

Performance of MobileNetV3 Transfer Learning on Handheld Device-based Real-Time Tree Species Identification

Ambreen Hussain*, Bidushi Barua, Ahmed Osman, Raouf Abozariba, and A. Taufiq Asyhari*
School of Computing and Digital Technology, Birmingham City University, UK

E-mail: {ambreen.hussain, bidushi.barua, ahmed.osman, raouf.abozariba}@bcu.ac.uk, taufiq-a@ieee.org

Abstract—Detailed information on tree species constitutes an essential factor to support forest health monitoring and biodiversity conservation. Current deep learning-based mobile applications for tree and plant identification require excessive computation. They largely depend on a network connection to perform computing tasks on powerful remote servers in the Cloud. Many forestry areas are remote with limited or no cellular coverage, which is an obstacle for these applications to recognize trees and plants in these areas in real-time. This paper investigates existing CNN-based machine learning applications for plant identification tailored for handheld device usages. Driven by network independence, reduced computation, size and time requirements, we propose the use of MobileNet (a mobile computer vision architecture) transfer learning to improve the accuracy of offline leaf-based plant recognition. We then carry out experimental validation of state-of-the-art MobileNet. Our findings reveal that using MobileNetV3 transfer learning, accuracy up to 90% can be achieved within fewer iterations than end-to-end CNN-based models for plant identification. The lightweight model comes with reduced computation that runs independently within a smartphone application without internet access, ideal for tree species identification in rural forests.

Index Terms—MobileNet, CNN, plant identification, mobile devices, transfer learning

I. INTRODUCTION

According to a report in [1], forests cover around one-third of the world's land, providing a range of ecosystem services, including energy, building materials, and climate regulation. Tree species identification is an important aspect of forest ecological research to support forest health monitoring and biodiversity conservation [2]. Many tasks, such as endangered species monitoring, assessment of the impact of climate change on species distribution, and analysis of weed control actions, depend on accurate tree species recognition [3]. For experts with domain knowledge, manual identification may be a resource-intensive task, and for non-experts, it will be error-prone and time-consuming [4]. The results from manual observation are less accurate due to the unavailability of expert knowledge in many cases [5]. Accelerating this task and making it possible for non-experts is beneficial in less effort and low cost. To complement the manual identification, DNA barcoding has received growing research interests. However, it requires DNA extractions from harvested tissue and lacks recorded data and reference databases in most countries [6]. Most recently, image recognition technologies have

gained momentum to determine tree species. Compared to DNA technologies, computer vision methods are non-intrusive, which mainly use machine learning algorithms. Many machine learning methods for tree species classification, such as deep learning models, e.g. Convolutional Neural Networks (CNN), have been applied to different datasets. The primary advantage of CNN for image classification is that the entire system is trained completely automatically from raw pixels of images to eventual categories. These images are acquired using remote sensing or handheld devices and processed by powerful Cloud-based machine learning algorithms. We aim at tree species identification using a CNN-based lightweight model, which could embed within a handheld device resolving computation, battery and network constraints.

CNN comes with the disadvantages of a huge number of labelled training samples requirement for weighting parameters and powerful GPUs for a faster learning process. With limited training samples, learning too many parameters directly will result in overfitting even after using over-fitting prevention techniques such as Dropout [7]. Overfitting of a machine learning model includes more terms and uses more complicated approaches than necessary [8]. To make deep CNN fit for small datasets while keeping high accuracy is a challenge. To overcome this challenge, transfer learning [9] offers the solution, which implies reusing previously attained knowledge in similar tasks. Some studies analyzed the use of transfer learning for many deep learning algorithms. These evaluations [10] concluded that transfer learning improves the accuracy of deep learning models, which apply deep features and fine-tuning.

Various research studies, reviewed in the next section, aimed at deep learning methods for plant species identification. However, these studies focused on utilizing complex deep models, such as InceptionV3, AlexNet, and VGG-16, which require a considerable number of iterations and computation power to train the network. Moreover, the existing plant identification applications such as PictureThis [11] and Pl@ntNet [12] depend upon the network connection to evaluate the captured image by analyzing trained models on servers. These applications fail in remote forestry areas where mobile networks provide limited/no patchy connection [13]. Due to low power and computation capabilities, a model with reduced computation

and size is required for smartphones, which is offered by CNN-based MobileNets [14]. To the best of our knowledge, MobileNet in general and MobileNetV3, in particular, has not emerged in any study for tree species identification using handheld devices in rural forests. Therefore, in this paper, we focus on measuring MobileNetV3 transfer learning performance compared to MobileNetV2 on images captured via smartphone in real-time for tree classification. The lightweight model will be embedded in the mobile application to perform the tree classification tasks in rural areas where internet infrastructure is sparse. Our work contributes to the following:

- Integrating state-of-the-art MobileNet with transfer learning for efficient and timely offline plant and tree species identification.
- Feature extraction and fine-tuning of pre-trained MobileNet architecture models, i.e. MobileNetV3 releases and MobileNetV2 on leaf dataset via transfer learning. The last layers of pre-trained models were removed and replaced with layers suitable for the leaf dataset with learning parameters.
- Training of new models on a new dataset containing leaves. Conversion of the newly trained network into handheld devices compatible versions for offline leaf species identification using a mobile app in rural areas.
- Comparison of the performance of the new models in terms of accuracy, loss, training time and size.

II. LITERATURE REVIEW

The use of machine learning algorithms for tree species identification is relatively new. In this regard, variations of leaf characteristics are utilized mainly by computer vision techniques for automated tree identification. Leaves are easy to access, observe and describe as compared to other parts of a tree. Many surveys have been performed on automated plant identification, such as those given by Ahmed, Khan and Asif in [15] and Thyagarajan and Raji in [16]. However, still plant identification is a challenging task because all classical computer vision techniques use handcrafted features for plant's leaf images. For example, an approach presented by [17] used the image binarisation technique to separate the leaf from its background, detect contours and geometrical derivations of leaf tooth features. Using this approach, the accuracy of leaf classification was between 72.8% and 79.3%. The handcrafted features for the leaf's images can be categorized into shape, texture, and venation. In this regard, a mobile app LeafSnap [18], was proposed to identify trees by analyzing leaves. Similarly, the Pl@ntNet app [19] was introduced by Goëau et al. to use citizen science to collect and integrate leaf imagery data. Both applications depend on chosen handcrafted features to measure similarities between the novel and stored images of known species. The top-1 accuracy rate of recognizing plant species for both applications lies between 69% and 73%.

To complement these mobile applications' handcrafted feature selection approach, deep learning CNN is applied in many studies, especially for image classification [20]. The main advantage of CNN is the automatic learning of input

data which replaced the traditional feature-based methods. A CNN approach for plant classification was given by [21] with an accuracy of 99.7% on a dataset of 44 species. Similarly, [22] and [23] used CNN for Flavia [24] dataset classification and reported 94.69% - 99% accuracy with 12,000 and 50,000. LeafNet [25] presented the usage of CNN on LeafSnap, Flavia, and Foliage [26] datasets and acquired accuracy of 79.66%, 98.69% and 98.75% with iterations of 200,000, 30,000 and 100,000, respectively. For automatic detection and recognition of plant diseases through mobile devices, a comparison of different CNN architectures was performed by [27]. The result of this comparison was that AlexNet achieves higher accuracy of 99.1% with 12,000 iterations; however, regarding complexity, MobileNet gave the best performance under the same epochs. A comparison of neural networks was performed by [28] for the mobile phone-based image classification of citrus leaf diseases. The MobileNetV2 transfer learning was implemented in a GPU environment for the disease recognition with a validation accuracy of 92% at epoch 10, which is comparatively faster than traditional CNN architectures with good accuracy. To demonstrate the efficiency and accuracy of MobileNetV3 for mobile devices, [29] compared the image classification performance of MobileNetV3 with other pre-trained CNN models on various image datasets such as fruits with 99.9% accuracy within 10 iterations and birds with up to 80% accuracy within 25 epochs. The use of MobileNetV3 based on the You Only Looked Once v3 (YOLOv3) [30] model has appeared in other domains, such as pedestrian detection [31] on Unmanned Aerial Vehicles (UAV) imagery.

Comparing all the above studies, we deduce that the applications used end-to-end CNN architecture trained the network with thousands of iterations to achieve a high range of accuracy. In contrast, the approaches that used MobileNet transfer learning achieved similar accuracy within few epochs under considerably less time, which makes them cost and time-efficient. Therefore, in this study, we focus on exploring MobileNetV3 releases and MobileNetV2 transfer learning performance compared to previous CNN-based leaf classification models. Our experience shows MobileNet models take a considerably fewer number of iterations to train through transfer learning with a good percentage of accuracy. Furthermore, the reduced models' size allows it to immerse in the mobile application to recognize plant species offline in remote forest sites.

III. METHODOLOGIES

A. Transfer Learning from Pre-trained Models

When data used to train deep learning networks is not enough, and it is expensive to create labelled data, transfer learning, a promising paradigm, is applied to acquire the knowledge learned in previous settings. The most common strategies of transfer learning are feature extraction and fine-tuning. For feature extraction, input data is provided to the pre-trained model, and activation values of many layers are stored to use as features. For fine-tuning, the initial layers of the model are fixed, and the final layers of the model

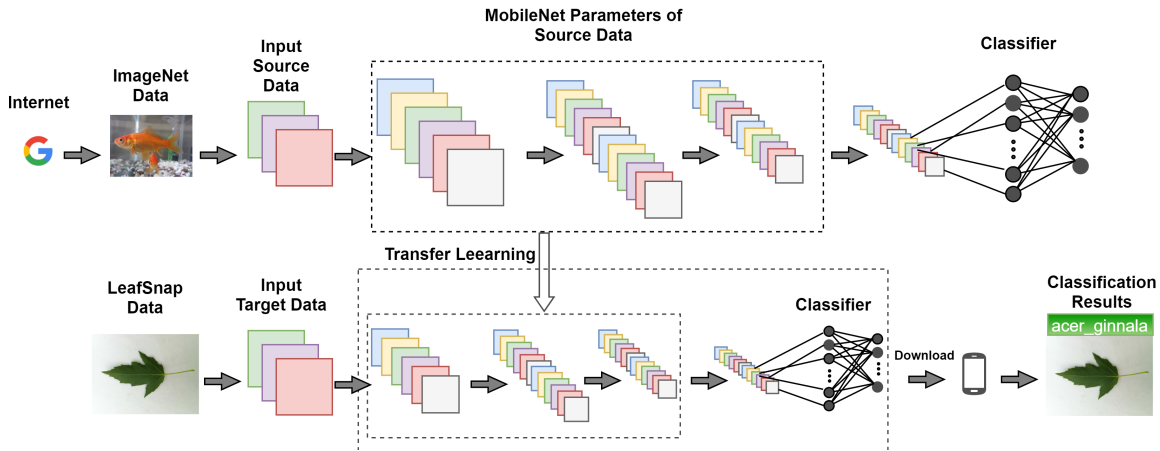


Fig. 1. Transfer Learning of MobileNet-based Model for Leaf Classification.

learn the characteristics of the new dataset. This way, the deep learning model is trained for a new similar task. The weights of the model are updated according to the new dataset. Transfer learning is faster than training a new network because all the parameters are not estimated from scratch. In an analysis of transfer learning methods [10], the authors compared deep learning models such as CNN, AlexNet, VGG-16, RNN for plant classification and concluded that transfer learning improves the efficiency of deep learning models, in particular those that apply deep features and fine-tuning as compared to other transfer learning techniques such as those based on feature spaces and domain given by [32].

B. Feature Extraction Model

MobileNet: MobileNet is the most popular CNN-based model for smartphone applications and embedded devices. It was introduced as a mobile-first computer vision model by the Google Research team [14]. The core idea is to use depthwise separable convolution to build a lightweight model with less computational complexity. The reason for high computation in traditional CNN architecture is the point to point convolution is performed between the image and filter. The MobileNet’s depthwise convolution uses a single filter per input channel, and pointwise convolution applies 1×1 convolution to create a linear combination of the output of the depthwise layer. The factorized convolutions allow a substantial reduction in the computation and model size. This makes the model ideal for mobile and embedded devices.

MobileNet Versions: MobileNetV2 introduced the inverted residual structure and linear bottleneck to make an efficient layer structure. MobileNetV3 [33] is the latest and most efficient combination of its predecessors, i.e. MobileNetV1 and MobileNetV2. MobileNetV3 is tuned to mobile CPU by fusing two technologies: MNasNet to select the optimal configuration and NetAdapt to discover and optimize network architecture. MobileNetV3 model has two releases: MobileNetV3-Large and MobileNetV3-Small. It is claimed that compared to V2, MobileNetV3-Large is 3.2% more

accurate on ImageNet classification with latency reduced by 20%, whereas MobileNetV3-Small is 6.6% more accurate with similar latency. The overall architecture of transfer learning using a version of the MobileNet-based model for leaf classification is shown in Fig. 1. In this study, we have compared MobileNetV3 releases with MobileNetV2 transfer learning on the LeafSnap dataset. The resultant model is downloaded into the smartphone for efficient tree species identification without an internet connection.

C. Dataset

For test and trial purposes in the lab experiments, we explored datasets available online in different data repositories such as Kaggle [34], Tensorflow [35] and Google [36]. For the plant identification use case, we accessed the LeafSnap dataset from Kaggle. The dataset consists of 184 tree species, showing two subsets, (1) 23,915 lab images of pressed leaves with dimensions of 800×700 pixels (2) 7,719 field images with 800×600 dimensions taken outdoors by different users using handheld devices. The field images contain varying amounts of noise, blur, shadows, and illumination patterns. Each plant category contains 20-30 RGB images of leaves. We downscaled all the images to 224×224 to achieve faster learning of the deep learning models. For this paper, only field images are used because of their similarity to real images taken from mobile phones.

D. Model Training

Feature Extraction: The hardware to train the model included a Lenovo laptop equipped with an 8265U CPU at 1.80 GHz of Intel Core i5, 8 GB of RAM running on a Windows 10 64-bit system. The software tools included Anaconda3 Jupyter Notebook with Python 3.8, where the Tensorflow, OpenCV-python3 [37] and Keras [38] libraries were used. The dataset was divided in a ratio of 75:25 into a training set of 5789 images and a test set of 1930 images. To obtain the pre-trained weight parameters, we built the base models from the MobileNetV2, MobileNetV3-Large and MobileNetV3-Small

models, pre-trained on 1.4M images from 1000 classes. First, the intermediate layer of all the models was selected to use for feature extraction. For this purpose, the output of the last layer (bottleneck layer) is commonly used prior to the flatten operation. The reason is that all the fully connected layers are specialized to the task network was trained on; thus, the features learned by these layers will not be useful for the new task. The bottleneck features are more generic. Therefore, the base models were frozen, loaded with weights trained on the ImageNet dataset and used as a feature extractor. We added a dropout layer with a dropout rate of 0.2 to reduce the overfitting of the model. Finally, the top classifier softmax layer was added for predicting the class for an input image. The model with the base model of MobileNetV2 was compiled and optimized with Adam optimizer. The other with the base model of MobileNetV3 were optimized with Gradient descent optimizer (SGD) with a learning rate of 0.5 and momentum of 0.9. All models were trained on a training set within 10 and 100 epochs (iterations).

Fine-Tuning: We un-froze the base models and trained the weights of the top layers of these models along with the classifier layer added in the previous step. The models were trained in 10 iterations to force the weights to be tuned from generic feature maps to feature associated with the leaf dataset. For a fair comparison, both V2 and V3 models were trained with the same dataset and training parameters, i.e. 235,704. The models were fine-tuned and re-evaluated. The models were saved in the machine with all the class labels.

IV. RESULTS AND DISCUSSION

Model Validation: The performance of the models is greatly influenced by the number of times it goes under training. Using sparse categorical cross-entropy loss, the models were validated with a test dataset. The performances of the models were evaluated by calculating accuracies and losses of training samples in each epoch. The model based on MobileNetV2 showed (Fig. 2) the training accuracy up to 97% and validation accuracy up to 84%. The model based on MobileNetV3-Large exhibit a training accuracy of 94% and validation accuracy up to 87%.

Fine-Tuning Results: After fine-tuning, the training accuracy of the MobileNetV2-based model was improved to 97%, and validation accuracy remained at 86% at each epoch. The training accuracy of the MobileNetV3-based model was improved to up to 96% (Fig. 3), and validation accuracy up to 89% after 10 epochs. Note the gap between training and validation curves for the MobileNetV3-based model is less than the MobileNetV2-based model.

Discussion: We managed to increase the validation accuracy of the MobileNetV3-Large based model to 91% after 100 epochs, whereas the MobileNetV2-based model could show only 1% improvement, i.e. 87%. The MobileNetV2 and MobileNetV3 took the almost same time to train the network; however, it is notable that MobileNetV3 contains 5.3M parameters, whereas MobileNetV2 contains 3.4M parameters. On selecting MobileNetV3-Small as a base model, the training time

of the model was reduced to 60%, trainable parameters were declined to 188,600 and training, and validation accuracies were between 92%-94% and 82%-86%, respectively.

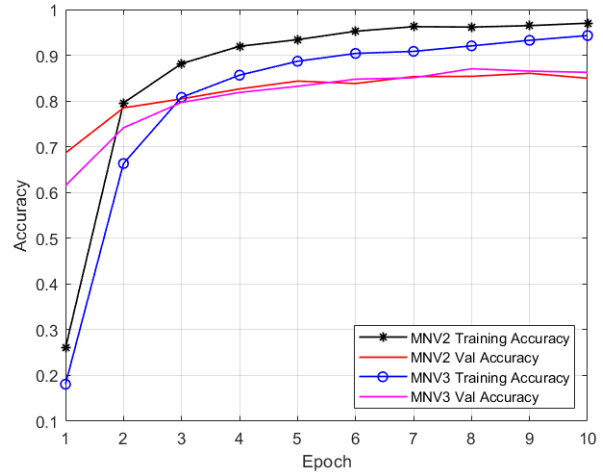


Fig. 2. Training and Validation Accuracy of MNV2/MNV3-based Transfer Learned Models.

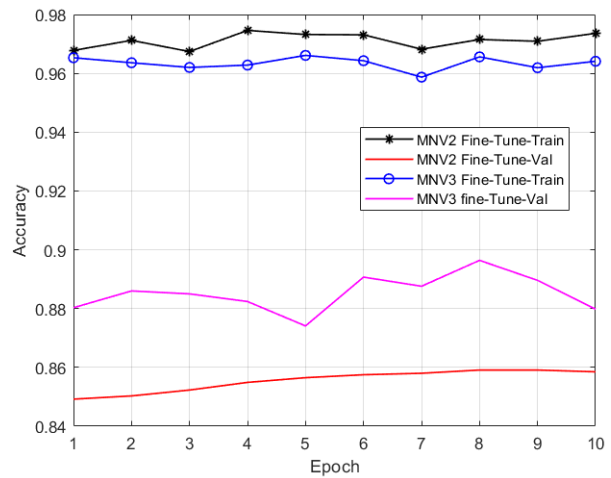


Fig. 3. Training and Validation Accuracy of MNV2/MNV3-based Fine-tuned Models.

Our results show that the performance of MobileNetV3 is better than MobileNetV2. The MobileNetV3-Small is time and cost-efficient with 3% less accuracy. The accuracy of all the MobileNetV3 models can be increased with the data augmentation techniques as in [28] and training iterations. The training time of MobileNetV3-Large can be reduced with the use of a GPU. In Table I, we compare the results obtained from our trained models based on MobileNetV3-Large, MobileNetV3-Small, and MobileNetV2 with the previously mentioned CNN-based models for leaf classification. It can be seen that we achieved a good accuracy using our trained models with fewer iterations and significantly less time, which makes our models time and cost-effective. Moreover, the size of the models is

TABLE I
COMPARISON OF TRAINED MNV3/MNV2 MODELS WITH OTHER CNN-BASED APPROACHES

Model	Dataset	Overall Iterations	Accuracy %	Loss %	Training Time (minutes)	Size
[22] CNN	Flavia	12,000	94.69	0.7-2.5	-	-
[23] CNN	Flavia	50,000	98.69	-	252	-
[25] CNN	LeafSnap	200,000	79.96	-	1920	-
[25] CNN	Flavia	30,000	98.69	-	-	-
Our work MobileNetV2	LeafSnap	20	86	0.4	40	9.86 MB
Our work MobileNetV3-Large	LeafSnap	20	89	0.3-0.4	40	4.88 MB
Our work MobileNetV3-Small	LeafSnap	20	86	0.5-0.6	16	1.99 MB

much reduced, which makes them ideal to be inserted in handheld devices for offline plant and tree species recognition.

V. MOBILE APP IMPLEMENTATION

App Development: For tree species identification in a remote area, an end-user app is essential. For the app development, Android Studio [39] was used. The saved models were converted into the TensorFlow Lite [40] version using the TensorFlow Lite converter. TensorFlow Lite is a set of tools to perform deep learning on smartphone and IoT devices. The architecture of the mobile app is shown in Fig. 4. The squares marked with dotted red will be included in future iterations. The size of the models based on MobileNetV3-Large and MobileNetV3-Small was 4.88 MB and 1.99 MB, respectively, which are lightweight compared to AlexNet (227.5 MB) [27]. The app is developed using the Java programming language. The leaf detection module in the app contains a Tensorflow Lite interpreter for the TensorFlow Lite model.

App Testing: For testing the application, Android studio comes with built-in phone emulators, or the app could be deployed to an actual device. From an online search, we downloaded leaf images belonging to the categories on which the model was trained. These downloaded images were not included in the training dataset. The functionality was tested with the WiFi and data connections being turned off (flight mode). This was to ensure that the app is not using any online computation resources to run the machine learning algorithm for classifying the images. This way the app can be used in remote areas with no/limited network.

VI. CONCLUSIONS

This paper analyzed CNN-based mobile apps for plant identification and compared the performance of MobileNet versions for their suitability for tree species recognition using smartphones in remote forests. The MobileNetV3 releases and MobileNetV2-based models were trained and tested on the LeafSnap dataset to detect tree species. We evaluated the performances of the models using the accuracy and loss of the training and test sets. The best validation accuracy of MobileNetV3 was 89% and 91% at 10 and 100 epochs, respectively, whereas MobileNetV2 had 86%. The MobileNetV3-

Small took 60% less time to train the model as compared to the MobileNetV3-Large and MobileNetV2. We believe that the accuracy of our models can be improved with data augmentation methods within 20 epochs. Our evaluation indicates that MobileNetV3-based transfer learned models are lightweight, cost and time-efficient, making them perfect candidates to be embedded in smartphone apps and IoT devices to perform deep learning tasks in rural areas without mobile networks.

VII. FUTURE WORK

As an extension of this work, we plan to investigate the accuracy of these models on images taken at different heights. This will involve applying data augmentation techniques to increase the dataset with rescaled images specified through the zoom range. The mobile app will be updated in subsequent iterations with a camera option to obtain images, percentage of accuracy, detailed view of the identified species with images of other parts of the tree, and permission to upload tree health-related data along with pictures into the Cloud.

VIII. ACKNOWLEDGMENT

This work was funded by the Department for Digital, Culture, Media & Sport (DCMS) as part of the 5G Connected Forest (5GCF) project under its 5G Testbeds and Trials Program. *The corresponding authors for this work are Ambreen Hussain and A. Taufiq Asyhari.

REFERENCES

- [1] Food and A. O. [FAO], "Global forest resources assessment 2020. terms and definitions," 2018.
- [2] E. J. Farnsworth, M. Chu, W. J. Kress, A. K. Neill, J. H. Best, J. Pickering, R. D. Stevenson, G. W. Courtney, J. K. VanDyk, and A. M. Ellison, "Next-generation field guides," *BioScience*, vol. 63, no. 11, pp. 891–899, 2013.
- [3] J. Wäldchen, M. Rzanny, M. Seeland, and P. Mäder, "Automated plant species identification—trends and future directions," *PLoS computational biology*, vol. 14, no. 4, p. e1005993, 2018.
- [4] G. E. Austen, M. Bindemann, R. A. Griffiths, and D. L. Roberts, "Species identification by experts and non-experts: comparing images from field guides," *Scientific Reports*, vol. 6, no. 1, pp. 1–7, 2016.
- [5] N. MacLeod, M. Benfield, and P. Culverhouse, "Time to automate identification," *Nature*, vol. 467, no. 7312, pp. 154–155, 2010.
- [6] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: A review," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7562–7573, 2012.

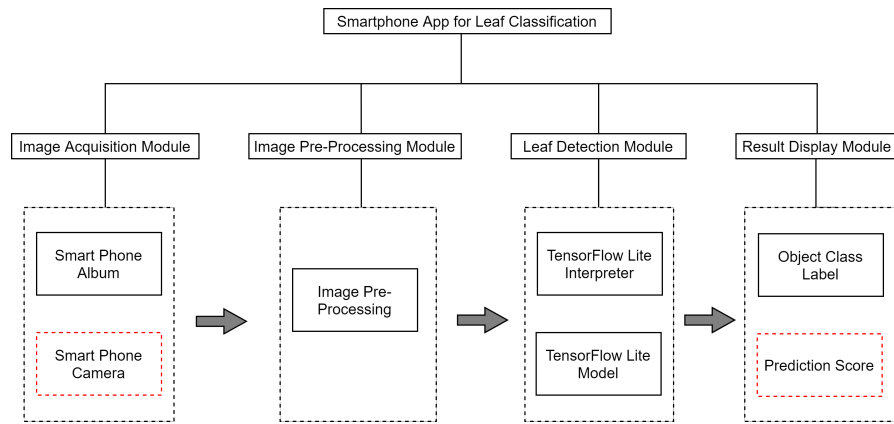


Fig. 4. The Architecture of Mobile App.

- [7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [8] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.
- [9] D. Han, Q. Liu, and W. Fan, "A new image classification method using cnn transfer learning and web data augmentation," *Expert Systems with Applications*, vol. 95, pp. 43–56, 2018.
- [10] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and electronics in agriculture*, vol. 158, pp. 20–29, 2019.
- [11] "plant encyclopedia + catalogue of plant + taxonomic tree." <https://plant.picturethisai.com/en/plant>, accessed: 04.22.2021.
- [12] "Accueil - pl@ntnet," <https://plantnet.org/en>, accessed: 04.22.2021.
- [13] R. Abozariba, E. Davies, M. Broadbent, and N. Race, "Evaluating the real-world performance of 5G fixed wireless broadband in rural areas," in *2019 IEEE 2nd 5G World Forum (5GWF)*. IEEE, 2019, pp. 465–470.
- [14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [15] N. Ahmed, U. G. Khan, and S. Asif, "An automatic leaf based plant identification system," *Science International*, vol. 28, no. 1, 2016.
- [16] K. Thyagarajan and I. K. Raji, "A review of visual descriptors and classification techniques used in leaf species identification," *Archives of Computational Methods in Engineering*, vol. 26, no. 4, pp. 933–960, 2019.
- [17] T. Jin, X. Hou, P. Li, and F. Zhou, "A novel method of automatic plant species identification using sparse representation of leaf tooth features," *PLoS one*, vol. 10, no. 10, p. e0139482, 2015.
- [18] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *European conference on computer vision*. Springer, 2012, pp. 502–516.
- [19] H. Goëau, P. Bonnet, A. Joly, V. Bakić, J. Barbe, I. Yahiaoui, S. Selmi, J. Carré, D. Barthélémy, N. Boujemaa *et al.*, "Pl@ntnet mobile app," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 423–424.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [21] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in *2015 IEEE international conference on image processing (ICIP)*. IEEE, 2015, pp. 452–456.
- [22] C. Zhang, P. Zhou, C. Li, and L. Liu, "A convolutional neural network for leaves recognition using data augmentation," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 2143–2150.
- [23] C. Wick and F. Puppe, "Leaf identification using a deep convolutional neural network," *arXiv preprint arXiv:1712.00967*, 2017.
- [24] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *2007 IEEE international symposium on signal processing and information technology*. IEEE, 2007, pp. 11–16.
- [25] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "Leafnet: A computer vision system for automatic plant species identification," *Ecological Informatics*, vol. 40, pp. 50–56, 2017.
- [26] A. Kadir, L. E. Nugroho, A. Susanto, and P. I. Santosa, "Leaf classification using shape, color, and texture features," *arXiv preprint arXiv:1401.4447*, 2013.
- [27] Z. Tang, J. Yang, Z. Li, and F. Qi, "Grape disease image classification based on lightweight convolution neural networks and channelwise attention," *Computers and Electronics in Agriculture*, vol. 178, p. 105735, 2020.
- [28] U. Barman, R. D. Choudhury, D. Sahu, and G. G. Barman, "Comparison of convolution neural networks for smartphone image based real time classification using citrus leaf disease," *Computers and Electronics in Agriculture*, vol. 177, p. 105661, 2020.
- [29] S. Qian, C. Ning, and Y. Hu, "Mobilenetv3 for image classification," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. IEEE, 2021, pp. 490–497.
- [30] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [31] X. Zhang, N. Li, and R. Zhang, "An improved lightweight network mobilenetv3 based yolov3 for pedestrian detection," in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. IEEE, 2021, pp. 114–118.
- [32] K. Weiss, T. M. Khoshgofaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [33] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.
- [34] "Leafsnap dataset kaggle," <https://www.kaggle.com/xhlulu/leafsnap-dataset>, accessed: 04.21.2021.
- [35] "Tensorflow," <https://www.tensorflow.org/datasets>, accessed: 04.21.2021.
- [36] "Dataset search," <https://datasetsearch.research.google.com>, accessed: 04.21.2021.
- [37] "Opencv-python · pypi," <https://pypi.org/project/opencv-python>, accessed: 04.21.2021.
- [38] "The functional api," https://keras.io/guides/functional_api, accessed: 04.21.2021.
- [39] "Android studio and sdk tools," <https://developer.android.com/studio>, accessed: 04.21.2021.
- [40] "Tensorflow lite, ML for mobile and edge devices," <https://www.tensorflow.org/lite>, accessed: 04.21.2021.