes used. In this research, three types of regularizers that implemented in architecture that proposed. The first architecture is transfer learning architecture with the VGG16 model using Dropout. The second architecture is transfer learning architecture with the VGG16 model using Batch Normalization. The last of type is Regularizes Kernel.

5. Results of Numerical Experimental

This study shows the effect of regularizes techniques on transfer learning in reducing overfitting of the fruit maturity dataset, with the architecture that stated in the proposed transfer learning sub-chapter. The regularizes technique that is the concern of this research is Dropout, Batch Normalization, and Regularizes kernel. In the process of forming the model in each training data the Data Augmentation process is performed using ImageDataGenerator from the Keras library with parameter settings rotation_range = 40, shear_range = 0.2, horizontal_flip = true, width_shift_range = 0.2, height_shift_range = 0.2, zoom_range = 0.2, and fill_mode = 'nearest'. While the data validation and testing not done in the Data Augmentation process. The number of images for each category used for image training, validation, and testing is 100, 20, 20 images, respectively.

In the implementation of the three types of transfer-learning architecture proposed to use the python and Keras library. The process of forming the model uses the callback function from the Keras library. The callback function uses EarlyStoping and ModelCheckpoint. The property of EarlyStoping used is monitor = val_loss, patience = 100. The property of the ModelCheckpoint used is monitor = val_accuracy, mode = max. The optimizer used is Adadelta with the property given is learning rate = 1.0, rho = 0.95, epsilon = 1e-08, and metrics = [‘accuracy’]. The number of epochs used in each model formation is 200 epochs.

As machine learning and deep learning, It needs to know the performance of each proposed architecture. The performance of the three proposed architecture considered based on the accuracy, precision, recall, and F-Measure value achieved by each architecture. The formula for calculating the performance value is as follows [40]:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F - Measure = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \tag{4}$$

where:
TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

By following the stages of each type of architecture proposed, the process of forming the model is considered based on the value of loss and accuracy. In this study, the research also looks for the best Dropout value and also considers the effect of Dropout value on system performance. Therefore, in this experiment, a Dropout value was changed between Dropout 0.2 to 0.7. It found that the best Dropout value was Dropout 0.5 based on the experimental result conducted on the fruit ripeness dataset. The representation of the loss value comparison between training and validation for 200 epochs using Regulator Dropout 0.5 is as stated in Fig. 3. The accuracy values comparison between training and validation during the formation of the model shown in Fig. 4.

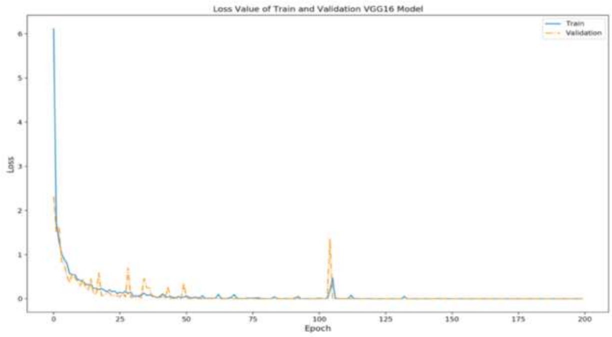


Fig.3. Comparison of optimizer loss of train and validation data.

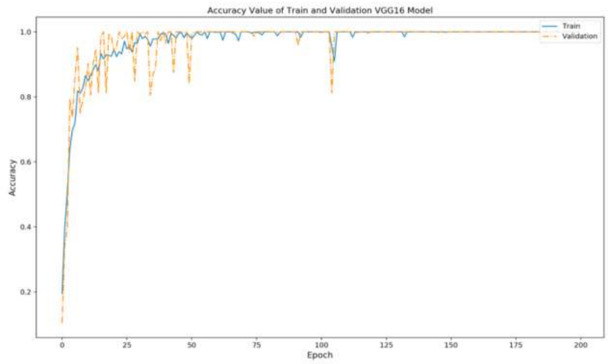
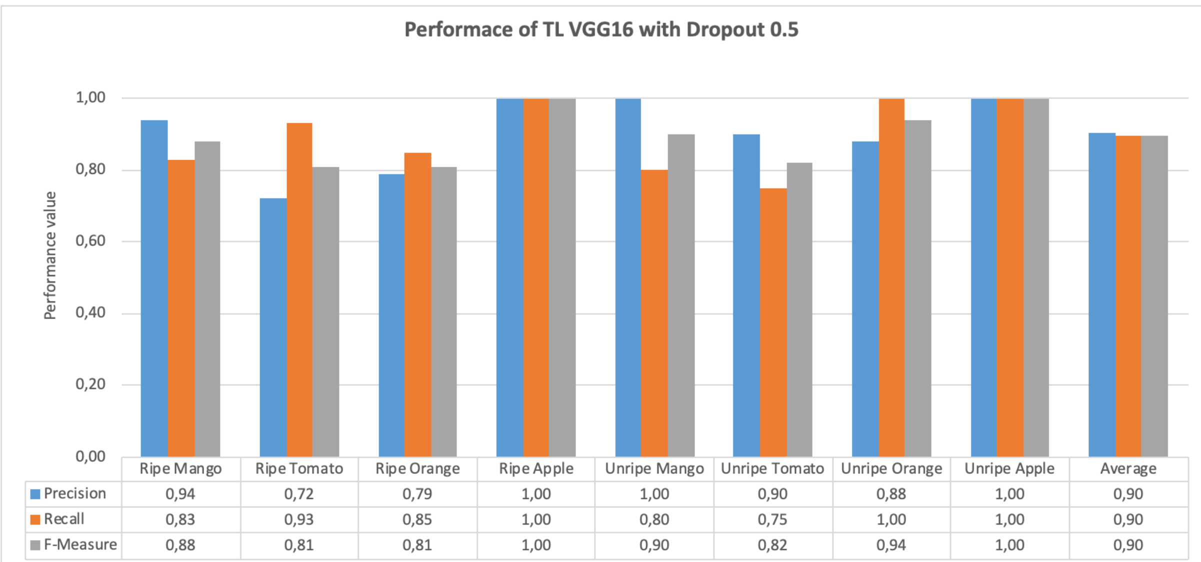


Fig.4. Comparison of optimizer accuracy of train and validation data.

Based on the experimental results obtained, the best accuracy value for architecture using Regulator Dropout 0.5 is 90%, with an average precision value is 0.90. The average recall and F-Measure at Dropout 0.5 are 0.90 dan 0.90, respectively. The average precision value, recall value, and F-Measure value for each type of fruit maturity at Dropout 0.5 are as shown in Fig. 5. The best performance of the architecture proposed in Ripe Apples and Unripe Apple. The comparison of the value of accuracy, precision, recall, and F-Measure for Dropout between 0.3 to 0.7 stated in Table 1. Based on the experimental results conducted, the performance of the proposed system resulted in better performance, namely accuracy values above 81%. While the accuracy value obtained in the [13] study only reached 76%. It shows that the proposed performance produces better results. The achievement of the proposed system with Dropout 0.5 has increased the accuracy value by 18.42% if it compared to the machine learning using extraction traditional color feature. A comparison of performance between color feature extraction [13] and transfer learning VGG16 using Dropout 0.5 is as represented in Fig. 6.



	Ripe Mango	Ripe Tomato	Ripe Orange	Ripe Apple	Unripe Mango	Unripe Tomato	Unripe Orange	Unripe Apple	Average
Precision	0,94	0,72	0,79	1,00	1,00	0,90	0,88	1,00	0,90
Recall	0,83	0,93	0,85	1,00	0,80	0,75	1,00	1,00	0,90
F-Measure	0,88	0,81	0,81	1,00	0,90	0,82	0,94	1,00	0,90

Fig.5. Performance of VGG16 Transfer Learning for each category with Dropout 0.5.

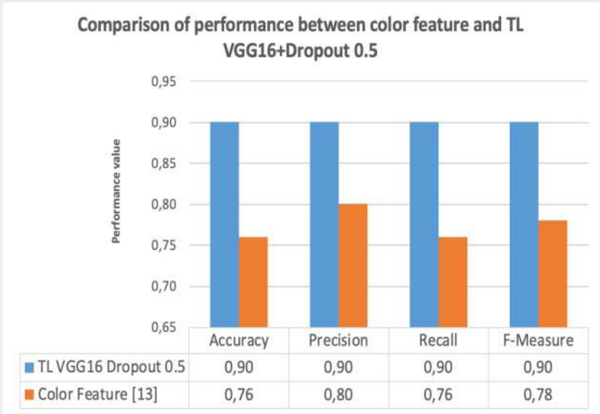


Fig.6. Comparison of performance between color feature and TL VGG16 with Dropout 0.5.

Table 1. Performance of system to the effect of the Dropout value

Performance	Dropout				
	0.3	0.4	0.5	0.6	0.7
Accuracy	0.82	0.83	0.90	0.82	0.81
Precision	0.84	0.84	0.90	0.84	0.83
Recall	0.83	0.82	0.90	0.83	0.82
F-Measure	0.83	0.82	0.90	0.82	0.81

On the second architecture, we used the same procedure with the first architecture before with the Batch Normalization regularizer as replace the Dropout. Based on the experimental results conducted on the same dataset, we found that the proposed performance only reaches an accuracy value of 84%. The average precision values, the average recall value, and the average value of the architectural F-Measure proposed using Batch Normalization are 0.84, 0.84, and 0.84, respectively. While the performance for each category of fruit maturity dataset is as shown in Table 2.

Table 2. Performance of each category from transfer learning VGG16 with Batch Normalization

Fruit Ripeness	Precision	Recall	F-Measure
Ripe Mango	0,77	0,91	0,83
Ripe Tomato	0,59	0,62	0,61
Ripe Orange	0,93	0,88	0,90
Ripe Apple	0,95	1,00	0,97
Unripe Mango	1,00	0,71	0,83
Unripe Tomato	0,57	0,57	0,57
Unripe Orange	0,94	1,00	0,97
Unripe Apple	1,00	1,00	1,00

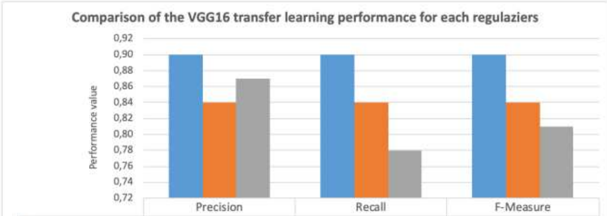
In this study also, testing the use of the Regularizer kernel to reduce overfitting. By using the same steps in Dropout and Batch Normalization but replacing the regularizer type with the Regularizer kernel, the test results for each fruit maturity type category stated in Table 3. The accuracy value of the proposed architecture with uses Regularizer kernel reaches is 78%. The average precision of the proposed architecture using the Regularizer kernel reaches 0.87. The average value of recall and F-Measure reached 0.78 and 0.81, respectively.

Table 3. Performance of each category from transfer learning VGG16 with the Regulazier kernel

Fruit Ripeness	Precision	Recall	F-Measure
Ripe Mango	0,92	0,73	0,81
Ripe Tomato	0,42	1,00	0,60
Ripe Orange	0,86	0,80	0,83
Ripe Apple	0,92	0,92	0,92
Unripe Mango	1,00	0,71	0,83
Unripe Tomato	1,00	0,07	0,13
Unripe Orange	0,85	1,00	0,92
Unripe Apple	1,00	1,00	1,00

6. Discussion

Based on the experimental results carried out, we compare the performance of the proposed architecture based on the average value of accuracy, precision, recall, and F-measure of each type of regularizer. The performance comparison between Dropout, Batch Normalization, and Regularizes kernel is as stated in Fig. 7. The best performance of the proposed architecture is the transfer learning architecture using Dropout 0.5. The lowest accuracy performance is using the Regularizer kernel. However, when compared to its performance with machine learning with the extraction of traditional color features showing its performance is still better, namely, its accuracy increases by 10.52%.



	Precision	Recall	F-Measure
Dropout	0,90	0,90	0,90
Batch Normalization	0,84	0,84	0,84
Regularizer kernel	0,87	0,78	0,81

Fig.7. Comparison of performance of TL VGG16 for each regularizers.

The second-best performance is to use Batch Normalization that reaches an accuracy value of 84%. When compared with the performance in previous studies [13] that uses color feature extraction, architecture with Batch Normalization achieved an increase in accuracy of 10.52%. But when compared to the proposed architecture using Dropout, using Batch Normalization has poor performance. Based on experimental results carried out on the fruit maturity dataset, it concluded that Dropout has better performance than Batch Normalization. It shows that Dropout cannot be exchanged with Batch Normalization, as stated in [28]. But to get the best performance from the proposed architecture in use Dropout, it is necessary to test various specific Dropout values. Because when compared with other Dropout values, regularizes using Batch Normalization can produce better performance also, for example, Dropout 0.7.

For future research, it is necessary to pay attention to how to determine the value of a Dropout appropriately. How to form adaptive regularizes without using manual customization. It is important to dig more into what affects the Dropout value for better performance.

7. Conclusions

In this study, researchers have compared the performance of regularizes results to reduce overfitting. Regularizes used in the proposed VGG16 transfer learning architecture is Dropout, Batch Normalization, and Regularizes kernel. The best accuracy system performance of VGG16 transfer learning architecture with Dropout, Batch Normalization, and Regularizer kernel is 0.90, 0.84, and 0.76, respectively. If the accuracy performance of VGG16 transfer learning architecture with Dropout, Batch Normalization, and Regularizer kernel is compared to the accuracy performance of color feature extraction then the performance of Dropout, Batch Normalization, and Regularizer kernel increases by 18.42%, 10.52%, and 2.63%, respectively. If we consider the best performance of the precision, recall, and F-measure, for the VGG16 transfer learning architecture using Dropout go up the amount 12.50%, 18.42%, and 15.38%, respectively. If we look at the performance of the best precision, recall, and F-measure, VGG16 transfer learning architecture with Batch Normalization increases respectively by 5.00%, 10.52%, and 7.69%. If we verify at the performance of the best precision, recall, and F-measure, VGG16 transfer learning architecture with Regularizer kernel, upward by 8.75%, 2.63%, and 3.97%, respectively.

The experimental results show that the performance of deep learning using transfer learning always results in better performance when compared to using machine learning with traditional feature extraction. This study shows the effect of regularizer use to reduce overfitting. With transfer learning using regularizer can improve system performance in terms of the average value of accuracy, the average value of precision, the average value of recall, and the average value of F-Measure on the test data substantially. The results of this experiment also show that regularizer using Batch Normalization cannot replace Dropout. Based on experimental results obtained that the best performance on fruit ripeness test data using Dropout 0.5. It’s declaring that the best technique to reduce overfitting in transfer learning is Dropout.

References

[1] V.E. Nambi, K. Thangavel, A. Manickavasagan, and S. Sharir, “Comprehensive ripeness-index for prediction of ripening level in mangoes by multivariate modelling of ripening behavior”, Int. Agrophys, vol. 31, pp. 35-44, 2017

[2] M. Dadwal and V.K. Banga, "Color Image Segmentation for Fruit Ripeness Detection: A Review", 2nd International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2012), 2012

[3] S. Naik and B. Patel, "Machine Vision based Fruit Classification and Grading - A Review", International Journal of Computer Applications, Vol. 170(9), July 2017.

[4] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review", Journal of King Saud University – Computer and Information Sciences, 2018.

[5] Y. Onishi, T. Yoshida, K. Kurita, et al., "An automated fruit harvesting robot by using deep learning". Robomech J 6, 13 (2019) doi:10.1186/s40648-019-0141-2

[6] I. Hussain, Q. He, and Z. Chen, "Automatic Fruit Recognition Based on DCNN For Commercial Source Trace System", International Journal on Computational Science & Applications (IJCSA), vol. 8(2), pp. 1 – 14.

[7] H. Muresan and M. Oltean, "Fruit recognition from images using deep learning", Acta Univ. Sapientiae, Informatica, 2018, pp. 26-42.

[8] L. Danev, "Fruit and vegetable classification from live video", MInf Project Report, University of Edinburgh, 2017.

[9] F.M.A. Mazen and A.A. Nashat, "Ripeness Classification of Bananas Using an Artificial Neural Network", Arabian Journal for Science and Engineering, Springer, 2019.

[10] S.R. Dubey and A.S. Jalal, "Fusing Color and Texture Cues to Categorize the Fruit Diseases from Images", Computer Vision and Pattern Recognition, https://arxiv.org/abs/1405.4930, 2014.

[11] H.M. Zawbaa, M. Abbass, M. Hazman, and A.E. Hassenian, "Automatic Fruit Image Recognition System Based on Shape and Color Features", Advanced Machine Learning Technologies and Applications. AMLTA 2014.

[12] S. Arivazhagan, R.N. Shebiah, S.S. Nidhyanandhan, L.Ganesan, "Fruit Recognition using Color and Texture Features", Journal of Emerging Trends in Computing and Information Sciences, vol.1, no.2, 2010.

[13] J.Pardede, M.G. Husain, A.N. Hermana, and S.A. Rumapea, "Fruit Ripeness Based on RGB, HSV, HSL, L*a*b* Color Feature Using SVM", International Conference of Computer Science and Information Technology, 2019.

[14] A. Patil and A.Zore, "Deep Learning based Computer Vision: A Review", IJIRT, vol.5, no,6, 2018.

[15] M.Z. Alom, T.M. Taha, C. Yakopcic, S. Westberg et.al., "A State-of-the-Art Survey on Deep Learning Theory and Architectures", Electronics, vol.8, no.3, 2019.

[16] F. Chollect, "Deep Learning with Python", 2018.

[17] K. Weiss, T.M. Khoshgoftaar, and D.D.Wang, "A survey of transfer learning", Journal of Big Data, vol. 3, no.9, 2016.

[18] S. J. Pan and Q. Yang, "A Survey on Transfer Learning", IEEE Transactions on Knowledge and Data Engineering, Volume 22 Issue 10, October 2010, Pages 1345-1359, 2010.

[19] M. Shaha and M. Pawar, "Transfer Learning for Image Classification", 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018.

[20] N. Houlsby, A Giurgiu, S. Jastrzebski, B. Morrone at al., "Parameter-Efficient Transfer Learning for NLP", Machine Learning, arXiv: 1902.00751, 2019.

[21] H.C. Shim, H.R. Roth, M.Gao, L. Lu et all., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning", vol. 35, no. 5,2016.

[22] J.Y. Lee, F. Dernoncourt, and P. Szolovits, "Transfer Learning for Named-Entity Recognition with Neural Networks", Computation and Language, arXiv:1705.06273, 2017.

[23] X. Ying, "An Overview of Overfitting and its Solutions", Journal of Physics: Conference Series, Intelligent system and control technology, vol. 1168, no. 2, 2019.

[24] S. Salman and X. Liu, "Overfitting Mechanism and Avoidance in Deep Neural Networks", Machine Learning, arXiv: 1901.06566, 2019.

[25] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, "A Study on Overfitting in Deep Reinforcement Learning", Machine Learning, arXiv: 1804.06893, 2018.

[26] B. Wu, Z. Liu, Z. Yuan, G. Sun et al., "Reducing Overfitting in Deep Convolutional Neural Networks Using Redundancy Regularizer", Artificial Neural Networks and Machine Learning – ICANN 2017, Lecture Notes in Computer Science, 2017.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever et al., ""Dropout: A Simple Way to Prevent Neural Networks from Overtting", Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.

[28] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Machine Learning, arXiv: 1502.03167, 2015.

[29] A. Khan, A. Sohail, U. Zahoora, and A.S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", Computer Vision and Pattern Recognition, arXiv: 1901.06032, 2019.

[30] S. Shuvaev, H. Giaffar, and A.A. Koulakov, "Representations of Sound in Deep Learning of Audio Features from Music", arXiv:1712.02898, 2017.

[31] W. Liu, Z. Wang, X. Liu, N. Zeng et al., "A survey of deep neural network architectures and their applications", Neurocomputing, vol. 234, pp. 11-26, 2017.

[32] J. Pardede, B. Sitohang, S. Akbar, M.L. Khodra, "Improving the Performance of CBIR Using XGBoost Classifier with Deep CNN-Based Feature Extraction", International Conference on Data and Software Engineering, 2019.

[33] R. Raina, A. Battle, H. Lee, B. Packer et al, "Self-taught learning: Transfer Learning from Unlabeled Data", Proceedings of the 24th international conference on Machine learning, pp. 759-766, 2007.

[34] C. Tan, F. Sun, T. Kong, W. Zhang et al., "A Survey on Deep Transfer Learning", Machine Learning, arXiv: 1808.01974, 2018.

[35] Z. Huang, Z. Pan, and B. Lei, "Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data", remote sensing, vol. 9, no. 8, 2017.

[36] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review", Computational Intelligence and Neuroscience, 2018.

[37] J. Zhang, W. Li, P. Ogunbona, and D. Xu, "Recent Advances in Transfer Learning for Cross-Dataset Visual Recognition: A Problem-Oriented Perspective", Computer Vision and Pattern Recognition, arXiv: 1705.04396, 2017.

[38] S. Francis, J.V. Landeghem, and M.F. Moens, "Transfer Learning for Named Entity Recognition in Financial and Biomedical Documents", Information, vol. 10, 2019.
[39] M. Hussain, J.J. Bird, and D.R. Faria, "A Study on CNN Transfer Learning for Image Classification", Advances in Intelligent Systems and Computing, vol. 840. Springer, Cham, 2018.
[40] J. Pardede and M.G. Husada, "Comparison of VSM, GVSM, and LSI in Information Retrieval for Indonesian Text, Journal Teknologi, vol. 78, vol. (5-6), pp. 51-56, 2016.

Authors' Profiles

Jasman Pardede received Bachelor degree in science math from Universitas Andalas (Unand), Indonesia, in 2001, and the M.Eng degree in Informatic Engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia in 2005. Since 2005 until now, he is active as a lecturer at Institut Teknologi Nasional Bandung. He currently continuing his doctoral studies at Institut Teknologi Bandung (ITB), Bandung, Indonesia since 2016 in field Image Retrieval.

Benhard Sitohang is a professor in field Software Engineering and Knowledge. He received Bachelor degree from Institut Teknologi Bandung, Indonesia, in 1978. He received the Master degree in filed Software Engineering from USTL-Montpellier, Montpellier, France in 1981, and he received Doctoral degree from the same institute, in 1983. He currently works at Institut Teknologi Bandung (ITB) as a lecturer and researcher.

Saiful Akbar received Bachelor degree in Department of Informatics, School of Electrical and Informatics, Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 1997, and the M.Eng degree in Informatic Engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia in 2002. Doctoral degree in Engineering Science from Johannes Kepler Universitat Linz – Austris, in 2007. Since 1998 until now, he is active as a lecturer at Institut Teknologi Bandung (ITB), Bandung, Indonesia. He currently works at Institut Teknologi Bandung (ITB) as a lecturer and assistant professor.

Masayu Leylia Khodra received Bachelor degree in Department of Informatics, School of Electrical and Informatics, Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 1998, and the M.Eng degree in Informatic Engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia in 2006. Doctoral degree, She received in the same institute, in 2012. Since 2008 until now, She is active as a lecturer at Institut Teknologi Bandung (ITB), Bandung, Indonesia. She currently works at Institut Teknologi Bandung (ITB) as a lecturer and associate professor.

How to cite this paper: Jasman Pardede, Benhard Sitohang, Saiful Akbar, Masayu Leylia Khodra, "Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection", International Journal of Intelligent Systems and Applications(IJISA), Vol.13, No.2, pp.52-61, 2021. DOI: 10.5815/ijisa.2021.02.04