International Journal of Computer Applications (0975 – 8887)
Volume 184– No.28, September 2022

Tomato Leaf Disease Identification using Deep Reinforcement Learning

B.I. Madhubhashinie
Department of Computing and Information Systems,
Wayamba University of Sri Lanka,
Kuliyapitiya, Sri Lanka

W.H.C. Wickramaarachchi
Department of Computing and Information Systems,
Wayamba University of Sri Lanka,
Kuliyapitiya, Sri Lanka

ABSTRACT
In the agriculture sector, tomato is a widely cultivated, most popular edible plant that contains rich nourishment and distinct flavor. Various factors, including bacteria, viruses, and fungus, are frequently responsible for tomato diseases. These diseases can be considered a prominent threat to cultivation. Therefore, the identification of leaf diseases plays a crucial role in taking disease control as well as increasing the quality and quantity of crop yield. With the idea of preserving harvest quality, the research aims to identify and categorize the diseases of tomato plant leaves. The initial focus of the research was to perform a comparative analysis between some existing Convolutional Neural Network (CNN) models to identify the best model for image recognition. The second phase of this research introduces a recurrent network to construct the model descriptions of Neural Networks (NN) and train this NN with Reinforcement Learning (RL) to optimize the anticipated accuracy of the constructed architectures on a dataset.

General Terms
Convolutional Neural Network, Pattern recognition, Leaf disease identification

Keywords
Reinforcement Learning, Convolutional Neural Network, Deep Learning, Tomato Leaves, Disease

1. INTRODUCTION
Agriculture has been Sri Lanka's foremost source of income from ancient times. It is the livelihood of 70% rural population. The agriculture sector contributes to about 12% of the GDP and 32% of the employment in Sri Lanka. Tomato is a vital and seasonal cash crop for Sri Lankan farmers. It is an essential source of vitamin. In 2017 Sri Lanka's tomato production quantity is 80,839 tons and it is increasing yearly. However, 25% - 30% of tomato harvest is lost because of insect pests, diseases, and weeds. The main reason for these losses of cultivation due to the unawareness of those diseases and therefore not identifying in advance before spreading on the farmland. The common diseases of tomato plants include tomato target spot disease, early blight, late blight, leaf mold disease, tomato mosaic virus, Septoria leaf spot, two-spotted spider mite, yellow leaf curl disease, etc. and sample images of how these diseases influence a tomato plant (see Figure 1).

Fig 1: Example images of the damages caused by the diseases in tomato plants

Among those diseases, Early Blight, Late Blight, Leaf Mold, Bacterial Spot, Septoria Leaf Spot, are Bacterial and fungus deceases and Mosaic disease and Yellow Leaf Curl disease are virus diseases. The two-spotted spider mite is the most common mite species that attacks tomato.

The context of this problem is typically associated with the results of the climate variations in the atmosphere and the way it alerts an ecosystem. Weather changes affect regional climate factors, like humidity, cloud cover, temperature, wind, and precipitation that accordingly support as a vector in which viruses, bacteria and microorganism can harm the cultivation. As the result of this climate variations, it directly impacts on the population by health, economic and livelihood impact. In earlier days, the identification of plant diseases was usually accomplished by regular monitoring of plants by farming experts. In the case of small farms or home garden cultivations, it was possible to spot the diseases easily and take immediate preventive on control measures. But in the case of enormous farms, it is not an effective way. On the other hand, at present lots of farmers are lack knowledge and experience of those diseases. Therefore, looking for an automatic, accurate, fast, and less expensive technology for disease identification is very important.

Image processing in agriculture had been applied in many areas such as sorting, grading, and detection of defects like dark spots, cracks, and bruises in fruits and seeds, etc. These classification tasks were done by using semantic features like corners, edges, shapes, etc. Remarkable progress has been made in image recognition, primarily due to the availability of large-scale annotated datasets, and the new advances in technology have enabled the development of Deep Convolutional Neural Networks (DCNN) and their number of applications such as image identification and image classification. Recently, Convolutional Neural Network (CNN) is the widely applied image classification method which makes a tremendous breakthrough. It enables learning data-driven, highly representative, hierarchical image features from sufficient training data. Compared to traditional image classification methods, CNN can automatically learn feature representation related to the classification target from the raw image. The last few years have seen much success of deep neural networks in image recognition and different architectures were invented such as LeNet [12], AlexNet [10], VGGNet [15], GoogleNet [17], ResNet [8], Xception [9] and SqueezeNet [5]. Therefore, this research initially focuses on a comparison between these existing architectures for the proposed purpose.

Although it has become easier, designing architectures still requires a lot of expert knowledge and takes ample time [20]. Typically, CNN architecture consists of several convolution layers, pooling layers, and fully connected layers. While constructing a CNN model the designer has to perform various design choices such as the number of layers of each type, the ordering of layers, and the hyper-parameters for each type of layer, e.g., the receptive field size, and stride for a convolution layer. Although there have been some automated or computer-aided neural network architectures, new CNN designs or network design elements are still primarily created by researchers using new theoretical insights gained from experimentations.

Therefore, this research is seeking to automate the procedure of CNN design choices based on RL. It introduces a recurrent network to construct the model descriptions of NNs and train this NN with RL to optimize the anticipated accuracy of the constructed architectures on a dataset. Its aim is to observe CNN models that execute effectively on a given machine learning problem with no human interference.

2. RESEARCH PROBLEM

Tomato is a vital and seasonal cash crop for Sri Lankan farmers and 25% - 30% of tomato harvest is losses because of insect pests, diseases, and weeds. According to statistics by different sources, the rate of harvest losses increasing year by year. The main reason for these losses of the cultivation due to the unawareness of those diseases and therefore not identifying in advance before spreading on the farmland. So, the most effective method of reducing losses, is early detection and proper immediate treatments. Proper, early identification of those diseases and necessary treatments can prevent spreading these diseases all over the farmland. The identification of plant diseases was usually accomplished by regular monitoring of plants. In the case of small farms or home garden cultivations, it was possible to spot the diseases easily and take immediate preventive on control measures. But in the case of enormous farms, it is not an effective way. On the other hand, at present lots of farmers are lack of knowledge and experience of those diseases. Therefore, looking for an automatic, accurate, fast, and less expensive technology for disease identification is very important.

CNN is one of the Artificial Intelligence that is popular in feature learning and image classification. Many studies show how CNN surpasses diseases detection in many computer vision tasks. Therefore, this study proposes to construct a tomato leaf diseases detection system with CNN and RL techniques. The outcome of this study will help to assist the farmers in identifying different tomato diseases which is very challenging for them.

3. LITRETURE REVIEW

For the past few years, plant leaf disease detection has been a prominent study area. It is an essential research area in which both image processing and machine learning mechanisms are widely used for obtaining accurate automated classification. Recently, these researches are expanded using Deep Learning (DL) techniques to further enhance of the classification performance. Few of the most significant CNN based leaf disease detection procedures are discussed in this section.

CNN is a feed-forward Artificial Neural Network (ANN) that mostly used to analyze images in Machine Learning (ML). S. Adhikari et al. [1] presented an image processing-based method for detecting tomato plant diseases. Three diseases that often affect tomato plants were discovered. YOLO object detection algorithm was used to train the model and determine diseases in the tomato plant. To manipulate the raw input image, the Python programming language and the OpenCV library were utilized. B. A. Ashqar et al. [2] developed a method for classifying plant diseases using state of art DL technique and demonstrated the feasibility using DCNN. The dataset of 9000 images of healthy and diseased tomato leaves were used to construct the model. That model could be used in smartphone applications to recognize 5 different types of tomato leaf diseases, with the accuracy of 99.84%. M. Brahimi et al. [4] proposed a DL based classification and symptoms visualization system for tomato diseases. The dataset consists of 14828 of tomato leaves images infected with 9 diseases. To understand patterns and to locate diseased areas in leaves, the system employed visualization approaches. For diagnosing diseases in tomato plant leaves, the suggested method had a 99.18 percent accuracy rate. H. Drums et al. [6] developed a robot system with a deep learning algorithm to detect the various diseases that occur and spread in plants in tomato fields or greenhouses. A robot detected the diseases of the tomato plants while roaming around manually on the field by using the close-up photographs taken from plants by its sensors. AlexNet and SqueezeNet got a test accuracy of 95.65% and 94.30% respectively. E. Suryawati et al. [16] assess the consequence of various CNN model depths on the detection accuracies of the plant disease detection. Several CNN designs with varying depths were investigated. The proposed system used colored images of tomato leaves from PlantVillage dataset to train the model by using AlexNet, GoogleNet and VGGNet and got a test accuracy of 91.52%, 89.68%, and 95.25% respectively. A. Kumara et al. [11] used the same dataset to train and differentiate the results of the CNN models (LeNet, VGGNet, ResNet50 and Xception) using colored and segmented images. All the tested CNN models achieved above 91% test accuracy K. Zhang et al. [19] used transfer learning to diagnose leaf diseases using DCNN. The backbone of the model was AlexNet, GoogleNet, and ResNet. The relative performance of these networks was compared by using Stochastic Gradient Descent (SGD) and Adam optimization method, revealing that the ResNet with SGD optimization method obtains the highest result with the best accuracy, 96.51%. Then, the performance evaluation of batch size and number of iterations affecting the transfer learning of the ResNet was conducted and suggested that, for a particular task, neither large batch size nor large number of iterations may improve the accuracy of the target model. The height accuracy of 97.28% for identifying tomato leaf diseases was achieved by the optimal model ResNet with SGD the number of batch size of 16 and the training layers from the 37 layers to the fully connected layer.

RL is targeted at learning with sequential decision making and reward offered procedure. It has been broadly used in several applications such as games, robotics control, medical diagnosis and treatments, and finance. B. Baker et al. [3] developed a meta-modeling approach based on RL to automate the process of CNN design selection. That system was capable to construct tailored CNN models for various image classification tasks. R. Furuta et al. [7] proposed a Fully Convolutional Network using RL for image processing. For different image processing applications, they expand deep RL to pixelRL. The suggested technique was applied to image processing tasks that require pixel-wise manipulations, where deep RL has never been used. In addition, it was able to identify the type of operation that was employed for each pixel at each iteration and that was helped to understand why and how that particular operation was selected. That technology improved the explainability and interpretability of the Deep Neural Network (DNN). Using remote sensing pictures, Y. Li et al. [13] presented an airplane identification framework based on RL and CNN. The developed frameworks could successfully identify the unfixed number of aircraft in the remote sensing images without the previous action. According to the research study of “Deep Reinforcement Learning for Greenhouse Climate Control”, the researchers Lu Wang, Xiaofeng He, Dijun Luo et al [18] proposed Deep Reinforcement learning based climate control method, which could model future reward explicitly.

4. PROPOSED SYSTEM

The study was carried out with three parts. In the first part existing CNN architectures tested with the dataset. So, in here existing CNN models such as LeNet [12], AlexNet [10], VGGNet [15], GoogleNet [17], ResNet [8], Xception [9] and SqueezeNet [5] were tested with the dataset.

In the second part of this study, one existing model architecture (LeNet) was tested with some changed parameters such as BatchSize, StepsPerEpoch and ValidationSteps and check whether how the results were change.

As the last part of this study, the study was automating the process of CNN architecture selection based on Reinforcement Learning. In this section this study was introduced a recurrent network to generate the model descriptions of neural networks and train this NN with reinforcement learning to maximize the expected accuracy of the generated architectures on a dataset.

5. TECHNOLOGY STACKS

Keras, TensorFlow, Python, and Google Colab used as the technologies in implementing the solution.

5.1 Keras and Tensorflow

Keras is a high-level API used by TensorFlow for building and training deep learning models which is written in python. Fast prototyping, state of the art research and production can be done using Keras. It is user friendly and can extend. Keras provides clear feedbacks to users. It is easier for the users to import different APIs from the Keras rather than implementing it from scratch. TensorFlow is a machine learning platform which is open source. It consists of comprehensive, libraries, flexible tools etc.

5.2 Python

Python is an object-oriented, simple programming language. Its high-level built-in data structures, combined with dynamic typing and binding, make python very attractive for Rapid Application Development [14]. Python contains different modules and packages which provides code reusability and easy coding. Python is a highly productive language and contain features of both Java and C languages. We use python in data science because it has various 43 packages like NumPy, SciPy, pandas, and matplotlib. For develop this system, pandas and NumPy libraries are used. These packages are very useful in machine learning. Google Colab-Jupytor notebook provides the environment to develop data science applications based on python.

5.3 Google Colab

Colab is a free notebook environment that runs entirely in the cloud. Google colab works as the way which Google docs work. Google Colab is a hosted Jupyter notebook service which is a product from Google Research. We can execute python codes via the browser.

This is well suited when we do machine learning. It provides the service for free to a certain no. of GPUs and if need more we can pay and buy (Colab Pro). It provides many features like sharing notebooks, connecting with google drive, omit outputs of the cells when saving etc.

6. DATASET

The models in this study were trained using the PlantVillage dataset. It contains 14 crop species leaves, and in this work images of tomato leaves are used. Tomato subset has 22,929 images of tomato leaves and classified into ten classes. There are one class of healthy tomato leaves (2,406) and nine class of tomato leaf diseases. Nine types of tomato leaf diseases are Bacterial Spot (2,127 images), Tomato Leaf Mold (2,352 images), Tomato Target Spot (2,284 images), Tomato Mosaic Virus (2,238 images), Tomato Early Blight (2,400 images), Tomato Septoria Leaf Spot (2,181 images), Tomato Late Blight (2,314 images), Yellow Leaf Curl Virus (2,451 images) and Tomato Spider Mites (2,176 images).

7. RESULTS

The study was carried out in several steps. In the first step this study was focused on the existing CNN models such as LeNet, AlexNet, VGGNet, GoogleNet, ResNet, Xception and SqueezeNet. These existing CNN architectures tested with the dataset and the training results were plotted according to the Number of Epochs upon Accuracy. These models were checked with the same hyperparameters, and the training results obtained after completing this part was as follows (see Figure 2).

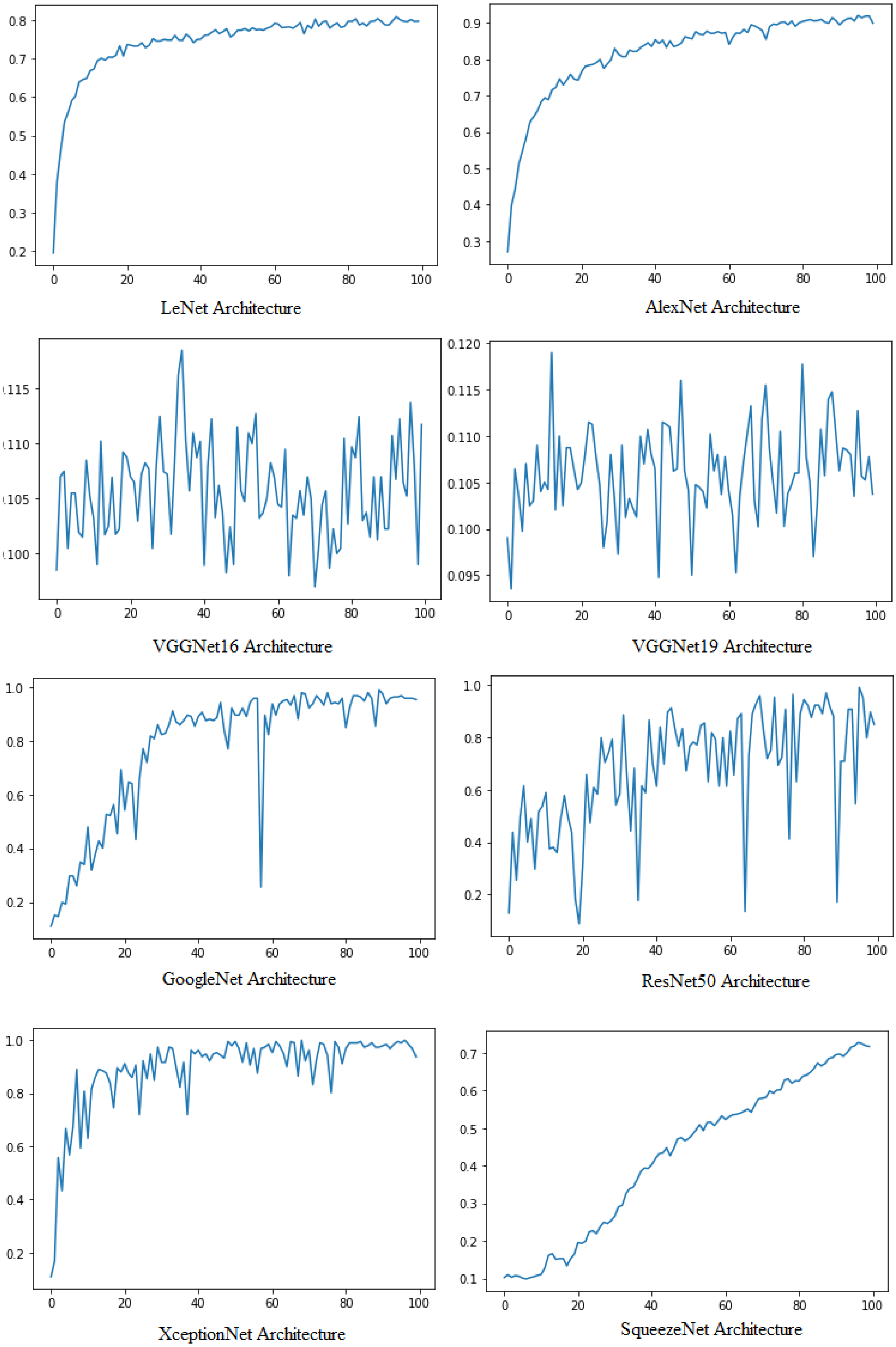

Fig 2: Training results of existing models

Table 1. Training Results

Model Name	Training Results (%)
LeNet	91
AlexNet	86
VGGNet16	14
VGGNET19	15
GoogleNet	95
ResNet50	84
Xception	93
SqueezeNet	78

The accuracies of training models are 91%, 86%, 14%, 15%, 95%, 84%, 93% and 78% respectively (see Table 1). The most probable reason for getting a significant difference in these accuracies is, using the same parameters for all the models. As a result, machine cannot learn the specific features of the images to do the predictions accurately.

Therefore, in the second part of this study results were checked with changed parameters such as Batch Size, Steps Per Epoch and Validation Steps in the one existing model architecture (LeNet). As the results, when batch sizes increase as 8, 16 and 32 the training results are 88%, 91% and 95% respectively. So, in here when the batch sizes increase it will increase the training results accordingly (see Figure 3). According to the training results the steps per epoch (see Figure 4) and validation steps (see Figure 5) are not much effect on the final training value.

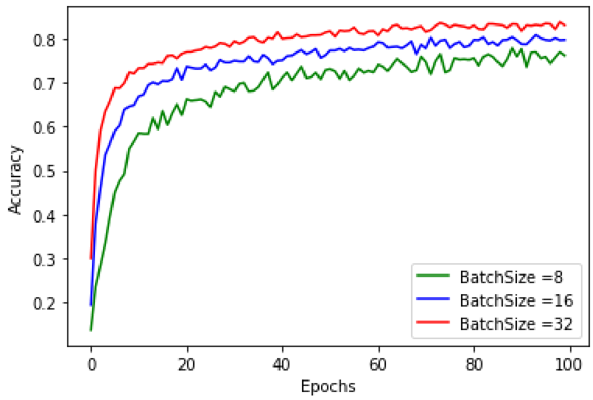

Fig 3: Result variation with batch size

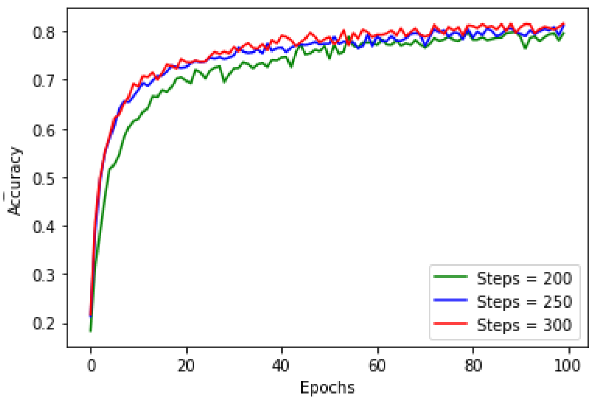

Fig 4: Result variation with steps per epoch

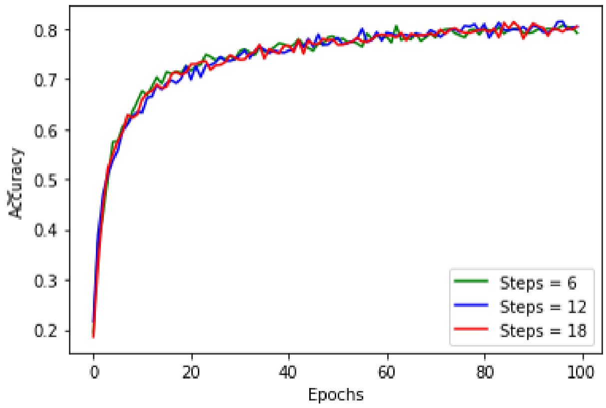

Fig 5: Result variation with validation steps

In the final part of this research was seeking to automate the process of CNN model selection using RL. It introduces a recurrent network to construct the model descriptions of NNs and train this NN with RL to optimize the anticipated accuracy of the constructed architectures on a dataset. Its objective is to observe CNN models that execute effectively without human assistance on a specific machine learning task. The trial 1 results (see Figure 6), trail 2 results (see Figure 7) and trail 3 results (see Figure 8) obtained after completing this part was as follows.

```
Manager: EWA Accuracy =  0.6143947839736938
Rewards :  0.1 Accuracy :  0.6143948
Total reward :  0.1
State input to Controller for training :  [1 0]
Training RNN (States ip) :  [1, 32, 1, 16, 1, 32, 1, 16]
Training RNN (Reward ip) :  [0.1]
Trial 1: Controller loss : 9.220693
```

Fig 6: Trail 1 results

```
Manager: EWA Accuracy =  0.6611559629440307
Rewards :  0.1 Accuracy :  0.8482007
Total reward :  0.2
State input to Controller for training :  [1 0]
Training RNN (States ip) :  [1, 16, 3, 64, 3, 16, 3, 64]
Training RNN (Reward ip) :  [0.1]
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/
Instructions for updating:
Use standard file APIs to delete files with this prefix.
Trial 2: Controller loss : 14.708861
```

Fig 7: Trail 2 results

```
Manager: EWA Accuracy =  0.6969509434700012
Rewards :  0.1 Accuracy :  0.84013087
Total reward :  0.30000000000000004
State input to Controller for training :  [0 2]
Training RNN (States ip) :  [3, 16, 3, 64, 3, 32, 3, 16]
Training RNN (Reward ip) :  [0.1]
Trial 3: Controller loss : 12.741738

Total Reward :  0.30000000000000004
```

Fig 8: Trail 3 results

8. CONCLUSION

In the agriculture sector, tomato is a widely cultivated, most popular edible plant which contains rich nourishment and ideal unique taste. Tomatoes usually provide a good harvest, and it plays a major role in agricultural sector and also in the international trade. The tomato diseases are often caused by various factors like bacteria, viruses, fungus, etc. Most of the farmers are unknown to such diseases. Therefore, these diseases can cause an influential threat to cultivation. Hence the identification of theses leaf diseases plays an essential role in taking disease control and increase the quality and quantity of crop yield. Addressing this practical scenario, research aims to detect and classify the diseases of tomato plants' leaf.

A computerized system can dramatically increase the accuracy of diseases detection process and simultaneously improve the throughput. With the development in machine learning implementing a convolutional neural network with image processing techniques can be consider as a better solution for the above-mentioned problem. Throughout this study, seven existing CNN models were trained.

Although CNN has achieved significant success in image classification, neural networks are still hard to design. It still requires both human expertise and labour for the developing process. So, in the last part of this research it was trained and tested result according to three various parameters such as batch size, steps per epochs and validation steps.

After identifying strong and weak points, the final phase of the research it was introduced a recurrent network to construct the model descriptions of NNs and train this NN with RL to optimize the anticipated accuracy of the constructed architectures on a dataset That automatically creates high-performing CNN architecture for a particular training task.

9. FUTURE WORKS

The proposed solution's detection is not applicable for all the types of tomato leaf diseases. It only focuses on the nine common tomato leaf diseases, such as Bacterial Spot, Tomato Leaf Mold, Tomato Target Spot, Tomato Mosaic Virus, Tomato Early Blight, Tomato Septoria Leaf Spot, Tomato Late Blight, Yellow Leaf Curl Virus, and Tomato Spider Mites.

However, this can be improved by training the model with different disease images. This study can be extended to classify any type of plant disease.

Currently the system only contains a CNN model for testing purposes. This can be expanded by implementing a mobile application, so that users can input an image of the disease and then the application will show the result in a table. Since mobile phone is usage is vastly increased in the past decade, implementing a mobile application will be more beneficial for the end users.

10. REFERENCES

[1] Adhikari, S., Shrestha, B., Baiju, B. and Kumar, S., 2018. Tomato plant diseases detection system using image processing. In KEC Conference on Engineering and Technology (pp. 1-6).

[2] Ashqar, B.A. and Abu-Naser, S.S., 2018. Image-based tomato leaves diseases detection using deep learning.

[3] Baker, B., Gupta, O., Naik, N. and Raskar, R., 2016. Designing neural network architectures using reinforcement learning. arXiv preprint arXiv:1611.02167.

[4] Brahimi, M., Boukhalfa, K. and Moussaoui, A., 2017. Deep learning for tomato diseases: classification and symptoms visualization. Applied Artificial Intelligence, 31(4), pp.299-315.

[5] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[6] Durmuş, H., Güneş, E.O. and Kırcı, M., 2017, August. Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International Conference on Agro-Geoinformatics (pp. 1-5). IEEE.

[7] Furuta, R., Inoue, N. and Yamasaki, T., 2019. PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing. IEEE Transactions on Multimedia.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[9] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[11] Kumar, A. and Vani, M., 2019, July. Image Based Tomato Leaf Disease Detection. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[12] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[13] Li, Y., Fu, K., Sun, H. and Sun, X., 2018. An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images. Remote Sensing, 10(2), p.243.

[14] Python, https://www.python.org/doc/essays/blurb/

[15] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[16] Suryawati, E., Sustika, R., Yuwana, R.S., Subekti, A. and Pardede, H.F., 2018, October. Deep Structured Convolutional Neural Network for Tomato Diseases Detection. In 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 385-390). IEEE.

[17] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[18] Wang, L., He, X., & Luo, D. (2020, August). Deep reinforcement learning for greenhouse climate control. In 2020 IEEE International Conference on Knowledge Graph (ICKG) (pp. 474-480). IEEE.

[19] Zhang, K., Wu, Q., Liu, A. and Meng, X., 2018. Can deep learning identify tomato leaf disease?. Advances in Multimedia, 2018

[20] Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.