International Journal of Engineering & Technology, 7 (4.11) (2018) 90-94

International Journal of Engineering & Technology

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

Research paper



VGG16 for Plant Image Classification with Transfer Learning and Data Augmentation

Mohamad Aqib Haqmi Abas*, Nurlaila Ismail, Ahmad Ihsan Mohd Yassin, Mohd Nasir Taib

Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
**Corresponding author E-mail: Mohdaqib93@yahoo.com*

Abstract

This paper discusses the potential of applying VGG16 model architecture for plant classification. Flower images are used instead of leaves as in other plant recognition model because the structure of shape and color of leaves are similar in nature. This might be disadvantageous when we want to use only leaves images as a sole feature of plants to classify the species. Previous work has demonstrated the effectiveness of using transfer learning, dropout and data augmentation as a method to reduce overfitting problem of convolutional neural network model when applied in limited amount of images data. We have successfully build and train the VGG16 model with 2800 flower images. The model able to achieve a classification accuracy score of 96.25% for training set, 93.93% for validation set and 89.96% for testing set.

Keywords: *convolutional neural network; transfer learning, dropout; data augmentation; deep learning.*

1. Introduction

Plants are important in earth ecosystem and known as the backbone of all life on earth. They are crucial for overall balance ecology in an area. They help to give off oxygen and use carbon dioxide while they are undergoing photosynthesis process which benefits tremendously for human and our atmosphere. Good understanding of plants is crucial especially in identifying rare and new plant species [1]. Plant identification is known to be a challenging and demanding task, mainly due to the large number of plant species. The traditional plant identification method normally done by expert botanists, which involves manual measurement for feature of plants is known to be a very tedious, slow and expensive task [2]. Some features of plant that is known to be useful to identify the species is the leaves, fruits, seeds, flowers and bark.

Since the past decade, there has been an expansive development and advancement of technology that can be used to identify the plant as a faster and cheaper alternative. The advancement of deep learning, especially for computer vision benefits tremendously in identification and classification of plant species. Deep learning technique that is used for plant identification and classification was mainly done by focusing the leaves image of a plant [1, 3, 4]. However, classification of plant based on other feature or parts of the plant is also required and necessary as only having leaves images to classify a plant maybe disadvantageous due to the structure of leaves of plant that has a very similar shape and color representation in nature [5].

In the past few years, convolution neural networks (CNN) have contribute a lot in many large-scale image and video recognition. The main reason for this success is due to a great advancement of computing system such as GPUs and large public image dataset repositories such as ImageNet and Pascal VOC [6]. There are multiple benefits of using CNN over traditional neural networks, mainly they can learn spatial hierarchies of patterns and the patterns that they learn are translational invariant [7]. Both of these are important as it makes CNN to be very efficient in learning increasingly complex and abstract visual concepts and efficient when processing images.

In this paper, we propose on using pre-trained VGGnet which is a variant of CNN model to identify and classify plant species based on the flower images. Flower images are used instead of normal leaves dataset as structure of leaves of some plants have similar shape and color in nature [5]. The flower dataset used in this paper is an open source dataset that contains a total of 2800 images of 4 classes which are dandelion, daisy, sunflower and tulip. Transfer learning technique is used in this paper because deep learning requires significant amount of data to train the neural network model. Transfer learning is possible to be applied as deep learning models are highly reusable by nature, where a trained model of image classification or speech to text model trained on a large-scale dataset can be reused easily on a significant different problem with only minor fine tune changes on certain parameters [7, 8]. VGGnet model used in this paper is a pretrained model, which has been trained previously in ImageNet dataset.

Data augmentation technique and dropout is also used in the experiment during the process of building the network model to reduce the problem of overfitting [7, 9]. When there is small number of samples for the model to learn from, the model will be unable to generalize to new and unseen data. Hence, data augmentation technique is used to generate more samples for the model to learn from. It works by augmenting the existing training samples via a few random transformations. The main purpose of data augmentation is to ensure the model will not see the same picture twice during training time and expose the model to much more aspects of data and thus generalize better [7, 10]. However, the augmented samples are still heavily intercorrelated with the original samples as they are produced by transformation process on the same original images. Therefore, it is impossible to produce a new information by applying data augmentation technique.

Dropout is used to reduce overfitting, especially in large networks. The main idea of the technique is to randomly drop units from the

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

network during training. Dropout has been proven to significantly reduce overfitting and provide a huge improvements over any other regularization methods [9]. It is a general technique in neural networks and is not specific to any type of network or any domain.

2. Literature Review

Based on literatures, there have been a few studies carried out on identifying and classifying plant species using images [1–4, 11, 12]. In [4], the author proposes a CNN model to be used to identify leaf based on Foliage and Flavia dataset. The author uses dropout and transfer learning method to reduce overfitting problem of the network as the dataset size is too small for training a new network from scratch. The results from the study gives a state-of-the-art performance of above 99% classification accuracy for images from both of the dataset.

Author in [1] uses two deep learning technique as a plant identification system, which are CNN model to learn the leaf features to perform plant classification and deconvolution networks (DN) to visualize the learned feature of the model and gives visual aid and understanding to identify leaf features that are important between different classes. The classification accuracy result achieved 99.6% which outperform current conventional solutions and further investigation based on visualization shows the contributing factor of the misclassification is caused by the condition of the leaves such as insect damages and wrinkled surface.

In [2], the author proposed the use of deep learning architectures as a method for leaf counting from direct RGB image source without image segmentation. This is due to the traditional plant phenotyping method that involves manual measurement of features of plant is very tedious, slow and expensive. Using deep learning allow to achieve a satisfactory level of accuracy that can reduce the cost in the future by having fewer field technicians for manual measurement each year. The results demonstrated by the author shows that the architecture proposed are successfully able to learn and estimate correctly the number of leaves without having any prior knowledge on the species or surrounding of the plant.

VGGnet was introduced in [13] as a mean to improve the CNN architecture. The author address the important aspect to be used in previous CNN design, which is the depth of the network. There are two variants of the model architecture proposed, which are with depth of 16 weight layers and 19 weight layers. The results demonstrated in the paper shows that the model proposed able to generalize well with a wide range of tasks and datasets. The author also made both of the model architecture trained on large dataset to be publicly available for any future research among the computer vision community. The pre-trained architecture can be reused in the form of transfer learning.

Transfer learning [8] has been widely used among deep learning practitioners and researchers as it reduces the needs on having a large dataset and computation power to train the model from scratch. Transfer learning also act as a need to reduce the possibility of overfitting for the model as any neural network model that is trained on limited data will easily get overfit. Dropout [9] and data augmentation [10] is also some of the widely adopted technique used in training deep learning model architecture, especially in computer vision field. Both of these methods have proven to be exceptionally effective to counter overfitting problems of neural network model.

3. Methodology

This experiment uses Python programming software and libraries mainly Keras with Tensorflow backend and scikit-learn. Most of the computations workload performed from this software is offloaded to graphics processing unit (GPU) instead of central processing unit (CPU) because deep learning model heavily benefits the use of GPU as it significantly reduces the computation time. The GPU used for the experiment in this paper is GTX 1080 Ti.

The flower image dataset used is an open source dataset that contains a total of 2800 images. The dataset is split with ratio of 60% for training set, 20% for validation set and 20% for testing set. The images consist of 4 classes of flowers which are dandelion, daisy, sunflower and tulip. Before feeding the model architecture with the dataset, the images will first undergo data augmentation process that applies transformation to the images which involves rotation, shifting, resizing, zooming and rescaling. The parameters used for data augmentation transformation is 20 degrees of rotation range, horizontal flip and mirroring of image and 0.2 fraction of shifting (both height and width), shearing range and zoom range. Data augmentation is done to artificially boost the size of training set. Thus, helps the model to be exposed with more aspects of the data and generalize better.

VGGnet with 16 layers (VGG16) is used as the model architecture for this paper. Transfer learning method is used get the pre-trained configuration of the architecture from ImageNet dataset. Having an architecture model trained on large image dataset such as ImageNet is advantageous as it contains 1.4 million labelled images with 1000 different classes. Pre-trained network with such large and general data enables it to be effectively act as generic model of the visual world as the spatial hierarchy of features learned from the dataset is huge.

Normal VGG16 consists of 5 convolutional blocks before it is connected to multilayer perceptron (MLP) classifier. In this paper, the first 3 convolutional blocks in VGG16 is preserved from previously learned while the fourth and fifth convolutional block is trained based on our flower image dataset. This process is called fine-tuning and it is required when training the pre-trained model to a data with different representation that the original trained data. This is because it allows the model architecture to slightly adjust the abstract representations of model to make them more relevant for the current new dataset. In a pre-trained model architecture, the early and bottom layers in convolutional block only encode a generic and reusable features while higher up layers encode specialized features based on the trained dataset. Thus, the idea of retraining the fourth and fifth convolutional block in VGG16 is to enable the model to fine-tune to a more specialized features in our dataset.

Next, it is connected to three MLP layers which consists of two hidden layers and one output layer. The first hidden layer consists of 128 nodes, while the second hidden layer consists of 64 nodes. Both hidden layers use 40% dropout rate and rectified linear unit (ReLU) activation function. The output layer consists of 4 nodes which directly represents the 4 classes and with softmax activation function. The model architecture is train with stochastic gradient descent (SGD) optimizer with learning rate of 5e-3 (0.005) and a total of 40 epochs. The batch size for all 3 training, validation and testing set is 20 samples. The steps for each epoch perform on validation and testing set is 28 times to complete 560 samples for each of the set, while for training set, the steps per epoch required to completely train all 1680 samples is with 84 steps.

The model architecture is configured to reduce the learning rate by factor of (0.5) after the performance of validation loss plateau for 2 epochs and to stop training the model for further epoch if the validation accuracy of the model does not improve in 3 epochs iteration.

The model's classification accuracy score and loss score for testing and validation set is checked for every epoch performed. The fundamental idea of having loss score in deep learning is to use this score as a feedback signal to adjust the value of weights of the model in order to minimize the loss score in next future train epoch. Validation set is required to act as an indicator of the generalizability performance of the model to a new and unseen data. Having a model with high accuracy score in training set but poor accuracy score performance in validation set indicates overfitting, where the model is over-optimizing its learning representation specific to the training set data.

The classification accuracy score formula used is as follow.

$$classification\ accuracy = \frac{correctly\ predicted\ samples}{total\ samples\ in\ the\ set} \quad (1)$$

Other than classification accuracy score, this experiment also used confusion matrix based performance measure score such as precision, recall and F_1 measure score. The formula for precision and recall are given as follow [14]

$$precision_i = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l}(TP_i + FP_i)} \quad (2)$$

and

$$recall_i = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l}(TP_i + FN_i)} \quad (3)$$

Precision score indicates the measure of how many samples predicted positive (the class with index i) by the model is truly positive, while recall score indicates how many number of truly positive samples that has been successfully predicted as positive by the model. However, in classification task of machine learning problems, there will always be a trade-off between optimization of precision score and optimization of recall score. Thus, F_1 measure score is usually used together with precision and recall score as it acts as a harmonic mean and provide a more proper and balance score indicator [15]. The formula of F_1 score is given as

$$F_1\ score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (4)$$

The final step of the experiment is to make an assessment of the trained model based on the results obtained from the accuracy score and the confusion-matrix based performance measure, whether the model is viable to be used to classify future plant species based on the flower images.

4. Results and Discussion

Images in Figure 1 shows some of the images generated by using the data augmentation parameters as stated as in experimental methodology. The batches of data generated will be looped over in batches randomly with the variation parameters used.

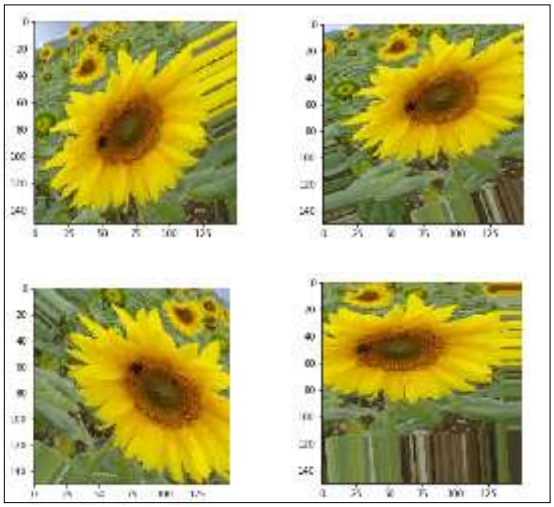

Fig. 1: Images generated using data augmentation configuration

Figure 2 shows the training set accuracy score and validation set accuracy score for epoch of 1 to 29. Note that even though the model is configured to train to 40 epochs, it stopped at 29 epochs as the validation accuracy score (performance monitored) stopped improving for 3 epochs. The training set accuracy score at epoch 29 reach 96.25%, while the validation set accuracy score at epoch 29 reach 93.93%. It can be seen that for earlier epochs the validation set accuracy score performed better than training set accuracy score. However, after the 17th epoch the scores begins to converge and training set accuracy begins to increase and performs better. Overall, the scores indicate a very good image classification model performance.

Figure 3 shows the training set loss score and validation set loss score for the model from epoch 1 to 29. The pattern for the loss score is similar to the accuracy score in Figure 2. In earlier epoch, validation set has a much lower loss score compared to training set. However, after 17th epoch, the loss score of training set is lower compared to validation set. After 22nd epoch, there is not much improvement for the validation set loss score while the loss score of training set keeps on decreasing slowly. If the pattern continues, it might indicate the model architecture to overfit with the training set data during learning.

Figure 4 shows the confusion matrix score of the training set. In this experiment, the classes are represented with number of 0, 1, 2 and 3. Number 0 represents dandelion, 1 represents daisy, 2 represents sunflower and 3 represents tulip class. From Figure 4, it can be seen that the model accurately predicted the class of dandelion and daisy by same 401 images and class of sunflower and tulip by same 409 images. The actual (true) images for each class is 420 images. The highest wrongly predicted is when the images is daisy but predicted as tulip for 10 images. There is also 0 images predicted as daisy when it is sunflower. This shows the model have learned to distinguish the pattern in the representation of sunflower and daisy when making prediction for sunflower class.

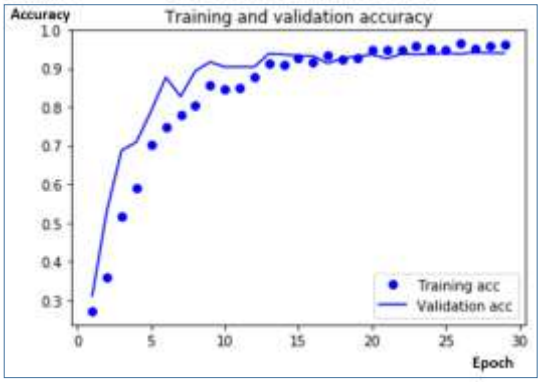

Fig. 2: Classification accuracy score for training and validation set

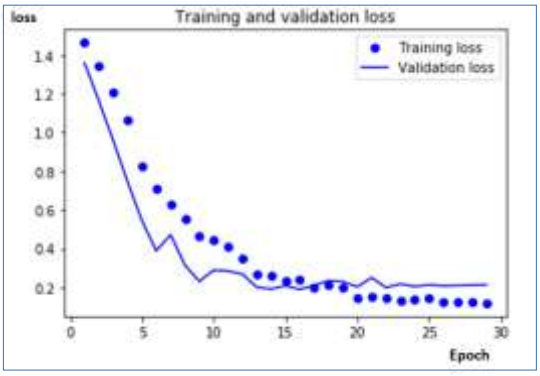

Fig. 3: Loss score for training and validation set

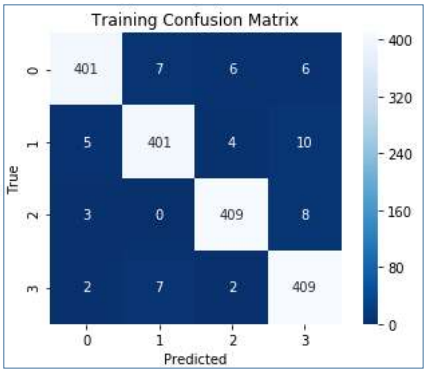

Fig. 4: Confusion matrix for training set

The confusion matrix for validation set is shown in Figure 5. The best successful prediction made is with the daisy class, which is 136 images and the lowest is sunflower class with only 127 images. The highest wrong class prediction is when actual class is sunflower but the model wrongly predicted as dandelion by 10 images. Overall from the prediction made on validation set, it shows the model are able to achieve good classification performance of 93.9% score.

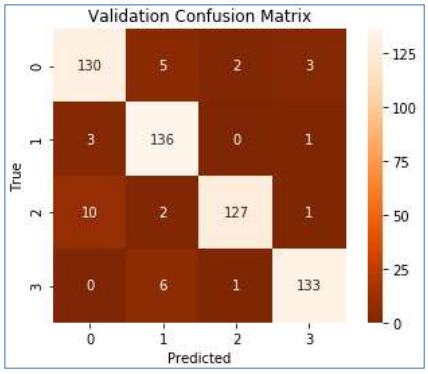

Fig. 5: Confusion matrix for validation set

The confusion matrix on testing set is shown in Figure 6. The best classification made is for tulip class with 130 images while the worse is daisy class with only 115 images. The highest misclassification is when the model are required to predict daisy class, but wrongly predicted as dandelion by 11 images and tulip by 11 images. Overall, the classification accuracy of the model made in predicting the testing set achieve 89.6%.

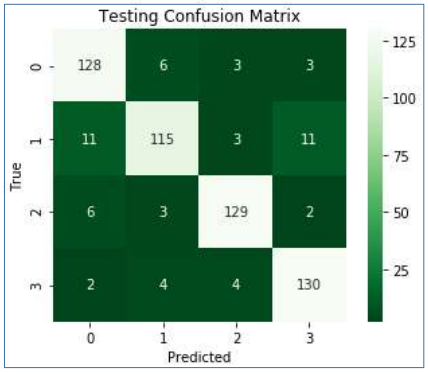

Fig. 6: Confusion matrix for testing set

Table 1 tabulates the precision, recall and F_1 score of the classification model on testing set. For precision score, the highest is score is by sunflower class with score of 0.93, followed by daisy with 0.90, tulip with 0.89 and lastly dandelion class with only 0.87. Precision score indicates how many sample of images that are predicted by the model are the actual class. The recall score of the model have the highest score for tulip class which is 0.93, followed by sunflower at 0.92, dandelion with score of 0.91 and lastly daisy class with only 0.82. The recall score shows how many of the actual sample images have been correctly predicted by the model. Lastly, for the F_1 measure score that acts as a harmonic mean of both precision and recall have the highest score for sunflower class with score of 0.92, then tulip with 0.91, followed by dandelion with 0.89 and lastly daisy class with 0.86. The average of all these 3 scores shows an average score of 0.90, which indicates a very good classification model.

Table 1: Precision, recall and F_1 measure score of the model

	Precision	Recall	F_1 Score	Samples
Dandelion	0.87	0.91	0.89	140
Daisy	0.90	0.82	0.86	140
Sunflower	0.93	0.92	0.92	140
Tulip	0.89	0.93	0.91	140
Average/Total samples	0.90	0.90	0.90	560

5. Conclusion

In this paper, we have demonstrated that given an adequate amount of data, our model are able to learn to classify the species of plant accurately from the flower images using transfer learning and fine tuning on VGG16 with dropout and data augmentation. From the experimental results, we justified that our model trained with 29 epochs is able to achieve high classification accuracy of all 4 classes. For future work, the current model architecture will be compared in terms of performance with advanced convolutional neural network methods.

Acknowledgement

The authors' appreciation for all ASPRG group of Faculty of Electrical Engineering, UiTM Shah Alam as well as BESTARI grant code no. 600-IRMI/DANA 5/3/BESTARI (054/2017) awarded to Dr Nurlaila Ismail for financial support.

References

[1] S. H. Lee, C. S. Chan, P. Wilkin, & P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," *Proceedings of the IEEE International Conference on Image Processing*, pp. 452-456, 2015.

[2] S. Aich, & I. Stavness, "Leaf counting with deep convolutional and deconvolutional networks," *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 22-29, 2017.

[3] I. Heredia, "Large-scale plant classification with deep neural networks," *Proceedings of the Comput. Front. Conf.*, pp. 259–262, 2017.

[4] C. Wick, & F. Puppe, "Leaf identification using a deep convolutional neural network," 2017, https://arxiv.org/pdf/1712.00967.pdf.

[5] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, & J. VB Soares, "Leafsnap: A computer vision system for automatic plant species identification," *Proceedings of the Computer Vision ECCV*, pp. 502–516, 2012.

[6] Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, & A. Y. Ng, "Large scale distributed deep networks," *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 1223–1231, 2012.

[7] F. Chollet, *Deep learning with Python*. Manning Publications, 2017.

[8] L. Torrey, & J. Shavlik, "Transfer learning," in E. Olivas, J. Guerrero, M. Martinez-Sober, J. Magdalena-Benedito, & A. Serrano López (Eds.), Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. Pennsylvania: IGI Global, pp. 242-264, 2010.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, 15, 1929–1958, 2014.

[10] L. Perez and J. Wang, "The effectiveness of data augmentation in

image classification using deep learning," 2017, https://arxiv.org/pdf/1712.04621.pdf.
[11] S. Aich, A. Josuttes, I. Ovsyannikov, K. Strueby, I. Ahmed, H. S. Duddu, C. Pozniak, S. Shirtliffe, & I. Stavness, "DeepWheat: Estimating phenotypic traits from crop images with deep learning," *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 323-332, 2017.
[12] M. Sadeghi, A. Zakerolhosseini, & A. Sonboli, "Architecture-based classification of plant leaf images," 2018, https://arxiv.org/ftp/arxiv/papers/1801/1801.02121.pdf.
[13] K. Simonyan, & A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/pdf/1409.1556.pdf%20http://arxiv.org/abs/1409.1556.pdf.
[14] M. Sokolova, & G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, 45(4), 427–437, 2009.
[15] J. D. Kelleher, B. Mac Namee, & A. D'Arcy, *Fundamentals of machine learning for predictive data analytics*, MIT Press, 2015.