

MODEL INFERENCING PIPELINE/CONCEPTUAL DIAGRAM

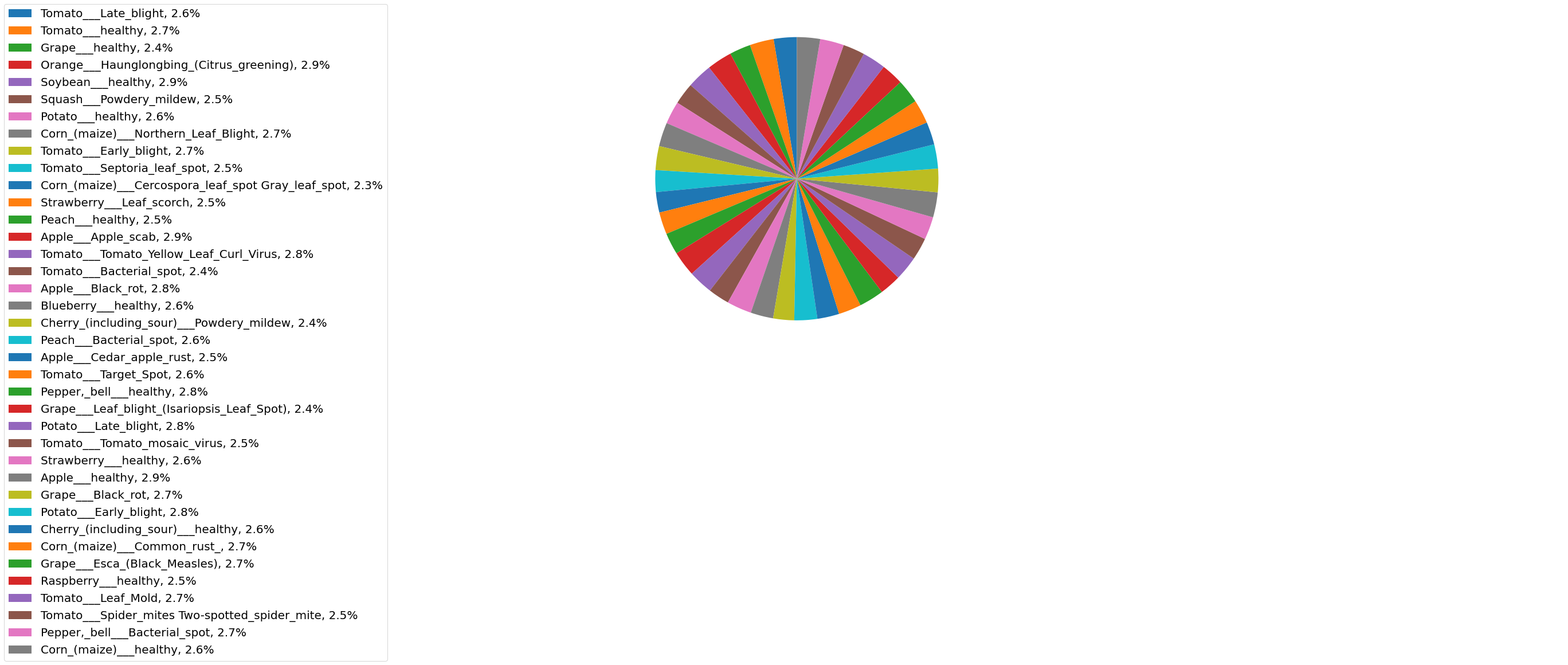
EXPERIMENTAL PROCEDURE DATA

The data was sourced from Kaggle. A popular site where a lot of datasets are hosted for experimental challenges.

This dataset is recreated using offline augmentation from the original dataset. The original dataset can be found on [this](https://github.com/spMohanty/PlantVillage-Dataset" \t "_blank) github repo. This dataset consists of about 87K rgb images of healthy and diseased crop leaves which is categorized into 38 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purpose.

This 38 classes are listed below

* Tomato\_\_\_Late\_blight
* Tomato\_\_\_healthy
* Grape\_\_\_healthy
* Orange\_\_\_Haunglongbing\_(Citrus\_greening)
* Soybean\_\_\_healthy
* Squash\_\_\_Powdery\_mildew
* Potato\_\_\_healthy
* Corn\_(maize)\_\_\_Northern\_Leaf\_Blight
* Tomato\_\_\_Early\_blight
* Tomato\_\_\_Septoria\_leaf\_spot
* Corn\_(maize)\_\_\_Cercospora\_leaf\_spot Gray\_leaf\_spot
* Strawberry\_\_\_Leaf\_scorch
* Peach\_\_\_healthy
* Apple\_\_\_Apple\_scab
* Tomato\_\_\_Tomato\_Yellow\_Leaf\_Curl\_Virus
* Tomato\_\_\_Bacterial\_spot
* Apple\_\_\_Black\_rot
* Blueberry\_\_\_healthy
* Cherry\_(including\_sour)\_\_\_Powdery\_mildew
* Peach\_\_\_Bacterial\_spot
* Apple\_\_\_Cedar\_apple\_rust
* Tomato\_\_\_Target\_Spot
* Pepper,\_bell\_\_\_healthy
* Grape\_\_\_Leaf\_blight\_(Isariopsis\_Leaf\_Spot)
* Potato\_\_\_Late\_blight
* Tomato\_\_\_Tomato\_mosaic\_virus
* Strawberry\_\_\_healthy
* Apple\_\_\_healthy
* Grape\_\_\_Black\_rot
* Potato\_\_\_Early\_blight
* Cherry\_(including\_sour)\_\_\_healthy
* Corn\_(maize)\_\_\_Common\_rust\_
* Grape\_\_\_Esca\_(Black\_Measles)
* Raspberry\_\_\_healthy
* Tomato\_\_\_Leaf\_Mold
* Tomato\_\_\_Spider\_mites Two-spotted\_spider\_mite
* Pepper,\_bell\_\_\_Bacterial\_spot
* Corn\_(maize)\_\_\_healthy



A DIAGRAM SHOWING THE DISTRIBUTION OF CLASSES IN THE DATASET

From the diagram above the dataset looks balanced and evenly distributed.

**DATA PREPROCESSING**

The dataset contains images of shape (224,224,3) where the last dimension specifies the number of channels corresponding to red, blue and green. Each channel contains numbers ranging from 0 to 255.

We normalize this channels from (0…255) to (0…1) by dividing them all by 255.

We also applied some transformations randomly on the images.

We use Tensorflow image data generators to fetch the dataset and perform preprocessing and augmentation.

By specify some parameters, we can tell the image generator how we want the augmentation and preprocessing to be done.

These parameters include

* Zoom\_range

Float or [lower, upper]. Range for random zoom. If a float, [lower, upper] = [1-zoom\_range, 1+zoom\_range].

We used this parameter to randomly zoom into some images by 2%

* Horizontal\_flip

Boolean. Randomly flip inputs horizontally.

* Shear\_range

This means some images would be distorted along the x-axis. This parameter was set to 0.2

* Rescale

This parameter is used to specify how we want to normalize the images.

Here we just give it a value of 1/255. Telling it that we would want to divide everything by 255.

**METHODOLOGY**

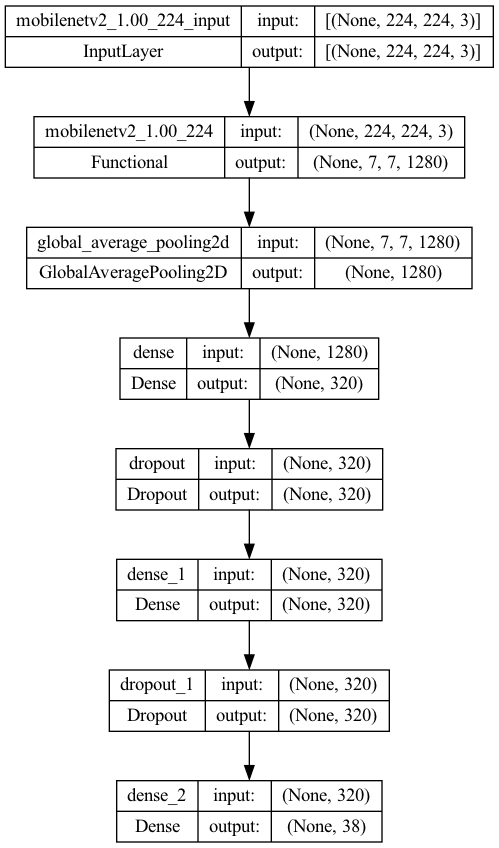
We use a pretrained model known as mobilenetv2 which was trained on the imagenet dataset.

The model has a feature extraction and classification layer.

The feature extraction layer uses CONVOLUTIONAL NEURAL NETWORKS to get the features before passing them to the classification layers which are a series of FULLY CONNECTED LAYERS.

We dropped the classification layer which originally was outputting 1000 classes and added our own classification layer on top of the previous feature extraction layer.

Our classification layer produces 38 outputs corresponding to the number of classes in our dataset.



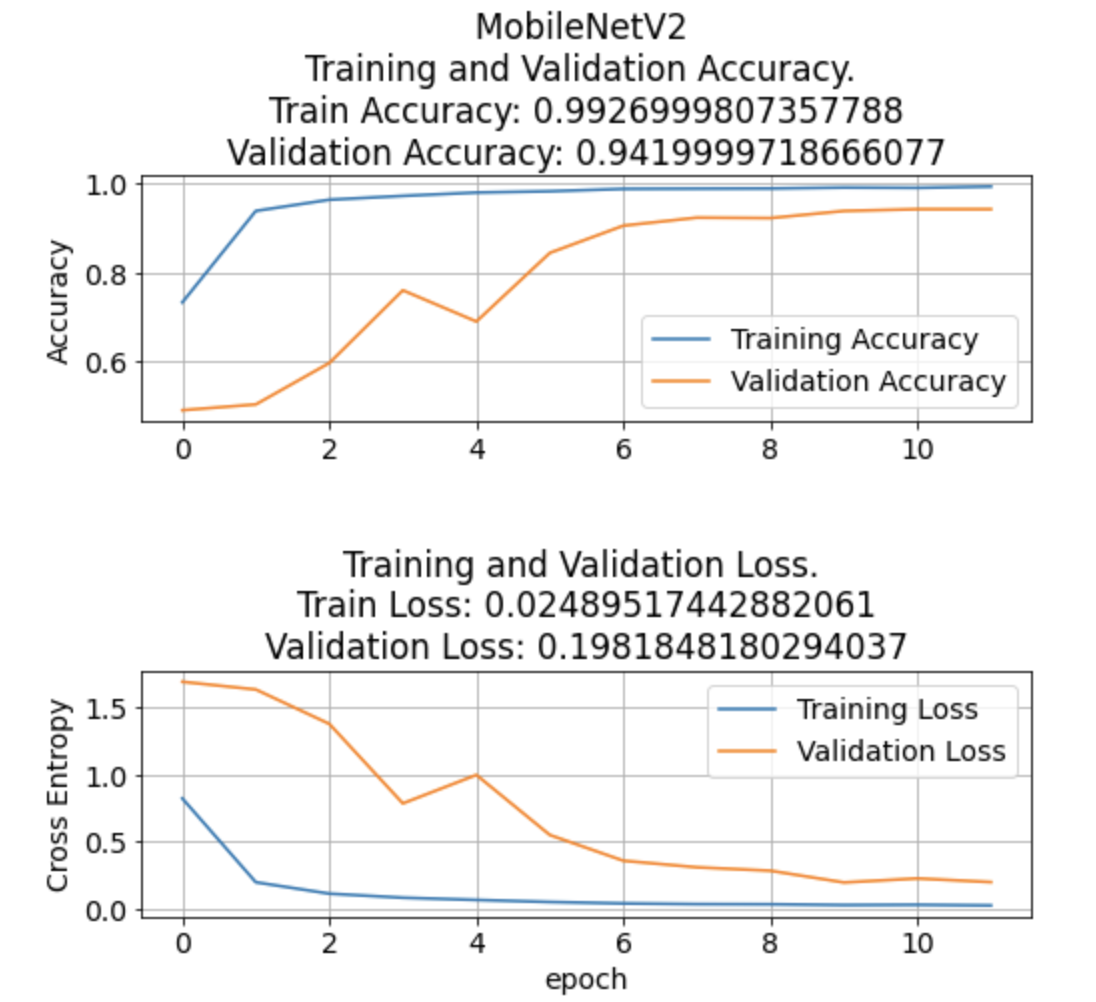
A DIAGRAM SHOWING OUR MODEL

The first 3 layers are the feature extraction layers gotten from our pretrained model.

EXPERIMENTAL SETUP

The dataset has already been split into 80 percent train and 20 percent validation.

We use 2 percent of the validation dataset for testing.



The model produced very amazing test results with a validation accuracy of 0.99

These is the classification report generated on the test dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASS | F1-sore | support | precision | recall |
| Tomato\_\_\_healthy | 1.0 | 5 | 1.0 | 1.0 |
| Grape\_\_\_healthy | 1.0 | 1 | 1.0 | 1.0 |
| Orange\_\_\_Haunglongbing\_(Citrus\_greening) | 1.0 | 4 | 1.0 | 1.0 |
| Soybean\_\_\_healthy | 1.0 | 4 | 1.0 | 1.0 |
| Squash\_\_\_Powdery\_mildew | 1.0 | 3 | 1.0 | 1.0 |
| Potato\_\_\_healthy | 1.0 | 3 | 1.0 | 1.0 |
| Corn\_(maize)\_\_\_Northern\_Leaf\_Blight | 1.0 | 6 | 1.0 | 1.0 |
| Tomato\_\_\_Early\_blight | 0.95 | 5 | 0.83 | 1.0 |
| Tomato\_\_\_Septoria\_leaf\_spot | 0.92 | 7 | 1.0 | 0.86 |
| Corn\_(maize)\_\_\_Cercospora\_leaf\_spot | 1.0 | 4 | 1.0 | 1.0 |
| Strawberry\_\_\_Leaf\_scorch | 1.0 | 5 | 1.0 | 1.0 |

Confusion matrix generated for test dataset.

It is a perfect diagonal matrix

