ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2022.0401

IMPROVING MEDICAL IMAGE PROCESSING USING AN ENHANCED DEEP LEARNING ALGORITHM

C. Kiran Kumar[1], R. Gayathri[2], S. Thirukumaran[3] and P.T. Kalaivaani[4]

[1]*Department of Data Science, Codecraft Technologies, Bangalore, Karnataka, India.*
[2]*Electronics and Communication Engineering, Rajalakshmi Engineering College, India*
[3]*Department of Data Science, School of Science, Jain University, India*
[4]*Department of Electronics and Communication Engineering, Vivekanandha College of Engineering for Women, India*

Abstract

The use of ML methods with the objective of selecting wheat varieties that have a higher level of rust resistance encoded in their genomes is referred to as rust selection. In addition to that, the categorization of wheat illnesses by means of machine learning It has been attempted to classify wheat diseases by making use of a wide variety of machine learning techniques. In this paper, we develop an enhanced deep learning model to classify the disease present in the wheat plant. The study uses an improved convolutional neural network to classify the plant disease using a series of layers. The simulation is conducted in terms of the accuracy, precision, recall and f-measure. The results show that the proposed method achieves higher rate of accuracy than its predecessor.

Keywords:

ML, Wheat Varieties, Rust Resistance, Disease

1. INTRODUCTION

There is currently a significant amount of research being carried out in the sector with the intention of improving wheat disease detection produced an inventory of diseases that can impact wheat and developed a method for diagnosing wheat diseases by using a semi-supervised diagnostic methodology. One of the potential benefits of genetic modification is an increase in the individual resistance to certain diseases. It can be considered an independent academic subfield. On the other hand, the use of AI may make it less difficult to uncover novel methods.

It has been demonstrated that support vector machines and decision forests are the most effective feature selection-based techniques that have been developed to this day. Because feature extraction is done implicitly throughout the learning process, deep learning models do not require it because it is superfluous for these models and cannot function properly without it.

These approaches have the capability to extract and learn from these qualities on their own. As a direct result of this, the methods of deep learning are now referred to as end-to-end machine learning approaches, and the classification of agricultural illnesses is one of the many areas in which they are extraordinarily useful. When it comes to sorting images into a variety of categories, deep neural networks, and more specifically convolutional neural networks, do exceptionally well.

When it comes to categorization, recent findings have demonstrated that these algorithms perform far better than more conventional supervised learning approaches. Because of the efforts of researchers, the amount of progress that has been made in the models of deep learning networks has significantly increased. This is because researchers have been able to make substantial strides. However, recurrent neural networks, also known as RNNs, do not have these problems. Convolutional neural networks, commonly known as CNNs, have a number of limitations. The integration of both long-term and short-term memory into the RNNs has resulted in considerable performance enhancements (LSTM).

Deep learning classifiers, on the other hand, are limited by their own special set of parameters. These approaches require a substantial quantity of data and processing resources during the phase in which they are being trained. The second major issue is referred to as overfitting, and it happens when there is a considerable discrepancy between the accuracy of the model when it is trained and when it is tested. The data has been permanently stored in memory, which prevents the model from learning from its past experiences and developing because of these learnings. Having a high sample size, using regularization techniques like dropout, and integrating extra data can all be beneficial in minimizing the likelihood of overfitting occurring in a model. Having a big sample size is very important. Transfer learning is a method that can help improve classification performance [10], particularly in situations in which there is a scarcity of both data and processing resources. In these kinds of situations, transfer learning can be especially helpful

Transfer learning can be seen as the process of applying the parameters of a model that has already been trained, despite the fact that the previous model may not have been constructed using the same dataset as the present-day model. One can choose and select from a broad number of deep learning architectures that have previously been trained to complete straightforward tasks such as the classification of photos. A wide variety of parameters, including as the number of classes, the diversity of the data, and the degree to which there is an imbalance between the classes, can all have an effect on the accuracy of the classification that is carried out by a deep learning model [7]. The extent to which the quantity and composition of the dataset influence the accuracy of the model is described in [8]. The classification of disorders is achieved through the application of deep learning strategies [9]. provided an overview of the many different approaches to deep learning that can be utilized to grow a variety of plant species.

2. RELATED WORKS

The author evaluated the performance of the traditional network models AlexNet and GoogleLeNet on the identification of 14 crop species and 26 illnesses by using a public database that contained 54,306 photographs that were taken under controlled conditions. This allowed the researchers to compare the accuracy of the models. This database can now be accessed via the internet. The fact that it was feasible to achieve a maximum accuracy of

99.35% was evidence of the approach applicability in real-world situations. When the accuracy of the model was put to the test using a series of photographs that had been taken under conditions that were different from those that were used for training, the model suffered a significant setback in terms of its accuracy. Under this experiment, only healthy plants with single leaves pointing upward against a uniform background were employed; as a result, the success rate of diagnosing plant illnesses in situations more similar to those found in the natural world would be substantially lower [5].

Researchers Fuentes et al. wanted to design a reliable detector based on DL that could identify tomato diseases and pests in real time. Their goal was to accomplish this goal. Although the backdrops, lighting, and sizes of the individual items depicted in the photographs of plant pathogens and pests may vary from one another, each photograph was taken in the natural environment in which the subject was discovered. If it were implemented in real life, the precision would be poorer because there were so few samples to begin with [2]. The availability of adequate datasets has a considerable bearing on how the implementation will actually be carried out. The process of classifying photographs is laborious and time-consuming in and of itself, and the collection of data is susceptible to being easily influenced by external factors such as the time of year and the weather. Due to the enormous number of aspects that need to be taken into consideration, the process of creating a useful dataset is riddled with complexity and can be rather challenging. The five ways that are now available for tackling issues with datasets include transfer learning, data augmentation approaches, few-shot learning, citizen science, and data sharing. Each of these five tactics is designed to help solve certain difficulties.

The term transfer learning comes from the field of machine learning and refers to a process in which previously learned knowledge can be applied in an effective manner in a variety of different contexts. It is helpful for lowering the need for massive datasets [3] when only a few layers of previously trained networks are retrained with the new databases.

It will be able to identify plant illnesses by utilizing a transfer learning model that was constructed on ResNet 50. The full collection of photographs that they have available in their database comes to a total of 87,867. Two distinct parts of the dataset were extracted: a twenty percent validation segment and an eighty percent training piece. They were able to achieve an accuracy rate of 99.80 percent at their finest [1].

A method that makes use of transfer learning was developed, in order to assess whether or not pearl millet is affected with mildew. The well-known VGG-16 traditional CNN model served as the basis for the development of this strategy during its development. The model was initially trained using the ImageNet dataset, which is available to the general public. The experiment was a success, with an accuracy rate of 95% and a recall rate of 94.5%, respectively [4].

In their study on the semantic segmentation of images of oilseed rape, [5] used three distinct approaches to transfer learning, each of which led to exceptionally successful results. The findings of the experiment demonstrated that transfer learning has the potential to be effective in situations such as this one, which involves the segmentationtic segmentation of photos of oilseed rape. The findings of the experiment demonstrated that transfer learning has the potential to be beneficial in a position such as this one, which involves segmentation. The results of the trial showed that it was accurate 96% of the time.

3. METHODS

A live stream of the training dataset is sent to the deep learning model so that it can learn more effectively. The model is educated to be able to identify between ten distinct kinds of data with the assistance of these pieces of information.

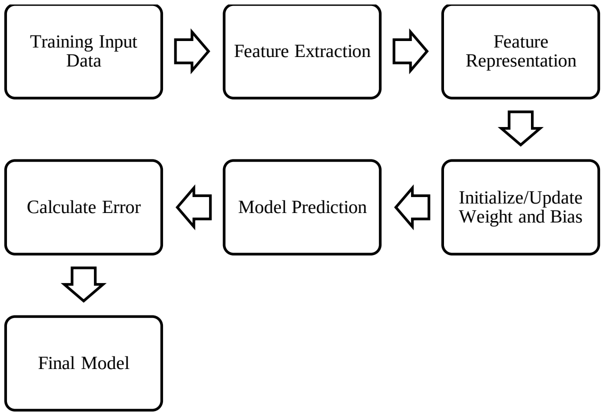

Fig.1. Proposed Method

During the forward pass of the model, the weights (parameters) of the model are updated so that they can express the error that was determined by making use of both the predicted value and the expected value. This is done so that the model can more accurately represent the data. The process of training is carried out repeatedly throughout the course of an established number of epochs (also known as periods). When the training step is complete, both the model and the accuracy metrics associated with it are written to disk. Following that, this model is utilized in the process of labeling the samples that are kept hidden from the view of the public.

The classification of wheat diseases was accomplished through the application of the deep convolutional model that was proposed. When appropriately utilized at multiple stages of the signal processing pipeline, such as convolution, pooling, and regularization, convolutional neural networks have shown to be particularly effective in the resolution of difficulties related to the categorization of images. In particular, this has been demonstrated to be the case in the resolution of difficulties relating to the classification of images. Nearly 12,000 pictures of various wheat illnesses were included in the training process for the proposed model. In the end, the trained model is able to take in an image and affix a disease categorization label (or a healthy wheat label, if one exists) to the picture. This is done by taking in the image and running it through the trained model. This capability is only attainable in the event that a disease classification label is present.

It requires an RGB image with the necessary dimensions and 224x224 pixels as the input for the training process. The convolutional layers of the model each have a kernel size that is equal to three of the dimensions that were provided. The convolutional layers of the model each have a kernel size of three, which is equivalent to three. A strategy known as max pooling can be utilized to bring the size of the training vector space down to a more manageable level. Twenty-one convolutional layers, seven max-pooling layers, and three fully linked layers make up the model that is being presented right now. In the convolutional layer of the network, Rectified Linear Units (ReLU) and Leaky Rectified Linear Units are utilized as activation functions. Leaky Rectified Linear Units are also used. In the more densely packed layers of the network, the ReLU is used to implement the activation function. The SoftMax classifier comes after a dropout layer and two layers that are entirely coupled to each other. The structure is finished off with a dropout layer at the very bottom (0.5). Adam is the name of the optimizer that was utilized in this process.

One-pixel convolution strides are applied. The width of the network is projected to start at 64 nodes and then rise by a factor of two after each pooling layer. This is according to estimates. The suggested architecture can be learned using any number of parameters in the range from 25 to 356, inclusive. The convolutional and fully connected layers of the neural network are the only ones for which training weights are currently accessible.

After the size of the input image has been decreased with the assistance of a Max pooling layer, the softmax layer is the one that makes the final determination. After these feature-identifying layers of convolution and pooling, the following stage will be to proceed to the dense layers for the purposes of learning and prediction. These dense layers will be comprised of several smaller layers.

An activation function that normalizes a distribution that is dynamically created from K composite functional concatenations is utilized at the very end of the dense layer of the SoftMax algorithm. This function is called at the very end. Each successive layer of nodes in the network learns from the output of the layer that came before it. There are several layers of nodes in the network. Because of this, nodes at deeper and deeper layers acquire the ability to distinguish characteristics that are becoming progressively more complex.

4. IMPROVED CONVOLUTIONAL NEURAL NETWORK

One example of a well-known architecture for deep learning is the Convolutional Neural Network (CNN), which stands for convolutional neural network. This architecture incorporates feed-forward connections between each of its tiers. The hierarchical representation of an input image is learned in its entirety by the CNN architecture, starting from the very beginning and continuing all the way through to the very end. In order to extract both local and global information from each image, this architecture has a significant number of additional layers compared to the one that came before it. Recent advancements in the performance of these models have been of assistance to a variety of diverse applications, such as object categorization [41], surveillance [42], and medical imaging [43]. Some of the most well-known layers in a typical CNN architecture are the convolutional, ReLu, pooling, and fully connected layers. Other well-known layers include the fully connected layer. The fully connected layer is another well-known layer that can be found (FC). The convolutional layer, the pooling layer, and the fully connected layer are the three main parts that make up the CNN model. These three parts are what hold the model together. There is a way to fix the issues of overfitting and generalization that arise in CNNs, and that way is to design the architecture so that it includes dropout and batch normalization layers [44]. The completion of this design comes in the form of the removal of elements on an abstract level, which is then followed by the reintroduction of a score.

In the construction of a CNN, one of the most important components is the inclusion of a convolutional layer that allows for the parameterization of its weights. The use of these weights allows for the acquisition of spatial information such as edges and contours, in addition to higher-level properties such as complex structures. This layer provides feature maps as output and is responsible for accepting input from HW3. In addition to that, feature maps are generated. Every single feature map may be described mathematically as the dot product of a field and a few weights. This is the case regardless of the type of map. The properties of each class are analyzed throughout training, and the results of those analyses are included into the weighting system. Backpropagation and the SDG method are both applied throughout the process of learning (gradient descent). The following is an example of the mathematical definition of this layer:

$$\psi_{ik}=Rel(\beta_{ik} + \mathbb{W}_{ik}\times\mathbb{X}_{i-1}), \tag{1}$$

where

ψ_{ik} - convolutional output,

β_{ik} - bias term,

$\mathbb{W}_{ik}$ - convolutional weight,

$\mathbb{X}_{i-1}$ - input terms, and

$Rel(.)$ - ReLu activation function.

The weights of a given layer are adjusted so that they accurately reflect the new information whenever the output of one layer is used as the input for the layer that comes after it. This capacity can be understood by looking at its definition, which can be found below:

$$Rel(\mathbb{x})=\max(0,\mathbb{x}). \tag{2}$$

Negative weights are reset to zero as part of the process of carrying out the aims of this function, whilst positive weights are not changed in any way over the course of the operation. To find the values that are the highest possible inside a rectangular area that is set by the size of the filter, a max-pooling layer is added into the design of a CNN. This is done to maximize the accuracy of the CNN. The value of the stride determines the amount that each rectangle is moved away from its starting position when it is relocated. Reduced requirements for the total number of features are possible thanks to the aid provided by this layer (weights). The specifications of this layer include the dimensions of the filter, the type of padding that is utilized, as well as the stride. When reduced to its mathematical form, it takes on the following appearance:

$$\psi_{ik}(\max)=\text{Poolmax}(Rel(\mathbb{x})),\ Rel(\mathbb{x})\in\psi_{ik}. \tag{3}$$

At some time in the future, each of these levels will transfer the output two-dimensional arrays that they generate to a layer that is completely connected. After that, this layer will simplify the output so that it just has one dimension. When assigning labels to the features that are located within this layer, the Softmax classifier is used as a method of classification.

$$\text{Softmax} = \psi_{ik}(\text{FC}) = \psi_k(\text{FC}), \tag{4}$$

where

$\psi_{ik}(FC)$ - FC layer output in Softmax and

K - classes.

4.1 PRE-TRAINED DENSENET 201 MODEL

Moving data from one layer to the next in a conventional CNN was a challenging and time-consuming procedure. During this procedure, there is a substantial possibility that mistakes will be made, which will result in an increase in the total amount of money spent on computing. The Resnet architecture manages to tackle the complexity problem while also making it easier to grasp. This is accomplished by skipping some of the layers in the architecture. This architecture does not make use of the requisite initial two tiers at any point in its development. Densenet is beneficial to the development of the model since it concatenates all of the characteristics in a sequential order, so linking all of the features together (linear form). This method is an advancement on the one that came before it, which only accumulated the characteristics in the output layer and passed them onto the layer that came after it. This current method, on the other hand, does not accumulate the characteristics in the output layer. The following is a description of the procedure that is being detailed in more detail:

$$S_m=Z_m(s_{m-1}), \tag{5}$$

$$S_m=Z_m(s_{m-1})+s_{m-1}, \tag{6}$$

$$S_m=Z_m([s_0,s_1,s_2,,\ldots,s_{m-1}]), \tag{7}$$

where

m - layer index,

Z - non-linear operation, and

S_m – m^{th} layer feature.

As is evident in the figure that follows, the ImageNet database served as the primary training set for this model when it was first developed. The process of classifying the data can be sped up in several ways, one of which is to make use of the skip option that is provided in each block. Convolution, ReLu, batch normalization, and pooling are some of the layers that are included in each block, which is made up of several different layers in total. In order to finally establish the characteristics that are associated with the transition layer, a softmax classifier is used to the data.

DenseNet201 was a model that had been trained previously and was developed with the intention of recognizing a wide variety of distinct forms of brain tumors. In this context, we made certain enhancements to that model. In order for this to be accomplished, we need to get rid of the two lowest layers and replace them with a layer known as functional connectivity (FC), which includes four distinct types of brain tumors. This will allow us to accomplish our goal. The new model, which makes advantage of transferred learning and training, maintains the same weights as the previous version. A more in-depth explanation of transfer learning is going to be provided.

4.2 TRANSFER LEARNING-BASED FEATURE EXTRACTION

Transfer learning (TL) is one of the most effective ways to put a deep learning model to work after it has already been trained [46]. This is because TL takes advantage of the fact that the model has previously been trained. In addition to that, it is one of the most difficult types of education. Much of the time, the TL method is utilized to train a model for a distinct endeavor using a reduced number of data points. On the other hand, in this instance, we are training a target model by making use of the information that we have previously gained from participation in the connected activity. Even if you just have a limited amount of training samples to deal with, you could still be able to produce a satisfactory answer with the help of this method. As a result, we are able to reach the realization that TL is useful in circumstances in which there are less data sets available for training the target in compared to the source.

Take into consideration the information sources labeled as PL=LP, LS, (dsm, esm), R, and a learning assignment. Ps=(ds1,es1),...,(dsi,esi),...,(dsn,esn)). In addition to the target labels (JL = LJ, LT, (dtn, etn)) that are associated with the educational activity, there is also target data. JT=(dt1,et1),...,(dti,eti),...,(dtm,etm)). where the designations nm, eP1, and eT1 originate; the training set is the source for them. PS and JT need to collaborate in order to improve JT teachability and make it possible for TL to achieve its goals. As a result of this, we are able to explain transfer learning by using the following terms:

$$Ps\neq JT \text{ and } PL\neq JL. \tag{8}$$

A visual representation of the process of transfer learning, which can be found further down this page. This image is an explanation of how the DenseNet201 source model was trained by utilizing the ImageNet dataset that contained a total of one thousand labels. Through the utilization of transfer learning, it is feasible to incorporate previously obtained knowledge into a more refined target model. Following that, we put this innovative new target model for the categorization of brain cancers through its training using the SGD learning approach. The following settings are applied during the training process: a learning rate of 0.0001, a mini-batch size of 64, and a total of 100 epochs. Both the BRATS2018 dataset and the BRATS2019 dataset are utilized in this process so that the appropriate model can be trained. In the end, the characteristics of the global average pool layer are factored into the subsequent analysis. This was done to ensure accuracy. The dimension of the feature vector, designated by the letter N, is equal to the number of training samples that are indicated by the number 2048. This dimension is referred to as Nk.

5. PERFORMANCE ASSESSMENT

Large image datasets are frequently used whenever a system is being trained on new content. The vast majority of these pictures came from a variety of sources that are available to the general public, with field photography accounting for roughly forty percent of the total. About 12,000 pictures, nine distinct classifications of wheat illnesses, and one normal classification are all included in the newly curated dataset (LWDCD2020). Dimensional uniformity has been achieved in the photographs that have been preprocessed. The Table.1, which may be located at this location, provides a more in-depth breakdown of the dataset. This compilation features graphic depictions of wheat in both its healthy and damaged phases, with the intention of making it simpler to differentiate between the two types of wheat.

Table.1. Wheat Disease Dataset

Wheat Disease Class	#Images
Karnal bunt	1150
Black Chaff	1100
Crown and Root Rot	1040
Fusarium Head Blight	1270
Healthy Wheat	1280
Leaf Rust	1620
Powdery Mildew	1230
Tan Spot	1220
Wheat Loose Smut	1100
Wheat Streak Mosaic	1150

As a result of the fact that this image is one of the few in LWDCD2020 that does not depict any type of the disease that is mentioned in the description, it has attracted some attention and comments as a direct consequence of this fact. The photographs that were shot for the LWDCD2020 were positioned in intricate settings and were captured in a variety of lighting circumstances; each image showed a different stage in the course of the disease, and yet, despite their own distinctions, they all share similarities. The subsequent step, which consisted in enhancing some of the photographs contained in the dataset so that it could be put to more productive use, was successfully finished. The most crucial goal of every single test is to acquaint the network with the distinctions that exist between the various categories. Because of this, the network chances of getting proficient with the appropriate characteristics increase in direct proportion to the quantity of augmented photographs that are utilized in a certain scenario.

5.1 EXPERIMENTAL SETUP

Both Keras and Tensorflow, which are used as the backend of the Experimental Environment 4.1, have had their code changed so that the suggested method may be implemented using any one of them. Experiments were carried out using a machine with a Ryzen 7 quad-core processor, 16 gigabytes of random access memory (RAM), an NVIDIA Geforce GTX 1660 Ti Max-Q graphics card with 6 gigabytes of RAM, and the Windows 10 operating system. Following the completion of training that lasted for up to a thousand iterations, the networks were put through an optimization process using the Adam optimizer. The total amount of time spent on the training came to approximately six hours.

In the process of determining whether or not our model was effective, we included a number of metrics in addition to the F1 score. These metrics included accuracy, recall, and precision. These evaluation metrics were produced by first determining the TN, then the TP, and then the FN and FP. The TN was determined first, followed by the TP, and finally the FN and FP. The following is a list of features that can be found in cases that were correctly classified as either negative (TN) or positive (TP). Cases that were wrongly categorized as positive or negative are counted as false positives and false negatives, respectively. False positives are also known as erroneous positives. The following equations match to each success criterion and can be found following their respective definitions in the next section:

The correctness of the analysis is determined by making a comparison between the total number of samples that were used in the research and the number of samples that were correctly categorised.

$$\text{Accuracy}=\text{TP}+\text{TN}/\text{TP}+\text{TN}+\text{FP}+\text{FN} \quad (3)$$

Measurements of sensitivity and recall are applied in order to compute the percentage of positive patterns that were successfully recognized.

$$\text{Recall}=\text{TP}/\text{TP}+\text{FN} \quad (4)$$

Calculating the percentage of times that positive patterns are detected can be done with the use of an instrument level of precision if the device is accurate enough.

$$\text{Precision}=\text{TP}/\text{TP}+\text{FP} \quad (5)$$

The F1-score approach is applied to discover the harmonic mean of recall and precision in a specific scenario. This is accomplished using an F1 score.

$$\text{F1score}=2\times\text{Precision}\times\text{Recall}/\text{Precision}+\text{Recall} \quad (6)$$

Table.2. Results of Accuracy

Classes	Precision	Recall	F1-Score
Karnal Bunt	0.97	0.98	0.97
Black Chaff	0.98	0.98	0.98
Crown and Root Rot	0.98	0.98	0.98
Fusarium Head Blight	0.98	0.98	0.98
Healthy Wheat	0.98	0.98	0.98
Leaf Rust	0.99	0.98	0.98
Powdery Mildew	0.98	0.98	0.98
Tan Spot	0.97	0.98	0.97
Wheat Loose Smut	0.97	0.97	0.97
Wheat Streak Mosaic	0.96	0.97	0.96

Here are displayed the graphs of loss and accuracy for the proposed model, with examples for VGG16 and ResNet50 presented in the center and bottom regions, respectively.

6. CONCLUSION

In this paper, a novel deep convolutional architecture for diagnosing wheat diseases together with a system that is capable of doing this task is proposed. The proposed model power lies in its capability to learn from a substantial amount of training data in an efficient manner while requiring only a small number of resources. This is where the model strength is found. Using the LWDCD2020 dataset, the proposed method for sickness classification was examined for its effectiveness. The outcomes were far better than expected. The recommended method achieved a test accuracy of 97.88%, which is satisfactory, and the average training accuracy was determined to be 98.62. The categorization of 10 wheat disease classes achieved these results. When compared to the performance of other deep learning systems, this one exhibits a significant improvement. On the other hand, there is a 7.1% and 15.92% improvement in accuracy, respectively. Because of this, the approaches that have been proposed provide an efficient instrument for the classification of wheat diseases.

REFERENCES

[1] Z. Li and B. Wang, "Plant Disease Detection and Classification by Deep Learning-A Review", *IEEE Access*, Vol. 9, pp. 56683-56698, 2021.

[2] B. Subramanian, V. Saravanan and S. Hariprasath, "Diabetic Retinopathy-Feature Extraction and Classification using Adaptive Super Pixel Algorithm", *International Journal of Engineering and Advanced Technology*, Vol. 9, pp. 618-627, 2019.

[3] A. Abbas and S. Vankudothu, "Tomato Plant Disease Detection using Transfer Learning with C-GAN Synthetic Images", *Computers and Electronics in Agriculture*, Vol. 187, pp. 106279-106287, 2021.

[4] R.K. Nayak, R. Tripathy and D.K. Anguraj, "A Novel Strategy for Prediction of Cellular Cholesterol Signature Motif from G Protein-Coupled Receptors based on Rough Set and FCM Algorithm", *Proceedings of 4th International Conference on Computing Methodologies and Communication*, pp. 285-289, 2020.

[5] M. Zia Ur Rehman and I. Hussain, "Classification of Citrus Plant Diseases using Deep Transfer Learning", *Computers, Materials and Continua*, Vol. 70, No. 1, pp. 1-12, 2021.

[6] R.D. Aruna and B. Debtera, "An Enhancement on Convolutional Artificial Intelligent Based Diagnosis for Skin Disease Using Nanotechnology Sensors", *Computational Intelligence and Neuroscience*, Vol. 2022, pp. 1-8, 2022.

[7] J. Annrose and D.G. Immanuel, "A Cloud-Based Platform for Soybean Plant Disease Classification using Archimedes Optimization based Hybrid Deep Learning Model", *Wireless Personal Communications*, Vol. 122, No. 4, pp. 2995-3017, 2022.

[8] J. Schuler, H. Rashwan and D. Puig, "Color-Aware Two-Branch Dcnn for Efficient Plant Disease Classification", *Nature*, Vol. 28, No. 1, pp. 55-62, 2022.

[9] E. Akanksha and K. Gulati, "OPNN: Optimized Probabilistic Neural Network based Automatic Detection of Maize Plant Disease Detection", *Proceedings of International Conference on Inventive Computation Technologies*, pp. 1322-1328, 2021.

[10] Z. Chen, S. Chen, Z. Yuan and X. Zou, "Plant Disease Recognition Model based on Improved Yolov5", *Agronomy*, Vol. 12, No. 2, pp. 365-373, 2022.