PLANT LEAF DISEASES IDENTIFICATION IN DEEP LEARNING

Md.Milon Rana, Tajkuruna Akter Tithy,
Nefaur Rahman Mamun and Hridoy Kumar Sharker

Department of Electronics and Communication Engineering,
Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh

ABSTRACT

Crop diseases constitute a big threat to plant existence, but their rapid identification remains difficult in many parts of the planet because of the shortage of the required infrastructure. In computer vision, plant leaf detection made possible by deep learning has paved the way for smartphone-assisted disease diagnosis. employing a public dataset of 4,306 images of diseased and healthy plant leaves collected under controlled conditions, we train a deep convolutional neural network to spot one crop species and 4 diseases (or absence thereof). The trained model achieves an accuracy of 97.35% on a held-out test set, demonstrating the feasibility of this approach. Overall, the approach of coaching deep learning models on increasingly large and publicly available image datasets presents a transparent path toward smartphone-assisted crop disease diagnosis on a large global scale. After the disease is successfully predicted with a decent confidence level, the corresponding remedy for the disease present is displayed that may be taken as a cure.

KEYWORDS

Plant leaf diseases; agriculture; mobile app; convolutional neural networks (CNN); support vector machine (SVM), deep learning.

1. INTRODUCTION

Modern technologies have given human society the flexibility to supply plants that demand quite scores of people in cities and towns. However, plant security remains threatened by many factors, including global climate change, the decline in pollinators plant diseases [1] et al [2], [3]. Plant diseases aren't only a threat to food security on a world scale but may have disastrous consequences for smallholder farmers whose livelihoods rely upon healthy crops. within the developing world, over 80 percent of the agricultural production is generated by smallholder farmers [3], and reports of yield loss of quite 50% thanks to pests and diseases are standard. Furthermore, the foremost significant fraction of hungry people (50%) lives in smallholder farming households [4], making smallholder farmers a gaggle that's particularly prone to pathogen-derived disruptions within the food supply. Various efforts are developed to forestall crop loss thanks to diseases. Historical approaches to widespread pesticide application have increasingly been supplemented by integrated pest management approaches within the past decade. Independent of the approach, identifying a disease correctly when it first appears is crucial for efficient disease management. Historically, disease identification has been supported by agricultural extension organizations or other institutions, like local plant clinics. in additional recent times, such efforts are supported by providing information for disease diagnosis online, leveraging the increasing Internet penetration worldwide. Even more recently, tools supported mobile phones have increased, taking advantage of the historically unparalleled rapid uptake of itinerant technology altogether parts of the planet. Smartphones, particularly, offer very novel approaches to assist identify diseases due to their computing power, high-resolution displays, and extensive built-in sets of accessories, like advanced HD cameras. it's widely estimated that there'll be between 5 and 6 billion smartphones on the world by 2020. In [04], 69% of the world's population already had access to mobile broadband coverage, and mobile broadband penetration reached 47% in 2015, a 12-fold increase since 2007 [5]. The combined factors of widespread smartphone penetration, HD cameras, and high-performance processors in mobile devices result in a situation where disease diagnosis supported automated image recognition, if technically feasible, will be made available at an unprecedented scale. Here, we demonstrate the technical feasibility employing a deep learning approach utilizing 4,306 images of 1 crop species with four diseases (or healthy) made openly available through the project PlantVillage [6]. An example of every crop—disease pair is seen in Figure 1

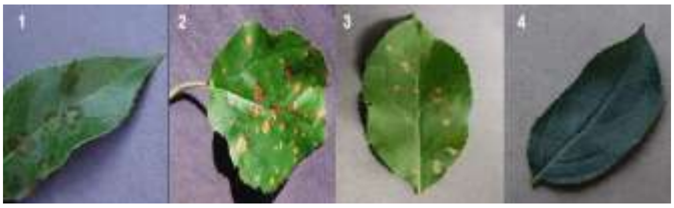

Fig. 1. Leaf images from the PlantVillage dataset, 1) Apple Scab, 2) Apple plant disease, 3) Apple Cedar Rust, 4) Apple healthy representing crop-disease pair used

In particular, computer vision and seeing have made tremendous advances within the past few years. While training large neural networks may be very time-consuming, the trained models can quickly classify images, making them also suitable for smartphone consumer applications. Convolutional neural networks have recently been successfully applied in many diverse domains as samples of end-to-end learning. Neural networks provide a mapping between an input—such as a picture of a diseased plant—to an output—such as a crop disease pair. The nodes in a very neural network are mathematical functions that take numerical inputs from the incoming edges and supply a numerical output as an outgoing edge [5]. Convolutional neural networks are simply mapping the input layer to the output layer over a series of stacked layers of nodes. The challenge is to form a deep network in such the way that both the structure of the network furthermore because the functions and edge weights correctly map the input to the output. Convolutional neural networks are trained by tuning the network parameters in such the way that the mapping improves during the training process [07]. This process is computationally challenging and has in recent times been improved dramatically by variety of both conceptual and engineering breakthroughs. so as to develop accurate image classifiers for the needs of disease diagnosis, we wanted an outsized, verified dataset of images of diseased and healthy plants. Until very recently, such a dataset failed to exist, and even smaller datasets weren't freely available. to deal with this problem, the PlantVillage project has begun collecting tens of thousands of images of healthy and diseased crop plants [08]. it's made them openly and freely available. Here, we report on classifying 4 diseases in one crop species using 4,306 images with a convolutional neural network approach. We measure the performance of our models supported their ability to predict the proper cropdiseases pair, given 38 possible classes. the simplest performing model achieves a mean F1 score of 0.9934 (overall accuracy of 99.35%), demonstrating our approach's technical feasibility. Our results are a primary step toward a smartphone-assisted disease diagnosis system.

The research paper has been organized as follows: Section 2 illustrates different object detection algorithm such as CNN, R-CNN. Section 3 describes the proposed method. The results and analysis have been demonstrated in the section 4. At last, the conclusion has been described in the section 5.

2. RELATED WORK ON PLANT DISEASE DETECTION

The disease recognition system helps within the identification of disease and provides remedies that may be used as a defence mechanism against the disease [11]. The dataset obtained from the PlantVillage is correctly segregated, and therefore the different plant species are identified and are renamed to create a correct database, then obtain a test database which consists of varied plant diseases that are used for checking the accuracy and confidence level of the project [12]. Then using training data, we are going to train our classifier, so the output are predicted with optimum accuracy. We use Convolution Neural Network (CNN), which comprises different layers, then we use SVM used for classification. A prototype android model is additionally designed that may be used for scanning. A high-resolution camera is attached and can capture images of the plant's leaves that may act as input for the software [13], supported which the software will tell us whether the plant are is healthy or not. With our code and training model, we've achieved an accuracy level of 78%. Our software gives us the name of the plant species with its confidence level and therefore the remedy that may be taken as a cure.

Support vector machine (SVM) classifier SVM was originally designed to make highly generalizable classifier for binary classification. SVM transforms the first feature space into a higher-dimensional space supported a user-defined kernel function so finds support vectors to maximise the separation (margin) between two classes [13]. SVM first approximates a hyperplane that separates both the classes. Accordingly, SVM selects samples from both the classes, referred as support vectors, that are closest to the hyperplane. the whole separation between the hyperplane and also the support vectors is referred as margin. SVM then iteratively optimizes the hyperplane and supports vectors to maximise the margin, thereby finding the foremost generalizable decision boundaries. When the dataset is separable by nonlinear boundary, certain kernels are implemented within the SVM to appropriately transform the feature space. For a dataset that's not easily separable, soft margin is employed to avoid overfitting by giving less weightage to classification errors round the decision boundaries. during this study, we use two SVM classifiers, one with linear kernel and therefore the other with a radial basis function kernel.

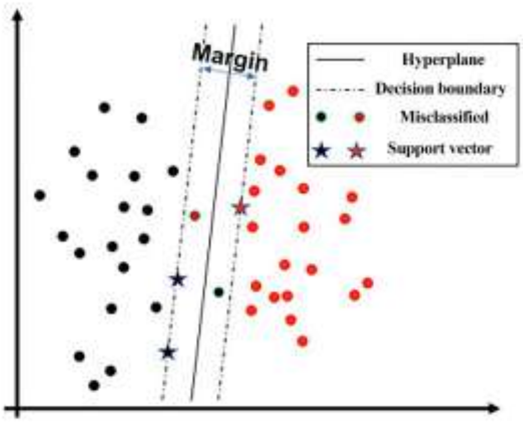

Fig. 2. Support vector machine: SVM

2.1. Artificial Neural Network

In a neural network because the structure says there's a minimum of one hidden layer between the input and output layers [12]. The hidden layers don't see the inputs. The word “deep” may be a relative term which implies what number hidden layers a neural network has. While computing the layer the input layer is ignored. for instance, within the picture below we've got a 3 layered neural network as mentioned input layer isn't counted. Layers in an ANN: 1 Dense or fully connected layers 2 Convolution layers 3 Pooling layers 4 Recurrent layers 5 Normalization layers 6 Many others Different layers perform different sort of transformations on the input. A convolution layer mainly wont to perform convolution operation while working with image data. A Recurrent layer is employed while working with statistic data. A dense layer may be a fully connected layer. in a very nutshell each layer has its own features and wont to perform specific task.

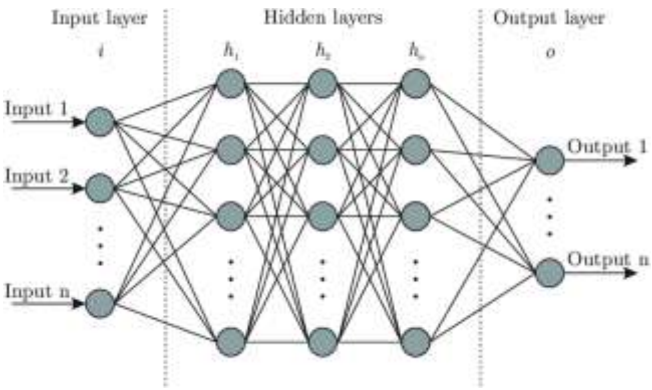

Fig. 3. Structure of artificial neural network Input layer

Each of the nodes within the input layer represents the individual feature from each sample within our data set which will pass to the model. Hidden layer: The connections between the input layer and hidden layer, each of those connections’ transfers output from the previous units as input to the receiving unit. Each connections have its own assigned weight. Each input are going to be multiplied by the weights and output are going to be an activation function of those weighted sum of inputs. To recap we've got weights assigned to every connection and that we compute the weighted sum that points to the identical neuron(node) within the next layer. That sum is passed as an activation function that transforms the output to variety which will be between 0 and 1. this may be passed on to the subsequent neuron(node) to the subsequent layer. This process occurs over and yet again until reaching the output layer. Let’s consider part1 connections between input layer and hidden layer, as from fig above. Here the activation function we are using is tanh function. $Z1 = W1\ X + b1$ $A1 = \tanh(Z1)$ Let’s consider part 2 connections between hidden layer and output layer, as from fig above. Here the activation function we are using is sigmoid function. $Z2 = W1\ A1 + b2$ $A2 = \sigma(Z2)$ During this process weights are going to be continuously changing so as to succeed in optimized weights for every connections because the model continues to be told from the info. Output layer: If it’s a binary classification problem to classify cats or dogs the output layer has 2 neurons. Hence the output layer are often consisting of every of the possible outcomes or categories of outcomes which much of neurons.

2.2. Convolutional Neural Networks (CNN)

CNN may be a style of deep learning model for processing data that features a grid pattern, like images, which is inspired by the organization of animal visual area and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN could be a mathematical construct that's typically composed of three sorts of layers (or building blocks): convolution, pooling, and fully connected layers [12]. the primary two, convolution and pooling layers, perform feature extraction, whereas the third, a totally connected layer, maps the extracted features into final output, like classification. In [14] A convolution layer plays a key role in CNN, which consists of a stack of mathematical operations, like convolution, a specialized form of linear operation. In digital images, pixel values are stored in a very two-dimensional (2D) grid, i.e., an array of numbers (Fig. 2), and alittle grid of parameters called kernel, an optimizable feature extractor, is applied at each image position, which makes CNNs highly efficient for image processing, since a feature may occur anywhere within the image. united layer feeds its output into the following layer, extracted features can hierarchically and progressively become more complex. the method of optimizing parameters like kernels is named training, which is performed so on minimize the difference between outputs and ground truth labels through an optimization algorithm called backpropagation and gradient descent, among others.

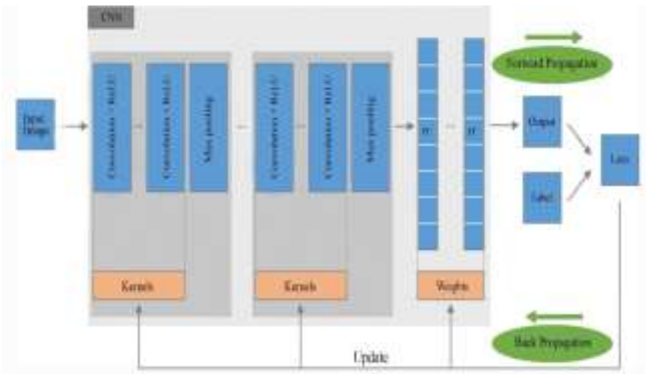

Fig. 4. Architecture of CNN model

An overview of a convolutional neural network (CNN) architecture and also the training process. A CNN consists of several building blocks: convolution layers, pooling layers (e.g., maxpooling), and fully connected (FC) layers. A model's performance under particular kernels and weights is calculated with a loss function through forwarding propagation on a training dataset, and learnable parameters, i.e., kernels and weights, are updated in line with the loss value through backpropagation with gradient descent optimization algorithm. ReLU, rectified long measure CNNs Layers Here's an summary of layers wont to build Convolutional Neural Network architectures.

2.3. Convolutional Layer

CNN works by comparing images piece by piece. Filters are spatially small along width and height but extend through the total depth of the input image [15]. it's designed in such a fashion that it detects a selected kind of feature within the input image. within the convolution layer, we move the filter/kernel to each possible position on the input matrix. Element-wise multiplication between the filter-sized patch of the input image and filter is finished, which is then summed.

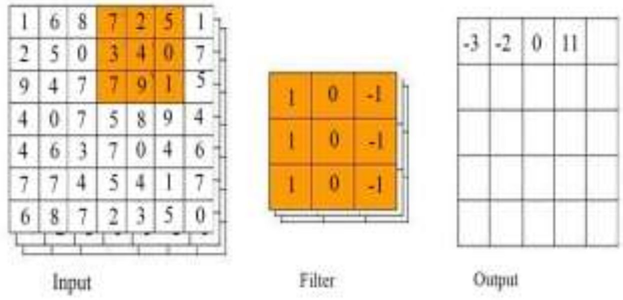

Fig. 5. Convolution operation

The translation of the filter to each possible position of the input matrix of the image gives a chance to find that feature is present anywhere within the image. The generated resulting matrix is named the feature map. Convolutional neural networks can learn from multiple features parallelly. within the ending, we stack all the output feature maps together with the depth and produce the output. as an example, a picture of a cat remains a picture of a cat whether or not it's translated one pixel to the right— CNNs take this property under consideration by sharing parameters across multiple image locations. Thus, we are able to find a cat with the identical feature matrix whether the cat appears at column i or column i+1 within the image.

2.4. Pooling Layer

Pooling layers are added in between two convolution layers with the only purpose of reducing the spatial size of the image representation. The pooling layer has two hyperparameters: • window size • stride

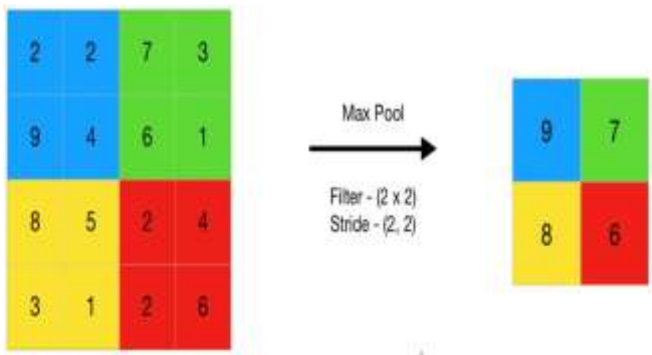

Fig. 6. Max Pooling

We take either the utmost value or the common of the values in each window depending upon the kind of pooling being performed. The Pooling Layer operates independently on every depth slice of the input, resizes it spatially and later stacks them together. sorts of Pooling: 1. Max Pooling selects the most element from each of the windows of the feature map. Thus, after the maxpooling layer, the output would be a feature map containing the foremost dominant features of the previous feature map. Average Pooling computes the typical of the weather present within the region of the feature map covered by the filter. It simply averages the features from the feature map.

2.5. Fully-Connected Layer

The Convolutional Layer, together with the Pooling Layer, forms a block within the Convolutional Neural Network [14]. the amount of such layers could also be increased for capturing finer details depending upon the complexity of the task at the value of more computational power

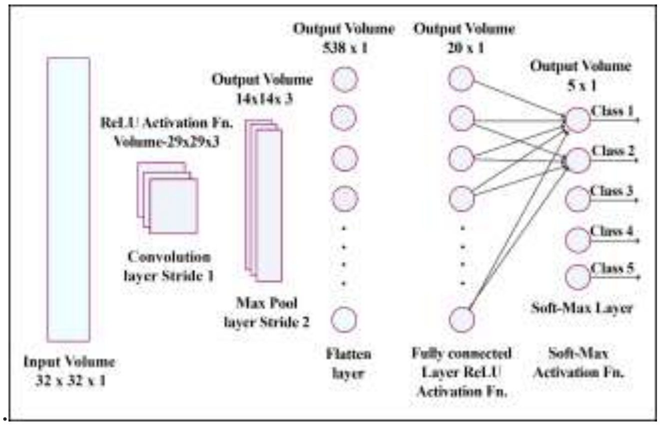

Fig. 7. Fully connected layer in CNN

3. PROPOSED METHOD OF SMARTPLANT DOCTOR

Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for disease image classification aren't so accurate. The suggested system altered the trainable classifier of the CNN by the SVM classifier. A convolutional network is helpful for extracting features information and SVM functions as a recognizer. it absolutely was found that this model both automatically extracts features from the raw images and performs classification. Additionally, we protected our model against over-fitting thanks to the powerful performance of dropout. during this work, the popularity on the disease images was evaluated; the training and test sets were taken from the PlantVillage datasets. Simulation results proved that the new design based-SVM of the CNN classifier architecture performs significantly more efficiently than standard SVM model or standard CNN classifier. The performance of our model is compared with image classification accuracies gained from state-of-the-art apple disease recognition, producing favorable results. Feature Extraction and Classification For the aim of evaluating the efficiency of the proposed CNN based-SVM trainable feature extractor model without and with dropout, we inquired its working performance to coach and recognize diseases of PlantVillage datasets. We observed that convolutional networks need a large number of samples to be told the parameters [07]. Hence, so as to best train the model on further data in order that we will better understand the variability of image shape, color and visibility, we displayed the dimensions of the training set ten times. One important thing before the feature extraction is preprocessing images [09]. We give the technical implementation details of the adopted system within the following sub-section. For the pre-processing, utilized during this experiment study, doesn't require to be normalized (noise reduction, segmentation). Some basic pre-processing tasks are a requirement so as to be performed during the database development. As for feature extraction, CNN is used during this experiment as a compact end-to-end model, consequently the input to the network is that the raw images. Finally, for the parameters setting: For the setting architecture, we've to define the amount of convolutional layers, size of the feature maps, weights, kernel, and biais in each layer of CNN. After that, defining the optimal kernel parameter and penalty parameter of SVM.

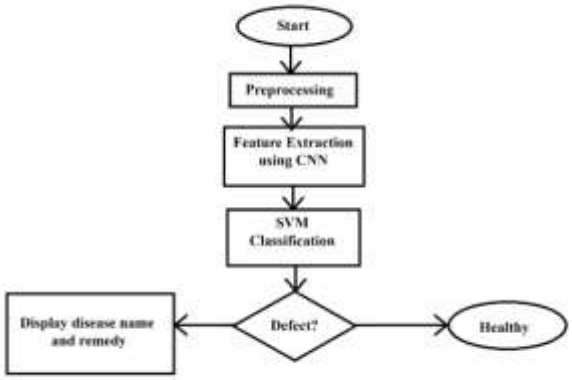

Fig. 8. Basic flow chart for the proposed system

we present the architecture of our Hybrid system supported CNN and SVM, wherein CNN is taken into account as a deep learning algorithm, on which the dropout technique has been applied during training. Our proposed system was tailored by altering the trainable classifier of the CNN with an SVM classifier. Deriving a flowchart of the proposed system Our target is to combine the CNN respective capacities and also the SVM to get a replacement effectual recognition system inspired by the 2 formalisms. We showed the specification of the CNN based-SVM model in Fig. 10. it had been noted that it sounds like as follow. Firstly, the primary layer welcomes raw image pixels as input. Secondly, the second and fourth layer of the network is convolution layers alternator with sub-sampling layers, which take the pooled maps as input. Consequently, they're ready to extract features that are more and more invariant to local transformations of the input image. FCL is that the sixth layer which consists of N neurons. the ultimate layer was substituted by SVM with an RBF kernel for classification. due to employing a huge number of knowledge and parameters, over-fitting can occur. So to stop our network from this problem and to boost it, dropout is applied. this system consists of temporarily removing a unit from the network. This removed unit is randomly selected only during the training. Dropout 16 is applied only at FCL layer and for more precisely, it's applied to feed-forward connections (perceptron). This choice is predicated on the actual fact that since the convolutional layers do not have lots of parameters, over-fitting isn't a controversy and so dropout wouldn't have much effect. The outputs from the hidden units are taken by the SVM as a feature vector for the training process. After that, the training stage continues till realizing good trained. Finally, classification on the test set was performed by the SVM classifier with such automatically extracted features. The structure of the CNN based-SVM model adopted in our experiments is presented here.

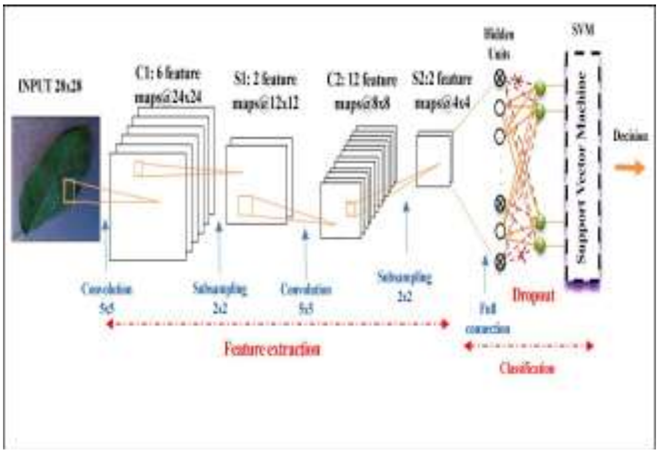

Fig. 9. CNN-SVM structure

CNN based-SVM spec with dropout is shown in Fig. 10, utilized in experiments applied to PlantVillage dataset with elastic distortion and is given within the following way: 28×28 represents a net with input images of size 28 × 28 pixels giving an input dimensionality of 784 with four Convolutional-Subsampling layers that's possible to be viewed as a trainable feature extractor. An SVM classifier substituted the ultimate output layer of CNN's fully connected hidden layers to classify disease images. Training parameters for CNN based-SVM model

3.1. Data Sheet Description

We analyze 4,306 images of plant leaves, which have a variety of 4 class labels assigned to them. Each class label may be a crop-disease pair, and that we make an effort to predict the crop-disease pair given just the image of the plant leaf. Figure 1 shows one example each from every crop-disease pair from the PlantVillage dataset. all told the approaches described during this paper, we resize the photographs to 32x32 pixels, and that we perform both the model optimization and predictions on these downscaled images. Data distribution among all class shown in fig: 11.

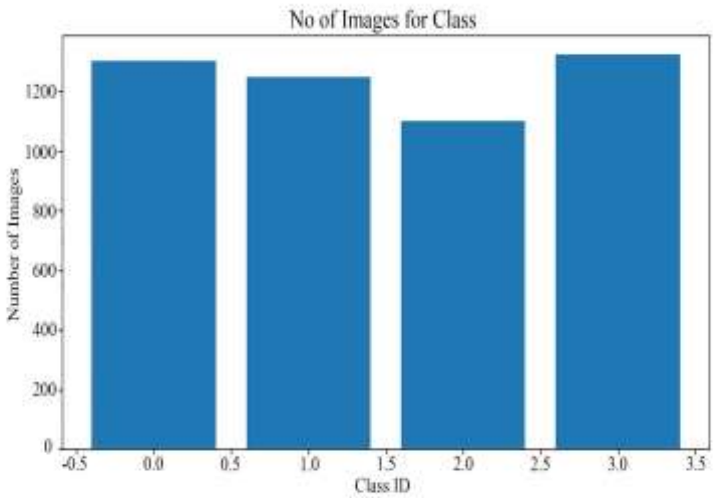

Fig. 10. Dataset distribution for each class

class Across all our experiments, we use three different versions of the full PlantVillage dataset. We start with the PlantVillage dataset because it is, in color; then we experiment with a gray-scaled version of the PlantVillage dataset, and at last we run all the experiments on a version of the PlantVillage dataset where the leaves were segmented, hence removing all the additional background information which could have the potential to introduce some inherent bias within the dataset because of the regularized process of knowledge collection just in case of PlantVillage dataset. Segmentation was automated by means of a script tuned to perform well on our particular dataset. We chose a way supported a collection of 19 masks generated by analysis of the colour, lightness and saturation components of various parts of the pictures in several color spaces. one in all the steps of that processing also allowed us to simply fix color casts, which happened to be very strong in a number of the subsets of the dataset, thus removing another potential bias. This set of experiments was designed to know if the neural network actually learns the “notion” of plant diseases, or if it's just learning the inherent biases within the dataset.

After loading the dataset, we split the info into i) training, ii) testing, iii) validation. Where the training and testing ratio are 80%:20% of total samples. and also the validation are going to be 20% of coaching data. we should always separate our data into train, validation, and test splits to stop our model from overfitting and to guage our model accurately. The training set the biggest corpus of your dataset that you just reserve for training your model.

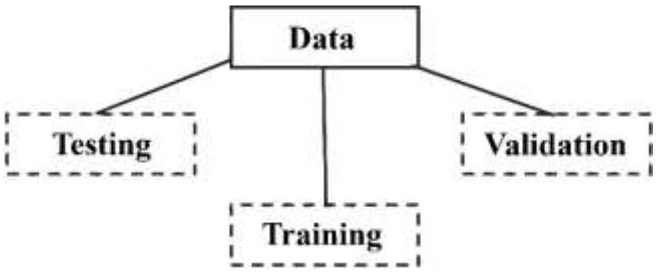

Fig. 11. Analysis the dataset

After training, inference on these images are loving a grain of salt, since the model has already had an opportunity to appear at and memorize the right output. The validation set may be a separate section of our dataset that you just will use during training to urge a way of how well our model is doing on images that aren't being employed in training. During the run evaluation metrics on the Test set at the very end of our project, to urge a way of how well our model will neutralize production. Pre-processing may be a vital step in CNN because the images within the dataset may have some inconsistency which can affect the accuracy of the system. the photographs within the dataset have noise and non-uniform lighting which must be rectified during this step. We did so by applying segmentation on the photographs to urge obviate uneven backgrounds. Through segmentation we extracted the relevant a part of the photographs which during this case were the image of leaves. Hence, after segmentation we had the photographs of leaves with black background. Later to rectify the non-uniform lighting we converted the pictures to grayscale images and sent it for further processing. Data augmentation could be a technique to artificially create new training data from existing training data. This was done by applying domain-specific techniques to examples from the training data that created new and different training examples. during a word we augmented the information to be more general.

4. RESULT AND DISCUSSION

The new architecture of CNN classifier, introduced during this work, enabled to couple a CNN method with SVM classifier. so as to boost this architecture. Our proposed system was compared to variety of other methods being recently proposed. the most effective performing convolutional neural network achieve a slip classification rate of test accuracy 97.43%. a unique architecture of CNN supported SVM similarly further reduces the error apple plant dataset with 4 classes. Adding dropout to the CNN based-SVM model, only to the fully connected hidden layers. Our network was trained using Adam optimizer for 100 epochs.Using the Convolutional neural network-Support vector machine architecture, we trained a model on images of plant leaves with the goal of classifying both crop species and therefore the presence and identity of disease on images that the model had not seen before. The accuracy curve and loss curve of proposed method is shown in figure 12. and 13

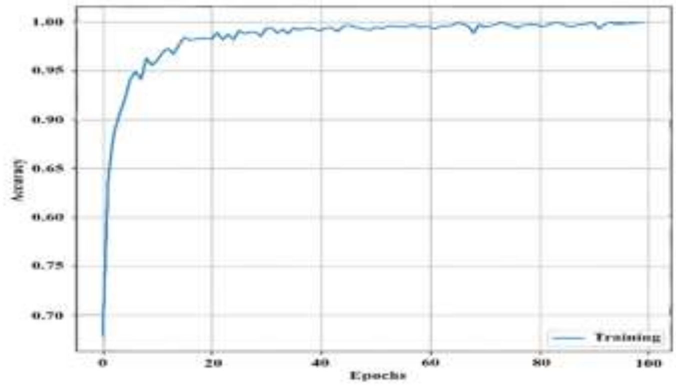

Fig. 12. Accuracy curve of proposed method

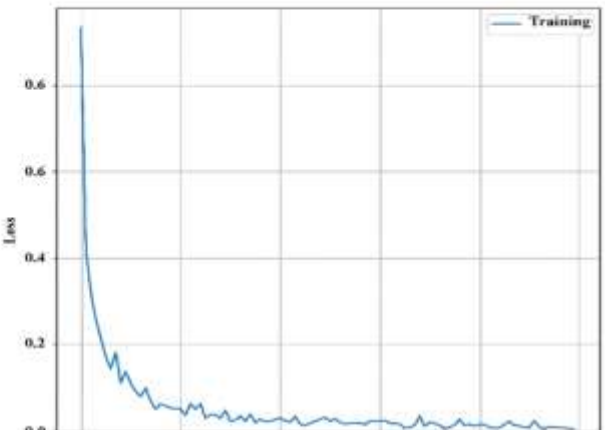

Fig. 13. Loss curve of proposed method

Within the PlantVillage data set of 4,306 images containing 4 classes of apple diseases (including healthy apple), this goal has been achieved as demonstrated by the highest accuracy of 97.43%. Importantly, while the training of the model takes lots of your time, the classification itself is incredibly fast (less than 3 seconds on a CPU), and simply be implemented on a smartphone. This presents a transparent path towards smartphone-assisted crop disease diagnosis on an enormous global scale.

5. Conclusion

Conventional disease recognition methods lack the employment of modal information aside from the image modality. within the present study, the disease text description information represented by continuous vectors was decomposed and recombined into graph structure data. For image data, the feature decomposition was implemented by randomly disarranging and recombining the image blocks after segmentation, which improved the robustness of the model to a particular extent. Specifically, the accuracy, precision, sensitivity and specificity of the fusion model were 97.62, 92.81, 98.54, and 93.57%, respectively. This research provides new ideas for disease recognition, and puts forward new insights and methodology in improving the robustness of disease recognition models.

References

[1] Madhulatha, G. & Ramadevi, O. (2020). Recognition of Plant Diseases using Convolutional Neural Network. 738-743. 10.1109/I-SMAC49090.2020.9243422.

[2] Gavhale, Ms& Gawande, Ujwalla. (2018). An Overview of the Research on Plant Leaves Disease detection using Image Processing Techniques. IOSR Journal of Computer Engineering. 16. 10-16. 10.9790/0661-16151016.

[3] Chen, JiaYou& Guo, Hong & Hu, Wei & He, JuanJuan& Wang, Yonghao& Wen, Yuan. (2020). Research on Plant Disease Recognition Based on Deep Complementary Feature Classification Network. 1685-1692. 10.1109/SMC42975.2020.9283299.

[4] Nigam, Sapna & Jain, Rajni. (2020). Plant disease identification using Deep Learning: A review. Indian Journal of Agricultural Sciences. 90. 249-57.

[5] Barbedo, Jayme. (2019). Plant disease identification from individual lesions and spots using deep learning. Biosystems Engineering. 180. 96-107. 10.1016/j.biosystemseng.2019.02.002.

[6] Kurumaddali, Krishna &Madhira, Aditya &Chinthamaneni, Vittal & Jilla, Kausthubha&Siddhantham, Vardhan. (2021). Detection of Plant Diseases Using Convolutional Neural Networks in International Journal for Research in Applied Science and Engineering Technology. 9. 1653-1657. 10.22214/ijraset.2021.37641. 29

[7] Zhang, S.W. & Shang, Y.J. & Wang, L. (2015). Plant disease recognition based on plant leaf image. Journal of Animal and Plant Sciences. 25. 42-45.

[8] Kaur, Jasmeet & Chadha, Raman & Thakur, Shvani& Kaur, Er.Ramanpreet. (2016). A REVIEW PAPER ON PLANT DISEASE DETECTION USING IMAGE PROCESSING AND NEURAL NETWORK APPROACH. 10.5281/zenodo.50392.

[9] Adelson, Edward H., Charles H. Anderson, James R. Bergen, Peter J. Burt, and Joan M. Ogden. "Pyramid methods in image processing." RCA engineer 29, no. 6 (1984): 33-41.

[10] M. Riedmiller and H. Braun (2016), “A direct adaptive method of faster backpropagation learning: The rprop algorithm”, in IEEE International Conference on Neural Networks, San Francisco, 1993, pp. 586–591.

[11] S. L. Phung, A. Bouzerdoum, and D. Chai, “Skin segmentation using color pixel classification: analysis and comparison,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 148–154, 2005.

[12] Yi Yang and Shawn Newsam, "Bag-Of-Visual-Words and Spatial Exten- sions for Land-Use Classification",ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010.

[13] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba (2008). “SUN Database: Large-scale Scene Recognition from Abbey to Zoo with machine learning”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[14] Tripathi, Anshul & Chourasia, Uday & Dixit, Priyanka & Chang, Victor. (2021). A Survey: Plant Disease Detection Using Deep Learning. International Journal of Distributed Systems and Technologies. 12. 1-26. 10.4018/IJDST.2021070101.

[15] Source for highway images [Online] National Highway Authority of India, nhai.org. link: https://www.computervision.

AUTHORS

Md. Milon Rana received the B. Sc. (Engineering) from the department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh in 2019. Currently, He is a student of M. Sc. (Engineering) in the same department. His research interest is performance analysis of communication protocols of IoT, Machine Learning, Deep Learning, Computer vision, Object detection and classifications, Object Segmentation, OFDM, PAPR.

Tajkuruna Akter Tithy received the B. Sc. (Engineering) from the department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh in 2019. Currently, she is a student of M. Sc. (Engineering) in the same department. Her research interest is performance analysis GSM, OFDM, PAPR, Antenna design.

Nefaur Rahman Mamun received the B. Sc. (Engineering) from the department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh in 2020. Currently, he is a student of M. Sc. (Engineering) in the same department. His research interest is performance analysis GSM, OFDM, PAPR.

Hridoy Kumar Sharker received the B. Sc. (Engineering) from the department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh in 2019. Currently, he is a student of M. Sc. (Engineering) in the same department. His research interest is in machine learning, Under water communication and MANET.