“E-KETHA” : ENRICHING RICE FARMER’S QUALITY OF LIFE THROUGH A MOBILE APPLICCATION.

Salika Madhushanka W.J
Department of Software Engineering
Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
salikamadhushanka33@gmail.com

P.Y.D Jayasinghe
Department of Software Engineering
Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
yasoja44@gmail.com

H.H.W.M.Binuka Sihan Paranagama
Department of Software Engineering
Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
shihanBinuka@gmail.com

K.M.Umesh Ranthilina
Department of Software Engineering
Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
umeshranthilina111@gmail.com

***Abstract*— In our country of Sri Lanka, rice is the most common type of food that is consumed daily. Due to that rice farmers face a huge amount of stress to supply according to the massive demand. This is happening while they are farming in poor conditions such as, amongst diseases and pests that harm rice crops with the inclusion of weeds that plague the field. They also have difficulties finding the correct fertilizers and the amount that are needed for the crops to grow properly. Another issue discovered, was that there some rice plants are underdeveloped, and farmers lack the understanding about proper treatment. These topics were chosen according to multitude of statistics including losses due to all insects, losses due to all diseases, losses due to all weeds, potential production harvested, and total potential production lost before harvest being found respectively at 34.4%, 9.9%, 10.8%, 44.9% and 55.1%. The aim is to develop a mobile application that will help farmers solve these problems. The application will use image processing to analyze crops to find solutions stored at a cloud database. Then after machine learning and deep learning will be used to recommend appropriate solutions.**

Keywords*— machine learning, image processing, deep learning

I. INTRODUCTION (*HEADING 1*)

Rice farming, which have been cultivated by humans since 3000-2500 BC have been one of the main staple sources of sustenance. This is due to multiple beneficial qualities that it provides such as being rich in carbohydrates, fiber, selenium and even vitamin B. Since rice mainly grows in hot and humid climates, Asia is the current most producer of rice in the world. In particular Chinese, Indian and Sri Lankan people tend to eat rice on a daily basis.

When we look at a country like Sri Lanka, due to rice being high on demand local farmers struggle massively to meet the said demand. This can be to the point of even having to import rice from overseas. As shown in graph below.

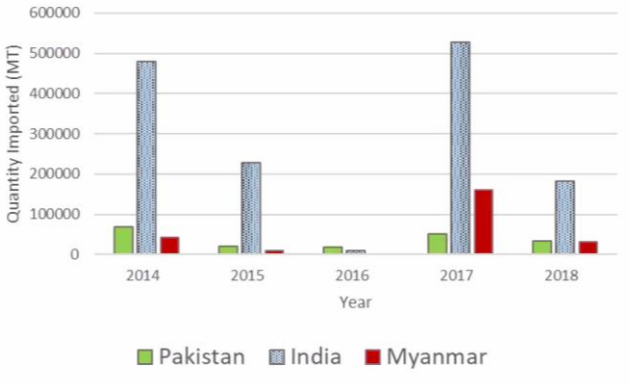


Figure 1:Rice importers by Sri Lanka (2014-2018), by main importing countries

There are multitude of reasons on why rice farming can be slowed down when it comes to local and foreign rice farming. However in particular they effect Sri Lanka farmers more due to lack of proper technology and knowledge. The four most important of the reasons are,

- Pests and dieseas – The many pests and dieseas that could harm rice crops.
- Weeds – The weeds that absorbs nutrients from the soil.
- Fertilizer missuse – The improper use fertilizers that harms rice crops.
- Growth problems – Issues when it comes to the growth of rice crops.

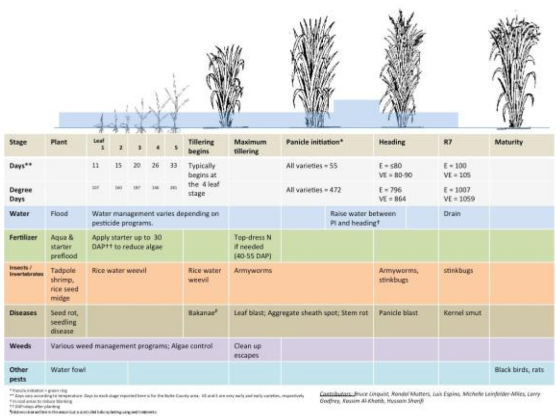


Figure 2:Rice growth

II. LITERATURE REVIEW

Intro ---------

A. Crop Farmers Mobile Application

Help the farmer with summary information about crops, fruits, and vegetables. Climatic and Soil Requirements, Avocado, Banana, Beans, Carrot, Jackfruit, Cucumber, Garlic, Irish Potato, Lettuce, Sorghum, Watermelon, Onion, Bell Peppers and Peppers, Pineapple and Eggplant sour Info explains how. The app also describes the most common causes of pests and diseases, symptoms, how they spread, and prevention and control measures. Where possible, app will advise on suitable farming methods to control crop pests. This app can be used as a guide for new farmers, or anyone involved in farming around the world. Learn new farming techniques/methods to avoid attacking your crops. It also provides information on best practices for to follow to improve farmers' performance in growing these crops[1].

B. Pest Identification using Image Processing using Neural Network

This study is done by Johnny L. Miranda, B. Gerardo, Bartolome T. Tanguilig, Sajad Sabzi with the goal of classifying pests in crops. Pest infestation in rice production is a challenging task for crop technicians and farmers. Pest infestation can cause serious losses and also affect the income of farmers. Decisions for pest predictions can be made by estimating the density of farmers. Existing detection techniques for these species involve the use of various traps to detect their presence. In this study, an identification system was developed for automatic detection of field insect pests. Continuous monitoring by a wireless camera for video recording is done by catching the insect with a sticky trap. Various imaging techniques are used to identify and extract the captured insect. Neural network was used to identify the extracted insect pests. The new automated detection system developed in this study provides reliable detection[2].

C. Weed Classification for Site-Specific Weed Management Using Automated Stereo Computer-Vision

This study is done by Mojtaba Dadashzadeh , Yousef Abbaspour-Gilandeh ,Tarahom Mesri-Gundoshmian , Sajad Sabzi with the goal of classifying weed in a specific site using stereo vision system to distinguish rice plants and weeds. This system is further augmented using an artificial neural network and two other metaheuristic algorithms, them being y particle swarm optimization (PSO) and the bee algorithm (BA). With stereo videos being recorded of the site beforehand and decomposed into singular frames, rice plants were extracted out using the color, shape and even texture. Then the previously mentioned metaheuristic algorithms were used to optimize the neural network and classify the weed detected as well. According to K-nearest neighbors (KNN) classifier this reached f 88.74% and 87.96% for right and left channels without accounting arithmetic or the geometric means as the basis and with it o 92.02% and 90.7% respectively[3].

D. A nutrient recommendation system for soil fertilization based on evolutionary computation

This study [11] is about predicting the fertilizers for different crops and give nutrients recommendations by analyzing the crop fertility and yield production. However, this application is limited to selected fertilizers (Nitrogen (N), Phosphorus (P), and Potassium (K)). This recommendation done by using improved genetic algorithm (IGA) which will uses time-series sensor data and recommends various crop settings. By analyzing the way that fertilizer works, the application will be able to give instructs farmers to get the maximum yield output[4].

E. Rice Crop Height Measurement Using a Digital Image Processing

This is a plant height identification method currently in operation in Thailand. It detects the height of the plant and shows the height of the plant to the user. But it does not use a mobile app.
Here is an automatic image processing method to identify the user based on the photos taken by a digital camera mounted on a field server, including a marker bar used to describe the height of the rice plant. Height can be assessed by analyzing the uploaded image obtained by the user. Digital image processing for analysis uses four steps to automatically measure rice crop height. Therefore, it is possible to get the height of the rice tree.

	A	B	C	D	E	E-Ketha
Detect diseases	Yes	No	No	No	No	Yes
Detect pests	No	Yes	No	No	No	Yes
Detect weeds	No	No	Yes	No	No	Yes
Provide guidance manage fertilizer	No	No	No	Yes	No	Yes
Detect growth	No	No	No	No	Yes	Yes

Table 1:Comparing existing application and our application features

III. RESEARCH OBJECTIVES

The main objective of this research project is to help farmers with their paddy fields and make life easier for them. The farmers will be receiving proper guidance and techniques so that producing a steady abundant yield of crops to match the great demand of consumers. Farmers will have the opportunity of exchanging information among one-another so as to regulate knowledge.

The Sub objectives are as follows;

A. Detection of pests and diseases using image processing and finding solutions by applying machine learning:

B. Detection of weeds using image processing and finding solutions by applying machine learning:

C. Provide fertilization solutions according to the size of paddy field and the fertilizer type using image processing, then after providing the instructions by applying machine learning:

D. Rice crop growth identification using image processing and giving solutions to debilitated crops by applying machine learning:

IV. METHODOLOGY

Intro

A. Detection of pests and providing solutions:

Customized CNN was used as the main model for the disease identification. CNN was chosen due to the it being one of the most basic deep learning models which can take input images and have them differentiated. Three input layers were added in order to customized according to the dataset and this was able to give the best outcome. The data was shuffled, resized and rescaled in order to perform preprocessing. Batch size 32 and 20 epochs were given as the hyperparameters for the best results to emerge. 0.8 and 0.2 split was made for the training and testing set.

For pest identification AlexNet was used as the model. The reason why AlexNet chosen was due to its relatively short training time compared to other deep learning models. This is because it allows multi-GPU usage thus making use of multiple GPUs if there are present. Normalization and label one hot encoding was performed as preprocessing. 20 epochs, 0.1 learning rate and 32 batch size are the hyper parameters used in this model. 0.8 and 0.2 split was made for the training and testing set

B. Detection of weeds and providing solutions:

For the purpose of Weed identification, the ResNet (Residual Network) model was used. This model was chosen in order to answer the issue of vanishing or exploding gradient which is a nuisance in deep neural networks that have a large number of layers. What is meant by vanishing or exploding gradient is the gradient becoming zero or becoming a large number with the increase of layers thus providing a high error rate on both training and test datasets. How ResNet archives this is by using the concept of residual blocks which utilize the technique of skip connections. This skip connections connect activation layers to oncoming layers by skipping the layers in the middle of them. How it decides to skip is by seeing if the next layer is damaging the performance. In particular ResNet50 model is used here due to the reasoning of giving the best results as well as 500x500 pixel size images being used as the dataset. The 50 after the model name is the amount of layers in the model as such ResNet50 contains 50 layers.

The dataset used for the training of the model has 17,509 belonging to 8 different weed species. Normalization was used to preprocess the data in order for better training. Mean and standard deviation was calculated in order to normalize. For training and testing purposes the dataset was split in a 0.8 and 0.2 ratio respectively. Another 0.1 was taken from the training set for validation to be done. As for the hyperparameters for this model, 10 epochs, 32 batch size were chosen as this gives the best accuracies, while the number of classes were 9 due to the 8 weed species and another for negative samples. The learning rate was then chosen to be 0.001 as the learning rate finder function gave that amount as the number with the lowest error rate.

C. Provide fertilization solutions according to the size of paddy field and the fertilizer type:

Customized CNN was used as the main model for the fertilizer identification. CNN was chosen due to the it being one of the most basic deep learning models which can take input images and have them differentiated. Layers of the model have been modified accordingly in order to get the maximum accuracy.

- 4 – Convolutional layers
- 4 – pooling layers

Then a flatten layer and dense layer with SoftMax activation function was added to convert the output of the model to a format which can make the prediction.

As for the preprocessing the dataset random vertical_flip, horizontal_flip, rescale and shuffle features were added.

Finally, in order to get the maximum test and the training accuracy hyperparameters were tuned accordingly,

- Batch size – 32
- Epoch - 90

Maximum accuracy was achieved, according to the previously mentioned configurations.

In order to calculate the area of the paddy field, Mobile device’s GPS has been used. Application was developed so that a user can easily calculate any paddy field part that they want to fertilize. User has to ping the 4 corner locations of the area that required for the fertilization. Then the application will get the latitude and longitude of each location, calculate the area of the paddy field.

D. Rice crop growth identification and providing solutions:

In Order to identify whether the plant is a rice plant or not. For this AlexNet model is used for its special quality which is speed. This is in part due to the multi gpu utilization ability. This enables several graphical processing units to work in tandem with each other. Another strength this algorithm posses nonlinearity which is provided by Rectified Linear Unit (ReLU). This also add to its already impressive speed.

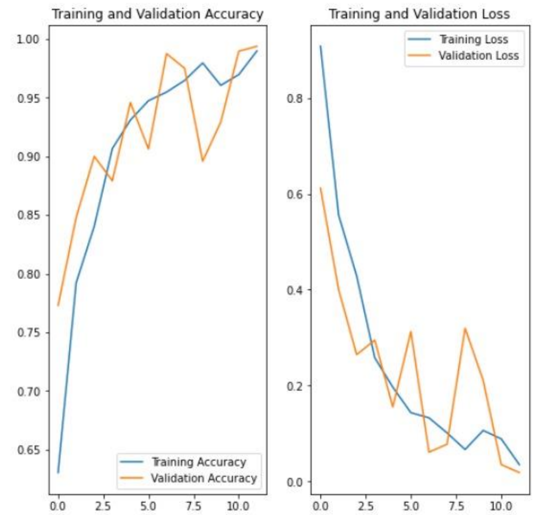
As for the preprocessing, first the integer numbers converted to the floats then performing normalization and finally one-hot encoding the labels. 10 epochs, 0.1 learning rate and 128 batch size are the hyper parameters which worked best and had the lowest error rate. 0.8 and 0.2 split was made for the training and testing set.

In order to measure the height of the rice plant, A python code has been implemented that has the capability to measure the height when the distance to the plant has been inputted.

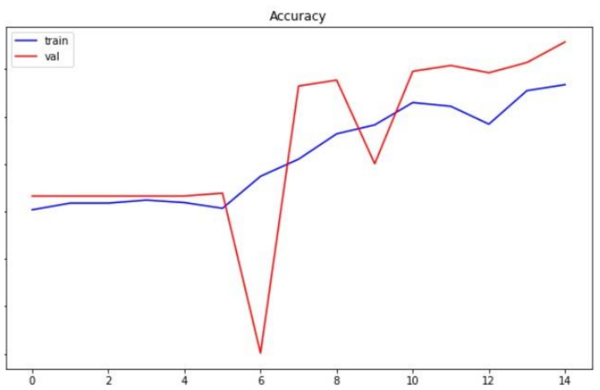
V. RESULT AND DISCUSSION

A. *Detection of pests and providing solutions:*

```
Epoch 4/12
120/120 [==============================] - 211s 2s/step - loss: 0.2581 - accuracy: 0.9066 - val_loss: 0.2948 - val_accuracy: 0.
8792
Epoch 5/12
120/120 [==============================] - 210s 2s/step - loss: 0.1966 - accuracy: 0.9309 - val_loss: 0.1552 - val_accuracy: 0.
9458
Epoch 6/12
120/120 [==============================] - 208s 2s/step - loss: 0.1434 - accuracy: 0.9474 - val_loss: 0.3126 - val_accuracy: 0.
9062
Epoch 7/12
120/120 [==============================] - 209s 2s/step - loss: 0.1327 - accuracy: 0.9547 - val_loss: 0.0611 - val_accuracy: 0.
9875
Epoch 8/12
120/120 [==============================] - 217s 2s/step - loss: 0.1013 - accuracy: 0.9647 - val_loss: 0.0775 - val_accuracy: 0.
9750
Epoch 9/12
120/120 [==============================] - 208s 2s/step - loss: 0.0666 - accuracy: 0.9796 - val_loss: 0.3196 - val_accuracy: 0.
8958
Epoch 10/12
120/120 [==============================] - 210s 2s/step - loss: 0.1064 - accuracy: 0.9605 - val_loss: 0.2108 - val_accuracy: 0.
9292
Epoch 11/12
120/120 [==============================] - 208s 2s/step - loss: 0.0888 - accuracy: 0.9696 - val_loss: 0.0351 - val_accuracy: 0.
9896
Epoch 12/12
120/120 [==============================] - 208s 2s/step - loss: 0.0351 - accuracy: 0.9898 - val_loss: 0.0186 - val_accuracy: 0.
9937

In [29]: scores = model.evaluate(test_ds)
         15/15 [==============================] - 17s 477ms/step - loss: 0.0461 - accuracy: 0.9771

In [ ]:
```



98.98% training accuracy and 97.71% test accuracy was able to be reached using this model as shown in the examples above with the predictions for the test data shown below.




```
Epoch 1/15
40/40 [==============================] - 57s 1s/step - loss: 0.6344 - accuracy: 0.7518 - val_loss: 0.5548 - val_accuracy: 0.7663
Epoch 2/15
40/40 [==============================] - 56s 1s/step - loss: 0.5451 - accuracy: 0.7588 - val_loss: 0.5211 - val_accuracy: 0.7663
Epoch 3/15
40/40 [==============================] - 53s 1s/step - loss: 0.5463 - accuracy: 0.7588 - val_loss: 0.5449 - val_accuracy: 0.7663
Epoch 4/15
40/40 [==============================] - 58s 1s/step - loss: 0.5384 - accuracy: 0.7620 - val_loss: 0.5106 - val_accuracy: 0.7663
Epoch 5/15
40/40 [==============================] - 59s 1s/step - loss: 0.5266 - accuracy: 0.7594 - val_loss: 0.4901 - val_accuracy: 0.7663
Epoch 6/15
40/40 [==============================] - 53s 1s/step - loss: 0.5408 - accuracy: 0.7533 - val_loss: 0.5396 - val_accuracy: 0.7693
Epoch 7/15
40/40 [==============================] - 57s 1s/step - loss: 0.4958 - accuracy: 0.7871 - val_loss: 0.7090 - val_accuracy: 0.6006
Epoch 8/15
40/40 [==============================] - 56s 1s/step - loss: 0.4725 - accuracy: 0.8052 - val_loss: 0.3608 - val_accuracy: 0.8824
Epoch 9/15
40/40 [==============================] - 56s 1s/step - loss: 0.4419 - accuracy: 0.8319 - val_loss: 0.2999 - val_accuracy: 0.8885
Epoch 10/15
40/40 [==============================] - 56s 1s/step - loss: 0.4280 - accuracy: 0.8413 - val_loss: 0.4405 - val_accuracy: 0.8003
Epoch 11/15
40/40 [==============================] - 56s 1s/step - loss: 0.3942 - accuracy: 0.8649 - val_loss: 0.3157 - val_accuracy: 0.8978
Epoch 12/15
40/40 [==============================] - 56s 1s/step - loss: 0.3645 - accuracy: 0.8610 - val_loss: 0.2362 - val_accuracy: 0.9040
Epoch 13/15
40/40 [==============================] - 52s 1s/step - loss: 0.4016 - accuracy: 0.8421 - val_loss: 0.2662 - val_accuracy: 0.8963
Epoch 14/15
40/40 [==============================] - 55s 1s/step - loss: 0.3234 - accuracy: 0.8775 - val_loss: 0.2476 - val_accuracy: 0.9071
Epoch 15/15
40/40 [==============================] - 52s 1s/step - loss: 0.2875 - accuracy: 0.8837 - val_loss: 0.1775 - val_accuracy: 0.9288
21/21 [==============================] - 5s 221ms/step - loss: 0.1775 - accuracy: 0.9288
92.879
```

88.37% training accuracy and 92.88% test accuracy was able to be reached using this model as shown in the examples above with the predictions for the test data shown below

B. *Pest Identi Detection of weeds and providing solutions:*

```
Epoch: 01 | Epoch Time: 8m 33s
	Train Loss: 0.828 | Train Acc @1:  73.24% | Train Acc @5: 100.00%
	Valid Loss: 0.405 | Valid Acc @1:  86.45% | Valid Acc @5: 100.00%
Epoch: 02 | Epoch Time: 8m 34s
	Train Loss: 0.354 | Train Acc @1:  87.96% | Train Acc @5: 100.00%
	Valid Loss: 0.343 | Valid Acc @1:  88.50% | Valid Acc @5: 100.00%
Epoch: 03 | Epoch Time: 8m 32s
	Train Loss: 0.308 | Train Acc @1:  89.94% | Train Acc @5: 100.00%
	Valid Loss: 0.417 | Valid Acc @1:  87.98% | Valid Acc @5: 100.00%
Epoch: 04 | Epoch Time: 8m 32s
	Train Loss: 0.231 | Train Acc @1:  92.47% | Train Acc @5: 100.00%
	Valid Loss: 0.225 | Valid Acc @1:  92.50% | Valid Acc @5: 100.00%
Epoch: 05 | Epoch Time: 8m 32s
	Train Loss: 0.182 | Train Acc @1:  93.92% | Train Acc @5: 100.00%
	Valid Loss: 0.350 | Valid Acc @1:  91.31% | Valid Acc @5: 100.00%
Epoch: 06 | Epoch Time: 8m 35s
	Train Loss: 0.122 | Train Acc @1:  96.04% | Train Acc @5: 100.00%
	Valid Loss: 0.166 | Valid Acc @1:  95.03% | Valid Acc @5: 100.00%
Epoch: 07 | Epoch Time: 8m 37s
	Train Loss: 0.064 | Train Acc @1:  97.86% | Train Acc @5: 100.00%
	Valid Loss: 0.119 | Valid Acc @1:  96.30% | Valid Acc @5: 100.00%
Epoch: 08 | Epoch Time: 8m 32s
	Train Loss: 0.042 | Train Acc @1:  98.53% | Train Acc @5: 100.00%
	Valid Loss: 0.095 | Valid Acc @1:  96.73% | Valid Acc @5: 100.00%
Epoch: 09 | Epoch Time: 8m 31s
	Train Loss: 0.021 | Train Acc @1:  99.35% | Train Acc @5: 100.00%
	Valid Loss: 0.082 | Valid Acc @1:  97.16% | Valid Acc @5: 100.00%
Epoch: 10 | Epoch Time: 8m 32s
	Train Loss: 0.019 | Train Acc @1:  99.49% | Train Acc @5: 100.00%
	Valid Loss: 0.088 | Valid Acc @1:  97.09% | Valid Acc @5: 100.00%
```

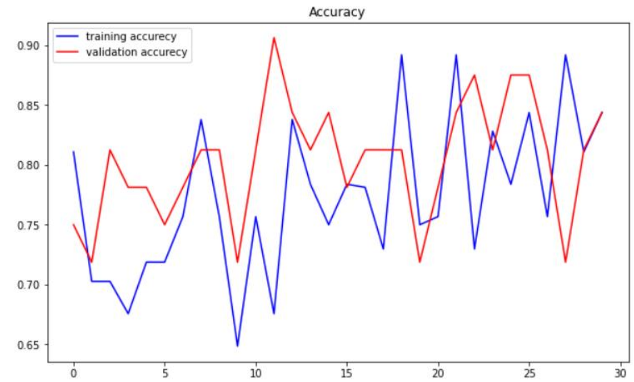
```
[ ] model.load_state_dict(torch.load('tut5-model.pt'))

    test_loss, test_acc_1, test_acc_5 = evaluate(model, test_iterator, criterion, device)

    print(f'Test Loss: {test_loss:.3f} | Test Acc @1: {test_acc_1*100:6.2f}% | ' \
          f'Test Acc @5: {test_acc_5*100:6.2f}%')

    Test Loss: 0.698 | Test Acc @1:  85.62% | Test Acc @5: 100.00%
```

99.49% training accuracy and 85.62% test accuracy was able to be reached using this particular model as shown in the examples above with the predictions for the test data shown below.

C. *Provide fertilization solutions according to the size of paddy field and the fertilizer type:*

```
2/2 [==============================] - 4s 1s/step - loss: 0.2148 - accuracy: 0.8919 - val_loss: 0.2958 - val_accuracy: 0.8750
Epoch 76/90
2/2 [==============================] - 4s 4s/step - loss: 0.1888 - accuracy: 0.9189 - val_loss: 0.1914 - val_accuracy: 0.9375
Epoch 77/90
2/2 [==============================] - 4s 1s/step - loss: 0.1154 - accuracy: 1.0000 - val_loss: 0.2521 - val_accuracy: 0.8750
Epoch 78/90
2/2 [==============================] - 4s 1s/step - loss: 0.2780 - accuracy: 0.8649 - val_loss: 0.2205 - val_accuracy: 0.8438
Epoch 79/90
2/2 [==============================] - 6s 3s/step - loss: 0.3761 - accuracy: 0.7812 - val_loss: 0.2141 - val_accuracy: 0.9062
Epoch 80/90
2/2 [==============================] - 5s 2s/step - loss: 0.3031 - accuracy: 0.8378 - val_loss: 0.4106 - val_accuracy: 0.7500
Epoch 81/90
2/2 [==============================] - 5s 5s/step - loss: 0.3072 - accuracy: 0.8108 - val_loss: 0.3842 - val_accuracy: 0.7500
Epoch 82/90
2/2 [==============================] - 4s 4s/step - loss: 0.2454 - accuracy: 0.8649 - val_loss: 0.2122 - val_accuracy: 0.8750
Epoch 83/90
2/2 [==============================] - 5s 2s/step - loss: 0.1635 - accuracy: 0.9189 - val_loss: 0.2611 - val_accuracy: 0.8438
Epoch 84/90
2/2 [==============================] - 5s 2s/step - loss: 0.3550 - accuracy: 0.7568 - val_loss: 0.3077 - val_accuracy: 0.8438
Epoch 85/90
2/2 [==============================] - 4s 4s/step - loss: 0.4292 - accuracy: 0.8108 - val_loss: 0.2468 - val_accuracy: 0.8438
Epoch 86/90
2/2 [==============================] - 4s 4s/step - loss: 0.2031 - accuracy: 0.8649 - val_loss: 0.2297 - val_accuracy: 0.8438
Epoch 87/90
2/2 [==============================] - 6s 3s/step - loss: 0.1995 - accuracy: 0.8906 - val_loss: 0.3161 - val_accuracy: 0.8125
Epoch 88/90
2/2 [==============================] - 7s 3s/step - loss: 0.2211 - accuracy: 0.8750 - val_loss: 0.2790 - val_accuracy: 0.8438
Epoch 89/90
2/2 [==============================] - 7s 4s/step - loss: 0.2056 - accuracy: 0.9062 - val_loss: 0.2989 - val_accuracy: 0.8438
Epoch 90/90
2/2 [==============================] - 5s 4s/step - loss: 0.2444 - accuracy: 0.8919 - val_loss: 0.3199 - val_accuracy: 0.8750
<keras.callbacks.History object at 0x0000026403E49AF0>
```

```
3/3 [==============================] - 2s 600ms/step - loss: 0.2241 - accuracy: 0.9420
Restored model, accuracy: 94.20%
(69, 4)
```

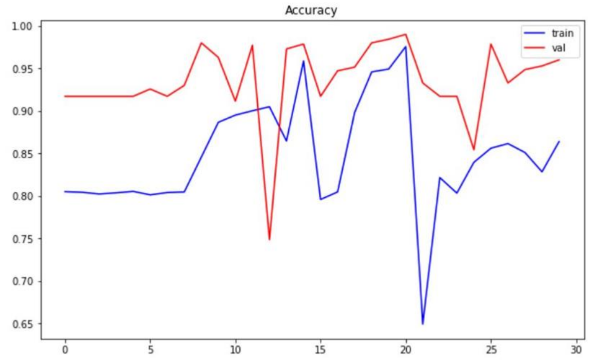


95.23% training accuracy and 94.20% test accuracy was able to be reached using this model as shown in the examples above with the predictions for the test data shown below.

D. *Rice crop growth identification and providing solutions:*

```
Epoch 16/30
24/24 [==============================] - 199s 8s/step - loss: 0.6790 - accuracy: 0.7959 - val_loss: 0.1375 - val_accuracy: 0.9171
Epoch 17/30
24/24 [==============================] - 201s 8s/step - loss: 0.2325 - accuracy: 0.8047 - val_loss: 0.1089 - val_accuracy: 0.9471
Epoch 18/30
24/24 [==============================] - 196s 8s/step - loss: 0.1969 - accuracy: 0.8981 - val_loss: 0.1259 - val_accuracy: 0.9514
Epoch 19/30
24/24 [==============================] - 197s 8s/step - loss: 0.1679 - accuracy: 0.9458 - val_loss: 0.0823 - val_accuracy: 0.9800
Epoch 20/30
24/24 [==============================] - 190s 8s/step - loss: 0.1831 - accuracy: 0.9492 - val_loss: 0.0665 - val_accuracy: 0.9843
Epoch 21/30
24/24 [==============================] - 189s 8s/step - loss: 0.1121 - accuracy: 0.9756 - val_loss: 0.0480 - val_accuracy: 0.9900
Epoch 22/30
24/24 [==============================] - 188s 8s/step - loss: 0.5736 - accuracy: 0.6493 - val_loss: 0.2730 - val_accuracy: 0.9329
Epoch 23/30
24/24 [==============================] - 187s 8s/step - loss: 0.5350 - accuracy: 0.8216 - val_loss: 0.4989 - val_accuracy: 0.9171
Epoch 24/30
24/24 [==============================] - 187s 8s/step - loss: 0.4657 - accuracy: 0.8033 - val_loss: 0.3801 - val_accuracy: 0.9171
Epoch 25/30
24/24 [==============================] - 192s 8s/step - loss: 0.4039 - accuracy: 0.8395 - val_loss: 0.3247 - val_accuracy: 0.8543
Epoch 26/30
24/24 [==============================] - 187s 8s/step - loss: 0.3699 - accuracy: 0.8561 - val_loss: 0.1020 - val_accuracy: 0.9786
Epoch 27/30
24/24 [==============================] - 198s 8s/step - loss: 0.3874 - accuracy: 0.8615 - val_loss: 0.2760 - val_accuracy: 0.9329
Epoch 28/30
24/24 [==============================] - 384s 16s/step - loss: 0.4067 - accuracy: 0.8510 - val_loss: 0.2438 - val_accuracy: 0.9486
Epoch 29/30
24/24 [==============================] - 198s 8s/step - loss: 0.4546 - accuracy: 0.8284 - val_loss: 0.2161 - val_accuracy: 0.9529
Epoch 30/30
24/24 [==============================] - 213s 9s/step - loss: 0.3642 - accuracy: 0.8639 - val_loss: 0.3518 - val_accuracy: 0.9600
22/22 [==============================] - 12s 548ms/step - loss: 0.3518 - accuracy: 0.9600
96.000
```

```
22/22 - 9s - loss: 0.3518 - accuracy: 0.9600 - 9s/epoch - 418ms/step
Restored model, accuracy: 96.00%
(700, 2)
```



86.39% training accuracy and 96.00% test accuracy was able to be reached using this model as shown in the examples above with the predictions for the test data shown b

References

[1] Johnny L. Miranda, B. Gerardo [Online]. Available: [PDF] Pest Identification using Image Processing Techniques in Detecting Image Pattern through Neural Network | Semantic Scholar. [Accessed: 20-Feb-2020].

[2] "Crop Farmers App - Apps on Google Play," Google. [Online]. Available: https://play.google.com/store/apps/details?id=com.bivatec.cropfarmersguide&hl=en_US. [Accessed: 20-Feb-2020].

[3] Dadashzadeh, Mojtaba, et al. "Weed Classification for Site-Specific Weed Management Using an Automated Stereo Computer-Vision Machine-Learning System in Rice Fields." *Plants*, vol. 9, no. 5, 1 May 2020, p. 559, www.mdpi.com/2223-7747/9/5/559, 10.3390/plants9050559.

[4] "Computers and Electronics in Agriculture | Journal | ScienceDirect.com by Elsevier," *www.sciencedirect.com*. https://www.sciencedirect.com/journal/computers-and-electronics-in-agriculture (accessed Jul. 31, 2021).

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.