Classification of Apple-leaf diseases for FGVC8 Plant Pathology

Amr Abdalla
Computer Engineering department
The American University in Cairo
amroadel@aucegypt.edu

Ismail Shaheen
Computer Engineering department
The American University in Cairo
i.shaheen@aucegypt.edu

Maha Shatta
Electronics and Communication Engineering
The American University in Cairo
mahashatta@aucegypt.edu

***Abstract*—Plant pathology detection have been one of the most active areas of research recently due to the increasing reliability of convolutional neural networks and its economic value. This paper presents a proposed method for classifying diseases in apple-leaf with a focus on decreasing computational resources and maintaining a competitive accuracy. Various computer vision techniques have been used to help the proposed deep learning model, and the results of our experimentation show an accuracy of 72% on FGVC8 Plant Pathology.**

Keywords—CNN, Augmentation, EfficientNet, Computer Vision

I. Introduction

Plant pathology is one of the most important fields and that is because its main responsibility is the diagnosis of crops diseases and utilizing various methodologies to control this damage. Image processing has been widely used to detect plant diseases to guarantee sustainable agriculture. This is because of its advantages in early capturing the disease and improving the productivity of the crops. To illustrate, in [1], [2], [3], [4] and [5], multiple computer vision techniques were conducted such as smoothing, segmentation or augmentation that are followed by a feature extraction model such as convolutional neural networks (CNN) and then classification layers. This has resulted in great accuracy that is above 90%. In addition, various challenges and competitions have been conducted such as FGVC8 plant pathology [7] which concentrates more on apple leaf diseases. The main challenge in this competition is that its dataset has multiple disease symptoms variations for each leaf due to differences in leaf color, leaf morphology, age of infected tissues, various light brightness in addition to the non-homogeneous and analogous to leaf-color background [4].

In this paper, we are proposing a methodology for detecting apple leaf diseases using FGVC8 plant pathology datasets. This paper is organized as follows, Section II includes the related work, Section III the proposed methodology, Section IV experiments and results, section V future work and finally Section VI the conclusion.

II. Related Work

In [1], the method introduced relied on using preprocessing techniques. This depends on removing any shadows of the images in the dataset, this is followed by a segmentation method. It is used to divide the images into various parts for showing the similarities. Segmentation is used to extract the plant from the image and mask the background to zero. This is done through grab cut segmentation technique in which the user has to add the estimated size of the bounding box of the targeted plant in addition to the input image for the segmentation purpose. This is followed by Morphological Processing to remove small noise parts and leaves the essential object. Two convolution and pooling layers were used for feature extraction followed by fully connected layers for classification. This model succeeded to achieve 0.91 accuracy range.

The dataset was collected for different plant types by which the user had to use the same illumination and avoiding shadows. It is also noticed that the plant in the image of the dataset has either some sort of contrast with the surrounding environment or the background is blurred.

In [2], a method introduced to detect different plant diseases. It also depends on conducting some image preprocessing which includes geometric transformations such as image scaling, rotation and translation, this step is then followed by Augmentation process. In this step, various versions of single image in datasets were generated, this is done to increase the datasets and avoid the overfitting problem. These variations include rotation, width shift, height shift and horizontal flip. ResNet50 model was used to extract the features. This model has multiple convolution and pooling layers and has proven its capabilities for accurate detections. VGG16, VGG19 and AlexNet models were also compared to ResNet50 and it was shown that ResNet50 achieved the best results and accuracy (0.998 Training Accuracy).

It is also noticed that the dataset used in this paper depends on having 38 categories of plants. The background of the leaves used in this dataset is in contrast with the leaf itself as shown in the below figure.

(a) (b) (c)

In [3], Data augmentation was also used with more variations than in [2]. These variations have included intensity disturbance, which are disturbances of sharpness, contrast and brightness in addition to Gaussian noise processing operation. And as a result, 12 images are generated for each image. And as real time detection is targeted with object detection, the model used is INAR-SSD explained in [3], which is divided into two steps, first is the single shot multi box detector (SSD) which is a basic pre network for the classification and bounding box detection. This is followed by two GoogLeNet Inception module that is using parallel layers of several kernel sizes of convolution for increasing the adaptability for multi-scale feature extraction. And finally, e Rainbow concatenation method is applied to improve the feature fusion performance.

The results for 26,377 images with uniform and complex background have reached 0.78% with a speed of 23.13 FPS.

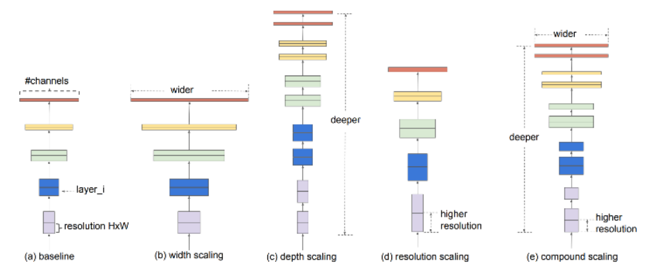
III. Proposed Methodology

Our problem statement is to classify the leaf images to decide if they have a disease or not. Our proposed solution combines computer vision techniques with deep learning to better approach the problem. The computer vision part is mainly applying filters and realistic transforms to the images to better help the deep learning model generalize and avoid overfitting. The details of this part will be elaborated in more details in the Data Augmentation section.

Our proposed deep learning architecture is based on EfficientNet [6]. EfficientNet is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. Unlike conventional practice that arbitrary scales these factors, the EfficientNet scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients. For example, if we want to use 2^N times more computational resources, then we can simply increase the network depth by α^N , width by β^N , and image size by γ^N , where α, β, γ are constant coefficients determined by a small grid search on the original small model. EfficientNet uses a compound coefficient $\emptyset$ to uniformly scales network width, depth, and resolution in a principled way.

The base EfficientNet-B0 network is based on the inverted bottleneck residual blocks of MobileNetV2 [8], in addition to squeeze-and-excitation blocks.

EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters.

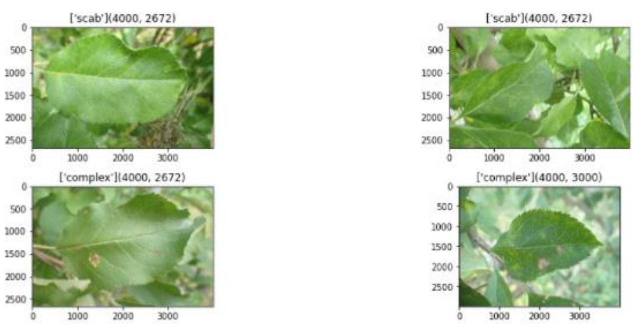


. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of netwo width, depth, or reso ut on. (c) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio

We used EfficientNet as our backbone by importing a pretrained model and removing its classification head. We then added dense layers to the end of the piplines that was trained independently at the begingin, then we used it to finetune the weights of the EfficientNet.

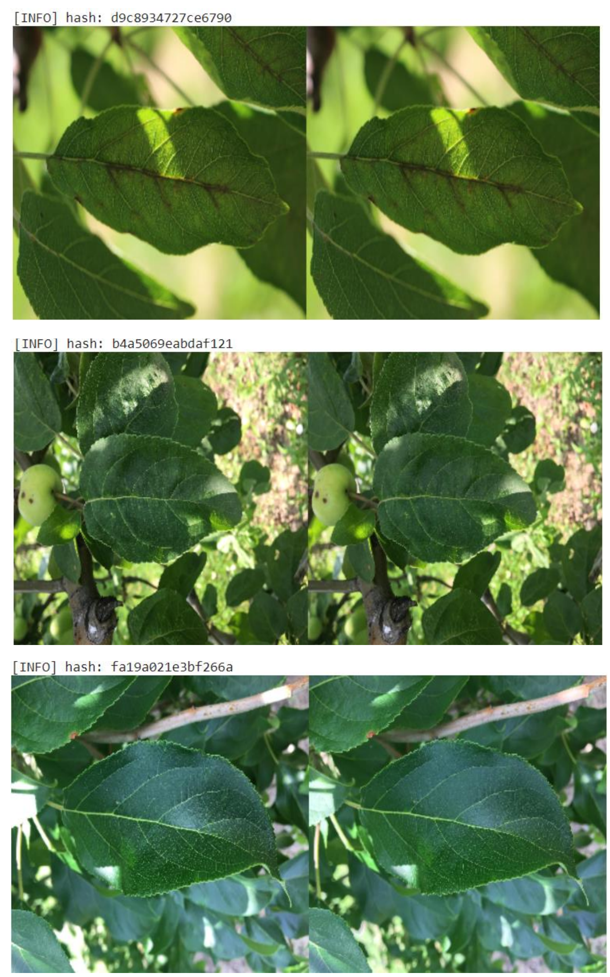
A. *Datasets*

The Dataset used in this architecture is the dataset of FGVC8 Plant Pathology. The dataset contains about 20K high-quality RGB images of apple foliar diseases, including a large expert-annotated disease dataset. This dataset reflects real field scenarios by representing non-homogeneous backgrounds of leaf images taken at different maturity stages and at different times of day under different focal camera settings. The images are labels with 12 different labels, some of which are a mixture of several abstract labels.



B. *Image Hashing*

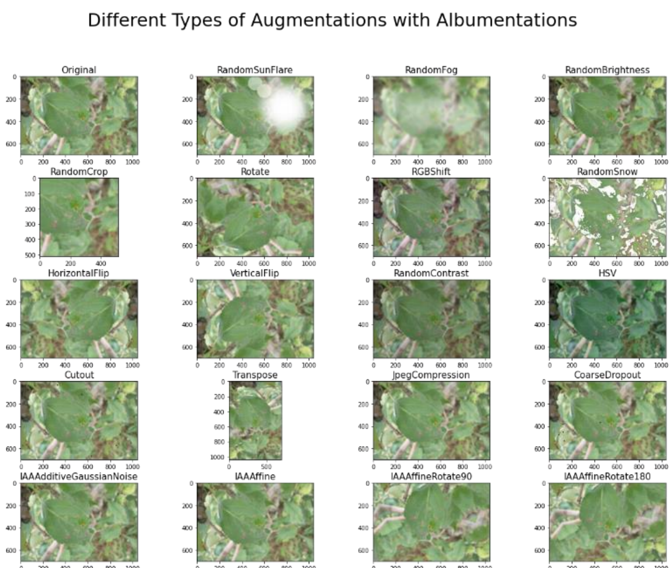
Duplicates in training datasets worsen the performance of the used model and this is because it causes biases to the duplicated images and hence, it would be more difficult for the model to generalize to new data. In addition, duplicates who are differently labeled would produce noise in the training phase. And so, Image hashing techniques were implemented to solve this issue. These include but not limited to Average hashing, Perceptual hashing, Difference hashing and Wavelet hashing. The Image Hash techniques rely on producing an identical hash for images which are similar with a small minor change. In [], these hashing methods were tested for different image modifications which include; contrast, scaling, brightness, gamma correlation, salt and pepper, gaussian smoothing, color adjustments and cropping. The result has shown that perceptual hashing is more robust against multiple types of errors with an error percentage of 23.7%, it has also shown good results in the average hamming distance however, it is calculation duration for each image is greater than almost other techniques. In this project, the perceptual hashing technique (phash) is chosen for its various advantages mentioned above. The phash works by calculating the gray value and then downsizing the image and calculating the discrete cosine transform (DCT) for each row and then for each column. Then, the average is calculated and then the pixels are scanned from left to right and compared with the mean. If it is larger then 1 is added for the hash, otherwise, zero is added..



C. Augmentation

Convolutional Neural Networks (CNNs) are usually used for scene understanding tasks such as image classification and object detection. One of the challenges that face these deep learning models is to gather and annotate enough training data. Different heuristic methods are typically used to solve overfitting such as DropOut, or early stopping the optimization algorithm. Apart from the regularization solutions, reducing overfitting can be obtained with data augmentation.

Data augmentation is the process of applying random yet realistic transformation to the images in the dataset to increase its diversity. Data augmentation is a critical component of training a deep learning model. Artificially increasing the dataset helps reducing the probability of overfitting and improves model's generalization. Simple image transformation like rotation and flipping can already improve the performance of the model in most vision tasks. However, using task specific augmentation will provide better gains. In our case, task specific augmentation can be flares, random noise, hue transformation, ...etc.

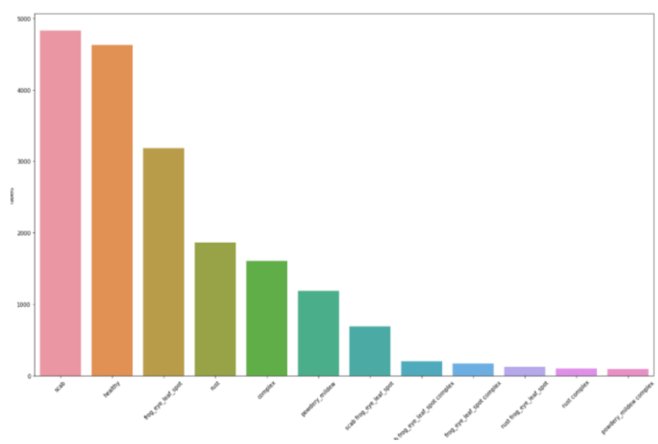


We built a data generator to apply image augmentation to our training set. In this case, the data augmentation will happen asynchronously on the CPU, and it will be non-blocking. This will enable us to overlap the training of our model on the GPU by prefetching.

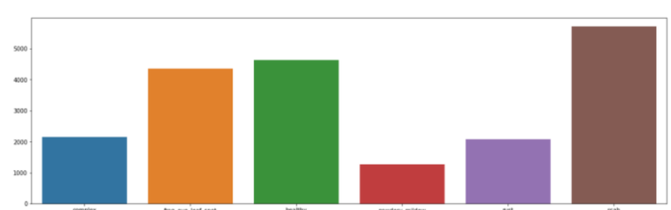
Our augmentation list included 17 different augmentation such as rotating, sun flare, contrast, and so on. We also applied Histogram Equalization to the images as it will provide more prominent images as shown in the figure below. Thus, after applying data augmentation, our training set was increased to be about 260K images.

In addition to data augmentation, we also tried to balance the dataset. The provided data contains 12 different classes for the images. When we plotted the number of classes, we found the results as the shown in the figure below. Notice that there is a huge imbalance in dataset with "scab" having the highest number of frequency and "powdery_mildew complex", the least; however, that was easy to fix. Since there are classes that are just a mix between different other classes, the problem becomes multilabel problem. Thus, there are just 6 abstract classes instead of 12:

1- Rust
2- Scab
3- Complex
4- From eye leaf spot
5- Powdery mildew
6- Healthy

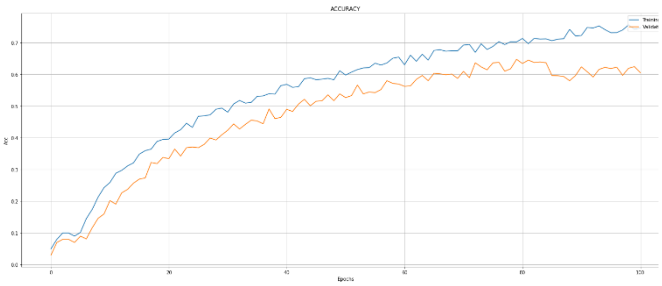


The next step was to use one-hot-encoding for the labels to represent the multilabel images. When we plot the results of reducing the number of classes, the figure below , we can observe that the data is now more balanced than before.

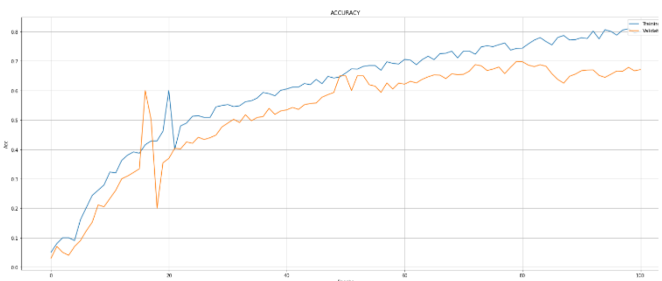


IV. Experiments and Results

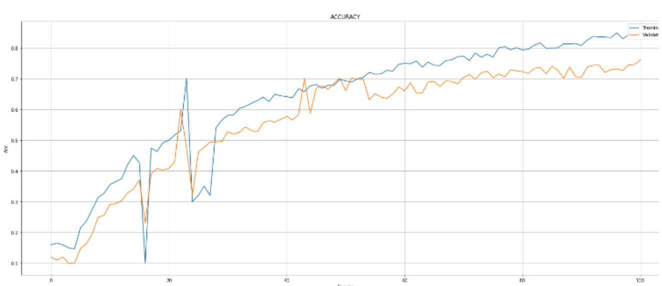
We have conducted 3 experiments to choose the best performing model. Frist, we used EfficientNet as a backbone and we froze its weights. The training took place only at the dense layers at the end of the pipeline. This produced accuracy of 63% without the augmented dataset.



The next experiment was similar to the first one but with the augmented dataset. This produced accuracy of 68% on the testing set. This experiment shows how critical augmentation is as it improved the accuracy by 5%.



Finally, trained the whole model end-to-end without freezing the weights of EfficientNet. We first trained our dense layer, then used it to fine tune the weights of EfficientNet. This was conducted in the augmented dataset, and it produced the highest accuracy of 72% on the test set.



The conclusion of the three experiments conducted is that augmentation and finetuning are important factors in enhancing the model accuracy. Augmentaion provides an efficient solution to the overfitting problem, and finetuning helps the backbone to adjust its weights based on the custom loss function of the application.

Experiment	Accuracy
Without finetuning and augmentation	63%
Without finetuning and with augmentation	68%
With finetuning and augmentation	**72%**

V. Future Work

There is a lot of potential in the data augmentation area, so one of our future strategies is to fine tune the augmentation to find the best combination the yields the best performance. An example of this is trying to include segmentation as one of our augmentations. Furthermore, special attention is a promising technique to increase the performance of vision models. Adding and attention layer to out model will make it easier for the CNN to focus on the important areas of the image. Finally, we will include ensemble to out pipeline as it will help our model avoid critical points while trying to converge.

VI. Conclusion

The results obtained by the experiments in this paper suggest that EfficientNet provides good results in vision problems; in addition, they show that data augmentation affect the results in a very desirable way as it increases the number of training examples. Our best achieving model yields accuracy of 72%, yet we assume that implementing the techniques recommended in the future work will improve the accuracy of the model.

References

[1] D. M. Sharath, S. A. Kumar, M. G. Rohan, Akhilesh, K. V. Suresh and C. Prathap, "Disease Detection in Plants using Convolutional Neural Network," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 389-394, doi: 10.1109/ICSSIT48917.2020.9214159.

[2] I. Z. Mukti and D. Biswas, "Transfer Learning Based Plant Diseases Detection Using ResNet50," 2019 4th International Conference on Electrical Information and Communication Technology (EICT), 2019, pp. 1-6, doi: 10.1109/EICT48899.2019.9068805.

[3] P. Jiang, Y. Chen, B. Liu, D. He and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 59069-59080, 2019, doi: 10.1109/ACCESS.2019.2914929.

[4] Y. Kashyap and S. S. Shrivastava, "Assessment of Soybean Leaf foliar Diseases using CNN & Weka tool," 2nd International Conference on Data, Engineering and Applications (IDEA), 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170719.

[5] R. Barosa, S. I. S. Hassen and L. Nagowah, "Smart Aquaponics with Disease Detection," 2019 Conference on Next Generation

Computing Applications (NextComp), 2019, pp. 1-6, doi: 10.1109/NEXTCOMP.2019.8883437.

[6] Srinidhi, V. V., Sahay, A., & Deeba, K. (2021). Plant Pathology Disease Detection in Apple Leaves Using Deep Convolutional Neural Networks : Apple Leaves Disease Detection using EfficientNet and DenseNet. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). doi:10.1109/iccmc51019.2021.9418268

[7] Plant Pathology 2021 - FGVC8. Kaggle. https://www.kaggle.com/c/plant-pathology-2021-fgvc8. Accessed 30 May 2021

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.