Predicting Potato Diseases Using Tensorflow in Mobile Apps Android

H Artanto and F Arifin
Department of Electronic and Informatics Engineering Education, Postgraduate, Yogyakarta State University, Yogyakarta, Indonesia
E-mail: herjunaartanto.2019@student.uny.ac.id

Abstract. Detecting disease is something that can help to determine how to cure it. Potato diseases, namely early blight and late blight, are the problem in agriculture. Both diseases have almost the same characteristics so it is difficult to recognize. The technique used by farmers only utilizes their eyesight. In addition, previous studies have mostly used the biological aspect approach. In this study utilizing Machine Learning to detect an object. The development of this application uses the training configuration used is SSD Mobilenet v1. Application test results obtained an average value of 90% can detect and distinguish the types of diseases in the potato plant.

1. Introduction
A disease in general can be treated if we know the type and how to cure it. Likewise with diseases in plants, we need to know the type of disease so that it can be treated in the right way. One way to find out the type of disease in plants is to observe the leaves. Potato plants give signs of disease to the leaves. The disease is early blight and late blight which can be recognized by looking at the leaves.

Early blight and late blight on potato plants is a type of disease that is often encountered by farmers. However, despite the name of the early and late disease does not necessarily come [2]. If at first glance there are also similarities so it is not easy to recognize. Early blight shows black marks on the leaves. While late blight also has black spots but is more widespread than early blight. Because the differences in the signs of disease are not so clear, so many farmers find it difficult to recognize.

Previous studies have been carried out using a biological approach. The solution provided is to identify signs of plant disease through vision. It still has many weaknesses because it only relies on vision and knowledge. Over time there are studies that use technology approaches to classify fruit types. However, there are no studies that use technology to solve the problem of potato plant diseases.

The development of technology has a good impact on human life. One technology that is developing rapidly is image processing and machine learning. Utilization of these technologies can be applied to help detect car license plate numbers via CCTV. It can also be applied to detect and recognize human faces. This does not rule out the possibility to be used to detect disease through image processing.

Machine learning is one of the tools in the deep learning method that is incorporated in the scope of artificial intelligence. It is said machine learning because learning is carried out by machines. In machine programming an application programming interface (API) can be used. Tensorflow API is one of the technologies that facilitates the development of machine learning.
processes include input processing, data training, data recognition, accuracy testing, and error correction.

![Figure 1. Difference of early blight leaves and late blight leaves](image)

In this research, an image processing and machine learning technology development was developed to recognize the types of diseases in potato plants. The image of a potato leaf is processed and used by the system for learning. The introduction of a type of disease to the system is a very decisive process. That is because the level of accuracy in recognizing the system is a measure of system success. Therefore, it needs high accuracy in the process.

2. **Proposed Algorithm**

Development of this type of disease detection in potato plants requires a lot of data on potato leaf images. In this study 100 data on potato leaf images were used consisting of healthy leaves, early blight, and late blight. As much as 80% of the data is used for system training and the rest is for testing. The data before being trained into the system is processed first so that it can be processed by TensorFlow. Following are the steps taken to develop an android application to detect potato diseases

![Figure 2. Development steps](image)
The initial step is data collection as in the previous explanation. The data used is a dataset that has been available on kaggle.com [4]. After the data is obtained then label images according to their description [5]. Figure 3 is the process of labeling the data of three types of leaves, namely early blight, late blight, and healthy leaves. Data labeling is done to all data used and then 80% is placed in the train folder and the rest is for testing. After being in the appropriate folder, convert the results of the labeling file into XML into CSV file.

![Figure 3. Labeling data](image)

After the file is divided into train and test folders along with the file that has been converted to CSV then create a labelmap. This labelmap intends to introduce the three data labeling results into Tensorflow in the form of TFrecord. The process of converting CSV files to TFrecord by editing the labelmap as shown in figure 5. When the data has been converted, the data is ready to be trained using Tensorflow.

![Figure 4. Result of conversion XML to CSV file](image)

![Figure 5. Configuration TFrecord conversion](image)

The training phase needs to configure the training pipeline first. The training pipeline used is SSD Mobilenet v1 0.75 Depth Quantized 300x300. This configuration is used because to be applied to Android, it can only use Mobilenet SSD [1]. This configuration uses the basis of the Convolutional Neural Network (CNN) training algorithm which is part of Deep learning and Machine learning [3]. Following are the details of the architectural model to be trained:

- **Box coder**: faster rcnn box coder
  - y scale: 10.0, x scale: 10.0, height scale: 5.0, width scale: 5.0
- **Matcher**: argmax matcher
- **Anchor generator**: ssd anchor generator
  - Layer: 6, min scale: 0.2, max scale: 0.95
- **Box predictor**: convolutional box predictor
Activation: RELU_6,

- Feature extractor: ssd mobilenet v1
  - Min depth: 16, depth multiplier: 0.75, activation: RELU_6

Figure 6. SSD Mobilenet architecture

A single convolutional network SSD used to answer an object. The SSD network is part of the CNN architecture as a feature extractor. In Figure 6, Mobilenet architecture uses an SSD as its extractor feature [3]. The detection takes only one shot for multiple objects in one image. So that the advantage of using an SSD is faster and more accurate even though it uses a small image input size [3]. The basic model of Mobilenet architecture is based on depthwise separable convolution and pointwise convolution. Layers The combined convolution, rail, and pooling in the Mobilenet architecture form 28 layers [1].

Then the training process will run for 2000 repetitions (epochs). However, when it reaches epochs to 506, a stop is made. The next step is to convert the training results with the appropriate error (loss) to the form of tflite graph. This graph form can already be checked for successful training using software on a laptop. The form still needs to be converted to tflite format so that it can be applied to Android. After getting the file with tflite format then it is copied to java folder on android along with the label file. The last step is to generate the apk file to deploy to the smartphone.

Figure 7. Configure to android apps

3. Experiment Simulation and Result Analysis

Based on the work steps performed, the mean average precision (mAP) and total error (loss) values are obtained. The mAP value is shown from the range 0 to 1. If the mAP value is 1 then the system can predict 100% correct. The mAP value obtained in the development of this application gets the final value on the 506th epochs close to 0.95. That means the system can predict an object of almost 95% correctly. Loss is the difference between the value obtained and the target value it should have.

The loss value is also shown in the range of 0 to 1. The smaller the loss value, the smaller the system error in predicting. The total loss value obtained in the development of this application gets the final value on the 506th epochs below 0.15. This means an error in detecting an object under 15%.
Application testing is done by detecting a sample leaf. A total of 20 test data will be provided to detect healthy leaves, early blight, and late blight. There are six healthy leaf test data, seven early blight leaf test data, and seven late blight leaf test data. The system tests the results of the training by detecting each of these leaf types. The following are the results of the test:

**Table 1. Result of testing healthy leaves**

| N | Leaves ID | Detection result (%) |
|---|-----------|----------------------|
| 1 | HL1965    | 93,75                |
| 2 | HL1935    | 77,34                |
| 3 | HL1762    | 90,23                |
| 4 | HL1898    | 85,55                |
| 5 | HL5399    | 84,38                |
| 6 | HL1813    | 89,45                |

**Table 2. Result of testing early blight leaves**

| N | Leaves ID | Detection result (%) |
|---|-----------|----------------------|
| 1 | EB7392    | 98,44                |
| 2 | EB6928    | 92,97                |
| 3 | EB7188    | 94,14                |
| 4 | EB7972    | 96,09                |
| 5 | EB7719    | 94,92                |
| 6 | EB7978    | 93,75                |
| 7 | EB7936    | 85,55                |

**Table 3. Result of testing late blight leaves**

| N | Leaves ID | Detection result (%) |
|---|-----------|----------------------|
| 1 | LB2554    | 97,66                |
| 2 | LB4416    | 95,31                |
| 3 | LB3950    | 96,09                |
| 4 | LB2746    | 96,88                |
| 5 | LB4833    | 99,22                |
| 6 | LB4784    | 97,66                |
| 7 | LB4414    | 97,66                |
Figure 10. Testing

The three tables above are all the results of tests on the detection of disease types in potato leaves. Based on the test results the apps can detected and obtained varying values. There is some low valued detection on healthy leaf. It is caused by several conditions including the use of training data and epochs that are few. In addition, the image quality used also has an effect. However, the overall test can show a high detection rate that is above 90%.

4. Conclusion
Based on the explanation above, it can be concluded that:
1) Detection of diseases in leaf plants can be done using Tensorflow through an application on android.
2) The quality of the data used for training influences detection results.
3) The number of repetitions (epochs) of the training also influences the detection results.

References
[1] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861v1. 2017
[2] W.E. Fry. The Canon of Potato Science: 10. Late Blight and Early Blight. Potato Research (2007) 50: 243-245
[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C-Y. Fu, A.C. Berg. SSD: Single Shot Multibox Detector. arXiv preprint arXiv:1512.02325v5
[4] T.O. Emmanuel. PlantVillage Dataset. https://www.kaggle.com/emmarex/plantdisease
[5] G. Tanner. Creating your own object detector with the Tensorflow Object Detection API. http://www.berttanner.com/blog/creating-your-own-objectdetector
[6] H.A. Afif. Object Detection Menggunakan Tensorflow-API. https://medium.com/@hafizhan.aliady/lihat-apa-yang-ada-di-box-hijau-begini-cara-membuat-object-detection-menggunakan-tensorflow-api-6d4a6d44e1a
[7] N. Toure. Convert a Tensorflow frozen graph to a TensorFlow lite (tflite) file (Part 3). https://medium.com/@teyou21/convert-a-tensorflow-frozen-graph-to-a-tflite-file-part-3-1ccdb3874c4a