Development of a mobile application for identification of grapevine (*Vitis vinifera* L.) cultivars via deep learning

Yixue Liu[1], Jinya Su[2], Lei Shen[1], Nan Lu[3], Yulin Fang[4], Fei Liu[5], Yuyang Song[4], Baofeng Su[1*]

(1. *College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China*;
2. *School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK*;
3. *College of Water Resources and Architectural Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China*;
4. *College of Enology, Northwest A&F University, Yangling 712100, Shaanxi, China*;
5. *College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China*)

Abstract: Traditional vine variety identification methods usually rely on the sampling of vine leaves followed by physical, physiological, biochemical and molecular measurement, which are destructive, time-consuming, labor-intensive and require experienced grape phenotype analysts. To mitigate these problems, this study aimed to develop an application (App) running on Android client to identify the wine grape automatically and in real-time, which can help the growers to quickly obtain the variety information. Experimental results showed that all Convolutional Neural Network (CNN) classification algorithms could achieve an accuracy of over 94% for twenty-one categories on validation data, which proves the feasibility of using transfer deep learning to identify grape species in field environments. In particular, the classification model with the highest average accuracy was GoogLeNet (99.91%) with a learning rate of 0.001, mini-batch size of 32, and maximum number of epochs in 80. Testing results of the App on Android devices also confirmed these results.

Keywords: deep learning, mobile phone, grapevine cultivar, vine leaf image, identification, *Vitis vinifera* L.

DOI: 10.25165/j.ijabe.20211405.6593

Citation: Liu Y X, Shen L, Su J Y, Lu N, Fang Y L, Liu F, et al. Development of a mobile application for identification of grapevine (*Vitis vinifera* L.) cultivars via deep learning. Int J Agric & Biol Eng, 2021; 14(5): 172–179.

1 Introduction

To meet the growing needs of the increasingly affluent and growing population, plant productivity must be improved, and resources must be managed more effectively. This problem is being severer in conditions of climate change and natural resources shrinkage in recent years. Precise breeding by selecting favorable genomic variants is helpful to improve plant productivity and efficiency. However, this method depends on a detailed understanding of the relationship between genotype and phenotype[1]. The framework of the automatic feature (phenotype) extraction and classification during the plant growth stage can significantly promote precise breeding.

As the pillar of the wine industry, the grape is receiving increasing interest and now has genome sequence from thousands of germplasm[2-4]. Conventionally, different germplasms (genotypes) were classified according to their working phenotypes by designated biologists, where "manual" images were used in terms of canopy architecture, leaf area, and other functions[5-8]. These functions can be calculated manually or through a customized image processing algorithm. The main disadvantage of using hand-made descriptors is that although being easy to interpret, they may fail to evaluate the actual characteristics of variability between germplasms. In addition, the customized image processing method for extracting manual features may not work well in other experiments and may be difficult to generalize to more heterogeneous datasets in practical applications[9-11]. Therefore, more advanced approaches are desirable for automatic genotype classification.

In the past few years, deep neural networks especially convolutional neural networks (CNN)[12-18] have been drawing increasing interest in both academia and industry due to their promising performance over conventional machine learning algorithms. In particular, the improved performance is mainly due to the complex structure of CNN, the substantially increased volume of the dataset and the significantly improved computation power. Different from conventional machine learning approaches, CNN can automatically extract the most descriptive and discriminating features from the image dataset in the training stage. In these approaches, the feature extraction and training steps are performed concurrently, where the algorithms attempt to automatically learn the features which can minimize the loss criteria for phenotype problems. As a result, the features for the problem of interest can be automatically generated. It is also noted that the training process for CNN approaches is usually time-consuming and also requires a large volume of the labelled dataset.

The traditional method of the identification of vine varieties by vine leaves is usually needed to do destructive sampling, followed by physical, physiological, biochemical and molecular measurements[19,20]. Although these methods have been proved to

Received date: 2021-03-19 **Accepted date:** 2021-07-27

Biographies: Yixue Liu, PhD candidate, research interest: plant phenotyping, Email: sunnyliu@nwafu.edu.cn; **Jinya Su**, PhD, Associate Professor, research interest: machine learning, Email: j.su@essex.ac.uk; Lei Shen, Master candidate, research interest: smart agriculture, Email: shenlei@nwafu.edu.cn; **Nan Lu**, PhD, Researcher, research interest: smart agriculture, Email: lunan@nwafu.edu.cn; **Yulin Fang**, PhD, Professor, research interest: horticulture, Email: fangyulin@nwsuaf.edu.cn; **Fei Liu**, PhD, Professor, research interest: smart agriculture, Email: fliu@zju.edu.cn; **Yuyang Song**, PhD, Associate Professor, research interest: viticulture, Email: yuyangsong@nwsuaf.edu.cn.

***Corresponding author: Baofeng Su**, PhD, Associate Professor, research interest: plant phenotyping. College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China. Tel: +86-29-87091527, Email: bfs@nwsuaf.edu.cn.

be effective, they are time-consuming, labor-intensive and usually require experienced grape phenotype analysts. To mitigate these problems, this study aimed to develop an application (App) running on Android client to identify the wine grape automatically and in real-time, which can help the growers to quickly obtain the variety information. This was achieved by seamlessly integrating some advanced tools and algorithms such as mobile devices for imaging, CNN for algorithm design and cloud servers for computing. Nowadays, there is a wide range of options offering mobile android devices. And even low-range devices with affordable prices provide excellent computing and photographic capabilities, which makes it possible to develop specialized applications in many fields. However, application for viticulture is not prolific in this sense yet, since there are only a few examples of available applications for managing vineyards[21,22]. In particular, mobile phones are first used to collect vine leaves of different cultivars to build the training datasets. Then CNNs of various architectures are trained on the labelled dataset to derive the suitable classification model. Then mobile App is built to automatically classify new vine leaves by the CNN model via cloud server so that different categories of the wine grape can be identified in complex field environments. To be more exact, the main contributions of this study are summarized as below:

1) A vine dataset including 5091 leaf images of 21 vine types was manually collected in field conditions;

2) Various CNN based image classification models were trained on the dataset to identify the suitable ones in terms of accuracy on validation dataset;

3) An Android App was developed which can automatically identify vine varieties by vine leaves via the cloud server.

2 Materials and methods

2.1 Vine dataset

In this study, the vine dataset contains twenty-one varieties covering common cultivars in vineyards. The images in the dataset were taken by Canon EOS 70D in different vineyards in Ningxia Hui Autonomous Region, China. As a consumer camera, Canon EOS 70D equipped with an approximately 20.2-megapixel CMOS sensor has strong adaptability for different scenes. So it is considered to be the representative of most lens of mobile phones in this study. Additionally, images (JPEG) stored on the camera's memory card were transferred and saved to cloud server. The images were taken without control conditions in the field. The image mainly consists of leaves, but also vines, soil and people. The sample images for twenty-one cultivars are displayed in Figure 1 and the details of cultivars are listed in Figure 2. The data set includes 225 images of Aglianico, 223 images of BeiMei, 308 images of Cabernet Sauvignon, 247 images of Aligote, 212 images of Welschriesling, 221 images of Pinot Noir, and 284 images of Riesling, 223 images of Malbec, 228 images of Marselan, 305

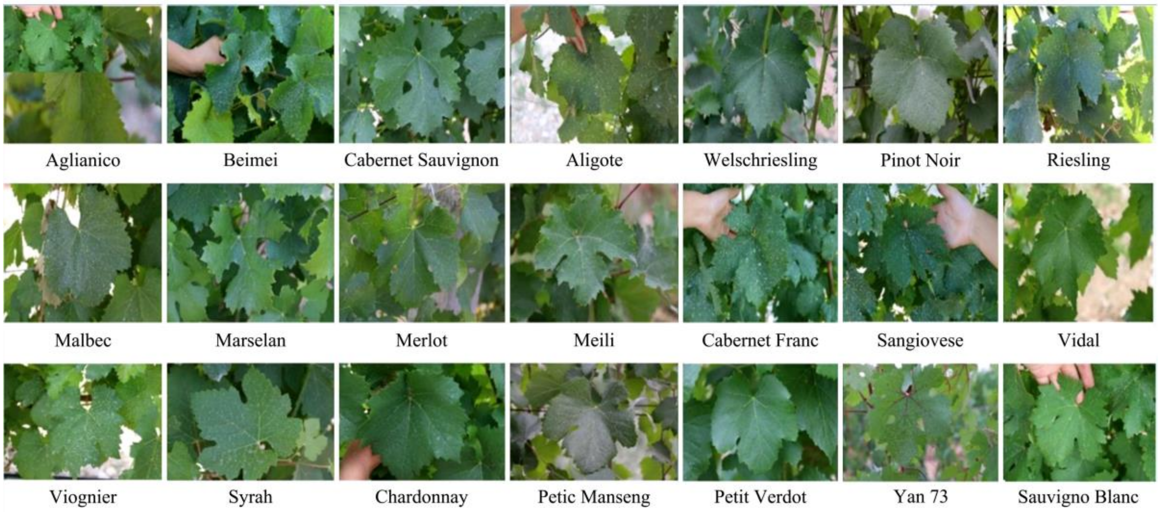


Figure 1 Example images for twenty-one cultivars vine taken in the field

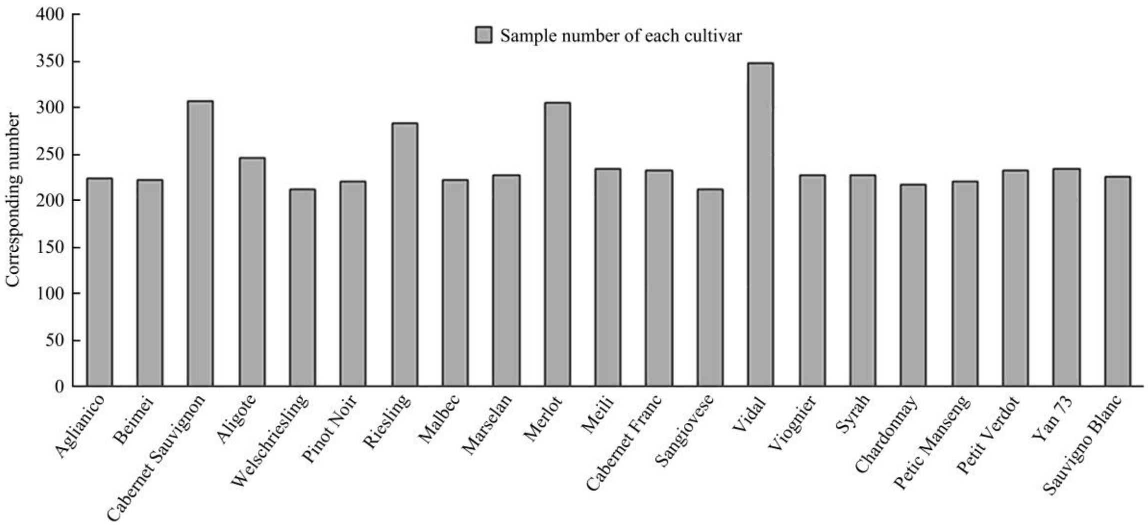


Figure 2 Twenty-one grapevine cultivars and corresponding numbers in the dataset

images of Merlot, 235 images of MeiLi, 232 images of Cabernet Franc, 212 images of Sangiovese, 348 images of Vidal, 228 images of Viognier, 228 images of Syrah, 218 images of Chardonnay, 221 images of Petit Manseng, 232 images of Petit Verdot, 235 images of Yan 73, and 226 images of Sauvignon Blanc.

2.2 Data preprocessing and augmentation

The performance of the vine classification task was improved by applying various image pre-processing techniques which can automatically obtain the strengthened localized image features from the original input image[23]. Firstly, raw images were processed into their complement image. Each color channel of the resulting image was the complement of the corresponding color channel in the original image[24]. In the original image, the leaves appear green because of a mixture of red and green signals. In the complement images, however, the leaves appear purple because the red and blue signals are higher than the green signal. Then, all the processed vine leaf images were resized to the same size as the input size for various CNN architectures.

Considering that deep learning algorithms usually have a high requirement on the volume of the labelled dataset (although a relatively large number of vine leaf images have been collected in this study), data augmentation techniques were further deployed to augment the labelled dataset so that the learned model can generalize well to unseen samples. In particular, geometric transformation techniques such as scaling, transposing, rotation and flipping were applied to expand the image dataset. This type of image augmentation technique is generic and computationally efficient to train the deep learning models effectively[25]. At the same time, Gaussian white noise with a mean and variance of 0.01 was also added to the image to improve algorithm robustness. 200 images were randomly selected for data augmentation of each variety as raw images. The total number of samples in the augmented dataset was finally 33 600 and 1600 images are contained per variety. 70% per variety including 1120 images are randomly selected as the training set to train the classifier. 30% per variety including 480 images were used as a validation set to evaluate the performance of the classifier. In addition, extra 10 images per variety that were not used for both training and validation were used as a test set to evaluate the performance of the App.

2.3 Transfer learning of deep networks

Deep learning is currently one of the most popular methods and has shown great performance on many image classification problems in the field of plant phenotyping. The concept of sharing weights in deep learning makes an effective image classification by discovering robust features in the images and reducing the vanishing gradient problem. The structure of CNN generally includes convolution layer, pooling layer, and fully connected layer. The convolutional layer acts as a filter, which aims to automatically extract image features. The convolutional layer is usually followed by the pooling layer, which performs downsampling and retains the most important information in the images. This layer reduces the spatial size of representation as well as the number of parameters and therefore prevents the problem of overfitting, making the model more effective. The last layer is the fully connected layer, which uses a softmax activation function and takes the high-level features from the images for classifying them into various categories[26].

In consideration of the limited number of labelled datasets, the concept of transfer learning was applied to retrain the deep learning classifier. This concept is not new and has been previously applied in a number of studies[27-29]. The vine classification tasks are evaluated in terms of accuracy and efficiency. In transfer learning, the layered architecture of the pre-trained models such as AlexNet, ResNet and VGG (without its final classification layer) can be used as a fixed feature extractor to achieve better vine classification performance with a shorter training time[30]. In this study, five deep learning networks were explored for vine leaf classification via transfer learning, which include AlexNet, ResNet, GoogLeNet, DenseNet and VGG that are pre-trained networks trained on images from the ImageNet database[31-35]. The concept of fine-tuning was adopted in this study, where the last fully connected layer was replaced and initialized with the target task class. The selection of all training parameters was based on the empirical observation of network training convergence and training effect.

In this study, the performance of the model was evaluated by comparing the classification results of the model with the actual tags. The two commonly used performance indicators included accuracy (ACC; Equation (1)), recall rate (RECALL; Equation (2))[36]. The data processing involved in the model construction was run in Matlab®2019b (The MathWorks, Inc., Natick, Massachusetts, USA). All experiments were carried out on a Linux machine with Ubuntu 16.04 preinstalled. It has a GTX 2080ti GPU, an Intel® core i7-5930k processor and 16 GB DDR4 RAM.

$$ACC = \frac{TP + TN}{TP + FP + TN +} \qquad (1)$$

$$\mathrm{RECALL} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad (2)$$

2.4 APP development

An App named VitisView was developed as a tool to accelerate the precise management of vineyards in the study. In particular, CNN based on object classification model was deployed on Android devices via cloud server, which can identify the leaf phenotype of wine grape in field environments and also in real-time. This app can be downloaded from http://175.27.144.115:8080/VitisView.apk. The VitisView was developed in the Android studio, and the main development tool JDK version is 11.0.7. It can call the camera of mobile phone to take images, then the app sends or receives data through socket. It can upload the image of the mobile library to the server and then receive the detection result of this image transmitted back by the server.

Two typical mobile phones were used to test the performance of the app. One is Vivo X9 with Android 7, 4GB RAM and Qualcomm® snapdragon 625 processor, the other one is Mi 9 with Android 10, 6GB RAM and Qualcomm® snapdragon 855 processor. The test set contains 210 images that were not used for training or validation and each category has 10 images. The performance evaluation was carried out in respect of the speed, the time from uploading images to getting classification results and the accuracy.

2.5 Cloud server

The trained CNN classification model was deployed on the cloud server. The communication between Android client and cloud server was realized through HTTP (Hypertext Transfer Protocol, HTTP). A java servlet using Apache Tomcat[37] was responsible for processing HTTP messages. The processing pipelines of cloud server are shown in Figure 3. Programs developed based on python provided the functions of downloading images and transferring files from Android devices to the cloud server. Matlab served as the function of running the classification model to process images and output recognition results. The cloud server was configured with Windows Server 2016 Datacenter, Intel® Xeon(R) Platinum 8255C CPU @2.5GHz and 8GB DDR4 RAM based on Tencent Cloud (Tencent, Inc., Shenzhen, China).

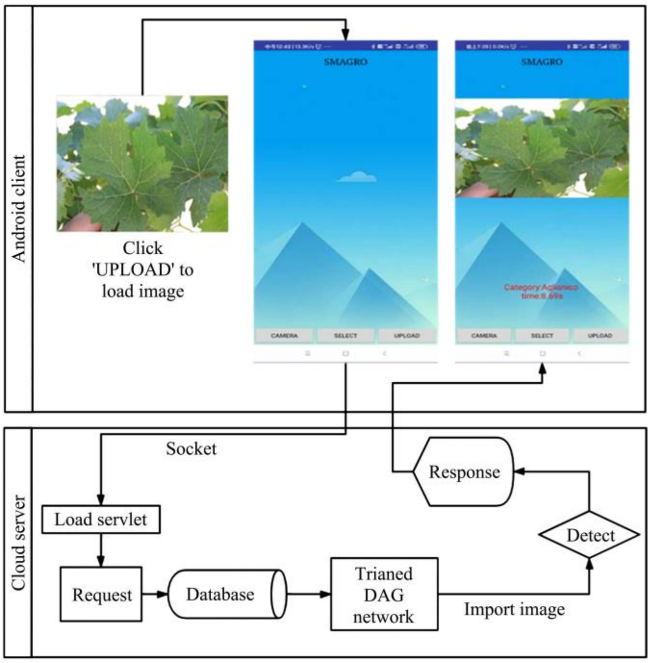


Figure 3 Cloud server working pipelines

3 Results and discussion

3.1 Classification model selection

To enable a fair comparison for all the experimental configurations, it was tried to standardize the parameters in experiments. In particular, a set of options was created for training a network using stochastic gradient descent with momentum and reduced the learning rate by a factor of 0.1 every 60 epochs. To set the maximum number of epochs for training to 80 and used a mini-batch of 32 observations at each iteration. To validate the network at regular intervals during training, the Validation Frequency value was chosen according to the size of mini-batch so that the network could be validated about once per epoch. The training was progressing by plotting various metrics was learned during training on Matlab.

The classification results of different CNN network structures were got, including AlexNet, Vgg-16, ResNet101, ResNet18, DenseNet and GoogLeNet. According to the evaluation metric, the accuracy of these classification models is shown in Table 1.

Table 1 Accuracy of six classification models

Network	ACC	Learning rate	Max epochs	Mini-batch size
Alexnet	94.70%	0.0001	80	32
VGG-16	96.85%	0.0001	80	32
ResNet101	97.24%	0.0001	80	32
ResNet18	95.68%	0.0001	80	32
DenseNet	94.70%	0.0001	80	32
GoogLeNet	99.66%	0.0001	80	32

The results showed that the accuracy on the validation dataset of the CNN detection algorithms can reach over 94% for 21 cultivars, which proved the feasibility of using transfer learning to identify grape cultivars in the actual growth environment. The detection model with the highest average test accuracy was GoogLeNet (97.4%). In particular, as shown in Figure 4, the prediction accuracy of 9 out of 21 categories of GoogLeNet detection model was 100%. It therefore could be drawn from the experimental results that the pre-trained models with transfer learning provided good results for the dataset with 21 labels of grapes, respectively.

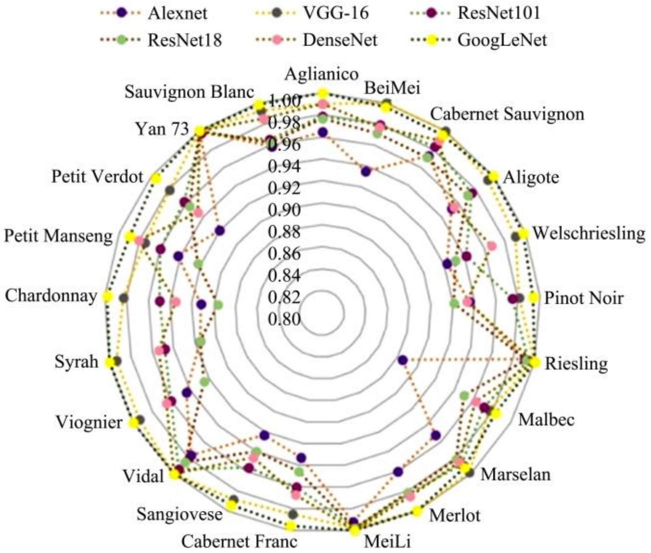


Figure 4 Recall rates of the twenty-one cultivars for different networks

3.2 Model parameter selection

Upon selecting GoogLeNet as the classification model in Section 2.1, the effects of different training parameters on the classification performance were tested. Particularly, the influences of learning rate and minimum batch size on training results were considered. Three experiments were set up with the value of learning rate being 0.001, 0.005 and 0.0001 while keeping other parameters unchanged. Table 2 shows the classification accuracy under different learning rate values. The results showed that when the learning rate was 0.001, the classification performance reached the best.

Table 2 Performance of GoogLeNet under different learning rate values

Model	Learning rate	Max epochs	Mini-batch size	ACC
T1	0.0001	80	32	99.66%
T2	0.001	80	32	99.91%
T3	0.005	80	32	99.41%

Note: T*i* represents the *i*-th trained model (*i*= 1, 2, 3, …), the same as below.

In consideration of the size of memory, increasing the minimum number of mini-batch sizes in a reasonable range can improve memory utilization and also increase training speed. For large-scale training, it is necessary to make sure that using larger batch size training gets the test accuracy similar to a smaller batch size under the same epochs. The reason why to keep the number of epochs unchanged was that from a statistical point of view, an epoch means that the algorithm contacts the whole dataset once; from a computational point of view, a fixed number of epochs means that the number of floating-point operations remains unchanged. Three experiments were set up with the value of mini-batch size being 16, 32 and 64 while keeping the learning rate as 0.001 and other parameters unchanged. Table 3 shows the classification accuracy under different mini-batch sizes for the GoogLeNet detection model. The results showed that the minimum batch size of 32 was enough.

Table 3 Performance of GoogLeNet under different mini-batch size values

Model	Mini-batch size	Learning rate	Max epochs	ACC
T4	16	0.001	80	99.84%
T5	32	0.001	80	99.91%
T6	64	0.001	80	99.40%

It was discovered in the process of training that the max epoch was an important factor that affected the learning progress of the model. Four experiments were set up with the maximum number of epochs being 30, 60, 80, 120 while keeping other parameters unchanged. Table 4 shows the accuracy under different maximum epoch values. The results showed that the ideal maximum number of epochs was 80.

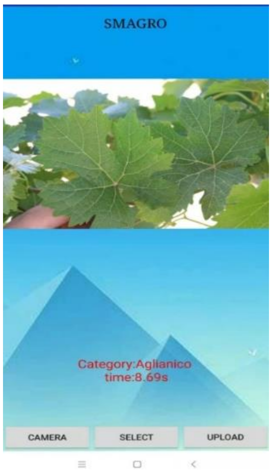
Table 4 Performance of GoogLeNet under different max epoch values

Model	Max epochs	Learning rate	Mini-batch size	ACC
T7	30	0.001	32	97.88%
T8	60	0.001	32	99.27%
T9	80	0.001	32	99.91%
T10	120	0.001	32	99.75%

Therefore, the optimized parameters for GoogLeNet classification model are the learning rate of 0.001, mini-batch size of 32 and the maximum number of epochs of 80.

3.3 App performance

The training and testing of CNN classification models were completed for grapevine identification on Matlab and uploaded to the cloud server. The cloud server performs the functions of caching images, detecting images and feedback the classification results. The app performance was carried on two Android devices mentioned in 2.4. The test results on Mi 9 showed that the accuracy was 98%. The recall rate of 19 categories reached 100%. The minimum recall was 70%. The average response time from uploading images to returning results was 8.25 s. The average speed of the test Internet environment was 1.19 Mb/s. Similarly, we repeated the same experiment on Android 7 mobile phone, and the overall accuracy was 98%, of which 19 categories achieved 100% and the minimum accuracy was 80%. The average response time is 8.28 s and the average speed was 1.33 Mb/s. The test results showed that the VitisView is stable and robust, which could help grapevine cultivar information in the field obtained in real-time.



Category:Aglianico
time:8.69s

Figure 5 Screenshot of the VitisView

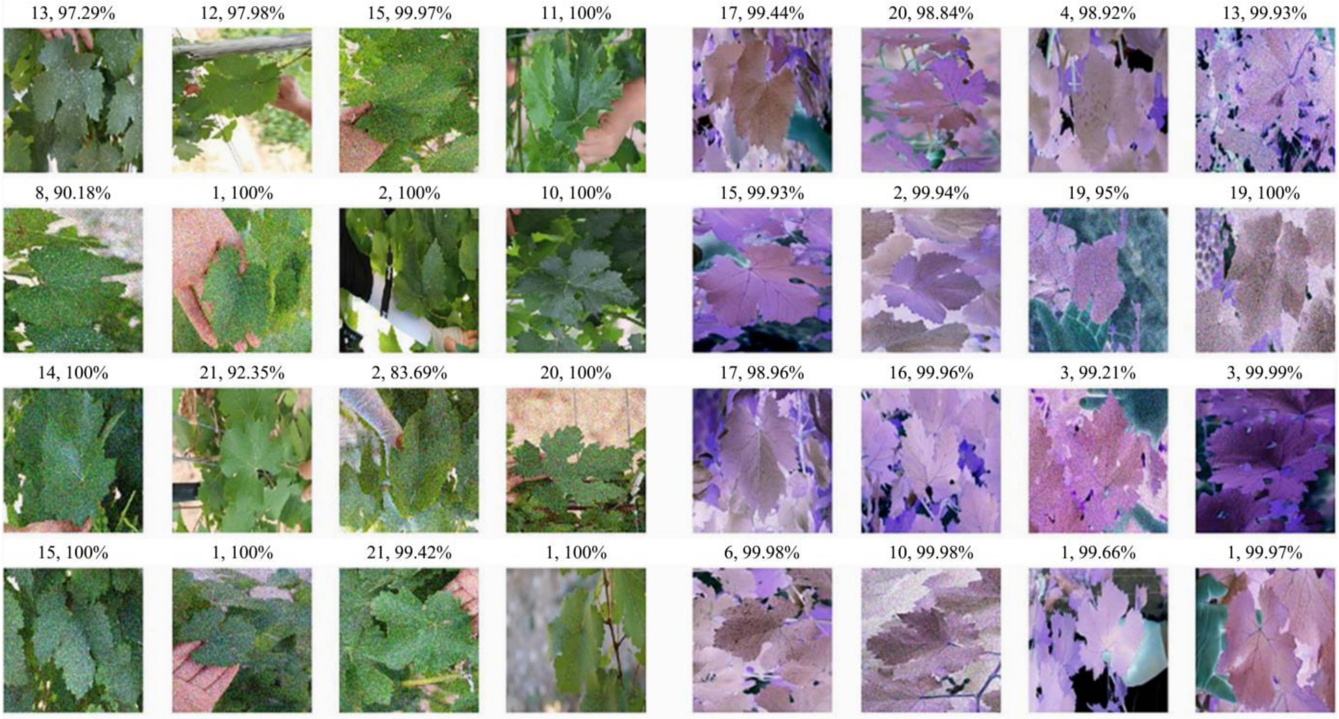
3.4 Discussion

Plant leaves show little difference even in different stages so that they can be often used as the feature in the case of plant cultivar identification. In this study, it has been shown that even in field environments, the images collected by typical smart phones or digital cameras can be used to train deep CNN by transfer learning for accurate wine grape cultivar identification.

There was very little literature on accurately identifying the phenotype of wine grape by nondestructive methods (e.g. imaging based approach) in the past few years. Carlos et al.[38] evaluated the performance of the transfer learning techniques based on AlexNet for the grapevine cultivar identification and reached a test accuracy of 77.30% on a dataset including six varieties with an average of 14 images captured in the field per variety. The accuracy of the trained model reached 99.91%. Besides a large number of images available for training, early dataset augmentation plays an important role in achieving performance. In the case of using transfer learning to solve the problem of object recognition, in order to improve the accuracy of the detection model, images are often acquired in a controlled environment in the laboratory[39]. The image preprocessing method was used to enhance the object or weaken the background on the other hand. Canny edge detection was used to detect the edge in the gray-scale image converted from RGB of insect and suppress the noise, which was a method of data augmentation to facilitate crop pest recognition using transfer learning[29]. Threshold processing, contour detection and watershed algorithm were used to eliminate the influence of natural background on object detection[28].

The raw dataset was tried to test without using the image complement processing mentioned in Section 5.2. Similarly, geometric transformation techniques such as scaling, transposing, rotation and flipping were applied to expand the number of image samples in the datasets. Datasets without image complement processing and datasets with image complement processing were trained on GoogLeNet model. While training, two datasets used the same number of images and the same training parameter setting. The results of CNN classification models by the two datasets were compared. The results show that the accuracy of the detection model is 97.49% for the dataset without image complement processing and 99.66% for image pre-processing. In other words, when using the deep learning algorithm to identify grape species in the field environment, the complement processing of the raw image has a certain impact on the performance. When using the classification model to identify the image obtained from Android devices on the cloud server, the image will be first processed by complement and then the result will be obtained by the detection model, which is not complicated.

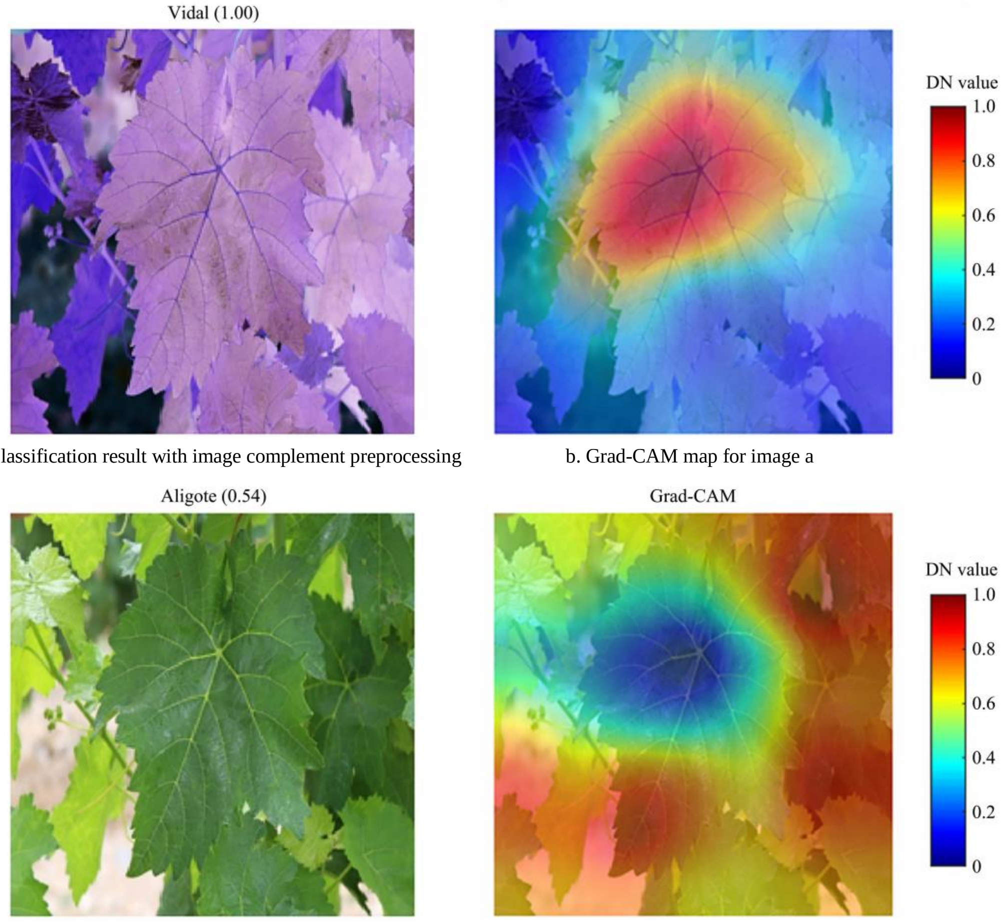
In addition, the Grad-CAM algorithm was introduced to analyze the impact of image complement preprocessing on classification results. The purpose of Grad-CAM was to calculate the gradient of the final classification score relative to the final convolution layer in the network[40]. The gradient of the output of the softmax layer is the largest part of the convolution Eigen graph, the final result mostly depends on the data. To this end, an image of Weddell was used as an example to assess the effects of image complement preprocessing. In addition to displaying the classification result, the gradient thermal image of the weight in the final convolution layer was obtained by Grad-CAM algorithm. The results in Figure 7 showed that the vein and the main part of leaves had the greatest influence on the classification. After the image complement preprocessing, the leaf vein was more easily recognized. While the model without image complement preprocessing does not work well. This is because the background of leaves has adverse effects on the final classification results. Therefore, image preprocessing plays an important role in achieving a satisfying classification result.



a. With image complement preprocessing b. Without image complement preprocessing

Note: The first data above each image represents the serial number corresponding to the grape cultivar, and the corresponding relationship is shown in the appendix. The second data above each image represents the confidence level of the classification results.

Figure 6 Examples of validation set classification results



a. Classification result with image complement preprocessing

b. Grad-CAM map for image a

c. Classification result without image complement preprocessing

d. Grad-CAM map for image c

Figure 7 Effect of image preprocessing on classification result by Grad-CAM map

In this study, the VitisView on Android devices was developed to realize automatic real-time identification of wine grape and help grapevine species information obtained in the field. However, this automatic real-time recognition algorithm for wine grapes can be applied not only to mobile devices but also to IoT devices, making it a part of the construction of unmanned farming. In addition, although the identification of different genotypes of wine grape constructed in this study involves 21 common cultivars in the vineyard, it is not enough for the wine industry in the long run. The data set can be enlarged and the algorithm can be modified for more cultivar identification in the future.

4 Conclusions

In this study, an App named VitisView running on Android client was developed to identify wine grape cultivars in real-time and field conditions, which can help the growers to quickly obtain the variety information of wine grapes. This is achieved by seamlessly integrating a number of techniques and algorithms such as phone imaging, convolution neural network and cloud server. In particular, a total of 5091 leaf images were first collected in the field environment which contain 21 cultivars of typical grapes. Then both image preprocessing and data augmentation were adopted to enhance image features and augment training dataset. On this basis, a number of typical CNN models (including VGG-16, DenseNet, ResNet101, ResNet18, and GoogLeNet) were compared in transfer learning training to identify the suitable one while with model parameter tuning. It is shown that GoogLeNet model outperformed other models in terms of accuracy, model complexity, and robustness with a fine-tuned accuracy of 99.91%. The effects of image complement preprocessing are also assessed by using Grad-CAM algorithm.

Acknowledgements

The authors would like to express their gratitude to the Teaching Experiment Farm of Ningxia University, for their kind help. This study was supported by the Key R & D projects of Ningxia Hui Autonomous Region (Grant No. 2019BBF02013).

[References]

[1] Furbank R T, Tester M. Phenomics--technologies to relieve the phenotyping bottleneck. Trends Plant Sci, 2011; 16(12): 635–644.

[2] Youens-Clark K, Buckler E, Casstevens T, Chen C, Declerck G, Derwent P, et al. Gramene database in 2010: updates and extensions. Nucleic Acids Research, 2011; 39: D1085–D1094. doi: 10.1093/nar/gkq1148.

[3] Joshua C, Pankaj K, Miklavcic S J. Land-based crop phenotyping by image analysis: consistent canopy characterization from inconsistent field illumination. Plant Methods, 2018; 14(1): 39. doi: 10.1186/s13007-018-0308-5.

[4] Wan Y, Schwaninger H R, Baldo A M, Labate J A, Zhong G Y, Simon C J. A phylogenetic analysis of the grape genus (*Vitis* L.) reveals broad reticulation and concurrent diversification during neogene and quaternary climate change. BMC Evolutionary Biology, 2013; 13: 141. doi: 10.1186/ 1471-2148-13-141.

[5] Berk P, Stajnko D, Belsak A, Hocevar M. Digital evaluation of leaf area of an individual tree canopy in the apple orchard using the LIDAR measurement system. Computers and Electronics in Agriculture, 2020; 169: 105158. doi: 10.1016/j.compag.2019.105158.

[6] Tian K, Li J, Zeng J, Evans A, Zhang L. Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm. Computers and Electronics in Agriculture, 2019; 165: 104962. doi: 10.1016/j.compag.2019.104962.

[7] Vanderwel M C, Lopez E L, Sprott A H, Khayyatkhoshnevis P, Shovon T A. Using aerial canopy data from UAVs to measure the effects of neighbourhood competition on individual tree growth. Forest Ecology and Management, 2020; 461: 117949. doi: 10.1016/j.foreco.2020.117949.

[8] Gao Z, Luo Z, Zhang W, Lyu Z, Xu Y. Deep Learning Application in Plant Stress Imaging: A Review. AgriEngineering, 2020; 2(3): 430–446. doi: 10.3390/agriengineering2030029.

[9] Shi Y, Zhu Y, Wang X, Sun X, Ding Y, Cao W, et al. Progress and development on biological information of crop phenotype research applied to real-time variable-rate fertilization. Plant Methods, 2020; 16(1): 11. doi: 10.1186/s13007-020-0559-9.

[10] Han L, Yang G, Dai H, Yang H, Xu B, Li H, et al. Combining self-organizing maps and biplot analysis to preselect maize phenotypic components based on UAV high-throughput phenotyping platform. Plant Methods, 2019; 15(1): 57. doi: 10.1186/s13007-019-0444-6.

[11] Chopin J, Kumar P, Miklavcic S J. Land-based crop phenotyping by image analysis: consistent canopy characterization from inconsistent field illumination. Plant Methods, 2018; 14(1): 39. doi: 10.1186/s13007-018-0308-5

[12] Bresilla K, Perulli G D, Boini A, Morandi B, Corelli Grappadelli L, Manfrini L. Single-shot convolution neural networks for real-time fruit detection within the tree. Frontiers in Plant Science, 2019; 10: 611. doi: 10.3389/fpls.2019.00611.

[13] Mohanty S P, Hughes D P, Salathé M. Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 2016; 7: 1419. doi: 10.3389/fpls.2016.01419.

[14] Lee S H, Goëau H, Bonnet P, Joly A. New perspectives on plant disease characterization based on deep learning. Computers and Electronics in Agriculture, 2020; 170: 105220. doi: 10.1016/j.compag.2020.105220.

[15] Desai S V, Balasubramanian V N, Fukatsu T, Ninomiya S, Guo W. Automatic estimation of heading date of paddy rice using deep learning. Plant Methods, 2019; 15(1): 76. doi: 10.1186/s13007-019-0457-1.

[16] Ubbens J, Cieslak M, Prusinkiewicz P, Stavness I. The use of plant models in deep learning: an application to leaf counting in rosette plants. Plant Methods, 2018; 14(1): 6. doi: 10.1186/s13007-018-0273-z.

[17] Kamilaris A, Prenafeta-Boldú F X. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 2018; 147: 70–90.

[18] Gao Z, Khot L R, Naidu R A, Zhang Q. Early detection of grapevine leafroll disease in a red-berried wine grape cultivar using hyperspectral imaging. Computers and Electronics in Agriculture, 2020; 179: 105807. doi: 10.1016/j.compag.2020.105807.

[19] Guo R, Xu X, Carole B, Li X, Gao M, Zheng Y, Wang X. Genome-wide identification, evolutionary and expression analysis of the aspartic protease gene superfamily in grape. BMC Genomics, 2013; 14(1): 554. doi: 10.1186/1471-2164-14-554.

[20] Huang Y-F, Doligez A, Fournier-Level A, Le Cunff L, Bertrand Y, Canaguier A, et al. Dissecting genetic architecture of grape proanthocyanidin composition through quantitative trait locus mapping. BMC Plant Biology, 2012; 12(1): 30. doi: 10.1186/1471-2229-12-30.

[21] Aquino A, Barrio I, Diago M-P, Millan B, Tardaguila J. vitisBerry: An Android-smartphone application to early evaluate the number of grapevine berries by means of image analysis. Computers and Electronics in Agriculture, 2018; 148: 19–28.

[22] Aquino A, Millan B, Gaston D, Diago M P, Tardaguila J. vitisFlower (R) : Development and testing of a novel Android-smartphone application for assessing the number of grapevine flowers per inflorescence using artificial vision techniques. Sensors, 2015; 15(9): 21204–21218.

[23] Cui X, Goel V, Kingsbury B. Data augmentation for deep neural network acoustic modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015; 23(9): 1469–1477.

[24] Youla D. Generalized image restoration by the method of alternating orthogonal projections. IEEE Transactions on Circuits and Systems, 2003; 25(9): 694–702.

[25] Montserrat D M, Lin Q, Allebach J, Delp E J. Training object detection and recognition CNN models using data augmentation. Electronic Imaging, 2017; 10: 27–36.

[26] Simonyan K, Science A Z J C. Very deep convolutional networks for large-scale image recognition. arXiv, 2014; arXiv: 1409.1556.

[27] Ramcharan A, Baranowski K, Mccloskey P, Ahmed B, Legg J, Hughes D P. Deep learning for image-based Cassava disease detection. Frontiers in Plant Science, 2017; 8: 1852. doi: 10.3389/fpls.2017.01852

[28] Li Y, Wang H, Dang L M, Sadeghi-Niaraki A, Moon H. Crop pest recognition in natural scenes using convolutional neural networks. Computers and Electronics in Agriculture, 2020; 169: 105174. doi: 10.1016/j.compag.2019.105174.

[29] Thenmozhi K, Srinivasulu Reddy U. Crop pest classification based on deep convolutional neural network and transfer learning. Computers and Electronics in Agriculture, 2019; 164: 104906. doi: 10.1016/j.compag.2019.104906.

[30] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012; 1: 1097–1105.

[31] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision & Pattern Recognition (CVPR), 2016; pp.770–778. doi: 10.1109/CVPR.2016.90.

[32] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vsion and Pattern Recognition, 2015; pp.1–9. doi: 10.1109/CVPR.2015.7298594.

[33] Huang G, Liu Z, Van Der Maaten L, Weinberger K Q. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp.2261–2269. doi: 10.1109/CVPR.2017.243.

[34] Ding P, Zhang Y, Deng W J, Jia P, Kuijper A. A light and faster regional convolutional neural network for object detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing, 2014; 141: 208–218.

[35] ImageNet. Available: http://www.image-net.org/challenges. Accessed on [2020-12-15].

[36] BS-ISO-5725-1. Accuracy (trueness and precision) of measurement methods and results. General principles and definitions. London: BSI, 1994; 17p.

[37] Apache Tomcat. Available: https://tomcat.apache.org. Accessed on [2020-12-15].

[38] Pereira C S, Morais R, Reis M. Deep learning techniques for grape plant species identification in natural images. Sensors, 2019; 19(22): 4850. doi: 10.3390/s19224850.

[39] Liu Z, Wang J, Tian Y, Dai S. Deep learning for image-based large-flowered chrysanthemum cultivar recognition. Plant Methods, 2019; 15: 146. doi: 10.1186/s13007-019-0532-7.

[40] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. International Journal of Computer Vision, 2020; 128(2): 336–359.