1. Total number of Images?

22930, compressed into 224\*224

2. Total number of diseases?

10 diseases

3. Total number of images of each disease?

training data set:18345

tomato bacterial spot: 1702

tomato early blight:1920

tomato healthy:1926

tomato late blight:1851

tomato leaf mold:1882

tomato septoria leaf spot:1745

tomato spider mites two spotted spider mite:1741

tomato target spot:1827

tomato tomato mosaic virus:1790

tomato tomato yellow leaf curl virus:1961

valid data set:4585

tomato bacterial spot: 425

tomato early blight:480

tomato healthy:481

tomato late blight:463

tomato leaf mold:470

tomato septoria leaf spot:436

tomato spider mites two spotted spider mite:435

tomato target spot:457

tomato tomato mosaic virus:448

tomato tomato yellow leaf curl virus:490

4. How did you divide the data set in training and testing w.r.t each disease?

Training 18345 testing 4585

5. Training and testing are performed using different combinations ([60%, 40%], [70%,30%], [80%,20%])

Accuracy :92.1%[80%,20%]

Computation time :20(165min),10(135 min)

Confusion matrix :Confusion Matrix[80 %,20%]

[[45 52 37 30 51 37 45 48 38 42]

[60 68 51 38 47 36 58 36 44 42]

[53 56 45 35 36 46 63 38 46 45]

[50 73 49 26 40 46 56 44 46 40]

[47 57 35 23 34 47 50 48 42 53]

[57 50 34 28 40 39 43 47 48 49]

[60 64 33 30 31 37 59 40 46 57]

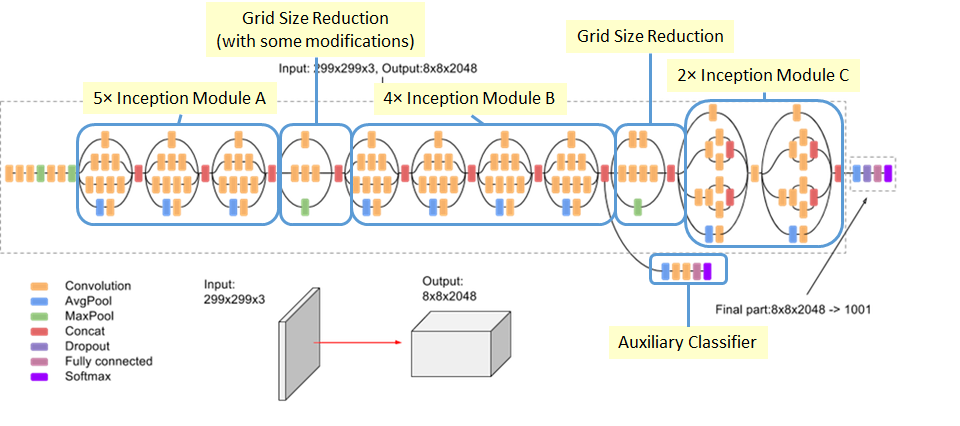
[51 62 46 40 36 53 65 47 49 41]

[58 49 39 38 43 39 50 41 47 44]

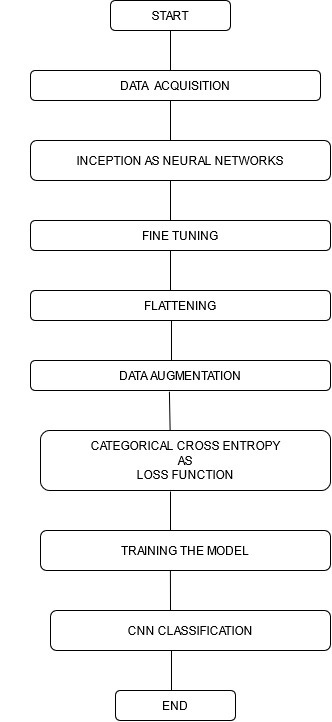
[60 60 46 35 33 35 63 50 46 53]]

6. Explain Complete methodology :

A.Transfer learning allows you to retrain the final layer of an existing model, resulting in a significant decrease in not only training time, but also the size of the dataset required. One of the most famous models that can be used for transfer learning is Inception V3. This model was originally trained on over a million images from 1,000 classes on some very powerful machines. Being able to retrain the final layer means that you can maintain the knowledge that the model had learned during its original training and apply it to your smaller dataset, resulting in highly accurate classifications without the need for extensive training and computational power.



7.Flow chart explanation :



**Description of flowchart-**

**1. Data acquisition**

Data acquisition is the way toward inspecting signals that action genuine states of being and changing over the subsequent examples into computerized numeric qualities that can be controlled by a PC.

**2. Inception Networks-**

When planning a convolutional neural organization, we frequently need to pick the kind of layer: CONV, POOL or FC. The commencement layer does them all. The consequence of the relative multitude of activities is then linked in a solitary square.

**3. Fine Tuning**

Unfreeze a few of the top layers of a frozen model base and Unfreeze a couple of the top layers of a frozen model base and together train both the recently added classifier layers and the last layers of the base model. This permits us to "fine-tune" the higher-request to include portrayals in the base model to make them more applicable for the particular assignment.

**4. Data Augmentation**

Image data augmentation is a method that can be utilized to falsely grow the size of a preparation dataset by making adjusted renditions of images in the dataset. Data augmentation is a procedure to falsely make new preparing data from existing preparing data. This is finished by applying space explicit procedures to models from the preparation data that make new and diverse preparing models. Image data augmentation is maybe the most notable kind of data augmentation and includes making changed adaptations of images in the preparation dataset that have a place with a similar class as the first image.

Changed incorporate a scope of tasks from the field of image control, like movements, flips, zooms, and considerably more. The expectation is to extend the preparation dataset with new, conceivable models. This implies, varieties of the preparation set images that are probably going to be seen by the model. Present day profound learning calculations, for example, the convolutional neural organization, or CNN, can learn highlights that are invariant to their area in the image. All things considered, augmentation can additionally help in this change invariant way to deal with learning and can help the model in learning highlights that are likewise invariant to changes. Image data augmentation is ordinarily simply applied to the preparation dataset, and not to the approval or test dataset. This is unique in relation to data arrangement, for example, image resizing and pixel scaling; they should be performed reliably across all datasets that associate with the model [14].

**5.Categorical cross entropy**

Categorical cross entropy is a loss function that is utilized in multi-class order errands. These are errands where a model can just have a place with one out of numerous potential classes, and the model should choose which one. Officially, it is intended to measure the contrast between two likelihood circulations..

**6. Training the model**

Training a model basically implies picking up (deciding) great qualities for every one of the loads and the predisposition from marked models. In regulated learning, an AI calculation fabricates a model by inspecting numerous models and endeavouring to track down a model that limits misfortune; this interaction is called exact risk reduction or minimising failure.

**7. CNN Classification**

The basic CNN classifier is explained in the image below.

Data acquisition :importing all necessary packages like keras,tensor flow

Pre-processing:inception,flattening,fine tuning,compiling the model

7. Cnn (Complete operations along with its layers)

1. Number of layers, - more than 48
2. Number of neurons on each layer-
3. Operation (Convolution, Operation, Filter, activation function applied on each layer )

Convolution:Inception

Filter:conv2d,max pooling,edge detection

activation:Softmax

8. Results and analysis

Without pre-processing

Epoch 20:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Disease Name | Number of images | Success rate % | Average error % | Number of images applied during testing |
| Disease 0 | 425 | 0 | 100 | 5 |
| Disease 1 | 480 | 0 | 100 | 5 |
| Disease 2 | 481 | 0 | 100 | 5 |
| Disease 3 | 463 | 0 | 100 | 5 |
| Disease 4 | 470 | 0 | 100 | 5 |
| Disease 5 | 436 | 20 | 80 | 5 |
| Disease 6 | 435 | 0 | 100 | 5 |
| Disease 7 | 457 | 0 | 100 | 5 |
| Disease 8 | 448 | 0 | 100 | 5 |
| Disease 9 | 490 | 0 | 100 | 5 |

After pre-processing

Epochs:10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Disease Name | Number of images | Success rate | Average error | no of images applied during testing |
| Disease 0 | 425 | 100 | 0 | 5 |
| Disease 1 | 480 | 100 | 0 | 5 |
| Disease 2 | 481 | 100 | 0 | 5 |
| Disease 3 | 463 | 40 | 60 | 5 |
| Disease 4 | 470 | 40 | 60 | 5 |
| Disease 5 | 436 | 60 | 40 | 5 |
| Disease 6 | 435 | 80 | 20 | 5 |
| Disease 7 | 457 | 100 | 0 | 5 |
| Disease 8 | 448 | 80 | 20 | 5 |
| Disease 9 | 490 | 100 | 0 | 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Overall success rate | Average error rate | Number of epoch | Time |
| Before Pre-processing | 0.2 | 9.8 | 20 | 40min |
| After pre-processing | 8.0 | 2.0 | 10 | 135min |

CNN Information

|  |  |
| --- | --- |
| Epoch | 20 |
| Batch size | 16 |
| Momentum |  |
| Weight decay |  |
| Learning |  |

9. Corresponding to each disease display the data with its picture.

After Preprocessing

**tomato bacterial spot**: 425



[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 1 |
| 1 | tomato early blight | 0 |
| 2 | tomato late blight | 0 |
| 3 | tomato leaf mold | 0 |
| 4 | tomato septoria leaf spot | 0 |
| 5 | tomato spider mites two spotted spider mite | 0 |
| 6 | tomato target spot | 0 |
| 7 | tomato tomato yellow leaf curl virus | 0 |
| 8 | tomato tomato mosaic virus | 0 |
| 9 | tomato healthy | 0 |

**tomato early blight**:480



[[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]]

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0 |
| 1 | tomato early blight | 1 |
| 2 | tomato late blight | 0 |
| 3 | tomato leaf mold | 0 |
| 4 | tomato septoria leaf spot | 0 |
| 5 | tomato spider mites two spotted spider mite | 0 |
| 6 | tomato target spot | 0 |
| 7 | tomato tomato yellow leaf curl virus | 0 |
| 8 | tomato tomato mosaic virus | 0 |
| 9 | tomato healthy | 0 |

**tomato late blight**:463



[0.0000000e+00 4.9230186e-22 1.0000000e+00 0.0000000e+00 0.0000000e+00

0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00]

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0.0000000e+00 |
| 1 | tomato early blight | 4.9230186e-22 |
| 2 | tomato late blight | 1.0000000e+00 |
| 3 | tomato leaf mold | 0.0000000e+00 |
| 4 | tomato septoria leaf spot | 0.0000000e+00 |
| 5 | tomato spider mites two spotted spider mite | 0.0000000e+00 |
| 6 | tomato target spot | 0.0000000e+00 |
| 7 | tomato tomato yellow leaf curl virus | 0.0000000e+00 |
| 8 | tomato tomato mosaic virus | 0.0000000e+00 |
| 9 | tomato healthy | 0.0000000e+00 |

**tomato leaf mold**:470



[[1.5966359e-34 0.0000000e+00 0.0000000e+00 6.5872800e-01 3.4127200e-01

0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00]]

array([3], dtype=int64)

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 1.5966359e-34 |
| 1 | tomato early blight | 0.0000000e+00 |
| 2 | tomato late blight | 0.0000000e+00 |
| 3 | tomato leaf mold | 6.5872800e-01 |
| 4 | tomato septoria leaf spot | 3.4127200e-01 |
| 5 | tomato spider mites two spotted spider mite | 0.0000000e+00 |
| 6 | tomato target spot | 0.0000000e+00 |
| 7 | tomato tomato yellow leaf curl virus | 0.0000000e+00 |
| 8 | tomato tomato mosaic virus | 0.0000000e+00 |
| 9 | tomato healthy | 0.0000000e+00 |

**tomato septoria leaf spot**:436



[[0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]]array([4]

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0 |
| 1 | tomato early blight | 0 |
| 2 | tomato late blight | 0 |
| 3 | tomato leaf mold | 0 |
| 4 | tomato septoria leaf spot | 1 |
| 5 | tomato spider mites two spotted spider mite | 0 |
| 6 | tomato target spot | 0 |
| 7 | tomato tomato yellow leaf curl virus | 0 |
| 8 | tomato tomato mosaic virus | 0 |
| 9 | tomato healthy | 0 |

**tomato spider mites two spotted spider mite**:435



[[0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00

1.0000000e+00 1.7044605e-23 0.0000000e+00 0.0000000e+00 0.0000000e+00]]

array([5], dtype=int64)

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0.0000000e+00 |
| 1 | tomato early blight | 0.0000000e+00 |
| 2 | tomato late blight | 0.0000000e+00 |
| 3 | tomato leaf mold | 0.0000000e+00 |
| 4 | tomato septoria leaf spot | 0.0000000e+00 |
| 5 | tomato spider mites two spotted spider mite | 1.0000000e+00 |
| 6 | tomato target spot | 1.7044605e-23 |
| 7 | tomato tomato yellow leaf curl virus | 0.0000000e+00 |
| 8 | tomato tomato mosaic virus | 0.0000000e+00 |
| 9 | tomato healthy | 0.0000000e+00 |

**tomato target spot**:457



[[0.0000000e+00 9.0116561e-26 0.0000000e+00 0.0000000e+00 1.6694503e-20

3.3591944e-21 1.0000000e+00 0.0000000e+00 0.0000000e+00 1.3849997e-28]]

array([6], dtype=int64)

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0.0000000e+00 |
| 1 | tomato early blight | 9.0116561e-26 |
| 2 | tomato late blight | 0.0000000e+00 |
| 3 | tomato leaf mold | 0.0000000e+00 |
| 4 | tomato septoria leaf spot | 1.6694503e-20 |
| 5 | tomato spider mites two spotted spider mite | 3.3591944e-21 |
| 6 | tomato target spot | 1.0000000e+00 |
| 7 | tomato tomato yellow leaf curl virus | 0.0000000e+00 |
| 8 | tomato tomato mosaic virus | 0.0000000e+00 |
| 9 | tomato healthy | 1.3849997e-28] |

**tomato tomato yellow leaf curl virus**:490



[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0 |
| 1 | tomato early blight | 0 |
| 2 | tomato late blight | 0 |
| 3 | tomato leaf mold | 0 |
| 4 | tomato septoria leaf spot | 0 |
| 5 | tomato spider mites two spotted spider mite | 0 |
| 6 | tomato target spot | 0 |
| 7 | tomato tomato yellow leaf curl virus | 1 |
| 8 | tomato tomato mosaic virus | 0 |
| 9 | tomato healthy | 0 |

**tomato tomato mosaic virus**:448



[[0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]]

array([8], dtype=int64)

|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0 |
| 1 | tomato early blight | 0 |
| 2 | tomato late blight | 0 |
| 3 | tomato leaf mold | 0 |
| 4 | tomato septoria leaf spot | 0 |
| 5 | tomato spider mites two spotted spider mite | 0 |
| 6 | tomato target spot | 0 |
| 7 | tomato tomato yellow leaf curl virus | 0 |
| 8 | tomato tomato mosaic virus | 1 |
| 9 | tomato healthy | 0 |

**tomato healthy**:481



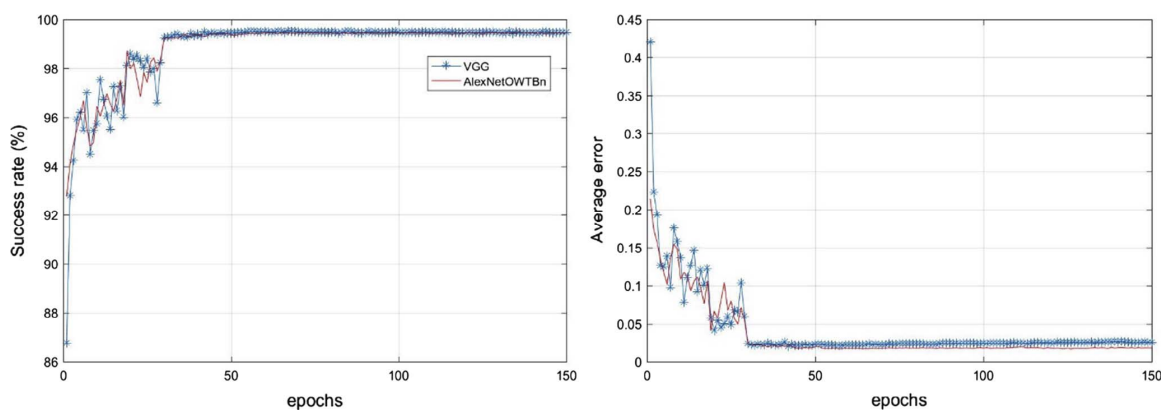
|  |  |  |
| --- | --- | --- |
| Rank | Class | Certainty |
| 0 | tomato bacterial spot | 0 |
| 1 | tomato early blight | 0 |
| 2 | tomato late blight | 0 |
| 3 | tomato leaf mold | 0 |
| 4 | tomato septoria leaf spot | 0 |
| 5 | tomato spider mites two spotted spider mite | 0 |
| 6 | tomato target spot | 0 |
| 7 | tomato tomato yellow leaf curl virus | 0 |
| 8 | tomato tomato mosaic virus | 0 |
| 9 | tomato healthy | 1 |





10 .Success rates and average errors of the final models on the testing dataset, during training.

Example



11. Python packages used for experiment. Hardware configure

Python packages:Keras,tensorflow

Hardware:Windows xp