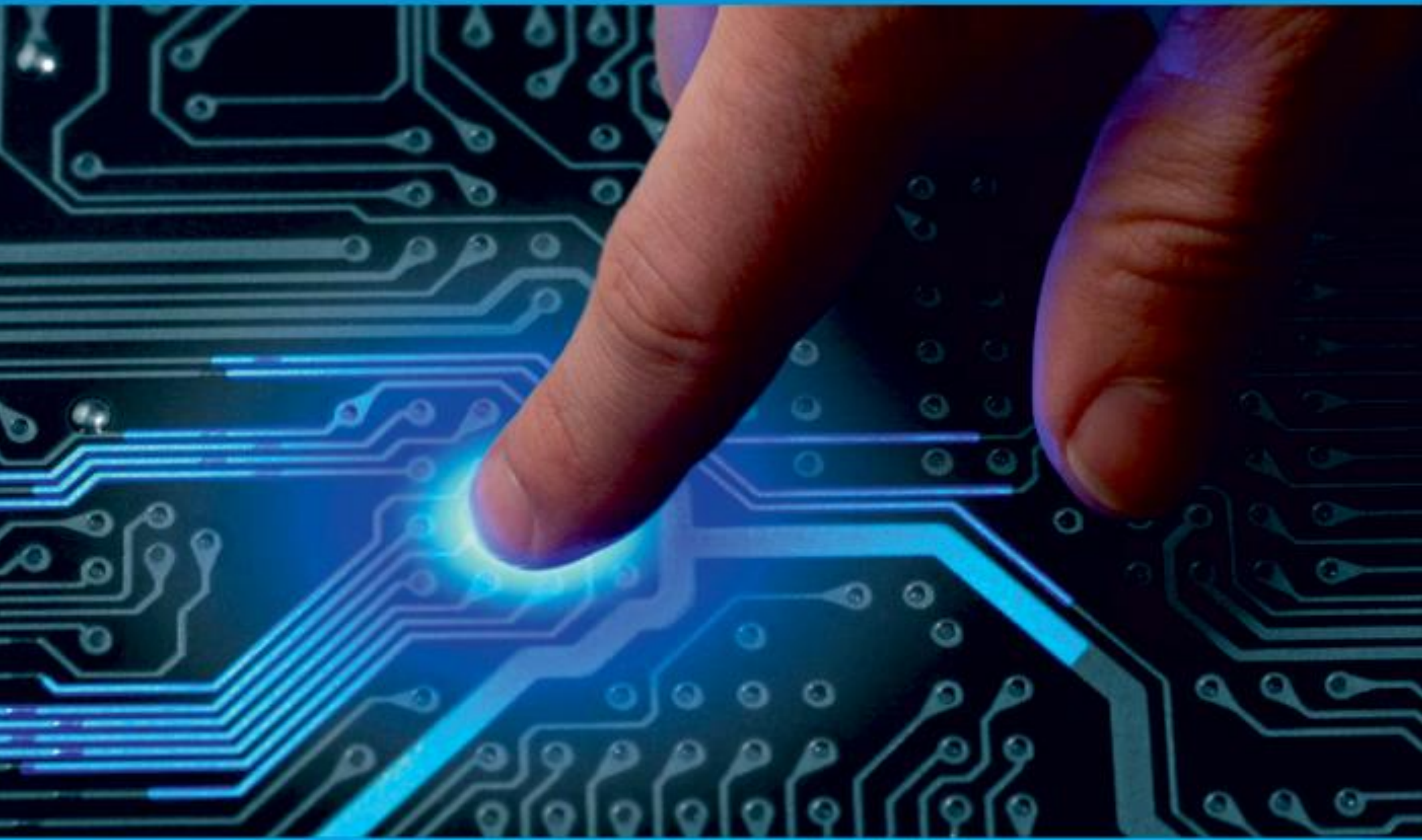
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 5, May 2023

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 5, May 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1105002 |

A Model to Predict Tomato Plant Leaf Disease

Pianos Gweme[1], Kudakwashe Zvarevashe[2], Raymond Zenda[3], Greenford W Katuruza[4], Paul Chikovo[5]

Software Engineering Student, Dept. of S.E, SIST, Harare Institute of Technology, Harare, Zimbabwe[1]

Data Science PhD, Dept. of C.S, University of Zimbabwe, Harare, Zimbabwe[2]

Software Engineering Student, Dept. of S.E, SIST, Harare Institute of Technology, Harare, Zimbabwe[3]

Software Engineering Student, Dept. of S.E, SIST, Harare Institute of Technology, Harare, Zimbabwe[4]

Software Engineering Student, Dept. of S.E, SIST, Harare Institute of Technology, Harare, Zimbabwe[5]

ABSTRACT: Plant disease monitoring or diagnosis are crucial since failing to do so could cause a significant decline in agricultural yield. The most vulnerable component of crops, the leaves, are where symptoms usually first appear. Machine learning algorithms were developed for identify various plant diseases using various image classifiers, nevertheless, they are costly, imprecise, and demand heavy computation. This can be attributed to issues with poor or wrong labelled datasets, redundant and irrelevant features, and poor feature extraction. This encourages scientists to use scientific techniques that are greater sophisticated and don't demand as much human involvement to recognize plant diseases.Under this study, deep learning (DL) algorithm was developed to detect crop disease using twenty six thousand tomato leaves, both health and unhealthy. Furthermore, systematic statistical literature review on machine learning (ML) and DL techniques for crop disease detection was rigorously conducted. The proposed innovative leaf detection technique algorithm was implemented and then compared to established plant leaf detection methods. Upon successfully implementation of the technique, the new algorithm recorded performance accuracy of 96 percentage. This research will go a long way in the scientific and community world in solving crop diseases throughlighter DL model of recognising plant diseases.

KEYWORDS: Plant Diseases Detection, Deep learning, leaf disease, Tomato, Machine learning.

I.INTRODUCTION

Tomatoes are among the most significant and commonly eaten crops in the entire globe[1].. The quantity of cropland used for tomato has expanded by 164% during the last 40 years, and in 2018 there were 182 metric tonnes produced worldwide. [3]. Several leaf infections can harm normal tomatoes, resulting in lower or lower-quality harvests.
Therefore, minimizing output losses and ensuring optimal output will be made possible by early and precise detection and categorization of tomato diseases[3].

The classification of plant leaf diseases has historically been done by qualified professionals visually examining plant tissues, however this technique has been shown to be expensive, time-consuming, and ineffective. ML algorithms have recently seen widespread use in machine learning for automating the classification of plant diseases. For classifying plant diseases, the most popular traditional ML techniques are neural networks (NN), logistic regression (LR), and random forest (RF).)[4], support vector machines (SVM)[4], adaptive boosting (AdaBoost)[5] and k-nearest neighbours (k-NN)[6]. The features we give these conventional machine learning methods, nevertheless, are really important.These features are physically extracted by the specialist, which adds cost and time to the process.

The restriction of the handmade features technique can be readily overcome by automated feature extraction utilizing DL. DL is often used for classifying plant diseases since it outperforms standard ML[7]. A deep convolutional neural network (DCNN) is the method that is most widely used to classify plant diseases[8]. DL models, meanwhile, rely on variables that must be adjusted prior to application. Hyperparameters are the name given to those parameters[9]. In several circumstances, the configurations of an algorithm's model parameters determine how well it performs on a specific learning job[9]. Machine learning engineers can adjust the hyperparameters to get the greatest results. The establishment of a search area, the list of hyperparameters, as well as the limits that must be taken into account are necessary for techniques for adjusting hyperparameters for the specific issue. The query method which is utilized to look for suitable hyperparameters inside the search area must also be defined. Leaf disease picture categorization has been studied using deep learning methods. There have been claims that a number of hyperparameters can affect how well they function, but there is no proof of which variable predicts leaf disease in plants images with the best accuracy.In addition, this is unclear how such hyperparameters might be improved to achieve the exact objective. Under this study, we want to determine the parameter best predicts the calculation and precision of leaf disease in plants image classification. In principle, adjusting is a computationally costly proposition that gets greater challenging even as size of the search area grows.The importance of tuning certain hyperparameters and with which hyperparameters produce equal efficiency whenever put to such a fair set point are just not presently supported by evidence.

In this study, hyperparameters that are utilized to classify plant leaf diseases will be identified, and their effects on classification techniques were evaluated. The effectiveness of an enhanced plant leaf disease classification method was discussed in the research. 6000 pictures of tomato leaves with 4 different symptoms make up the training set. This article responds to a query about hyperparameters used in image classification techniques for leaf disease. This picture classification were resolved using the following query. How can the effectiveness the algorithms for classifying images of plant leaves disease be improved holistically?

II RELATED WORK

To identify diseased plant leaves, Vijai Singh et al., used learning algorithm and image analysis. They concentrated on accurate illness classification segmentation into five sub-segments. After generating a randomly selected community and determining its efficiency, they chose chromosomal couples to produce an offspring and replace the initial species. An image with segments was the result. They discovered that all aspects of image analysis should indeed be combined for more precise and reliable detection, despite the fact that segmentation was their primary focus [21].

Visible plant disease recognition was examined by Rahul Das et al. RGB digital cameras were employed to capture images of the infected areas. RGB was converted to HSI and CIELAB. To classify plants and choose the best traits and identifiers, they employed edge detection method and k means. By employing clustering, the project will avoid using several classifiers. Transformation from RGB to HIS is effective [23].

To find wheat illnesses, Gaikwaad and Dr. Vijay employed image processing. Photographs of wheat were taken with a digital camera, and they were afterwards reduced utilizing subtraction. They used a 3*3 median filter to minimize noise during pre-processing. They took out features for color, form, and texture. In its application, SVM outperformed than NN. Since more acquired data boost accuracy, the suggested project would generate more statistical parameters from the gray level co-occurrence matrix [6].

A internet mobile app was created by Zarreen Naowal Reza et al. to identify jute plant disease utilizing image processing and ML. This was connected to an image processing Matlab engine and library. Their internal servers receive Matlab results, and the resultant image is uploaded to the internet. Clients get control measurements and image outcomes. RGB to HSV conversion, segmentation, masking, extraction and classification for texture analysis, and SVM classification were all part of the pre-processing. Through practical application, they achieved an accuracy of 80%. The proposed project will determine efficiency following certain practical trials and, if any loopholes are found, assist in improving the platform [7].

Nikos Petrellis' plant disease detection app employed a cutting-edge GPS localization technique to determine the plant's remote position. According to the study, a plant's backdrop should be lighter than its foreground. The app disseminated weather data. The proposal may include the latitude and longitude of the plant. They concentrated on establishing a link between the local weather and plant diseases [28]. In the upcoming projects, context removal will enable higher precision and efficiency, but computation might go up.

In order to estimate the extent and severity of plant diseases from leaf images, Ernest Mwebaze et al. employed ML.Cassava plant health was forecasted via a real-time smartphone app. The severity of the sickness was assessed. Just one data collected included colour and form. Colour properties were extracted using HSV transformation. Linear SVC, KNN, and highly stochastic trees are the three classifiers. To eliminate background concerns, this desktop application transforms the images [21].

III.PROPOSED ALGORITHM

A. *Dataset:*

The Kaggle Tomato Plant Leaf Disease Dataset is a brand-new reading comprehension dataset that includes the 10 classes BacterialSpot, EarlyBlight, Healthy, LateBlight, LeafMold, SeptoriaLeafSpot, SpiderMites, TwoSpottedSpiderMite, TargetSpot, TomatoMosaicVirus and TomatoYellowLeafCurlVirus. The dataset includes 10,000 training plant leaves—1,000 in each class—as well as 10,000 validation plant leaves—100 in each class. At https://www.kaggle.com/code/taha07/plant-disease-detection/data, the dataset is freely accessible. An illustration of the tomato leaf photos taken from the dataset utilised is shown in Figure 1.

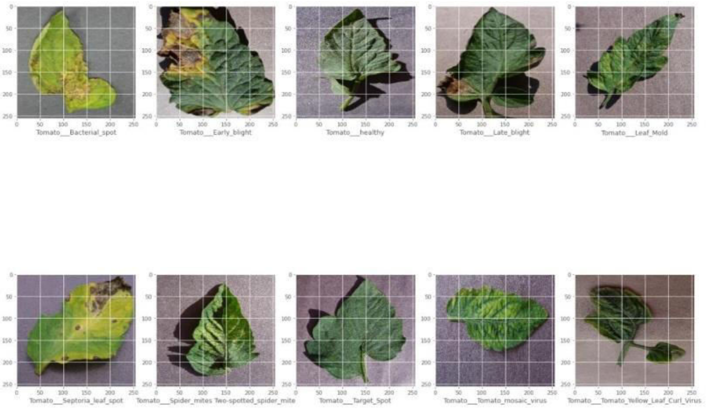

Tomato___Bacterial_spot
Tomato___Early_blight
Tomato___healthy
Tomato___Late_blight
Tomato___Leaf_Mold
Tomato___Septoria_leaf_spot
Tomato___Spider_mites Two-spotted_spider_mite
Tomato___Target_Spot
Tomato___Tomato_mosaic_virus
Tomato___Tomato_Yellow_Leaf_Curl_Virus

Figure 1: Dataset visualization

B. *Algorithm Setup:*

Step 1: Minibatch size
To test the algorithms, minibatch sizes of 32 and 64 were employed.

Step 2: Number of Epochs and Number of Iterations
The network was trained using a single iteration and many epochs beginning at 10.

Step 3: Learning Rate
For the purposes of comparison, the researcher used a learning rate of 0.1 with early halting in this investigation.

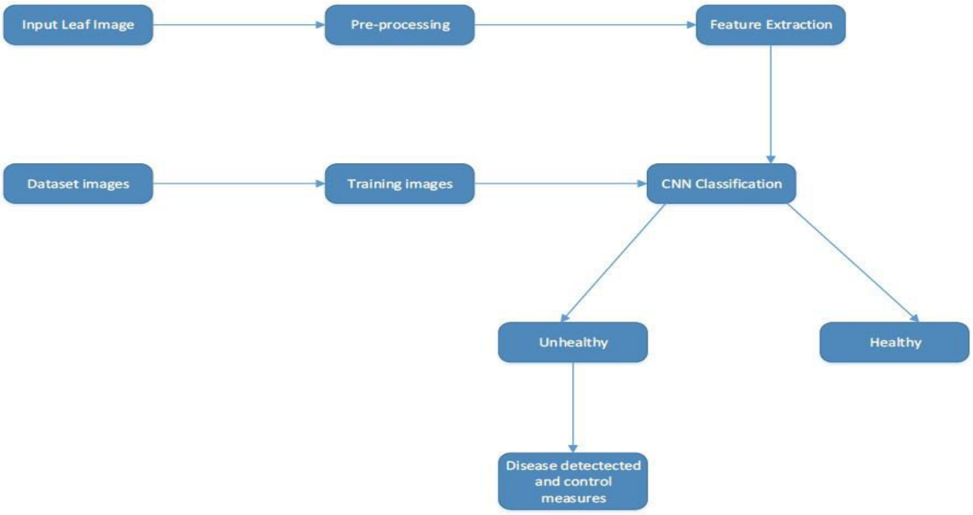
Step 3: Activation Function
Both the Softmax activation function and the Rectified Linear Unit (RELU) activation function were used.

Step 4: Loss Function
Each layer of deep neural networks uses loss functions, which can be pre-trained to discover more pertinent weights.

Step 5: Optimisation Algorithm
In this work, Adam was used as the default optimizer for the methods and later compared with rmsprop and adagrad.

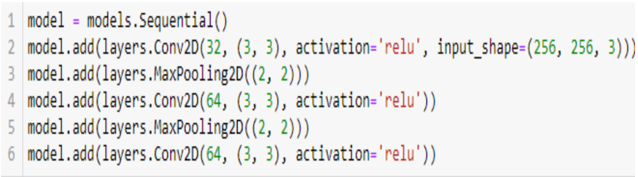
Step 5: Layers
In this experiment, the researcher closely connected the layers between the input and output.

C. *Proposed Methodology:*



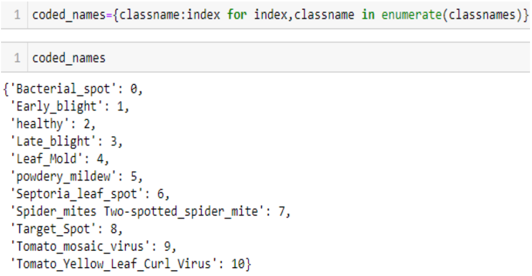
D. *Algorithm Design:*

The convolutional neural network was created using Tensorflow and the following model.



```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

The CNN model that was created has two convolved layers and a maximum pooling dimension of 2x2. The class index of the list's disease categories was defined using the following code.



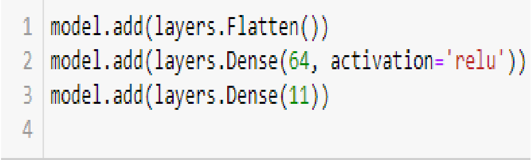
```
coded_names={classname:index for index,classname in enumerate(classnames)}
```

```
coded_names
```

```
{'Bacterial_spot': 0,
 'Early_blight': 1,
 'healthy': 2,
 'Late_blight': 3,
 'Leaf_Mold': 4,
 'powdery_mildew': 5,
 'Septoria_leaf_spot': 6,
 'Spider_mites Two-spotted_spider_mite': 7,
 'Target_Spot': 8,
 'Tomato_mosaic_virus': 9,
 'Tomato_Yellow_Leaf_Curl_Virus': 10}
```

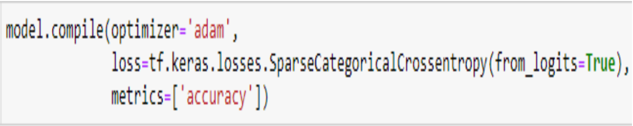
The results indicate that the studied leaves contain 11 different disease types. The diseases have a 0–10 index. As a result, the statistics are balanced in terms of the distribution of diseases. The train dataset and the indexed labels were combined using the code below.

The CNN model was given a flattening layer by using the following code.



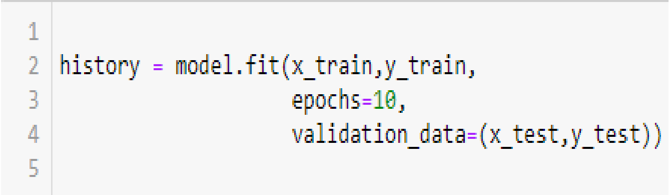
```
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(11))

```

The following summary extract illustrates the new model's summary with the flattening layer applied. The following code snippet was used to create the model:



```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

The model was built using Adam optimiser and loss as a metric for model loss as well as accuracy as a metric for model performance, as evidenced by the model compilation code in the aforementioned code. The following code was used to fit the model:



```

history = model.fit(x_train,y_train,
                    epochs=10,
                    validation_data=(x_test,y_test))

```

The history of the model was tracked, as seen in the code snippet, by fitting the model to the training and testing sets using 10 epochs of data validation.

II. Results

We put our system to the test on a Kaggle dataset on tomato plant diseases that was split 80:20 for training and validation. Our primary functional metric is accuracy. To assess it, the performance of the algorithm was examined.

A. Analysis of Algorithm Accuracy

Our algorithm's accuracy graph for training and validation is displayed. The training and validation accuracy are shown on the same axis in Figure 2.

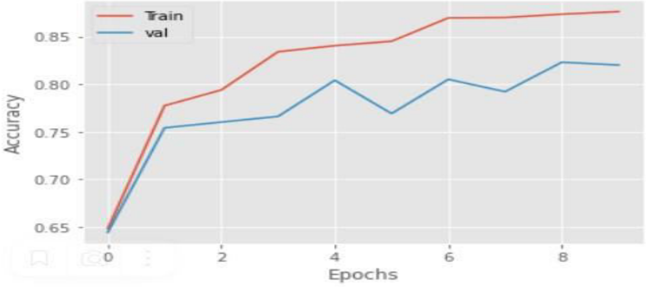


Figure 2: Training and Validation accuracy for our algorithm over ten epochs.

Figure 2 shows that training accuracy consistently beats validation accuracy during the displayed epochs. From the first through the tenth epochs, training accuracy increased gradually from 78% to 88%, whereas validation accuracy fluctuated between 76% and 82%. The model improved its training efficiency through increased accuracy, as evidenced by the positive expansion of the accuracy curves on the plane, according to the scores for both the training and validation sets. The accuracy of the model should, in theory, increase as one approaches

III. CONCLUSION AND FUTURE WORK

Various leaf diseases can harm healthy tomato plants, resulting in lower or lower-quality harvests. The early and precise identification and categorization of tomato leaf disease is crucial for ensuring optimal production and minimising yield losses. This study employs a deep learning algorithm to identify tomato plant leaf diseases. The algorithm's performance accuracy was assessed. We draw the conclusion that our algorithm can increase the accuracy of disease identification because it performed better than previous algorithms. 90% accuracy was attained by our algorithm. In future work, we intend to improve our algorithm's precision and broaden its use to include other plants besides

REFERENCES

[1] T. Biswas and F. Wu, “The US Tomato Industry : An Overview of Production,” pp. 1–4, 2017.

[2] I. Kibriya and M. Adnan, "Tomato Leaf Disease Detection Using Convolution Neural Network," no. January, 2021, doi: 10.1109/IBCAST51254.2021.9393311.

[3] F. Shang, M. Khayyat, and A. A. A. El-latif, "Tomato leaf disease classification by exploiting transfer learning and feature concatenation," no. October 2021, pp. 913–925, 2022, doi: 10.1049/ipr2.12397.

[4] A. S. Zamani *et al.*, "Performance of Machine Learning and Image Processing in Plant Leaf Disease Detection," *J. Food Qual.*, vol. 2022, pp. 1–7, 2022, doi: 10.1155/2022/1598796.

[5] U. N. Fulari, R. K. Shastri, and A. N. Fulari, "Leaf Disease Detection Using Machine Learning," vol. 15, no. 9, 2020.

[6] M. F. Ab Jabal, S. Hamid, S. Shuib, and I. Ahmad, "Leaf features extraction and recognition approaches to classify plant," *J. Comput. Sci.*, vol. 9, no. 10, pp. 1295–1304, 2013, doi: 10.3844/jcssp.2013.1295.1304.

[7] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.

[8] J. Trivedi, Y. Shamnani, and R. Gajjar, "Plant Leaf Disease Detection Using Machine Learning," *Commun. Comput. Inf. Sci.*, vol. 1214 CCIS, no. September, pp. 267–276, 2020, doi: 10.1007/978-981-15-7219-7_23.

[9] H. J. P. Weerts, A. C. Müller, and J. Vanschoren, "Importance of Tuning Hyperparameters of Machine Learning Algorithms," 2018.

[10] N. Reimers and I. Gurevych, "Reporting score distributions makes a difference: Performance Study of LSTM-networks for sequence tagging," *EMNLP 2017 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 338–348, 2017, doi: 10.18653/v1/d17-1035.

[11] R. T. Mutanga, "A Comparative Study of Deep Learning Algorithms for Hate Speech Detection on Twitter," 2020.

[12] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *J. Phys. Conf. Ser.*, vol. 1237, no. 2, 2019, doi: 10.1088/1742-6596/1237/2/022030.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[14] B. Taiwo and T. Bukola, "Colour Image Segmentation Using Perceptual Colour Difference Saliency Algorithm," no. July, 2018.

[15] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, "Introduction to Deep Learning," *Stud. Big Data*, vol. 57, pp. 1–11, 2020, doi: 10.1007/978-981-13-6794-6_1.

Disease Detection," *J. Food Qual.*, vol. 2022, pp. 1–7, 2022, doi: 10.1155/2022/1598796.

[5] U. N. Fulari, R. K. Shastri, and A. N. Fulari, "Leaf Disease Detection Using Machine Learning," vol. 15, no. 9, 2020.

[6] M. F. Ab Jabal, S. Hamid, S. Shuib, and I. Ahmad, "Leaf features extraction and recognition approaches to classify plant," *J. Comput. Sci.*, vol. 9, no. 10, pp. 1295–1304, 2013, doi: 10.3844/jcssp.2013.1295.1304.

[7] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.

[8] J. Trivedi, Y. Shamnani, and R. Gajjar, "Plant Leaf Disease Detection Using Machine Learning," *Commun. Comput. Inf. Sci.*, vol. 1214 CCIS, no. September, pp. 267–276, 2020, doi: 10.1007/978-981-15-7219-7_23.

[9] H. J. P. Weerts, A. C. Müller, and J. Vanschoren, "Importance of Tuning Hyperparameters of Machine Learning Algorithms," 2018.

[10] N. Reimers and I. Gurevych, "Reporting score distributions makes a difference: Performance Study of LSTM-networks for sequence tagging," *EMNLP 2017 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 338–348, 2017, doi: 10.18653/v1/d17-1035.

[11] R. T. Mutanga, "A Comparative Study of Deep Learning Algorithms for Hate Speech Detection on Twitter," 2020.

[12] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *J. Phys. Conf. Ser.*, vol. 1237, no. 2, 2019, doi: 10.1088/1742-6596/1237/2/022030.

[13] B. Taiwo and T. Bukola, "Colour Image Segmentation Using Perceptual Colour Difference Saliency Algorithm," no. July, 2018.

[14] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, "Introduction to Deep Learning," *Stud. Big Data*, vol. 57, pp. 1–11, 2020, doi: 10.1007/978-981-13-6794-6_1.

BIOGRAPHY

Pianos Gweme is a spatial data scientists at the Zimbabwe National Geospatial and Space Agency with vast experience in near real time spatial data collection using UAVs,mobile devices, satellites etc for various applications such as agriculture, mining, disease surveillance, environmental management etc. He is the chief investigator of ZIMSAT-1 the first satellite of Zimbabwe and Chairman of Technical committee of the Zimbabwe center for high performance computing. His research interests are in remote sensing, satellite communication, UAVs software architecture and precision agriculture using UAVs.

Kudakwashe Zvarevasheis a Chairman Computer Department, University of Zimbabwe. He received PhD in Data Science. His research interests are Data science and Big Data.

Raymond Zenda is a Masters student in the Software Engineering Department, School of Information Science and Technology, Harare Institute of Technology (HIT). He received Bachelor of Technology degree in Information Technology degree in 2019 from HIT, Harare, Zimbabwe. His research interests are Deep Learning, Big Data Analytics, and Natural Language Processing etc.

Greenford Walter Katuruzais a Software Engineer at Zimbabwe Centre for High performance Computing (ZCHPC) and Space Scientist at Zimbabwe National Geospatial and Space Agency (ZINGSA). He is currently pursuing Master of technology at Harare Institute of Technology, School of Information Science and Technology, Department of Software Engineering, He received Bachelors of Science Honours Geographyin Geospatial Intelligence degree in 2019 from University of Zimbabwe, Harare, Zimbabwe. His research interests are onDeep Learning, AI, Algorithms, Automation, Space technology etc.

Paul Confidence Chikovo is a Masters student in the Software Engineering Department, Harare Institute of Technology. Paul received his Bachelor of Science Honors in Computer Science in 2017 from National University of Science and Technology, Bulawayo, Zimbabwe. His research interests are in NLP, recommender systems, and Deep learning.