International Journal of Engineering & Technology, 7 (4.40) (2018) 75-80

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Evaluation of Deep Convolutional Neural Network Architectures for Strawberry Quality Inspection

Rika Sustika[1], Agus Subekti[2], Hilman F. Pardede[1], Endang Suryawati[1],
Oka Mahendra[1], Sandra Yuwana[1]

[1]Research Center for Informatics (P2I)
[2]Research Center for Electronics and Telecommunications (PPET)
Indonesia Institute of Sciences (LIPI), Bandung, Indonesia
**Corresponding author E-mail:rika002,agus075@lipi.go.id*

Abstract

Fruits quality inspection is important task on agriculture industry. Automated inspection using machine and vision technology have been widely used for increasing accuracy and decreasing working cost. Convolutional Neural Network (CNN) is a type of deep learning that had a great success in large scale image and video recognition. In this research, we investigate the effect of different deep convolutional neural network architectures on its accuracy in strawberry grading system (quality inspection). We evaluate different types of existing deep CNN architectures such as AlexNet, MobileNet, GoogLeNet, VGGNet, and Xception, and we compare them with two layers CNN architecture as our baseline. Here, we have done two experiments, the first is two classes strawberry classification and the second is four classes strawberry classification. Results show that VGGNet achieves the best accuracy, while GoogLeNet achieves the most computational efficient architecture. The results are consistent on both two classes classification and four classes classification.

Keywords: *CNN; deep learning; quality inspection; strawberry*

1. Introduction

Automatic fruit grades classification is very useful in sorting the harvest production based on the quality of the fruits. This grading classification for quality grading can be used for determining prices, fulfillment of orders with certain quality standards and also for other post-harvest processing. Some of the methods employed include high performance liquid chromatography [1], near-infrared imaging [2], and gas sensor [3]. However, these approaches require expensive devices and professional operators. In addition, the resulting accuracy may not be satisfactory. These studies report accuracies below 85 %.

Fruit classification using image data-based machine learning is another approach. This approach is cheaper because it only requires a digital camera for the acquisition of fruit images. Better accuracies are also reported (many systems achieve accuracies above 85 %). Support vector machine (SVM) based method was proposed in [4] with an accuracy of 88.2 %. A neural network based artificial bee colony (ABC) was proposed in [5] with an accuracy of 89.47 %. Better performance could be achieved by using more complex machine learning methods such as feed forward neural network in [6], or using more complex feature extraction such as using texture and shape features [7], and fractional Fourier entropy (FRFE) in [8].

Recent developments show that deep learning, as a newest technology in machine learning, provide better accuracy results than previous (shallow) machine learning algorithms. Deep learning has superior performance because of its ability to extract high-level features from raw input data due to the use of many non-linear functions. This feature extraction capability is obtained by statistical learning using neural network layers structure with input from large amount of image data. Recent ImageNet Large Scale Visual Recognition Competition (ILSVRC), an annual competition for object classification tasks [9], shows that winners from the recent years competition usually employ deep learning architectures, in particular Convolutional Neural Network (CNN). It shows excellent ability to recognize objects for ImageNet data [10] competition. The competition produced some superior deep CNN architectures such as AlexNet [11], GoogLeNet [12], Xception [13], VGGNet [14], MobileNet [15] compared to many shallow architectures such as SVM. Deep learning technology has been applied to many other fields such as abnormality detection using medical images [16], carcinoma nuclei grading [17], and others. In studies [18], 13 DNN layers are used to recognize the image of the fruit that reaches accuracy of 94.94 %. In [19], convolutional neural network (CNN), a variant of deep learning architectures, with the input of image elements of RGB (red, green, blue) and D (depth) are used to perform fruit and vegetable grading. The resulting accuracy reaches 97 %.

In this paper we evaluate CNN performance for implementation on strawberry quality inspection. We design a simple CNN architecture with two convolutional layers as baseline. Then we evaluate and compare five popular CNN architectures for this task. They are AlexNet, GoogLeNet, VGGNet, Xception, and MobileNet. The rest of the paper is organized as follows. Section 2 introduces the evaluated architectures. Section 3 details the experimental setup. Section 4 shows the result and discussion. We conclude the paper in section 5.

2. Evaluated Architectures

We evaluated the performance of some deep learning architectures, especially CNN algorithm for strawberry quality inspection. A typical CNN architecture consist of several convolutional layers followed by fully connected networks. First we design simple convolutional neural network with two convolutional layers and on top of that we stack two fully connected layers as baseline architecture. Then we also evaluate five popular CNN architectures: AlexNet, VGG-16, GoogLeNet, Xception, and MobileNet. The detail explanation of each network is described in this section.

2.1. Baseline CNN

As baseline on this research, we used simple CNN architecture with two convolutional layers and two fully connected layers on the top of the convolutional layers as depicted on Fig. 1. This architecture uses 3x3 convolution filter with stride 2. Max pooling was applied over a 2x2 pixel window on the second convolutional layer. Dropout 20% was applied to the first convolutional layer and dropout 50% was applied on first fully connected layer to avoid overfitting problem.

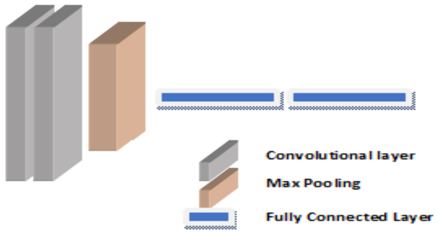


Fig. 1: Baseline CNN Architecture

2.2. AlexNet

AlexNet [11] was first proposed to perform classification of 1.2 million high resolution images in the LSVRC-2010 ImageNet contest. The big numbers of images have to be classified to 1000 different classes of object. The proposed deep neural network has 60 million parameters and 650,000 neurons. Training is made faster by using non-saturating neurons and a very efficient GPU implementation of the convolution operations. In the ILSVRC-2012, the proposed neural network system achieved the top 5 among participants. The networks achieved a test error rate of 13.3 %. After the publication, AlexNet has been used on many tasks such as object detection [20], image segmentation [21], and video classification [22].

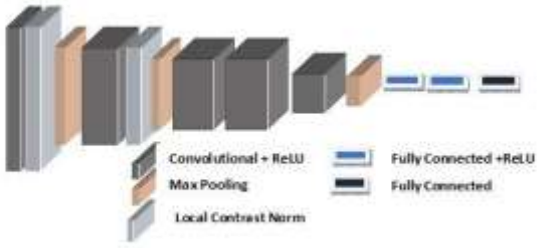


Fig. 2: AlexNet Architecture

AlexNet has relatively simple layout, as can be seen on AlexNet architecture on Fig. 2 [11]. It consists of five convolutional layers and three fully connected layers. Some of convolutional layers are followed by max-pooling layers. Convolutional process uses filter size 11x11 with stride 4 pixel on first layer, 5x5 on second layer, and 3x3 on the remaining layers. The output of the last fully connected layer is fed to a softmax function. ReLU (Rectified Linear Unit) is used as activation function in each of convolutional layer. AlexNet also apply local response normalization after applying ReLU in certain layers. AlexNet use data augmentation and dropout to reduce overfitting on image data. Data augmentation is artificially enlarge dataset using some transformation such as image translation, horizontal reflections, and altering the intensities of the RGB channels. Dropout is a process to set the output become zero value of each hidden neuron, to reduce complexity of the co-adaptations of neurons. AlexNet uses dropout in the first two fully connected layers with probability 0.5.

2.3. VGGNet

VGGNet is one of the deep learning architectures proposed by VGG team for their ImageNet Challenge 2014 submission [14]. VGGNet investigate the effect of the convolutional network depth on its accuracy for this contest. The submission achieved the first and the second places in the localization and classification. VGGNet improves AlexNet by adding the network depth. In VGGNet, the number of convolutional layers are added. There are some configurations of VGGNet, depend on number of convolutional layers in the networks. On this research, we used VGG-16, that consists of 13 stacks of convolution layer which is followed by 3 fully-connected layers and the final layer is the soft-max layer. VGG-16 makes the improvement over AlexNet by replacing large kernel size filters (11x11 in the first convolutional layer and 5x5 in the second convolutional layer) with multiple 3x3 sized filters with stride 1 for all convolutional processes. All hidden layers are equipped with ReLu (Rectified Linear Unit). Unlike AlexNet, there is no local response normalisation (LRN) in VGGNet because LRN leads to increased memory consumption and computation time [14]. Some of the convolutional layers is followed by max-pooling, performed over 2x3 pixel window with stride 2. VGG-16 architecture can be seen on Fig.3 [14].

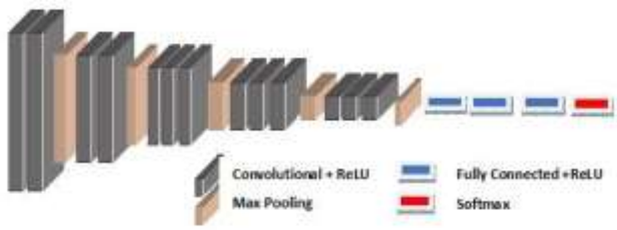


Fig. 3: VGGNetArchitecture

2.4. GoogLeNet

GoogLeNet is a deep convolutional neural network architecture that proposed by Szegedy et al. for ILSCVR 2014 competition [12]. GoogLeNet improved accuracy while keeping computational load constant by inscreasing not only the depth of the networks but also the width of the networks [12]. The performance at the 2014 ILSVRC achieved error rate of 6.67 %, put it in first place among participants.

The most common way for improving performance of deep neural network is by increasing the size of the network. It includes the number of layers (depth) and the number of units in each layer (width) of the network. Increasing size of network has some drawbacks. It would increase the number of parameters to train and as the consequences, it require more computational resources. These problems can be solved by moving from fully connected to sparsely connected architectures, even inside the convolutions [12]. GoogLeNet solves it by utilizing inception module. Inception module uses a parallel combination of 1x1, 3x3, and 5x5. 1x1 convolutions are used to compute reductions before expensive 3x3 and 5x5 convolutions. Single inception module can be seen on Fig.4 [12]. GoogLeNet architecture uses 9 inception modules, consists of 22 layers deep when counting only layers with parameters. Beside the 22 layers deep network if we count only layers with parameters, there are also 5 pooling layers (four max pooling layers and one average pooling layer). Average pooling with 5x5 filter size and stride 3 is used before the classifier. GoogLeNet use dropout layer with 70% ratio of dropped outputs. The ReLU is used in all convolutional layers, including inside the inception modules. The complete schematic of GoogLeNet architecture is depicted in Fig. 5 [23].

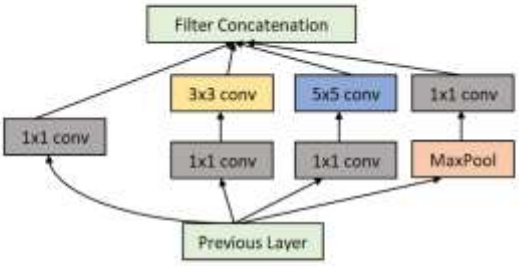


Fig. 4: Inception Module

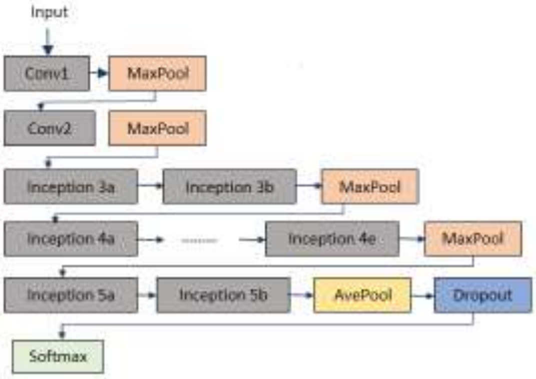


Fig. 5: GoogLeNet Architecture

2.5. Xception

Xception (short form of extreme inception) was proposed by Franois Chollet in 2017 as an improvement of GoogLeNet. On this architecture, the standard Inception modules are replaced by depthwise separable convolutions (extreme version of Inception module). This architecture slightly outperform inception V3 on the ImageNet dataset and outperforms Inception V3 on a larger image classification dataset significantly [13].

The Xception architecture based entirely on depthwise separable convolutional layers. Depthwise separable convolutional almost identical with extreme form of inception module such as depicted on Fig. 6 [13]. The differences between depthwise separable convolution and extreme inception is depthwise separable convolutions perform wise spatial convolution first, and then 1x1 convolution, whereas inception performs the 1x1 convolution first.

The Xception architecture has 36 convolutional layers. These layers are structured into 14 modules. All modules have linear residual connection, except for the first and last modules, as depicted on Fig. 7 [13]. The data first goes through four modules on entry flow, then through eight modules on middle flow, and finally through two modules on the exit flow.

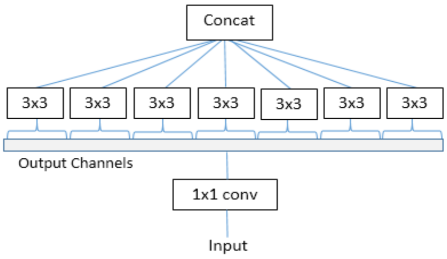
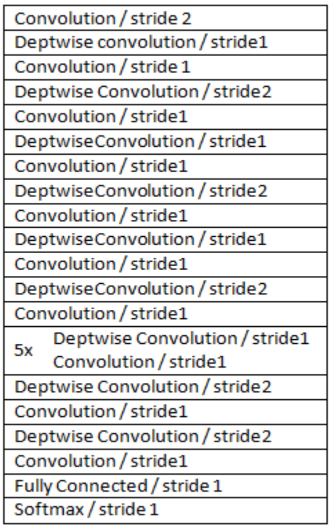


Fig. 6: Extreme form of inception module

2.6. MobileNet

The MobileNet is one of CNN architectures that is proposed for mobile and embedded vision applications [15]. MobileNet structure is built on depthwise separable convolutions except for the first layer which is a full convolution [15]. MobileNet has 28 layers. All layers, except the final layer, are followed by a batch normalization and ReLu. The final layer is a fully connected layer that feeds into a softmax layer for classification. Average pooling is used before the fully connected layer to reduce the spatial resolution to 1. MobileNet architecture is depicted on Fig. 8 [15].



Convolution / stride 2
Deptwise convolution / stride1
Convolution / stride 1
Deptwise Convolution / stride2
Convolution / stride1
DeptwiseConvolution / stride1
Convolution / stride1
DeptwiseConvolution / stride2
Convolution / stride1
DeptwiseConvolution / stride1
Convolution / stride1
DeptwiseConvolution / stride2
Convolution / stride1
5x Deptwise Convolution / stride1 Convolution / stride1
Deptwise Convolution / stride2
Convolution / stride1
Deptwise Convolution / stride2
Convolution / stride1
Fully Connected / stride 1
Softmax / stride 1

Fig. 8: MobileNet architecture

3. Experimental Setup

For experiments, we collect a number of image data that are grouped into binary classes and four classes labels. Then the data are preprocessed and some of them are used to train six types of CNN architectures, while the rest are used as test data. Simple diagram of our research methods can be seen on Fig. 9.

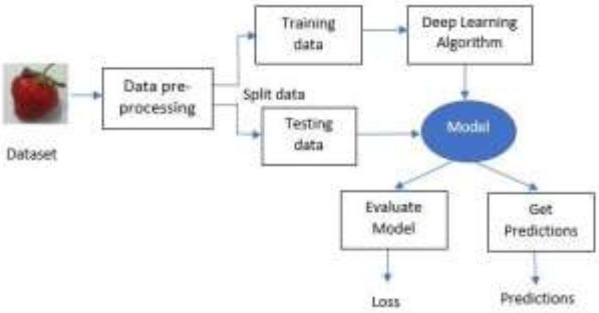


Fig. 9: Research Method

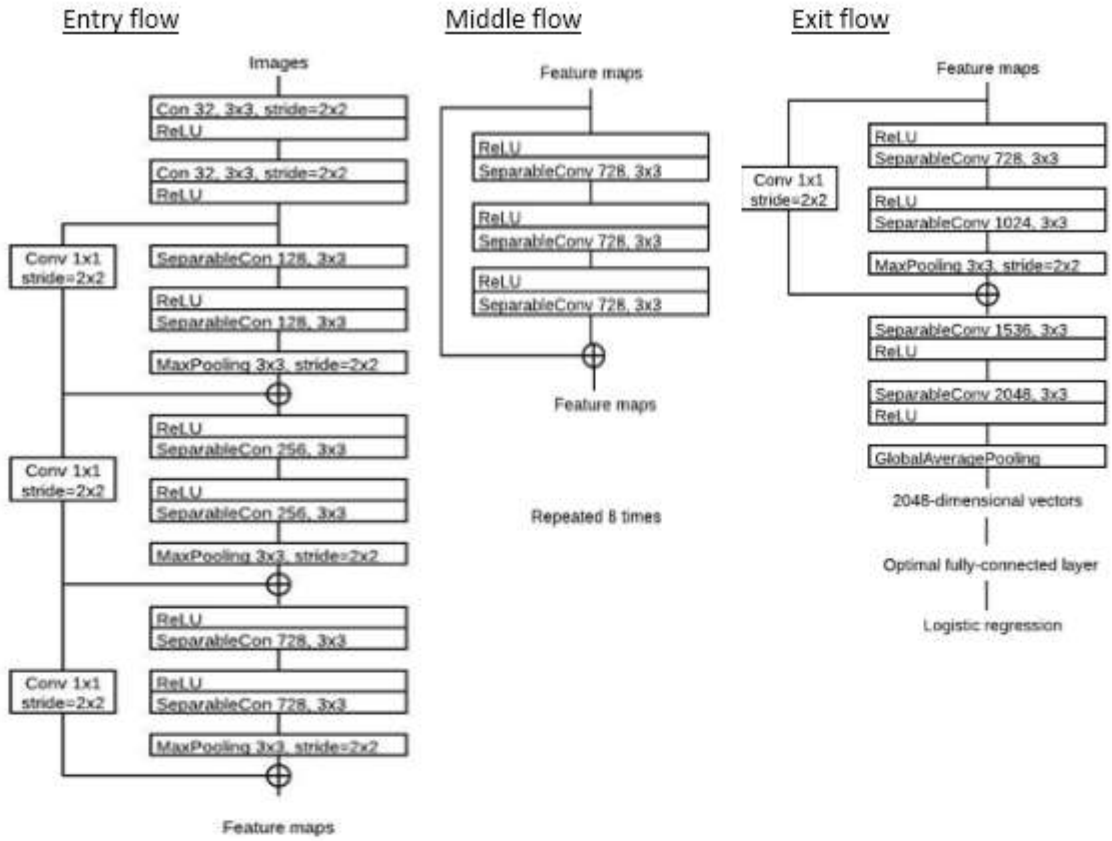


Fig. 7: Xception Architecture

On conventional machine learning algorithm, feature extraction process is a separate process from the classifier. It is important process that will determine classifier accuracy. Different with that, deep learning learns the features of image automatically. We do not need to manually extract features from the image. We only just feed the image to the network and the network learns to extract features and contextual details from the image on single process.

We collect 1870 images of fruit using two digital cameras and three smartphone cameras. Original dataset consists of RGB images with different resolution. We group the images on two classes first, they are bad and good class. Overripe, damage, and rotten strawberries are fall into bad class, and the rest are good class.

Then we also group the data into four classes labels, where we grade good class into three ranks (first, second, and third rank) and one class of bad strawberry (fourth rank). First rank is good strawberry with light red and normal shape, second rank is good strawberry with dark red and normal shape, and third rank is good strawberry with abnormal shape.

In total 1870 images, our dataset consists of 1000 images with good quality (523 images for 1st rank, 355 images for 2nd rank, and 122 images for 3rd rank) and 870 images for bad quality (4th rank). Sample of the image data we collect can be seen on Fig. 10.

On this research, we only preprocessed dataset by resizing all the images into fixed size 64x64 RGB images and extract the RGB values from each image as features. For training the system, we use 80% of data for training, 10 % is used for validation, and the remaining is used for testing. We use test accuracy to measure the performance of the system. We also compared training time and model size resulted from the training process. All of the architectures used same training and validation sets with number of learning epochs is 40 epochs and batch size is 10. This CNN training was implemented in python using Keras and Tensorflow packages.

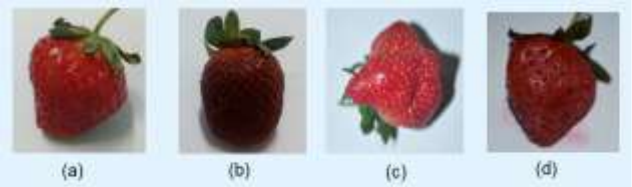


Fig. 10: (a) 1st rank, (b) 2nd rank, (c) 3rd rank, (d) 4th rank

4. Result and Discussion

The following section presents the results of the evaluated architectures for strawberry quality inspection. In the first experiment, we performed strawberry grading into two categories of good or bad quality. In the second experiment, strawberries was classified into 4 grade. The accuracies of all CNN network architectures are shown in Table I.

As expected, binary classifications achieve higher accuracy than four classes classifications. The class of good and bad are more distinctive in color and shape making it easier to classify, while for four classes classification, the classes are less separable.

Table 1: Accuracy (%) of CNN architectures for 2 and 4 classes strawberry grading

Architectures	Accuracy (%)	
	2 classes	4 classes
Baseline	85,61	73,33
AlexNet	96,48	87,37
GoogLeNet	91,93	85,26
VGGNet	**96,49**	**89,12**
Xception	92,63	87,72
MobileNet	83,51	64,56

We notice that adding the depth of the networks may benefit the classification accuracies. By comparing the baseline (with 4 layers), AlexNet (with 8 layers), and VGGNet (with 13 layers) that have quite similar architectures, it appears that VGGNet achieves the best, with AlexNet the second and baseline the third. The results are consistent on both two classes classification and four classes classification, indicating the effect of the network depth to the performance of the systems.

We also notice that the width of the layers may also contributing to the performance. By comparing GoogLeNet and Xception, we find that Xception is superior to GoogLeNet. GoogLeNet uses inception modules and Xception and MobileNet uses depthwise separable convolutional operation. Xception slightly outperform GoogLeNet on this experiment. This result inline with result from experiment on Imagenet dataset [13]. Increasing the depth of network and replacing inception modules with depthwise separable convolution make the system better.

Comparing between all of the architectures, as observed from Table I, VGGNet outperform all of the architecture on first and second experiment, with 96.49 % on first experiment and 89.12 % accuracy on the second experiment. VGGNet outperformed GoogLeNet, Xception, and MobileNet even though the three architectures are deeper and wider that VGGNet. These results are different with previous research on ImageNet dataset that GoogLeNet and Xception outperformed VGGNet. It maybe because the type of dataset used in this experiment has different characteristic with Imagenet dataset. MobileNet got the worst accuracy over all architectures maybe because the focus of the MobileNet architecture is for mobile and embedded vision applications that need smaller and faster model.

Besides test accuracy, we also evaluate complexity of all architectures by comparing training time and size of model resulted from completed training. The results can be seen on Table II. By comparing baseline CNN, AlexNet, and VGGNet that has almost similar architecture with different depth, we could observe that the accuracy of VGGNet must be paid with the size of the model and training time, which is larger than baseline. But VGGNet still produces smaller model than AlexNet. We could observe from these results that complexity of the architecture is linearly correlated with the depth of layer.

Next, we compare the size of the models from GoogLeNet, Xception, and MobileNet. Surprisingly, GoogLeNet is the fastest to train compared to others including MobileNet, and produce the smallest model. It maybe because of the used of 1x1 convolutions on the architecture. This property used for reducing the dimension and limiting the size of the network to decrease computational complexity. From these results we could say that GoogLeNet has the most computational efficiency.

Table II: Training time and size of models

Architectures	Training time	Size of Model (MB)
Baseline	1840.26	3271.42
AlexNet	4834.90	357.1
GoogLeNet	**853.29**	**0.6**
VGGNet	5391.08	167.2
Xception	9076.54	479.0
MobileNet	3271.42	479.0

From Tables I and II, we can see that every architecture has advantages and drawbacks. Among the six architectures, GoogLeNet shows fastest speed and smallest model size. GoogLeNet has computational efficiency, so that this model can be run on devices with limited computational resources, especially with low memory. VGGNet has the best accuracy but it need more training time and it has big size of model. We can choose type of architecture to be implemented, depend on the kind of model implementation. When the training time and memory devices is important concern, using GoogLeNet model could be a good choice. When accuracy is the most important thing, VGGNet is the best choice. On our research, model will be implemented for strawberry quality inspection with web and desktop based. In our system, size of model and computation complexity is not a big problem but accuracy is the most important thing. So we choose VGGNet as our classifier. Graph of train and validation accuracy of VGGNet on every epoch can be seen on Fig. 11.

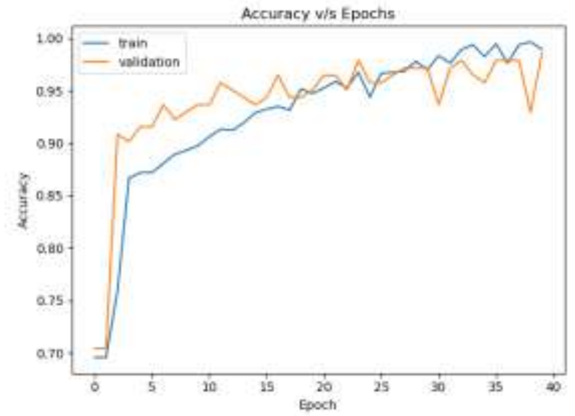


Fig.11: Accuracy vs epoch on VGGNet architecture

Fig. 11 shows that validation accuracy increased rapidly at the first 3 epoch and than increased slowly until around 30th epoch and tend to stable with a little fluctuation from 30 epoch to 40 epoch.

5. Conclusion

This paper present the evaluation of deep learning technology, especially deep convolutional neural network architecture for strawberry quality inspection. We evaluated six types of architecture, baseline CNN, AlexNet, GoogLeNet, VGGNet, Xception, and MobileNet. The performance of these architectures is measured on a dataset representing 4 class categories. From the experiment we have got that VGGNet achieves the highest accuracy. The results prove that the depth of layer in CNN can improve the accuracy. The used of inception module on GoogLeNet not only increasing the depth but also the width of the network. This property significantly reduce computational complexity without significant performance penalty.

Acknowledgement

This research is funded by Ministry of Research, Technology, and Higher Education, Indonesia. The experiments are conducted on High Performance Computing (HPC) facility in Research Center for Informatics, Indonesian Institute of Sciences (LIPI).

References

[1] Pardo-Mates, A. Vera, S. Barbosa, M. Hidalgo-Serrano, O. Nez, J. Saurina, S. Hernndez-Cassou, and L. Puignou, “Characterization, classification and authentication of fruit-based extracts by means of hplc-uv chromatographic fingerprints, polyphenolic profiles and chemometric methods,” Food Chemistry, vol. 221, pp. 29 – 38, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0308814616316508

[2] W. Shao, Y. Li, S. Diao, J. Jiang, and R. Dong, “Rapid classification of chinese quince (chaenomeles speciosa nakai) fruit provenance by near-infrared spectroscopy and multivariate calibration,” Analytical and Bioanalytical Chemistry, vol. 409, no. 1, pp. 115–120, Jan 2017. [Online]. Available: https://doi.org/10.1007/s00216-016-9944-7

[3] Radi, S. Ciptohadijoyo, W. Litananda, M. Rivai, and M. Purnomo, "Electronic nose based on partition column integrated with gas sensor for fruit identification and classification," Computers and Electronics in Agriculture, vol. 121, pp. 429 – 435, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169915003622.

[4] Y. Zhang and L. Wu, "Classification of fruits using computer vision and a multiclass support vector machine," Sensors, vol. 12, no. 9, pp. 12 489–12 505, 2012. [Online]. Available: http://www.mdpi.com/14248220/12/9/12489

[5] M. F. Adak and N. Yumusak, "Classification of e-nose aroma data of four fruit types by abc-based neural network," Sensors, vol. 16, no. 3, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/3/304

[6] Z. Yudong, P. Preetha, W. Shuihua, J. Genlin, Y. Jiquan, and W. Jianguo, "Fruit classification by biogeography-based optimization and feedforward neural network," Expert Systems, vol. 33, no. 3, pp. 239–253, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12146

[7] F. Garcia, J. Cervantes, A. Lopez, and M. Alvarado, "Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets," IEEE Latin America Transactions, vol. 14, no. 7, pp. 3434–3443, July 2016.

[8] S. Wang, Z. Lu, J. Yang, Y. Zhang, J. Liu, L. Wei, S. Chen, P. Phillips, and Z. Dong, "Fractional fourier entropy increases the recognition rate of fruit type detection," BMC Plant Biology, vol. 16, p. 85, October 2016.

[9] O. Russakovsky, J. D. H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, December 2015.

[10] E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, "Comparison of regularization methods for imagenet classification with deep convolutional neural networks," AASRI Procedia, vol. 6, pp. 89 – 94, 2014, 2nd AASRI Conference on Computational Intelligence and Bioinformatics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212671614000146

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 1–9.

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 1800–1807.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, vol. abs/1409.1556, 2014.

[15] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam., "MobileNets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint, vol. arXiv:1704.04861, 2017.

[16] M. Cicero, A. Bilbily, E. Colak, T. Dowdell, K. Gray, Bruce Perampaladas, and J. Barfett, "Training and validating a deep convolutional neural network for computer-aided detection and classification of abnormalities on frontal chest radiographs," Investigative Radiology: May 2017, vol. 52, no. 5, pp. 281–287, May 2017.

[17] S. Li, H. Jiang, and W. Pang, "Joint multiple fully connected convolutional neural network with extreme learning machine for hepatocellular carcinoma nuclei grading." Comput Biol Med, no. 84, pp. 156–167, May 2017.

[18] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, and S.-H. Wan, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," Multimed Tools Appl, September 2017.

[19] T. Nishi, S. Kurogi, and K. Matsuo, "Grading fruits and vegetables using rgb-d images and convolutional neural network," in 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Nov 2017, pp. 1–6.

[20] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy et al., "Deepid-net: Deformable deep convolutional neural networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2403–2412.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[22] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4694–4702.

[23] P. Pawara, E. Okafor, O. Surinta, L. Schomaker, and M. Wiering, "Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition," in ICPRAM, 2017.