

Prediction of Rice Diseases Using Convolutional Neural Network (in Rstudio)

A. Sony
Dept. of CSE

Sreenidhi Institute of Science and Technology

Abstract:- Detection of rice pest and diseases, and proper management and control of pest infested rice fields may result to a higher rice crop production. Using modern technologies, like smart phones, farmers can be aided in detecting and identifying the type of pests and diseases found in their rice fields. Convolutional Neural Network using r language are used to find the diseases in rice by using images of disease leaves. The disease images are collected from the UCI Machine Learning Repository contain the three types of diseases namely Bacterial Leaf Blight, Brown Spot, Leaf Smut.

Keywords:- Convolutional Neural Network, Rstudio, Paddy Image Dataset.

I. INTRODUCTION

Rice is the world's most important food crop and furnish food for more than half of the population. In the last 20 yrs, 3 lakhs farmers have committed suicides, due to yield loss. Farmers lose an estimated average of 37% of their rice crop due to pests and diseases every year. So, identifying, analysing, predicting the disease must be fast to upgrade production. The diseases are mostly caused by Bacteria, Fungus, Virus and others like mosquito, green hoppers etc. Mostly occurred bacteria diseases are Bacterial Leaf Blight, foot rot etc. and fungus are blast, brown spot, narrow brown spot, leaf smut, false smut etc. and Rice tungro is more occurred virus disease.

Convolutional Neural Network algorithm in r plays a key role to improve the accuracy for detecting diseases. R is a programming language which is used for many statistical modelling and data science tasks, it is dynamically typed, provides more than 10,000 default packages.

A. Plant Diseases on Paddy Crop

Paddy is the major crop for many farmers ,there are mainly two seasons in which paddy is grown, one is rabbi and other karif.Each season takes six months for production. Due to the environment changes diseases are attacking the crop result in yield loss. Below are the major paddy diseases:

- Bacterial Leaf Blight is mostly occurred diseases caused by “Xanthomonas Oryzae bacteria”.It occurred mostly in the area that have weeds, stubbles of infected plants and high use of nitrogen.More observed symptoms are Milky dew drops can be observed early in the morning.



Fig 1

- Blast is caused by “Magnaporthe grisea fungus”.These symptoms can be found on leaves,neck,seeds.Classic diamond shaped lesions seen on leaves,at the neck brown to chocolate colour spots .



Fig 2

- False smut is caused by” Ustilaginoidea Virenil fungi” at flowering stage.Individual rice grain transformed into a mass of velvety spores.



Fig 3

- Rice tungro is an virus disease caused by “RTBV”and “RTSV”.The major symptom is leaves turn yellow and more leafhoppers .



Fig 4

II. ABOUT DATASET COLLECTION

The dataset is collected from the UCI repository and the information about diseases are gained under the guidance of DR.Srinivas Prasad who was a scientist in IIRR (Indian Institute of Rice Research centre in Rajendranagar).

The link where image dataset collected is:
<https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases>

Based on the capacity of my system I have trained 15 images and tested three images for prediction.



Fig 5

After dividing dataset into training and testing, using Convolutional Neural Network we are going to predict the testing sample.

III. HOW CNN WORKS

In CNN it reduce the parameters this can help to save lot of time.

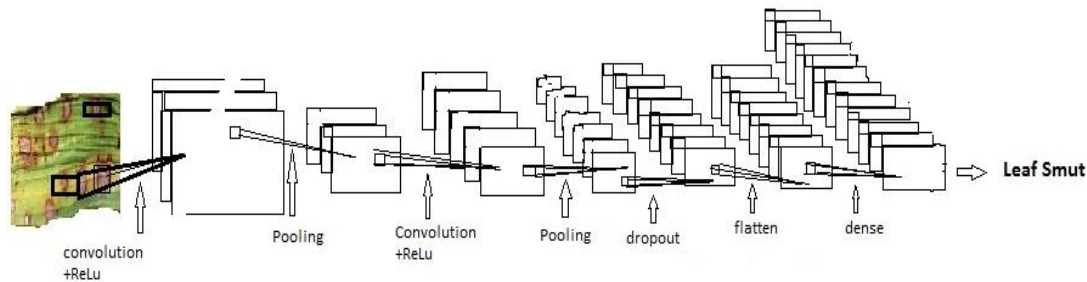


Fig 6

ConvNet is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex.

Have following layers:

- Convolution
- ReLU Layer
- Pooling
- Fully Connected

A. Steps Involved in Convolution Layer:

Here we will move the feature /filter to every possible position on the image.

- Step1: Line up the feature and the image.
- Step2: Multiple each image pixel by the corresponding feature pixel.
- Step3: Add them up and divide by total number of pixels in the feature.
- Step4: Now to keep track of where that feature was, we create a map and put the value of the filter at that place.
- Step5: Similarly we move the feature to every other positions of the image and will see how the feature matches that area.

B. Steps Involved in ReLU Layer:

Rectified Linear Unit is an activation function only activates a node if the input is above a certain quantity, while the input is below zero, the output is zero. In this layer we remove every negative values from the filtered images and replaces it with zeros.

$$F(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

C. Steps involved in Pooling Layer:

In this layer we shrink the image stack into a smaller size.

- Pick a window size.
- Pick a stride.
- Walk your window across your filtered images.
- From each window take the maximum value.

We can add one more layers of Convolution, ReLU and Pooling then go to the **Fully Connected Layer**. Here we take our filtered images and put them into a single list.

Compare the input image value to the default image values, the value which is near to the default image value then that image belongs to that set.

IV. IMPLEMENTATION DIAGRAM

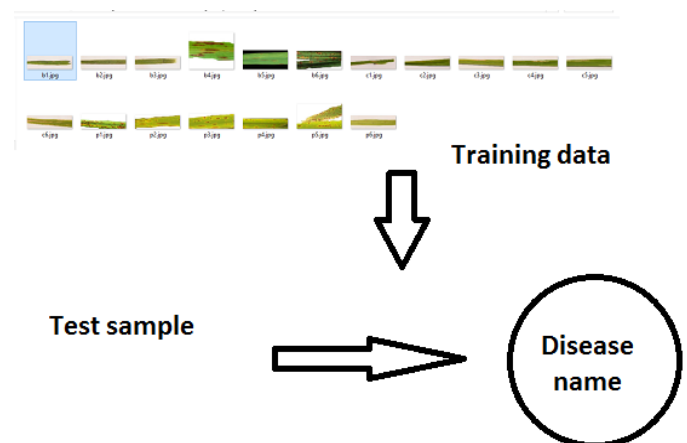


Fig 7

V. STEPS TO BE PERFORMED

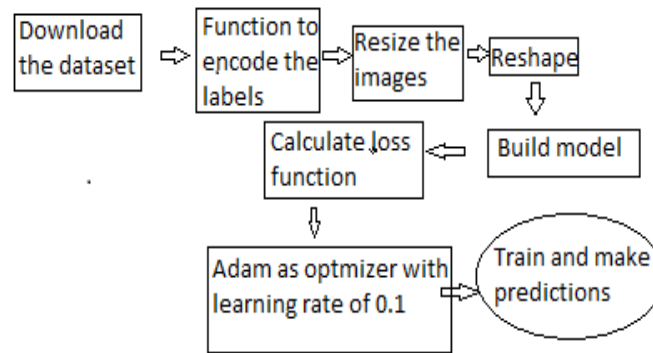


Fig 8

➤ *Load Packages and Read Images:*

We require tensor flow as the main package and sub packages are EBIImage, keras.

```

library(tensorflow)
library(keras)
library(EBImage)
  
```

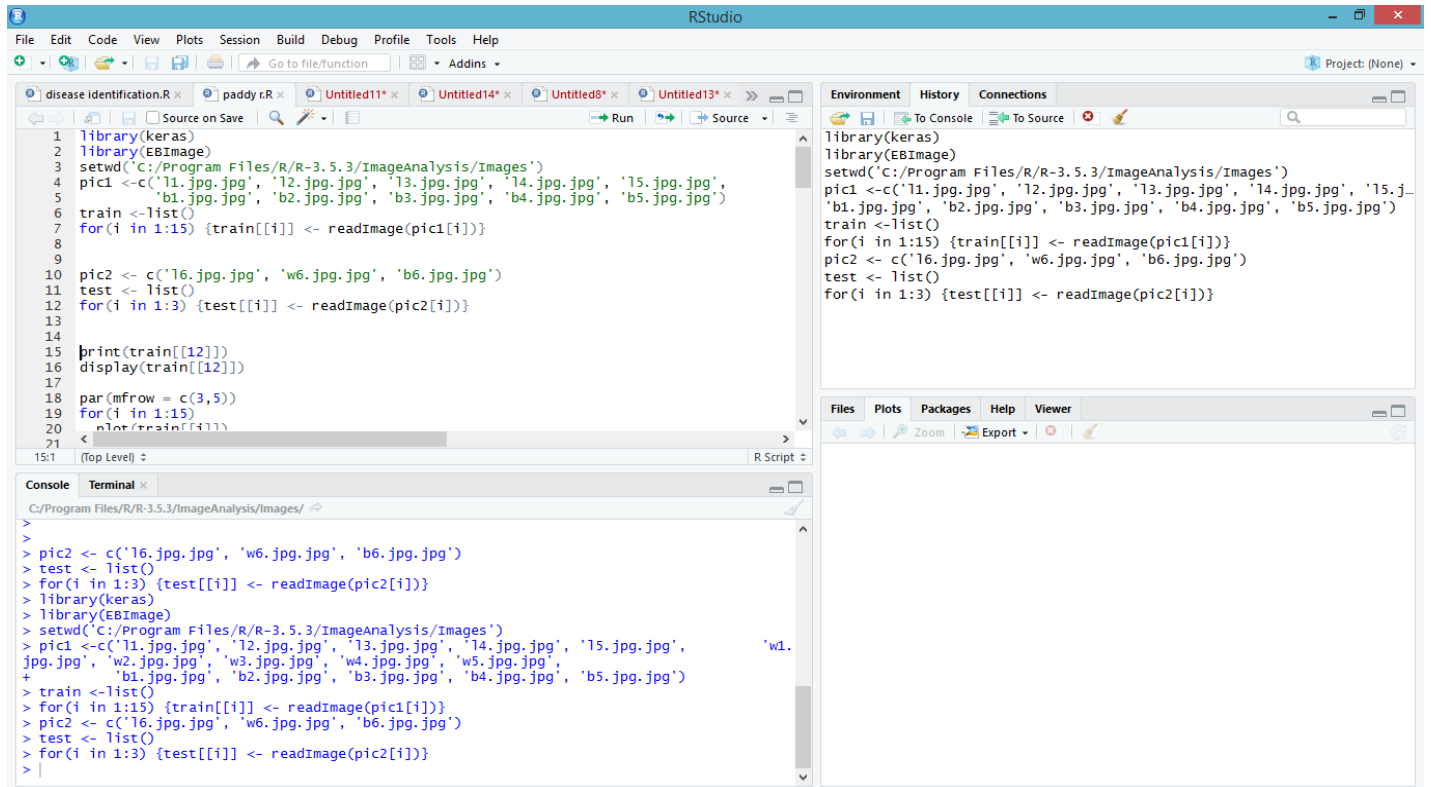


Fig 9

➤ *Explore, Resize & Combine:*

We resize all the images both train and test into 100 * 100.

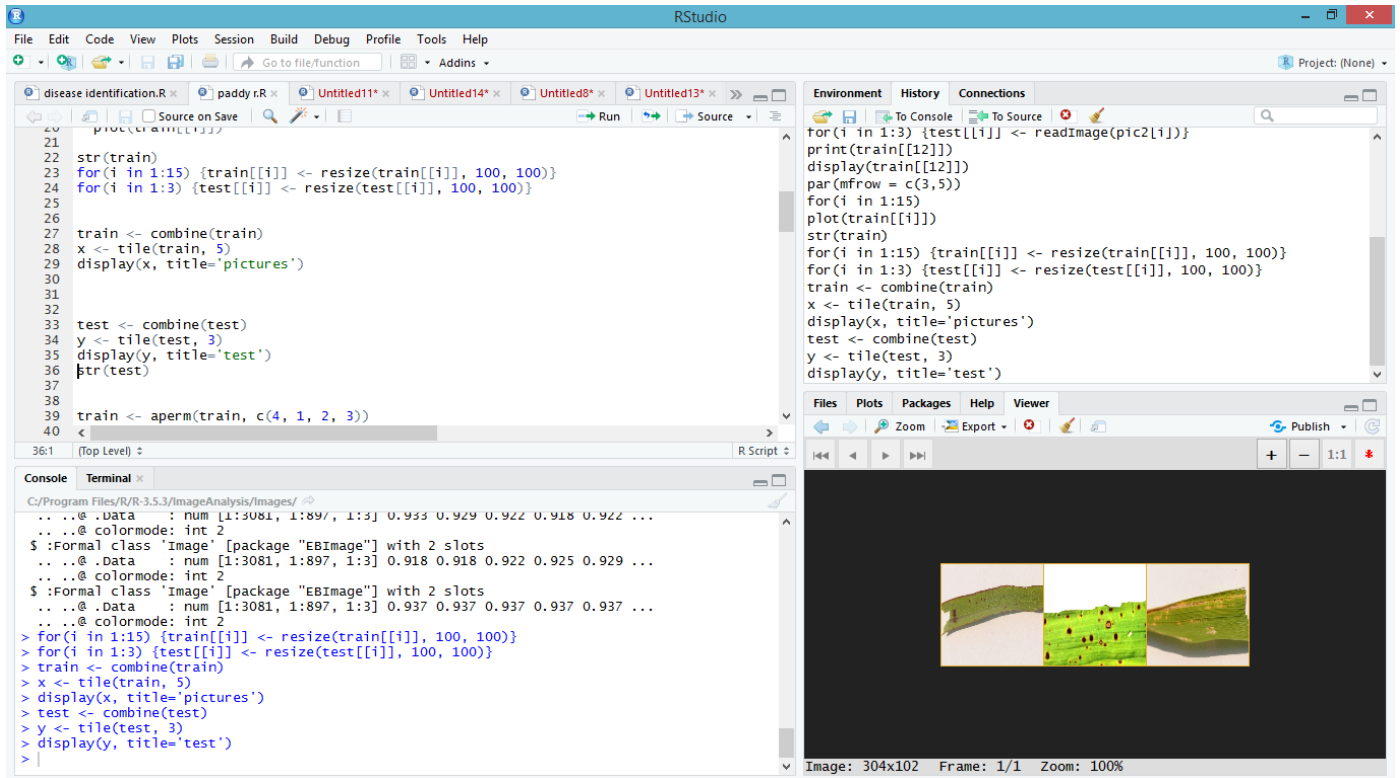


Fig 10

- *Reorder Dimension and Response:*
By using function “**aperm**” we change the order of dimensions.
- *One Hot Encoding:*
Using `to_categorical` we perform one hot encoding.

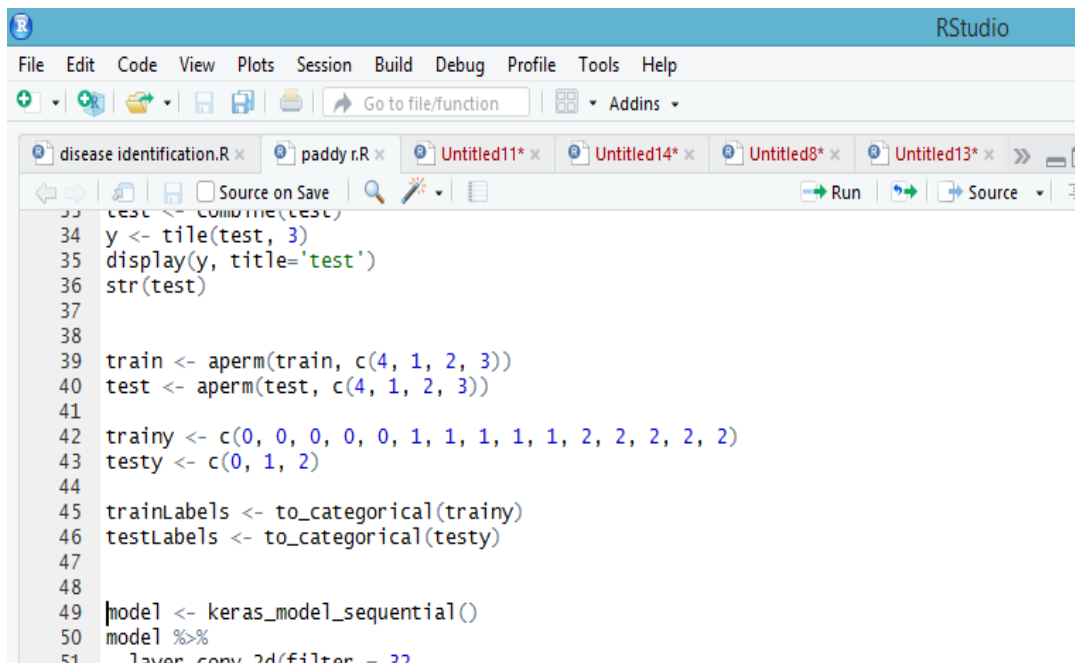


Fig 11

- *Model:*
Here I have used 32 filters, kernel size is $3 * 3$, activation is ReLu and input shape is $100 * 100 * 3$.

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
disease identification.R x paddy r.R x Untitled11* x Untitled14* x Untitled8* x Untitled13* x darra x Unti
45 trainLabels <- to_categorical(trainy)
46 testLabels <- to_categorical(testy)
47
48
49 model <- keras_model_sequential()
50 model %>%
51   layer_conv_2d(filter = 32,
52                 kernel_size = c(3,3),
53                 activation = 'relu',
54                 input_shape = c(100,100,3)) %>%
55   layer_conv_2d(filters = 32,
56                 kernel_size = c(3,3),
57                 activation = 'relu') %>%
58   layer_max_pooling_2d(pool_size = c(2,2)) %>%
59   layer_dropout(rate = 0.25) %>%
60   layer_conv_2d(filters = 64,
61                 kernel_size = c(3,3),
62                 activation = 'relu') %>%
63   layer_conv_2d(filters = 64,
64                 kernel_size = c(3,3),
65                 activation = 'relu') %>%
66   layer_max_pooling_2d(pool_size = c(2,2)) %>%
67   layer_dropout(rate=0.25) %>%
68   layer_flatten() %>%
69   layer_dense(units = 256, activation = 'relu') %>%
70   layer_dropout(rate = 0.25) %>%
71   layer_dense(units = 3, activation = 'softmax') %>%
72
73   compile(loss='categorical_crossentropy',
74           optimizer= optimizer_sgd(lr=0.01,
75                                   decay=1e-6,
76                                   momentum=0.9,
77                                   nesterov = T),
78           metrics=c('accuracy'))
79
80
81
82 summary(model)
83
49:1 (Top Level) R Script
Console

```

Fig 12

Specified learning rate as 0.01, decay is 1e-6 and validation split as 0.2 (it means 20 % data for validation). For better outcome re-run the model again.

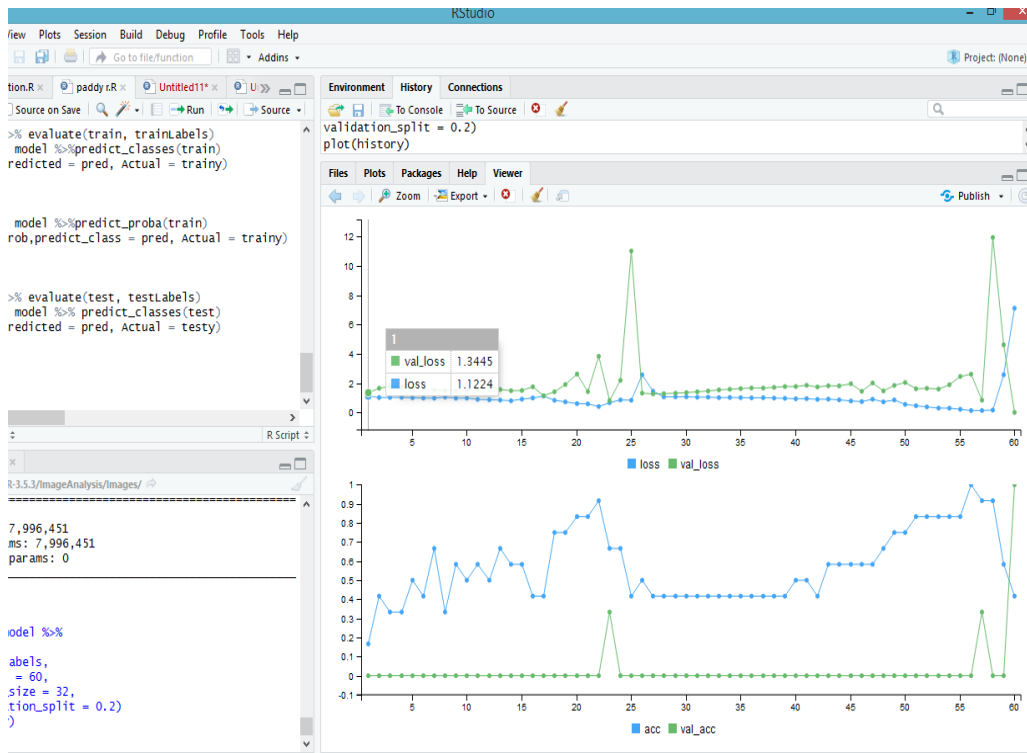


Fig 13

The graph below shows the loss and accuracy for training and validation data. Accuracy is top one and loss is at bottom. Accuracy should be larger the better and loss be smaller the better.

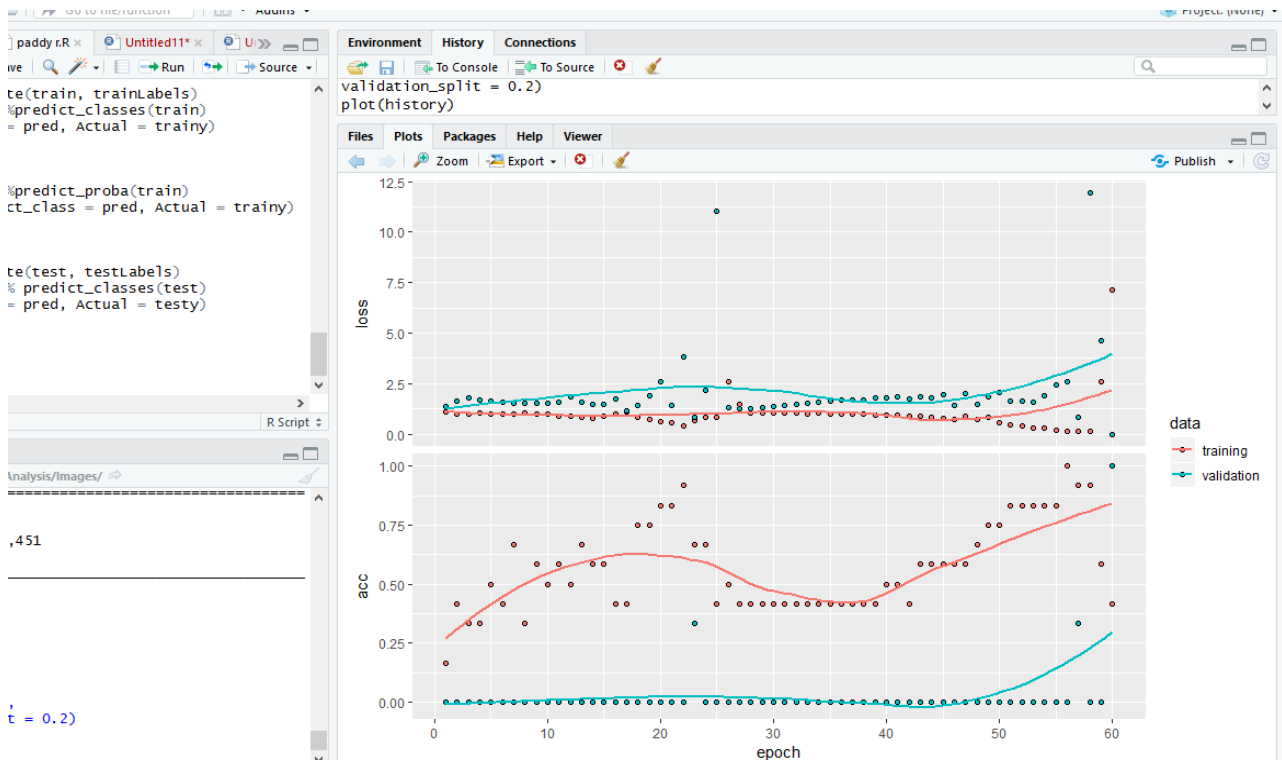


Fig 14

➤ *Evaluation & Prediction – Train and Test Data:*

The accuracy got is 0.866, if we involve more number of training samples we can improve performance. Here two samples are correctly predicted.

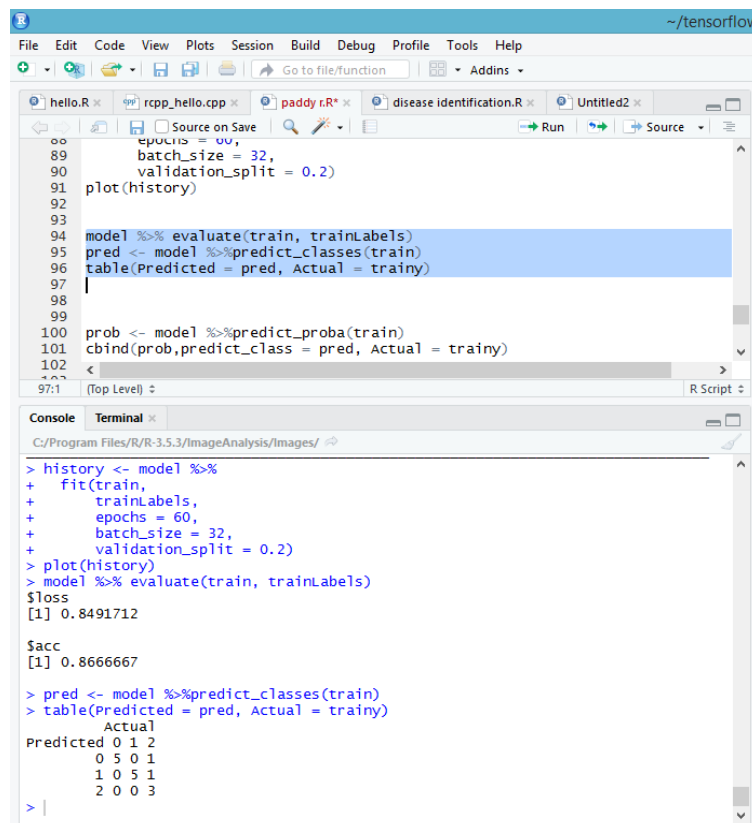


Fig 15

VI. CONCLUSION

In this paper, particularly focused on identifying the diseases which occur in paddy using r language .By improving the training images we achieve better results. In future, we can also predict disease name and provide suitable pesticides.

REFERENCES

- [1]. Dan, C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J. 2011. Flexible, high performance convolutional neural networks for image classification. Proceedings of the 22nd International Joint Conference on Artificial Intelligence, vol. 2, pp. 1237–1242.
- [2]. [http://refhub.elsevier.com/S0925-2312\(17\)31138-4/sbref0002](http://refhub.elsevier.com/S0925-2312(17)31138-4/sbref0002)
- [3]. [http://refhub.elsevier.com/S0925-2312\(17\)31138-4/sbref0001](http://refhub.elsevier.com/S0925-2312(17)31138-4/sbref0001)
- [4]. Hemanta Kalita, Shikhar Kr. Sarma, “Expert System for Diagnosis of Diseases of Rice Plants: Phototype Design and Implementation,” 2016
- [5]. International Conference on Automatic Control and Dynamic Optimization Technique (ICACDOT), International Institute of Information Technology, Pune, 2016. [2] Chiunnapa Buntaoet, Wongkot Sriurai. “Diagnosis System for Chili Disease by using Data Classification Technique,” The Eight National Conference on Computing and Information Technology, NCCIT, 2012.
- [6]. H. Park, J. S. Eun and S. H. Kim, “Image-based disease diagnosing and predicting of the crops through the deep learning mechanism”, In Information and Communication Technology Convergence (ICTC), IEEE 2017 International Conference on, pp. 129-131, 2017.
- [7]. K. Elangovan and S. Nalini, “Plant disease classification using image segmentation and SVM techniques”, International Journal of Computational Intelligence Research, vol. 13(7), pp. 1821-1828, 2017.
- [8]. S. H. Lee, C. S. Chan, S. J. Mayo and P. Remagnino, “How deep learning extracts and learns leaf features for plant classification”, Pattern Recognition, vol. 71, pp. 1-13, 2017.
- [9]. K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis”, Computers and Electronics in Agriculture, vol. 145, pp. 311318, 2018.