Improved Image Network with Uniform Distribution for Crop Classification

[1] *De-Yan Lu* (盧德晏), [2] *CS Fuh* (傅楸善)

[1] Institute of Communication Engineering,
National Taiwan University, Taipei, Taiwan,
*E-mail: qe59979022@gmail.com

[2] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
*E-mail: fuh@csie.ntu.edu.tw

ABSTRACT

In recent years, deep learning is the fastest-growing field of machine learning. Image classification is a common problem in deep learning. Among all deep neural networks, the Convolution Neural Networks (CNN) are the main tools for image classification such as AlexNet, Google-NETs, ResNet, EfficientNet, and so on. However, there are some challenging problems to tackle for image classification. In this paper, we focus on agriculture crops for classification tasks. To encourage further progress in challenging realistic agricultural conditions, we present the crop species dataset, consisting of 40,000 training data, 12000 valid data and 20,000 testing data from 14 classes. The classes include 13 types of crops and 1 types of conditions of farms. The images were collected with different situations and tools, providing a stronger benchmark for the deep learning networks. To further improve the accuracy for agricultural classification, we propose some technique about improved data augmentation, state of the art deep learning classification based on different deep neural network models. Furthermore, we can combine state of the art network with linear combination, using different weights.

Keywords: *Data Augmentation, Convolutional Neural Network, Agriculture, Image Classification, EfficientNet*

1. INTRODUCTION

AI has been considered as the further enabler and the important technological challenge in classifying the types of crops and ecological value of the whole agriculture. Thus, the development of deep learning technology has provided an effective approach to facilitating intelligent management and decision-making in many aspects of agriculture, such as visual crop categorization [1] and real-time plant disease and pest recognition [2]. Moreover, there is increasing agricultural success present in the near future, because deep-learning systems can easily take advantage of data increases in the number of available sensors, cameras, and smartphones. Inspired by the multi-level visual perception process of human brain, deep-learning allows computational models comprising multiple processing layers to learn representations of data with multiple levels of abstraction, obtained by non-linear modules (such as convolutional layers or memory units) that each transforms the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned, and tough structures in high-dimensional data can be discovered automatically to complete agricultural tasks.

Image classification of crops is a popular problem in the deep learning. In 2012, AlexNet [3] is a network with a deeper neural network for better performance. It consisted of 11x11, 5x5, and 3x3 convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD (Stochastic Gradient Descent) with momentum. It attaches ReLU activations after every convolutional and fully-connected layer.

In 2014, there was a competition held in ILSVRC (ImageNet Large Scale Visual Recognition Challenge), which is a race about deep learning. The winner of the competition was GoogleNet [4] from Google. The important key of GoogleNet is Inception architecture which is a combination of 1x1 convolution layer Their architecture consisted of a 22-layer deep CNN but reduced the number of parameters from 60 millions (AlexNet) to 4 millions. Briefly speaking, GoogleNet is deeper than AlexNet and specially featured as the design of Inception architecture. Therefore, the performance was outstanding in deep learning.

However, researchers were still not satisfied with current situations. They were eager to anticipate achieving better performances than GoogleNet. Therefore, Residue Net (ResNet) [5, 6] had flourished and attained the best performance in the issue of classification. At the ILSVRC 2015, the ResNet by Kaiming He et al. introduced a novel architecture with "skip connections" and features heavy batch normalization. Such skip connections are also known as gated units or gated recurrent units and have a strong similarity to recent successful elements applied in RNNs. Thanks to this technique, they were able to train a NN with 152 layers while still having lower complexity than GoogleNet..

In 2019, Google researchers have proposed a new network for boosting performance and reducing the parameters in training process, known as EfficientNet [7, 8]. As the size of network is increasing and the accuracy would be saturated in one dimension, EfficientNet used the scaling of three dimension, including depth, width and resolution of images. It can not only boost the accuracy of networks, but also reduce the complexity in training process.

In this paper, we will propose improved architecture based on the Efficient net including data augmentation, feature analysis, and other tips such as weighting sum of difference neural networks for enhancing performances. Weighting sum can be viewed as the linear combination type, and we will expect that it can combine other properties of neural network for better performance. Not only can preserve the attribute of any network, but also get different information from different networks. We think it can attain better performance. The overall steps include the pre-processing, neural network building, and post-processing. First, we need to cope with the dataset based on an appropriate analysis such as feature analysis or data augmentation. Second, building neural network and fine-tune parameters is an important task in our steps, related to the performance of the final test. Last, we can do something special like the linear combination of different augmentation, given any ratio λ to enhance the performance more.

The rest of the paper is organized as follows. Section 2 describes the related dataset to this study; Section 3 introduces the deep-learning classification network selected and our proposed method. Section 4 presents experimental results and discussion of the dataset to verify the application performance. Finally, we present our conclusion with further research aims in Section 5.

2. RELATED WORK

2.1. Transfer learning

Transfer learning or domain adaptation is an extensively studied topic [9]. Transfer learning is widely used for deep learning when a target task is short of labelled data. The most common deep transfer learning strategy is fine-tuning [10]: first train a base network using a large source data and then copy the first *n* layers to the corresponding layers of the target network, followed by randomly initializing the remaining layers and finally fine-tune only them or all layers. A systematic study is presented in [10] which examines how transferable features of different layers are between the source and target domains. It concludes that the generalization ability diminishes when the discrepancy between the source and target tasks increases.

Limited by our resource, it costs much time to train a large set of unknown data at beginning. According to transfer learning, we can fine-tune for multi-classification task, because the source data and target data are both labeled.

2.1. Focal loss

Initial goal of focal loss function proposed by Lin et al. [11] is to address the problem of extreme balance between foreground and background classes during training in object detection scenarios. Focal loss is mainly used for object detection, but we also show that its sparse-specific characteristics are also applicable for classification problem with imbalanced dataset. The focal loss is defined as follows

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{1}$$

Note that α_t and γ are two parameters indicating how sensitive it is to the easily classified samples and $p_t \in [0, 1]$ indicates the model's estimated probability for each class. In this work, we propose to apply this focal loss function to the end of our proposed model.

3. PROPOSED MODEL

3.1. Overview architecture

The overall flowchart can be viewed in Fig. 1 including preprocessing, building the model, transfer learning and post-processing. The reason why we need to preprocess is that dataset may be imbalanced or machine requires to be adapted to different view of images such as luminance, position or perspective, known as data augmentation. And the reason why we use transfer learning to build our model is that our resource is limited by our computer, the number of dataset is fewer, and it can saving a great deal of time in training process. Last, post-processing is a technique with boosting the final performance of our model including ensemble learning or Test-Time Augmentation (TTA) [12, 13]. Here we use ensemble learning and TTA simultaneously.

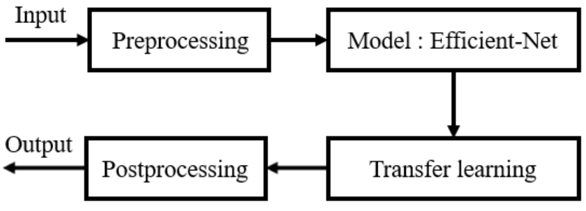


Fig. 1 The flowchart of our proposed method.

3.2. Preprocessing

First, we pre-process the image as our input data. To make the model more adapted to the situations in the real world, we generate many images by a series of transformation such as rotation, normalization, flip, affine, Gaussian blur, and so on. Data augmentation are shown in Fig. 2.

The series of transformation is also known as augmentation in deep learning. The most importance is that we use the uniform distribution to sample our dataset. Regarding dataset as random filed, we sample the dataset by transforming dataset into the field of uniform distribution. The technique known as removing imbalance effect often copes with imbalanced dataset. If the number of one of the classes is much smaller than the number of other classes, we need to sample them equivalently so as not to sample them bipolarly. Uniform distribution can be written as follows

$$P_{label}(i) = \begin{cases} \frac{1}{n(i)} & , \ i \in Classes \\ 0 & , \ \text{otherwise} \end{cases} \quad (2)$$

$$Sampler(i) = \sum_{i=1}^{number\ of\ class} P_{label}(i) \cdot D(i) \quad (3)$$

where P_{label}, n, D are the probability of each label, the number of each label and the number of dataset in training process respectively. The schematic diagram of removing imbalance effect can be described in Fig. 3.

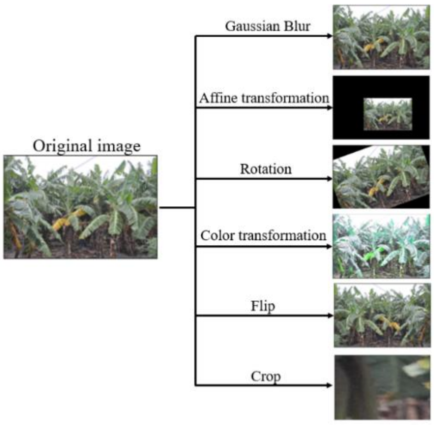


Fig. 2 Preprocessing with image processing.

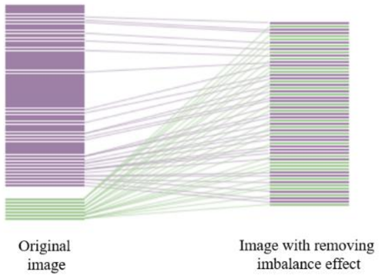


Fig. 3 Preprocessing with removing imbalance effect.

3.3. Model building

Second, we can use EfficientNet by transfer learning for saving time and getting good performance by fine tuning. Scaling network depth is the most common way used by many ConvNets. The intuition is that deeper ConvNet can capture richer and more complex features, and generalize well on new tasks. However, deeper networks are also more difficult to train due to the vanishing gradient problem. Although several techniques, such as skip connections and batch normalization, alleviate the training problem, the accuracy gain of very deep network diminish. The scaling depth can be shown in Fig. 5a.

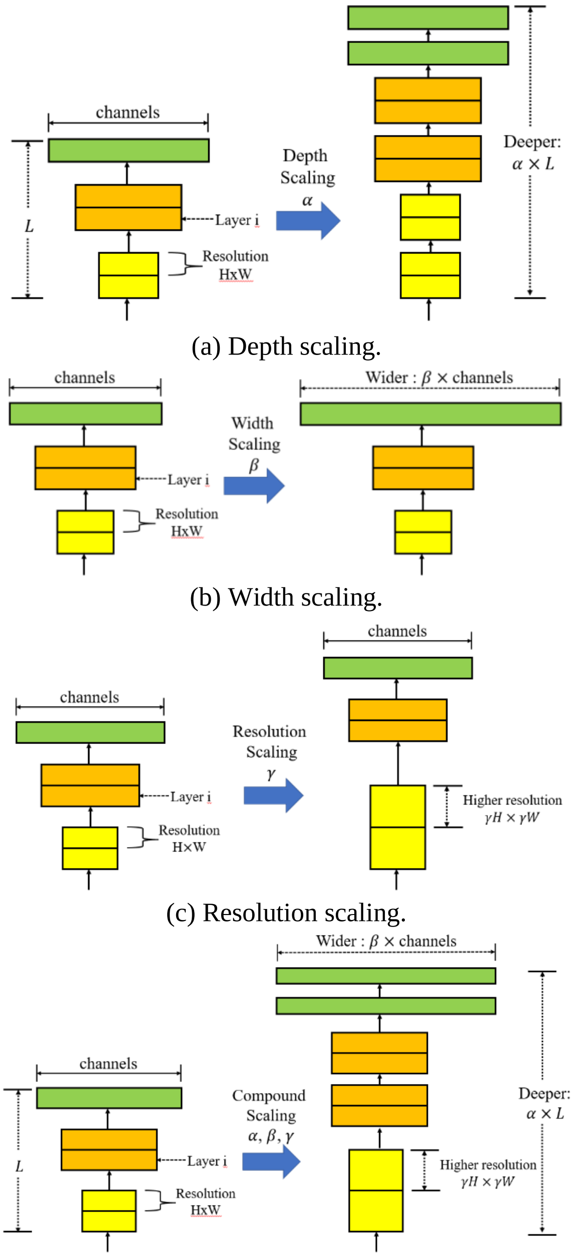
Scaling network width is commonly used for small size mode. Wider networks tend to be able to capture more fine-grained features and are easier to train. However, extremely wide but shallow networks tend to have difficulties in capturing higher level features. The accuracy quickly saturates when networks become much wider with larger. The scaling width can be shown in Fig. 5b.

With higher resolution input images, ConvNets can potentially capture more fine-grained patterns. Starting from 224x224 pixels in early ConvNets, modern ConvNets tend to use 299x299 or 331x331 pixels for better accuracy. Higher resolutions, such as 600x600 pixels, are also widely used in object detection ConvNets, which are introduced in [14, 15]. The results of scaling network resolutions are indeed higher resolutions improve accuracy, but the accuracy gain diminishes for very high resolutions. The scaling resolution can be shown in Fig. 5c.

We empirically observe that different scaling dimensions are not independent. Intuitively, for higher resolution images, we should increase network depth, such that the larger receptive fields can help capture similar features that include more pixels in bigger images. Correspondingly, we should also increase network width when resolution is higher, in order to capture more fine-grained patterns with more pixels in high resolution images. These intuitions suggest that we need to coordinate and balance different scaling dimensions rather than conventional single-dimension scaling. In order to pursue better accuracy and efficiency, it is critical to balance all dimensions of network width, depth, and resolution during ConvNet scaling. In fact, a few prior work have already tried to arbitrarily balance network width and depth, but they all require tedious manual tuning. The following equation (3) is compound scaling method, which use a compound coefficient φ to uniformly scale network width, depth, and resolution in a principled way:

$$\begin{gathered} d = \alpha^{\varphi} \\ w = \beta^{\varphi} \\ r = \gamma^{\varphi} \\ \text{s.t.} \quad \alpha\beta^2\gamma^2 \cong 2\,, \ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{gathered} \quad (4)$$

where α, β, γ are constants that can be determined by a small grid search. Intuitively, φ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution respectively. The compound scaling is shown in Fig. 5d.



(a) Depth scaling.

(b) Width scaling.

(c) Resolution scaling.

(d) Compound scaling.

Fig. 5 The architecture of Efficient net.

3.4. Post-processing

In image classification, while we are going to predict classes for our test set images after training our model, we will generate the confident probability for each test image for n number of times and finally we will assign the max average value among all the prediction classes to the image. This is so called test time augmentation (TTA).

Test-time augmentation (TTA) is popular because it is easy to use. It is simple to put into images with training augmentation and test augmentation, makes no change to the underlying model, and requires no additional data. Typically, test-time augmentation methods aggregate model predictions by averaging.

The overall flowchart of TTA is described in Fig. 6. α is the coefficient which tune the weight between test augmentation and train augmentation. Usually, we set α Larger than 0.5 because the dominant inference is determined in the test time phase.

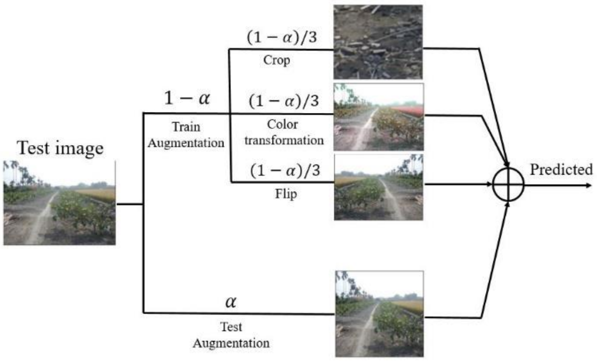


Fig. 6 Post-processing with TTA.

Although TTA can boost the better performance in the test time phase, the condition for further boosting performance is that we need to build much different training augmentation. It will cost much time to infer the performance and even has limit to enhance the accuracy if the number of training augmentation is fewer. The reason why TTA could not further boost the performance is that data distribution is random distribution and may be too large variance to concentrate the accuracy.

Thus, ensemble learning has been proposed to cope with the issues. Neural network models are nonlinear and have a high variance, which can be frustrating when preparing a final model for making predictions. A successful approach to reducing the variance of neural network models is to train multiple models instead of a single model and to combine the predictions from these models. This is called ensemble learning and not only reduces the variance of predictions but also can result in predictions that are better than any single model.

The ensemble learning, introduced in [16] involves multiple models combined in some fashion like averaging, voting such that the ensemble model is better than any of the individual models. To prove that average voting in an ensemble is better than individual model, Marquis de Condorcet proposed a theorem wherein he proved that if the probability of each voter being correct is above 0.5 and the voters are independent, then addition of more voters increases the probability of majority vote being correct until it approaches one [17]. Although Marquis de Condorcet proposed this theorem in the field of political science and had no idea of the field of

Machine learning, but it is the similar mechanism that leads to better performance of the ensemble models. Here we use average models for our final prediction with ensemble learning. The overall flowchart of ensemble learning can be described in Fig. 7.

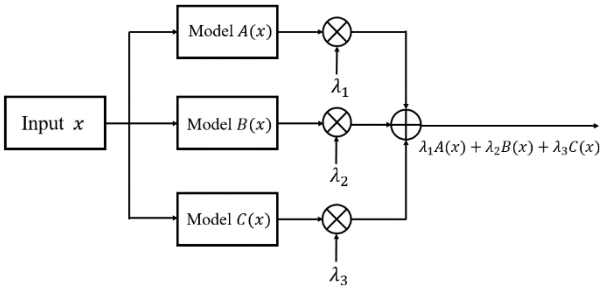


Fig. 7 Post-processing with ensemble learning.

Here, we take three training models for example in Fig. 7. Here we trained three different models with different augmentation, loss function or learning rate schedule. If the training steps are similar to each other, ensemble learning does not have an obvious effect to reduce the variance of predictions for better performances. Thus, the final predictions are the average of three predictions from those models. Here we set the weight for averaging as follows

$$\lambda_1 = \lambda_2 = \lambda_3$$
$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{5}$$

4. SIMULATION AND RESULTS

4.1. Dataset

Our crops dataset collected from AIDEA includes the banana, bareland, carrot, corn, dragonfruit, garlic, guava, peanut, pineapple, pumpkin, rice, soybean, sugarcane and tomato. All kinds of crops are composed of 14 classes. 13 classes are vegetables or fruits and 1 class is the status of the farmland, i.e. bareland. The kinds of dataset can be viewed in Fig. 8.

For our agriculture datasets consisting of vegetables and fruits, they were featured as various types of fruit and situations such as weather, luminance, and so on. According to these factors, it was very hard to differentiate different kinds of situations for the same crops by neural networks. Currently, the dataset covers common vegetables and fruits of 14 categories including 13 classes of fruits and vegetable and 1 kind of situations of farms, which are collected by visual cameras of IoT (Internet of Things), autonomous robots, and smartphones in greenhouses. It contains more than 40,000 training data which had been labeled, 12,000 valid data for validation in training process and at least 20,000 testing data which had not been labeled. The size of each image is at least one megabytes. Thus, the data is so large that it would cost much time in training process. Training one epoch would cost about one hour.

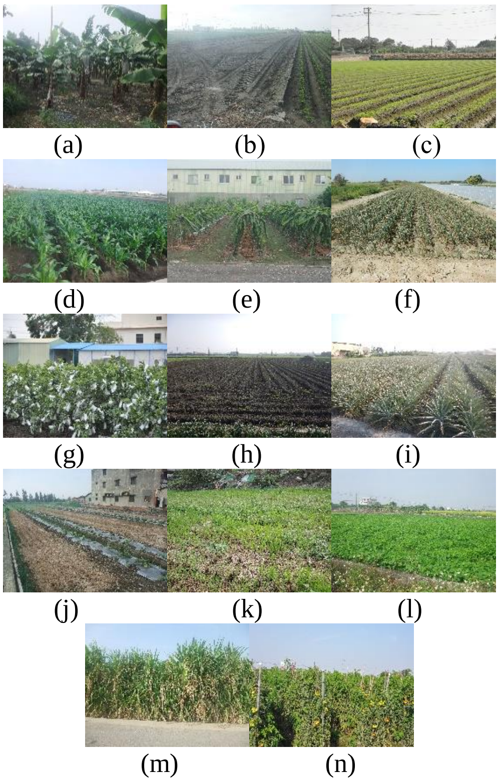


Fig. 8 Dataset for all classes. (a) banana (b) bareland (c) carrot (d) corn (e) dragonfruit (f) garlic (g) guava (h) peanut (i) pineapple (j) pumpkin (k) rice (l) soybean (m) sugarcane (n) tomato.

4.2. Metrics for inference

According to our proposed methods, we expect the loss would be small as much as possible and performance further better. First, we introduce the precision and recall. Precision should ideally be 1 for a good model. Precision becomes 1 only when the numerator and denominator are equal, i.e. $TP = TP + FP$. This also means FP is zero. As FP increases, the value of denominator becomes greater than the numerator and the value of precision will decrease.

Another metric is recall. Recall should ideally be 1 for a good model. Recall becomes 1 only when the numerator and denominator are equal, i.e. $TP = TP + FN$. This also means FN is zero. As FN increases, the value of denominator becomes greater than the numerator and the value of recall will decrease. Equation 6 and 7 are calculation of precision and recall respectively

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

where TP, FP, and FN are true positive, false positive, and false negative respectively. These indexes can be evaluated from confusion matrix measuring the difference of predicted value and ground truth. Confusion matrix can be described in Fig. 9 with 14 classes.

$$C = \begin{array}{cc} \text{Predicted} & \\ \begin{bmatrix} c_{1,1} & \cdots & c_{1,14} \\ \vdots & \ddots & \vdots \\ c_{14,1} & \cdots & c_{14,14} \end{bmatrix} & \begin{array}{c}\text{Ground}\\ \text{Truth}\end{array} \end{array}$$

Fig. 9 Confusion matrix with 14 classes.

where $\dim(C) \in \mathbb{R}^{14\times 14}$ because of 14 classes. Therefore, *TF*, *FP*, and *FN* for each class can be calculated from the confusion matrix C described as vector as follows

$$TP = diag(C) \tag{8}$$

$$FP = \sum_{i=1}^{14} c_{i,j} - TP \tag{9}$$

$$FN = \sum_{j=1}^{14} c_{i,j} - TP \tag{10}$$

For the multi-image classification, the performance cannot be evaluated only by precision or recall because they are usually trade-off. Thus for a good model, we want to consider both precision and recall to be one which also means *FP* and *FN* are zero. F1-score is a metric considering both precision and recall. It can be viewed as a good model for the task of multi-class if F1-score would be usually larger than 0.7.

The important metric for our evaluation is Weight Precision (WP). Weight Precision (WP) is usually applied into the task of multi-classification. F1-score and weight precision are defined in equation (11) and (12) respectively.

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

$$\text{WP} = \frac{\sum_{i=1}^{num\ of\ classes}\left(precision_i \times (TP_i + FN_i)\right)}{Total\ image\ counts} \tag{12}$$

4.3. Simulation and results

Epoch for training we set is 20. The optimizer we use is AdamW, introduced in [18], the loss function we use is cross-entropy or focal loss and the compile environment we use is Google Colab. In training process, we introduce the learning rate scheduler to seek to best local minimum point during the backward propagation.

The results of F1-score for the task of multi-classification are shown in Table 1 and Table 2 with imbalance effect and imbalance removal respectively. With the same model and parameter settings, our proposed method can cope with imbalance effect in training process indeed.

Last, the results of Weight Precision (WP) are shown in Table 3. The higher the WP is, the better the model we built is. In Table 3, our proposed method is higher than other models.

Table 1. F1-score of validation set and accuracy of test set for each class with imbalance effect.

Class	F1-score	Accuracy
Banana	0.987	99.3%
Bareland	0.988	100.0%
Carrot	0.945	92.0%
Corn	0.985	98.0%
Dragonfruit	0.969	97.8%
Garlic	0.978	98.3%
Guava	0.984	98.0%
Peanut	0.958	95.1%
Pineapple	0.994	99.5%
Pumpkin	0.939	94.3%
Rice	0.990	98.7%
Soybean	0.917	91.6%
Sugarcane	0.981	97.8%
Tomato	0.941	97.1%

Table 2. F1-score of validation set and accuracy of test set for each class with removal of imbalance effect.

Class	F1-score	Accuracy
Banana	0.993	99.4%
Bareland	0.992	99.0%
Carrot	0.953	97.2%
Corn	0.989	98.2%
Dragonfruit	0.985	98.4%
Garlic	0.984	98.0%
Guava	0.991	98.7%
Peanut	0.970	96.7%
Pineapple	0.995	99.5%
Pumpkin	0.974	97.6%
Rice	0.995	99.3%
Soybean	0.941	93.8%
Sugarcane	0.985	98.6%
Tomato	0.969	98.4%

Table 3. Weight precision.

Method	WP
Resnet18	0.975
Efficient net with imbalance effect	0.979
Efficient net with TTA removal of imbalance effect	0.981
Proposed method	0.988

Weight Precision (WP) and loss are plotted in training process and validation process with Fig. 10 and Fig. 11 respectively. Validation process is used to observe whether it is overfitting in training process. From the plot, our proposed method is better than other methods.

Weight Precision for different methods in Table 3 is shown in Fig. 12. In Fig. 12, we can know that our proposed method is better than other method in validation process.

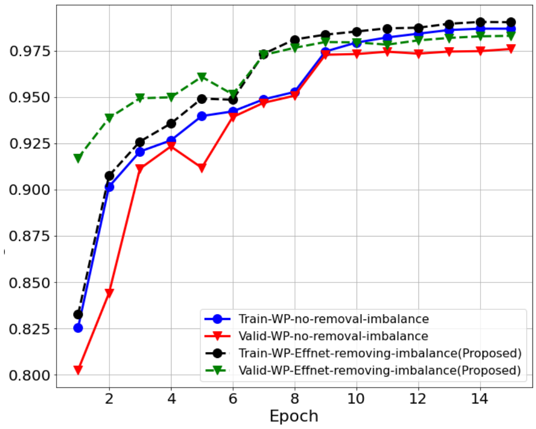


Fig. 10 Weight precision in training process.

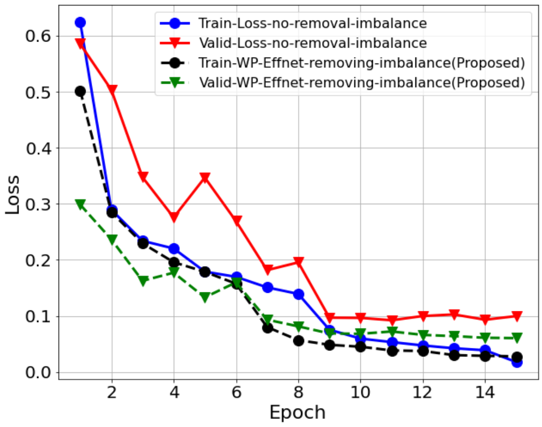


Fig. 11 Loss in training process.

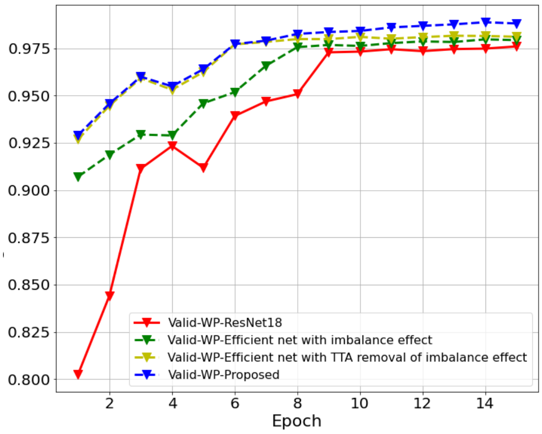


Fig. 12 WP for different methods in valid process.

5. CONCLUSION

In the real world, we often encounter with imbalance dataset in the task of multi-classification. Removing imbalance effect is an important preprocessing task before training our model. Thus, we propose a method which make the dataset with uniform distribution. The fundamental concept of our proposed method is that making small data sampling with lower probability and large data sampling with higher probability. With that we can get better performance in the phase of inference (currently ranking 29th among 151 teams).

Because of limited resource, it is impossible train a dataset with non-pre-trained model. Thus, transfer learning is a good technique without retraining the model and we can use the pre-trained weight for fast convergence of performance. However, the performance is not growing better as the network is more complicated. We must choose the appropriate model carefully to avoid overfitting.

Ensemble learning and TTA are most commonly used in post-processing. Especially for ensemble learning, it can reduce the influence of variance of dataset. With combining above techniques, the performance will be further enhanced.

6. FUTURE WORK

In our experiments, one of the most challenging task is that it costs one hour to train our neural networks in one epoch because of our limited resources. To accelerate our training process, we need to have more resources to speed up our training process. Another challenging task is that our RAM has a limitation in Google Colab. Google Colab only provides 128G RAM to train our network. Thus, there is limited capacity of our training data.

In the future, we would concentrate on Transformers such as Vision Transformers, Swin Transformer or vision transformers, introduced in [19, 20]. As Natural Language Programming (NLP) is growing fast, the transformers have been proposed with self-attention mechanism. This is popular in recent years.

Vision Transformers use multi-head attention mechanisms as the main building block to derive long-range contextual relation between pixels in images. First, images under analysis are divided into patches, then converted into sequence by flattening and embedding. To keep information about the position, embedding position is added to these patches. Then, the resulting sequence is fed to several multi-head attention layers for generating the final representation. At the classification stage, the first token sequence is fed to a soft-max classification layer.

Large Vision Transformers model attains state-of-the-art performance on multiple popular benchmarks, including 88.55% top-1 accuracy on ImageNet and 99.50% on CIFAR-10. Vision Transformers also performs well on the_cleaned-up version of the ImageNet evaluations set “ImageNet-Real”, attaining 90.72% top-1 accuracy. Finally, Vision Transformers works well on diverse tasks, even with few training data points. For example, on the VTAB-1k suite (19 tasks with 1,000 data points each), Vision Transformers attain 77.63%, significantly ahead of the single-model state of the art (SOTA) (76.3%), and even matching SOTA attained by an ensemble of multiple models (77.6%).

Swin Transformers is a type of Vision Transformer. It builds hierarchical feature maps by merging image patches in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window. It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. In contrast, previous vision Transformers produce feature maps of a single low- resolution and have quadratic computation complexity to the input image size due to computation of self-attention globally.

Swin Transformer makes it compatible with a broad range of vision tasks, including image classification (87.3% top-1 accuracy on ImageNet-1K) and dense prediction tasks such as object detection (58.7 box AP and 51.1 mask AP on COCO test-dev) and semantic segmentation (53.5 mIoU on ADE20K val). Its performance surpasses the previous state-of-the-art by a large margin of +2.7 box AP and +2.6 mask AP on COCO, and +3.2 mIoU on ADE20K, demonstrating the potential of Transformer-based models as vision backbones.

Briefly speaking, the transformers can not only be used in NLP, but also in image classification. This is a newly innovation in image classification. In the future, combining transformers with CNN networks may be a trend in overall architecture, composed of global information and local information respectively.

REFERENCES

[1] Patrício, D.I.; Rieder, R, “Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review,” *Comput. Electron. Agric*. vol. 153, pp. 69–81, 2018.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient based learning applied to document recognition,” *Proc. IEEE*, 1998.

[3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet classification with deep convolutional neural networks,” *NIPS*, 2012.

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, “Going deeper with convolutions,” *Proceedings, CVPR*, 2015, pp. 1-9.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep residual learning for image recognition,” *Proceedings, CVPR*, 2016, pp. 770-778.

[6] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu and Kaiming He, “Aggregated residual transformations for deep neural networks,” *Proceedings, CVPR*, vol. 2, pp. 1-8, 2017.

[7] Mingxing Tan, Quoc V.Le, “EfficientNet: Rethinking model scaling for convolution neural networks,” *Proceedings, ICML*, 2019, pp. 6105-6114.

[8] Gonçalo Marques, Deevyankar Agarwal and Isabel de la Torre Díez, “Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network,” *Appl. Soft Comput.* Vol. 96, 2020.

[9] S. Pan and Q. Yang. “A survey on transfer learning.” *IEEE TKDE*, vol. 22, pp. 1345–1359, 2010

[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. “How transferable are features in deep neural networks?” In *NIPS*, pp. 3320–3328. 2014.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal ́ loss for dense object detection,” in *Proceedings* of *IEEE* International Conference on Computer Vision (*ICCV*), pp. 2999–3007, 2017.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. “Automatic brain tumer segmentation using convolution neural networks with test-time augmentation,” *LNIP*, vol. 11384, pp. 61–72. 2019.

[13] Guotai Wang, Wenqi Li, Michael Aertsend, Jan Deprest, Sébastien Ourselinb, Tom Vercauteren, “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks,” *Neurocomputing,* vol. 338, pp. 34-45, 2019.

[14] Christian Szegedy, Sergey loffe, Vincent Vanhoucke, and Alex Alemi. “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” *Proceedings, AAAI conference*, pp. 4278–4284. 2016.

[15] Gao Huang, Zhuang Liu, Laurens vander Maaten, and Kilian Q. Weinberger, “Densely connected convolutional networks,” *Proceedings, CVPR*, pp. 4700–4708. 2017.

[16] M.A. Ganaie, Minghui Hu, A.K. Malik, M. Tanveer and P.N. Suganthan “Ensemble deep learning: A review,” *Proceedings, CVPR*, vol. 2, 2022.

[17] Marquis de Condorcet, “Essay on the application of analysis to the probability of majority decisions,” Paris: Imprimerie Royale, 1785.

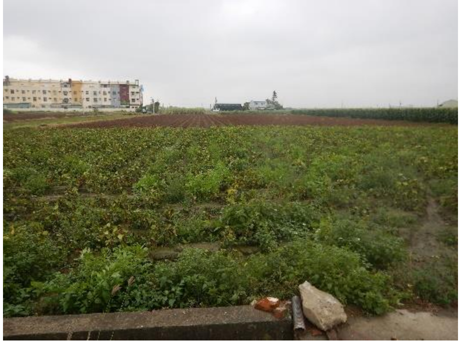
[18] Ilya Loshchilov and Frank Hutter, “Decoupled Weight Decay Regularization,” *ICLR*, vol. 3, pp. 1-8, 2019.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, “Swin Transformers: Hierarchical Vision Transformer using shifted windows,” *Proceedings, CVPR*, 2021.

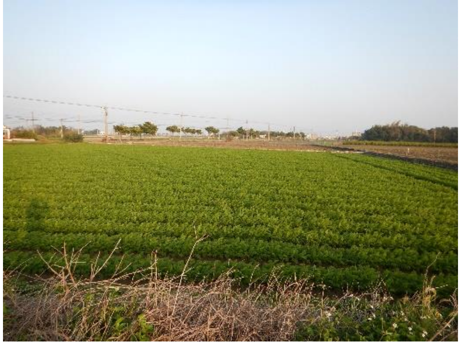
[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *Proceedings, CVPR*, 2021.

Appendix

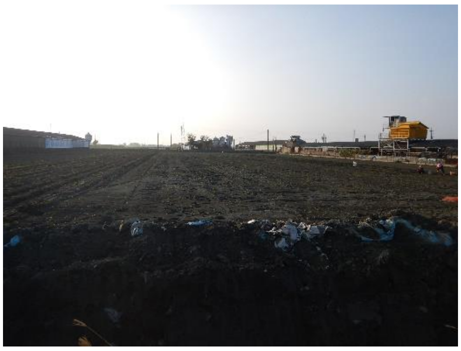
Predicted Correct (Proposed)



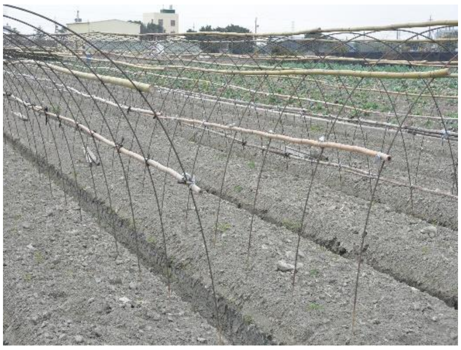
Ground truth: Soybean
Predicted: Soybean



Ground truth: Carrot
Predicted: Carrot

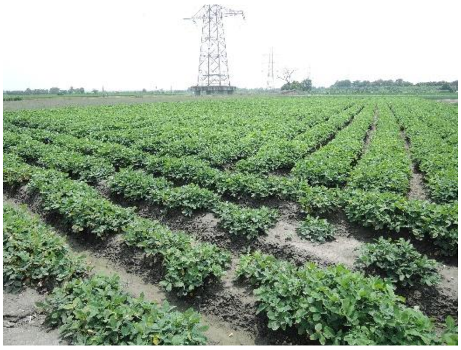


Ground truth: Peanut
Predicted: Peanut

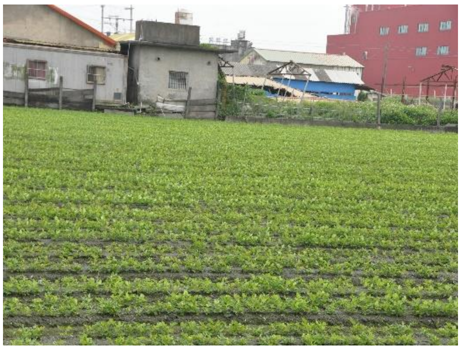


Ground truth: Tomato
Predicted: Tomato

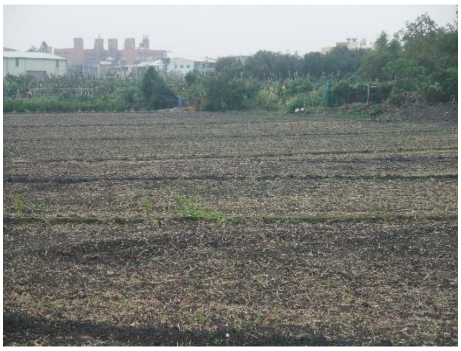
Predicted Wrong (ResNet 18)



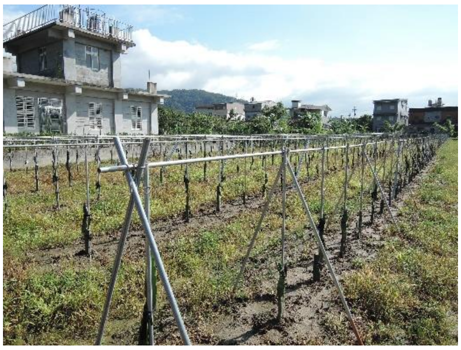
Ground truth: Soybean
Predicted: Peanut



Ground truth: Carrot
Predicted: Peanut



Ground truth: Peanut
Predicted: Bareland



Ground truth: Tomato
Predicted: Dragonfruit