Project Report

AI system for diagnosing plant diseases

74-Vedant Jumle | 83-Mandar Kasangottuwar | 81-Heer Kapadia

Problem statement:

Make a system that recognises plant diseases by taking images of plant leaves and gives a diagnosis about what disease the plant has.

Objective:

The objective of this project is to build a system that helps in diagnosis of plant diseases. It is particularly hard to train humans to identify differences between two different diseases of the same plant. Fortunately, most plant diseases have certain visual patterns that are developed on the leaves of the plant. Images of leaves can be taken for analysis by a classification algorithm that can built using various techniques of Machine Learning. In particular, 'Convolutional Neural Networks' are very useful to create this Artificially Intelligent system.

Scope:

The project only focuses on providing help in diagnosis of plant diseases. There will always be some margin of error in the algorithm and thus this is not a perfect system. This system is only designed for diagnosing 38 distinct diseases:

1. Apple:
    a. Apple Scab
    b. Black Rot
    c. Cedar Apple Rust
    d. Healthy
2. Blue berry healthy
3. Cherry:
    a. Healthy
    b. Powdery Mildew
4. Corn:
    a. Cercospora Leaf Spot (Grey Leaf Spot)
    b. Common Rust
    c. Northern Leaf Blight
    d. Healthy
5. Grape:
    a. Black Rot
    b. Esca (Black Measles)
    c. Leaf Blight (Isariopsis Leaf Spot)
    d. Healthy
6. Orange Haunglongbing(Citrus Greening)
7. Peach:
    a. Bacterial Spot
    b. Healthy
8. Bell Pepper (Capsicum):
    a. Bacterial Spot
    b. Healthy

9. Potato:
    a. Early Blight
    b. Late Blight
    c. Healthy
10. Raspberry Healthy
11. Soybean Healthy
12. Squash Powdery Mildew
13. Strawberry:
    a. Leaf Scorch
    b. Healthy
14. Tomato:
    a. Bacterial Spot
    b. Early Blight
    c. Late Blight
    d. Leaf Mould
    e. Septoria Leaf Spot
    f. Spider Mites (Two-spotted spider mite)
    g. Target Spot
    h. Mosiac Virus
    i. Yellow Leaf Curl Virus

This is system will be made robust enough so that new additions to repertoire will just require adding enough data to the dataset and training the system on the new data. The machine learning part of this system will be built using TensorFlow which is an open-source library for Machine Learning and Artificial Intelligence.

Literature Review:

- The Field of Applied Machine Learning is an ever-growing field with multiple approaches to solving the same problem.
- There were many technical papers and articles [1][2][3][4] which applied variations different techniques: Neural Architecture, Training Loops, Data Argumentation, and Transfer Learning.
- All the paper that we reviewed only discussed about how to build an AI model, but no one really discussed on how something like this can be deployed in a real setting.
- Our ML model takes in the consideration that our dataset is relatively large (=87867 images) for 38 total classes. Hence, the model architecture was build to ensure optimal performance.
- For the transfer learning aspect of the model, Inception V3 was used, which is trained on the ImageNet Dataset.

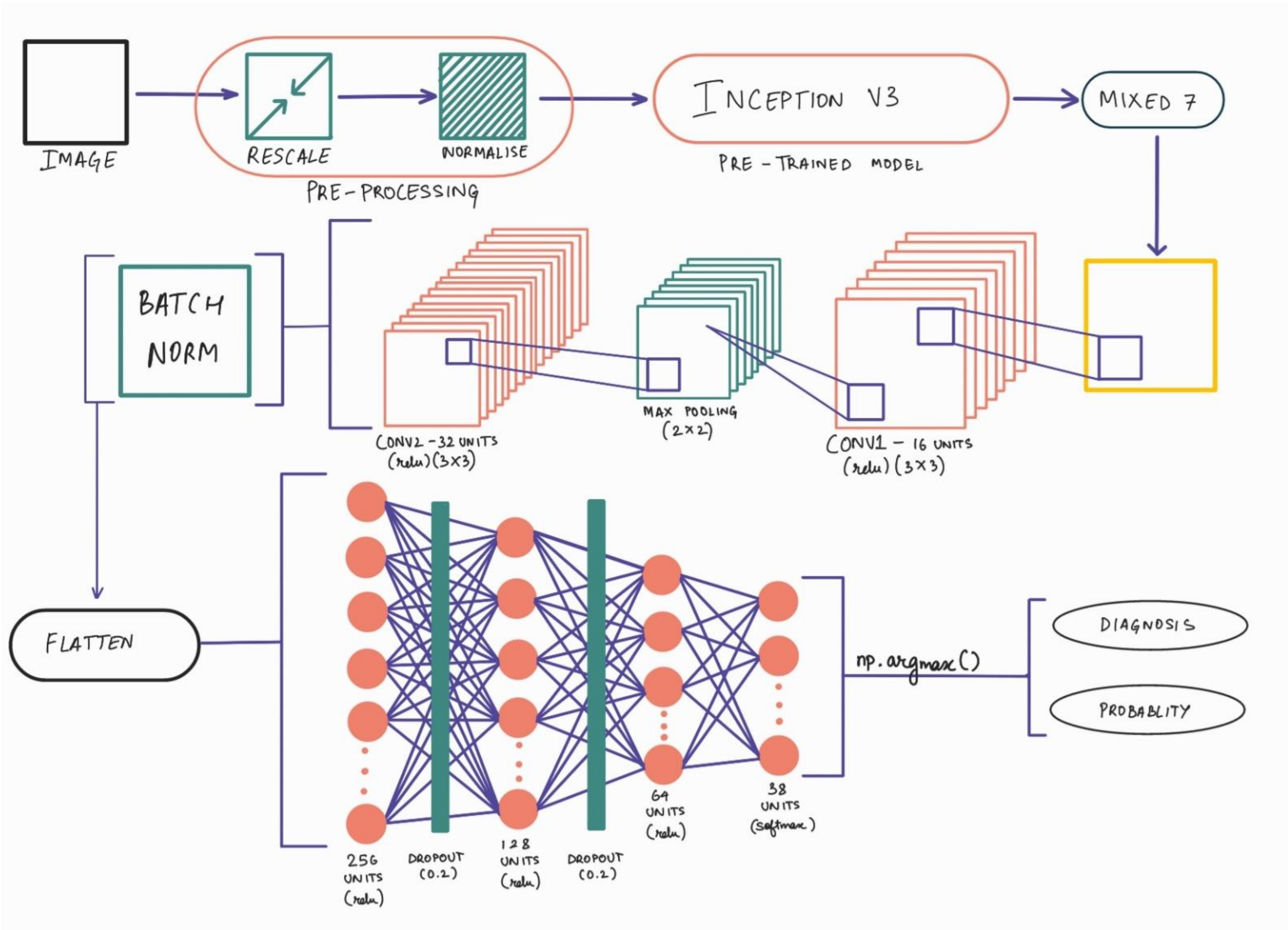
[1]S. Mohanty, D. Hughes and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection", Frontiers in Plant Science, vol. 7, 2016. Available: 10.3389/fpls.2016.01419 [Accessed 1 October 2021].
[2]R. Khan, K. Khan, W. Albattah and A. Qamar, "Image-Based Detection of Plant Diseases: From Classical Machine Learning to Deep Learning Journey", Wireless Communications and Mobile Computing, vol. 2021, pp. 1-13, 2021. Available: 10.1155/2021/5541859 [Accessed 1 October 2021].
[3]J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review", Plant Methods, vol. 17, no. 1, 2021. Available: 10.1186/s13007-021-00722-9 [Accessed 1 October 2021].
[4]D. Radovanovic and S. Dukanovic, "Image-Based Plant Disease Detection: A Comparison of Deep Learning and Classical Machine Learning Algorithms", 2020 24th International Conference on Information Technology (IT), 2020. Available: 10.1109/it48810.2020.9070664 [Accessed 1 October 2021].

Project outline:

1. Overview:

   This project will have two components:
   - An API that handles all the Artificial Intelligence part of the system.
   - A User Interface that will use the API to act as a bridge between the user and the AI.

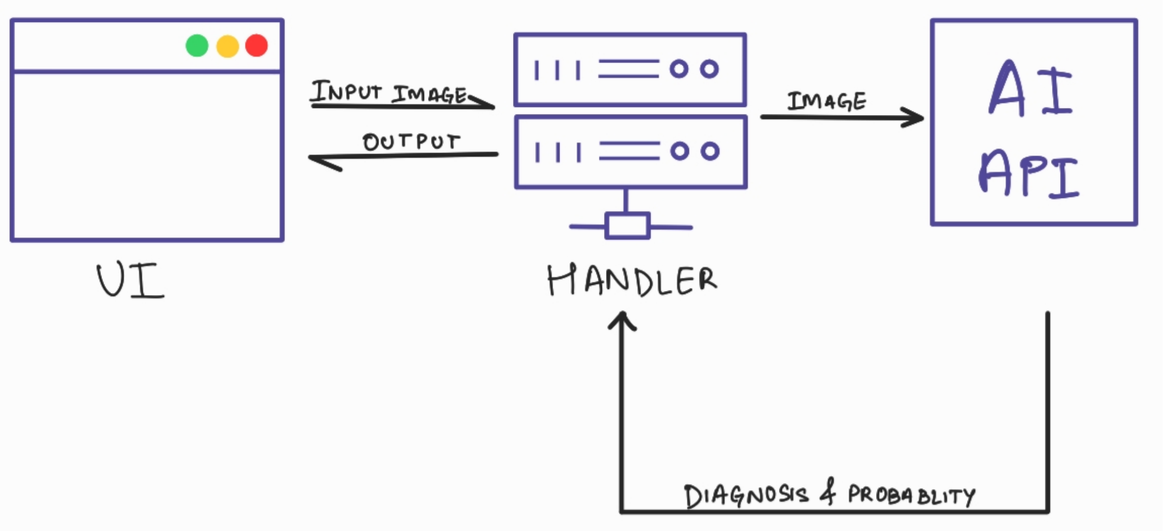
2. AI architecture:



The AI is build using a technique of Machine Learning called *Transfer Learning*. This technique basically means that we use a pre-trained model (in our case Inception V3) which has been trained on the *ImageNet* dataset, which is a dataset that consists of 1.4 million hand labelled images. Our AI model uses the output of the '*mixed 7'* layer of the model which is then passed to an untrained network which is fine tuned on our dataset. The Inception V3 model acts as a helping hand to easily recognise reoccurring patterns in the images and hence reducing training time and gives very accurate results.

It is important to mention that the trainable part of the model was evaluated on *categorical crossentropy loss function* and the backpropagation was optimized by using the *adam optimizer*. The final layer in the DNN is particularly interesting. This is because it consists of 38 units which are all activated using the *SoftMax function.* This is a widely used activation function that converts all activations of units into probabilities that sun to 1 over all the units.

3. System Outline:

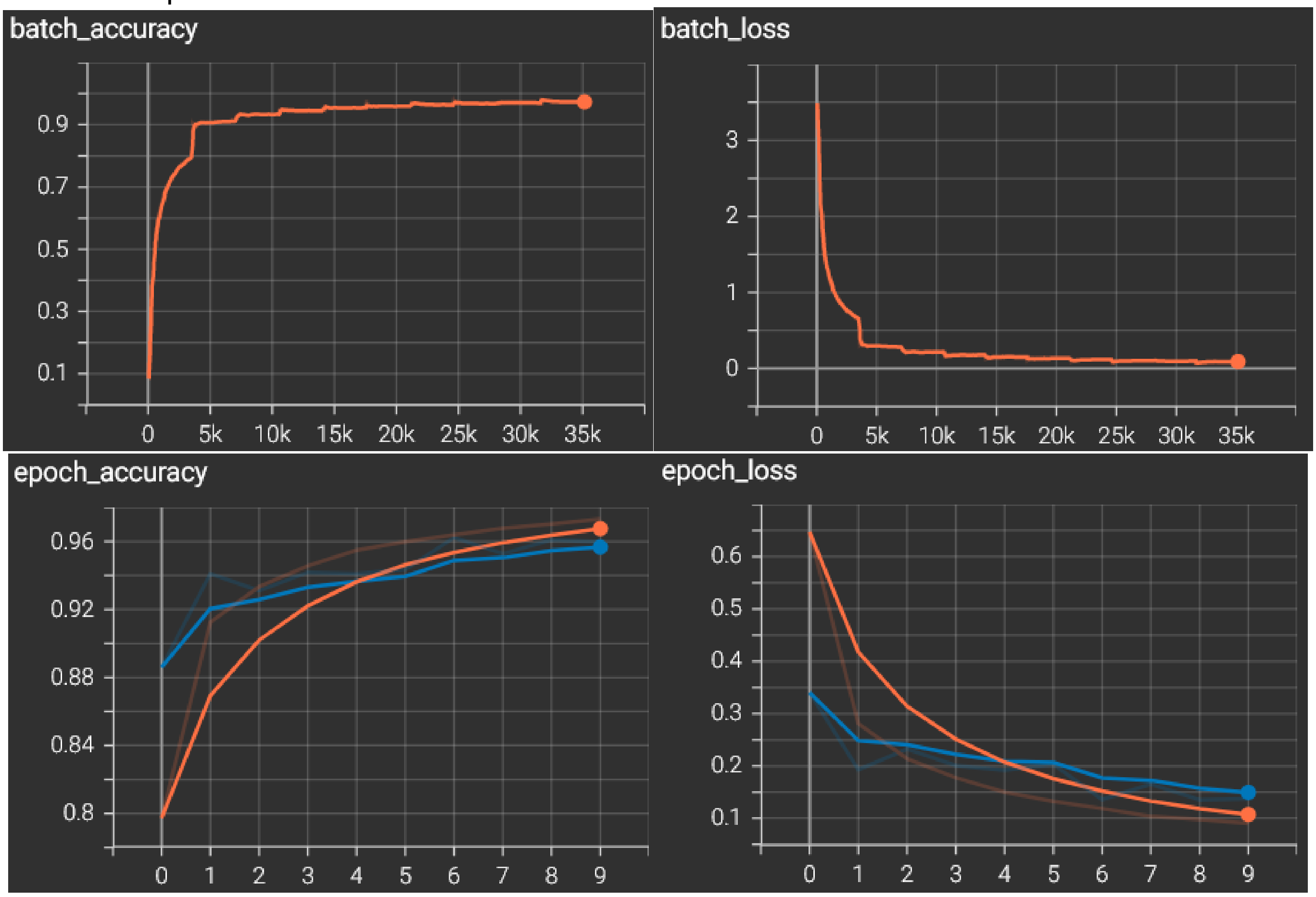


AI performance:

The AI algorithm used in this project ran for 10 epochs. At the end these were the results:

1. Training Loss: ~0.0865
2. Training Accuracy: ~97.47%
3. Validation Loss: ~0.1375
4. Validation Accuracy: ~95.97%

Here are the plots for the same:



NOTE:
Blue line corresponds to validation set.
Orange line corresponds to training set.

References:

Dataset (Kaggle): link
Convolutional Neural Networks (original paper): link
TensorFlow: link
Pandas: link
Numpy: link
Inception Model: link