

Research Article

A Novel Computer Vision Model for Medicinal Plant Identification Using Log-Gabor Filters and Deep Learning Algorithms

Stephen Opoku Oppong ¹, Frimpong Twum ², James Ben Hayfron-Acquah ²,
and Yaw Marfo Missah ²

¹Department of ICT Education, University of Education, Winneba, Ghana

²Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Correspondence should be addressed to Stephen Opoku Oppong; sooppong@uew.edu.gh

Received 24 April 2022; Revised 16 August 2022; Accepted 5 September 2022; Published 27 September 2022

Academic Editor: Kuruva Lakshmanna

Copyright © 2022 Stephen Opoku Oppong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Computer vision is the science that enables computers and machines to see and perceive image content on a semantic level. It combines concepts, techniques, and ideas from various fields such as digital image processing, pattern matching, artificial intelligence, and computer graphics. A computer vision system is designed to model the human visual system on a functional basis as closely as possible. Deep learning and Convolutional Neural Networks (CNNs) in particular which are biologically inspired have significantly contributed to computer vision studies. This research develops a computer vision system that uses CNNs and handcrafted filters from Log-Gabor filters to identify medicinal plants based on their leaf textural features in an ensemble manner. The system was tested on a dataset developed from the Centre of Plant Medicine Research, Ghana (MyDataset) consisting of forty-nine (49) plant species. Using the concept of transfer learning, ten pretrained networks including Alexnet, GoogLeNet, DenseNet201, Inceptionv3, Mobilenetv2, Resnet18, Resnet50, Resnet101, vgg16, and vgg19 were used as feature extractors. The DenseNet201 architecture resulted with the best outcome of 87% accuracy and GoogLeNet with 79% performing the worst averaged across six supervised learning algorithms. The proposed model (OTAMNet), created by fusing a Log-Gabor layer into the transition layers of the DenseNet201 architecture achieved 98% accuracy when tested on MyDataset. OTAMNet was tested on other benchmark datasets; Flavia, Swedish Leaf, MD2020, and the Folio dataset. The Flavia dataset achieved 99%, Swedish Leaf 100%, MD2020 99%, and the Folio dataset 97%. A false-positive rate of less than 0.1% was achieved in all cases.

1. Introduction

Computer vision is a broad term that describes the computer performing the function of an eye by using different mathematical algorithms on a digital image. This area is concerned with the automated processing of images from the real world, extracting features, and interpreting information in real-time based on the user's requirements [1, 2]. The fundamental task in computer vision is image recognition [3]. Human vision is unique and superior in that it detects and discriminates against the objects around it with ease. It can perceive 3-D structures with perfection and also categorize them efficiently [4]. Computer vision is modelled after the human visual system which nonetheless is superior

in detection, identification, and discrimination objects. With human vision being such a complex mechanism, computer vision can be thought of as an approximation of it [5].

Plant taxonomy is the science that aims in detecting, recognizing, describing, characterizing, and naming plants. Chemotaxonomic, anatomical, and morphological classifications are some of the techniques adopted for this science [6]. In comparison to chemotaxonomy, morphological and anatomical classifications are viewed as more traditional [7]. The key aspects that play vital roles in plant taxonomy are; plant identification which deals with the determination of an unknown plant in relation to a previously collected specimen and plant classification which places a known plant in a category based on its shared characteristics with other plants.

Despite the fact that the tropical vegetation of Ghana is abundant in medicinal plants, a thorough understanding of their spread and usage appears to be limited to the aged and herbalists [8, 9]. The majority of them learned about the plants through oral transmission or by employing them in traditional medicinal preparations. In general, information obtained on the subject of diversity and value of medicinal plants in Ghana, and their use, is made available through ethnobotanical means by a few individual researchers [10]. Herbal medicine is gradually becoming more widely accepted and used globally. This realisation is similar in Africa, where over 60% of its populace relies only on these plants for their primary healthcare needs, especially in underdeveloped nations [11, 12]. As a result, plants constitute a significant contributor to natural products and a vital component of health care. Traditional medicines are extremely important to the pharmaceutical business; in fact, traditional medicines account for a quarter of all prescribed pharmaceuticals worldwide. In comparison to synthetic medications, medicinal plants are chosen since they have fewer side effects and are more affordable [13]. Humans benefit from classifying medicinal plants in a variety of ways, thus it is critical to address this issue [14].

Attributes such as leaves, fruits, flowers, or the entire plant are mostly used to identify a plant. The use of the leaves is one most promising and reliable methods of identifying medicinal plants out of all the keys of identification. [15]. The use of plants as medicine has necessitated plant identification in order to identify whether or not a given plant has medicinal characteristics. When the untrained eye examines two plants closely, it is simple to confuse them. This makes plant identification a very important part of natural products and medicine that should not be overlooked since misidentification might have catastrophic effects. [10]. Plants go through several growth phases in different seasons and may have different shape characteristics as a result of environmental factors including climate change, topographical position, and so on. [16]. Furthermore, knowledge of plant species is vital for biodiversity protection. Using conventional keys to identify plants is complex, time-consuming and it is laborious for nonbotanists and provides a formidable obstacle for freshmen interested in obtaining specific expertise because it necessitates the use of scientific nomenclature. [17]. It is indeed difficult to discern between plants using their many morphological traits. High intraclass variability and small interclass variances are the key obstacles. [18]. Plant categories are tightly related, and some of their structural parts are closely related, resulting in low interclass differences. Furthermore, plants vary greatly in terms of size, colour, shape, and texture, and their appearance changes throughout the year, resulting in significant intraclass variance [19]. This study seeks to identify a plant using a proposed deep learning algorithm to perform the matching process that maps a leaf image to a plant category.

The rest of the study is outlined as follows: Section 2 is made of the literature review of the main concepts related to the study, Section 3 presents the methodology of the study which includes the feature extraction methods and the

proposed model as well as the performance metrics, Section 4 analyses and discusses the results and the last section presents the conclusion and future works.

2. Literature Review

2.1. Deep Learning. Deep learning, a branch of machine learning, uses varied layers of algorithms (ANNs, or artificial neural networks) to model high-level abstractions with complex structures in data [2], and it is also based on data learning representations [20]. For each layer, a different interpretation of data that has been fed into them is provided [21]. Deep learning, inspired by the human brain information processing patterns, utilizes huge data in order for a given input to be mapped to specific labels. Convolutional neural networks, deep belief networks, deep neural networks, and recurrent neural networks are all deep learning architecture. Deep learning has been applied in various areas including natural language processing, audio and video recognition, computer vision, and automatic speech recognition and have produced tremendous results [22, 23].

Traditional machine learning algorithms consist of various stages which are preprocessing, feature extraction, feature selection, and classification. Feature selection stage plays a great role in these types of algorithms and might lead to incorrect classifications of classes if it is biased. Deep learning, however, overcomes this problem by automating the learning of features that are rich and complex [24]. Deep learning combines both feature extraction and classification at a go. With the emergence of big data, the concept of deep learning has expanded greatly [25].

Deep learning algorithms provide some level of abstraction of data and this is achieved with the number of its layers and their sizes [26]. In the general interpretation of deep neural networks, universal approximation theorem and probabilistic inference are used [27]. Deep learning has the following advantages; universal learning approach [28], robustness [29], generalization [30], and scalability [31]. Some drawback concept of deep learning include; a large data is required; data modules are complex; it is very expensive to train and it requires a classifier for comprehending mere learning results [32].

In machine learning, overfitting is one common problem, especially in deep learning, when training a model due to the large volume of parameters needed to train this kind of complex model. This occurs when the trained model does not generalize or predict well to unknown test data [33]. One way of dealing with overfitting is regularization which allows the model to deduce better to unknown data when training on a finite training set, or with an imperfect optimization procedure [34]. DropConnect, dropout, data augmentation, stochastic pooling, batch normalization, weight decay, early stopping, and ℓ_1 and ℓ_2 regularization are some of the common regularization strategies used to prevent overfitting. [35].

Optimizers are used to minimize a loss function or to increase the production efficiency. They are dependent on the model's learnable parameters i.e., biases and weights [36]. Popular optimizers include Adam (Adaptive Moment

Estimation), RMSProp (Root Mean Square Propagation), Stochastic Gradient Descent (SGD), AdaGrad (Adaptive Gradient Descent), Momentum, and Adadelta [37]. The choice of the best optimizer relies, among others, on training data and a trade-off between speed and performance, the application, and network architecture [38]. Adaptive gradient algorithms in particular RMSprop and Adam, in modern machine learning models training, have demonstrated greater performance, e.g., deep neural networks [39].

2.2. Convolution Neural Networks. Convolutional Neural Network (CNN) is, now, the go-to method for pattern classification and image processing [40] and it has been proven to perform better than the traditional methods [41]. Although numerous unsupervised and recurrent variants have been developed, a CNN is a supervised feedforward artificial neural network. Literature has also demonstrated the importance of CNNs and its use in computer vision systems [2]. The visual cortex of the eye inspired the arrangement of neurons in CNNs [42]. Convolutional layers, pooling Layers, and nonlinear and fully connected (FC) layers are the components of CNN architecture [43].

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has seen an increase in several CNNs and deep learning techniques commonly called pretrained networks. The AlexNet model developed in 2012 brought about other advanced architectures such as VGG, Xception, ResNet, and DenseNet which have been among performing techniques in recent times [44]. Pretrained networks used in this study are presented in the next section.

2.2.1. Alexnet. AlexNet, considered largely as the first deep CNN architecture for classification and recognition tasks, has five convolution layers and three fully connected layers. A Local Response Normalization (LRN) was first introduced in AlexNet architecture with the ReLU activation function. AlexNet has a deeper architecture than its predecessor LeNet, which consists of five convolution layers, one pooling layer (max), a ReLU activation function, and three fully connected layers. It also used the dropout technique as its regularization method. [45].

2.2.2. Googlenet. Three convolution layers, nine inception modules, with two levels each, and one fully-connected layer make up GoogleNet's 22-layer network. An inception block is used in the first layer of the GoogleNet architecture, which uses filters of 1×1 , 3×3 , and the 5×5 sizes. To improve performance, errors are calculated at numerous intermediate stages. A 1×1 concatenation filter is placed between them to alter input computations before moving on to the following layers' convolution kernels for processing. The number of features in the last layer is 1,024. To decrease the parameters numbers, the fully connected layers were replaced with a pooling layer. This reduced the parameters from 138 million to 4 million [46].

2.2.3. Inceptionv3. An improved version of GoogleNet was introduced in 2015 which is the Inceptionv3. To improve performance, $n \times 1$ and $1 \times n$ convolutional layer factors are used instead of $n \times n$ factors. Filters with sizes of 5×5 were replaced with two of 3×3 filters in the architecture, which lead to a significant number of neurons reduction and parameters to 24 million. In this architecture, the convolutions are factorised into smaller convolutions [47].

2.2.4. Vggnet. In 2014, Oxford University researchers introduced Visual Geometry Group (VGG) which differs slightly from AlexNet in terms of the kernel size and the number of feature maps. VGG architecture has thirteen convolutional layers with a max-pooling layer and three fully-connected layers following it. Large filters were replaced with smaller ones because developers believed running filters with lesser sizes concurrently could perform the same task. The VGG-16 and VGG-19 increased the layers numbers in the network as indicated by the numbers. The VGG-16 and VGG-19 architectures are made up of successive 3×3 convolution layers after which a pooling layer follows. The increase in the depth of the layer increased trainable parameters [48].

2.2.5. Resnet. The theory behind designing the Residual Neural Network (ResNet) transformed the generation of CNN in 2015. It introduced the concept that higher layers learn new features from the previous layers. Connections added to layers are copied to the next layer's input without considering the extraction of features and identity from the previous layer. Even though it is having 152 layers which is 20 times more than AlexNet and 10 times more than VGG, it has a lower computational complexity than the other networks. The ResNet indicated a 3.57% error after training and implementation on the ImageNet dataset, which, as compared to the human error, is less [49].

2.2.6. Mobilenet. One drawback of deep learning is that it is very expensive to train therefore Google researchers in 2017 introduced MobileNet to solve this resource constraint problem. In MobileNet which is a small low-consumption model, a normal convolutional layer was used instead of a deep convolutional layer. Deep convolutional layers process individually on each colour channel making it computationally intensive. MobileNet consists of 28 layers which in the absence of computing power become more appropriate for mobile-based vision programs. To improve training performance, depthwise convolution layers are being replaced with convolution layers [50].

2.2.7. Densenet. DenseNet is a powerful neural network for image recognition that was first introduced in 2017. In DenseNet, features are rather transferred to all subsequent layers from all previous layers, as opposed to ResNet, which keeps each layer's information within the layers without offering to the next layer. This leaves subsequent layers with insufficient data and information to train. However, in a

DenseNet, each subsequent layer in a dense block receives maps of the previous feature after which it is obtusely connected to each preceding layer, potentially reducing gradient calculations, decreasing the parameters numbers, and allowing features to be reused [51]. The core of the ResNet model is to train deeper CNNs by establishing shortcuts (skip connections) between the front and back layers, which helps to backpropagate the gradient during training. The DenseNet model is developed based on the same basic idea as ResNet, but it establishes dense connections between all of the previous and subsequent layers, which is reflected in its name. These features allow DenseNet to achieve better performance than ResNet with fewer parameters and less computational cost [52].

2.2.8. LOG Gabor Filters. David Field proposed the Log-Gabor function [53]. Log-Gabor filters contain transfer functions that are consistent with the human visual system, which exhibits symmetric cell responses on the log frequency scale. Furthermore, observations on mammals' visual systems show that we have symmetric responses of cells on the log frequency scale, similar to the Log-Gabor function. Because the Log-Gabor transform has an extended tail with no DC component, it allows for the construction of infinitely broad bandwidth filters, which are able to encode natural images more effectively by expressing the higher frequency components.

When calculated on a logarithmic frequency scale, log-Gabor functions have Gaussian transfer functions. The 2D Log-Gabor filter is generated in the frequency domain because the log function at the origin has a singularity. The radial and angular filters make up the Log-Gabor function in polar coordinates. The response frequency of the radial filter is illustrated by the following equation [54]:

$$G_r(r) = \exp\left(-\frac{[\log(r/f_0)]^2}{2\sigma_r^2}\right). \quad (1)$$

And, the frequency response of the angular filter described by the following equation:

$$G_\theta(\theta) = \exp\left(-\frac{[\theta - \theta_0]^2}{2\sigma_\theta^2}\right). \quad (2)$$

2.2.9. Related Works. Many researchers have used hand-crafted features with supervised classifiers and deep learning models, particularly, CNN for plant identification.

Kan et al. [55] introduced an automatic system that uses the leaf for the classification of medicinal plants. The dataset contained 240 leaves of different plants from the medicinal plant specimen library of Anhui University of Traditional Chinese medicine. Five texture and ten shape features were extracted and using the SVM classifier and a 93.3 percent recognition rate was achieved. Begue et al. [56] extracted several leaf features such as a number of vertices, length, width, perimeter, and area of hull and colour on a dataset of 24 different plant species having 30 images each from the

tropical island of Mauritius. The highest accuracy achieved was 90.1% using the random forest classifier.

De Luna et al. [57] experimented with seven algorithms (logistic regression, naïve bayes, K-Nearest Neighbor (KNN), linear discriminant analysis, classification and regression trees, SVM, and Neural Networks (NN) in identifying Philippine herbal medicine plants using leaf features. Various leaf shape and venation structure features were extracted and resulted in a 98.6% recognition rate. Vijayshree and Gopal [58] introduced a system using neural networks to classify and identify herbal medicinal plants on a dataset containing 50 different species having 500 leaves. A total of 21 features were extracted using texture, colour, and shape. Experimental results gave 93.3% accuracy using only texture features and 99.2% using all three features. Dahi-gaonkar and Kalyane [59] identified ayurvedic medicinal plants using leaf based on its colour texture and shape feature using SVM on 32 different plants. Features extracted include entropy, solidity, eccentricity, contrast, extent, standard deviation, mean, and equivalent diameter. An accuracy of 96.66% was achieved.

Britto and Pacifico [60] compared the performance of the Extreme Learning Machine (ELM) algorithm with K-Nearest Neighbor, Decision Tree classifier, Support Vector Machine, Naive Bayes classifier, and a Multilayer Perceptron trained with Backpropagation algorithm in the context of plant classification. The datasets, Fisher's Iris Plant, Wheat Seed Kernels, and 100 Plant Leaves were used in this investigation. A Centroid Contour Curve form signature, a fine-scale margin feature histogram, and an interior texture feature histogram were among the characteristics extracted. ELM achieved the best performances with the Iris data set (97%) and Seed data set (96%). The texture was presented as the best individual discriminatory power.

Dissanayake and Kumara [61] performed a comparison of the performance of multiple machine learning algorithms to identify herbal, fruit, and vegetable plants using their leaves. A total of 3,150 leaf photos from 25 different herbal, fruit, and vegetable species were used. Color photos were transformed to grayscale images, and the image noise was reduced using a Gaussian filter. Shape, texture, and colour are the three feature categories from which 17 features were collected. Support Vector Machine, K-Nearest Neighbors, Multilayer Perceptron, Random Forest, and Decision Tree algorithms have classification accuracy of 85.82 percent, 75.45 percent, 82.88 percent, 80.85 percent, and 64.39 percent, respectively.

Naeem et al. [62] developed a machine learning (ML) based medical plant leaf classification utilizing multispectral and texture datasets. A total of six varieties of medicinal plant leaves are used. Out of 65, 14 features were selected using a chi-square feature selection strategy. Five machine learning classifiers were used (multilayer perceptron, random forest, logit-boost, basic logistic, and bagging), with the multilayer perceptron classifier showing the most promise at 99.01 percent accuracy.

Xue et al. [63] showed that an ANN model developed using the morpho-colourimetric parameters as inputs performed better (accuracy of 98.3%) than a visible (VIS)/Near

Infrared (NIR) spectral analysis (92.5% accuracy) when tested on 20 different Chinese medicinal plants which were sampled for their leaves. Kaur and Kaur [64] using the Swedish Leaf dataset applied the Gaussian filtering mechanism as a preprocessing technique after which texture and colour features were extracted. Classification using a Multiclass-support vector machine achieved an accuracy of nearly 93.26%. Singh [65] proposed the Local Binary Patterns—Support Vector Machine (LBP-SVM) methodology on the Swedish Leaf and compared it with the K-NN classifier and Binarized Neural Network (BNN). The LBP-SVM model provided a higher accuracy outcome of 84% while the existing BNN, and KNN models produced only 77% and 75%, respectively.

Nguyen et al. [66] used a pretrained network GoogLeNet to extract a 1024-dimensional feature vector of the last average pooling layer before the dropout layer. Ten classifiers are applied (Nearest neighbour (NNB), Linear Support Vector Machine (L_SVM), Nonlinear Support Vector Machine, Ada Boost (AB), Decision Tree (DT), Naïve Bayes (NB), Neural network (NN), Random Forest (RF), Quadratic discriminant analysis (QDA), and Softmax (SM), the default classification method of the GoogLeNet. The Linear SVM classifier achieved the best result of 87.34%. Jaiganesh et al. [67] proposed a Convolution Neural Network, which consists of four layers, convolution layer, dropout, max pooling, and average pooling. The proposed model was implemented using the Kaggle tool. The accuracy of the model was 86% which was accomplished with much less computational effort and shows the efficiency of the algorithm. Huynh et al. [68] suggested a five-layered Convolutional Neural Network (CNN) architecture in which the red channel of colors replaces the leaf vein shape data. To enhance the amount of training images, data augmentation was used to make three duplicates of each image after reflection and rotation. On the Flavia leaf data set and the Swedish leaf data, experimental findings showed that the suggested CNN model was successful for leaf recognition, with the greatest accuracy of greater than 98.22%.

Banzi and Abayo [69] proposed a CNN-LSTM network with 20 layers: 12 convolutional layers, one Fully Connected layer, five pooling layers, one Long Short-Term Memory (LSTM) layer, and one output layer with the softmax function for classification. Training of the models was performed by using an open database of 100 plant species images, containing 64 different element vectors of plants in a set of 100 distinct classes of plant species. Experiments showed that the proposed CNN-LSTM performs better in classifying plant species than the convolutional CNN as it attains an accuracy of 95.06%. Karahan and Nabiyevev [70] used a pretrained network MobileNetV2 to develop a plant identification system. A dataset containing 5,345 flowers and plant images belonging to 76 species was used. Preprocessing techniques used were centre cropping and normalizing. To expand the collection of images in the database and boost the model's generalization power, data augmentation techniques were also used. The suggested model attained a training set accuracy of 0.9971 and a test set accuracy of 0.9897 after 15 epochs. Pravin and Deepa [71] reviewed three

different convolutional neural network algorithms; Matrix-Based Convolutional Neural Network (M-bCNN), Dual-path CNN (DP-CNN), and Fine-tuned AlexNet model for the medicinal leaf identification. DP-CNN produced the highest accuracy of 95.67%, Fine-tuned AlexNet model had 93.31%, and Dual-path CNN, 91%.

Chung et al. [72] created a dual-path CNN model in which the two subnetworks are independent and receive individual input from either an original image or a centrally cropped image. The suggested model attained a 77.1 percent accuracy rate after training and validation on a plant dataset of 14 species of Taiwan's most prevalent trees. Adetiba et al. [73] used the Leafsnap image dataset of 185 plant species and five pretrained CNN models (AlexNet, GoogLeNet, VGG-19, ResNet50, and MobileNetV2) to produce an accurate plant species recognition. MobileNetV2 with ADAM optimizer has the greatest testing accuracy of 92.3% among the pretrained models.

Bao et al. [74] devised a system for recognizing plants based on their leaf patterns that use two methods: a Histogram of Oriented Gradient (HoG) and a deep convolutional neural network. HoG was used to classify the features, while CNN was utilized to identify them. Ghazi et al. [75] implemented three models of transfer learning to explain better the identity of the various plants. These three-model used were GoogleNet, VGGNet, and AlexNet applied on the LIFECLEF 2015 dataset. The overall accuracy was 80% on the validation set and an overall inverse rank score of 0.752 on the official test set was achieved with the best-combined model. Krause et al. [76] presented a What's That Plant (WTPlant) system for identifying plants in natural images using deep learning approaches. Preprocessing was done using the Watershed Transform and the GrabCut and the classification engines used are two AlexNets pretrained models. In preliminary tests, the WTPlant system detected 99.3 percent of plants in 17,000 natural photos.

Sulc and Matas [77] combined the ResNet152 and Inception-ResNetv2 architectures with LBP and achieved an accuracy of 99% on the Swedish Leaf dataset. Zhang et al. [78] proposed a seven-layer CNN to classify the Flavia dataset and reached 94% accuracy. Pawara et al. [79] fine-tuned the AlexNet and GoogLeNet architectures and achieved 94% accuracy on Flavia, 98% on Folio and 99% on the Swedish Leaf dataset. Barre et al. [80] used a 17-layer CNN architecture and obtained an accuracy of 97.9% validated on the LeafSnap, Flavia, and Foliage datasets. [79] fine-tuned the AlexNet and GoogLeNet architectures and achieved 94% accuracy on Flavia, 98% on Folio and 99% on the Swedish Leaf dataset.

Pearline et al. [81] utilized VGG19 architecture with a logistic regression classifier on the Folio, Flavia, and Swedish leaf datasets, achieving an accuracy of 96%, 96%, and 99%, respectively. Blesslin and Baulkani [82] developed a proposed network AousethNet by replacing the SoftMax classifier with supervised learning classifiers; support vector machine (SVM), Decision tree (DT), Naive Bayes (NB) and K Nearest Neighbor (KNN), Ensemble classifier (EC) and the Majority vote classifier (MVC). With the Mendeley









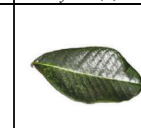
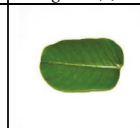




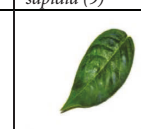



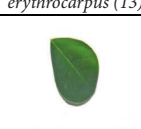

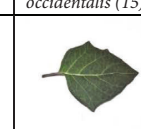
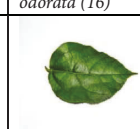
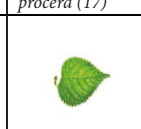

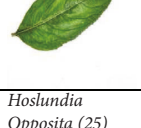

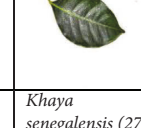
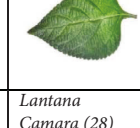
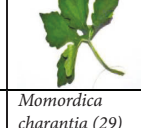


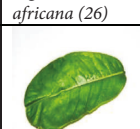
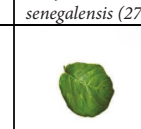
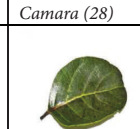
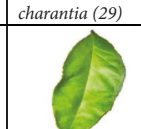
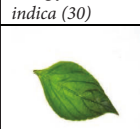
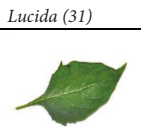

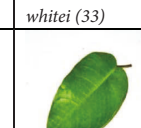
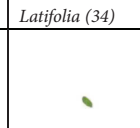
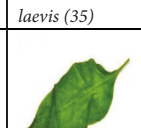
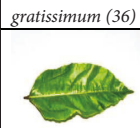
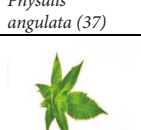

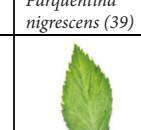
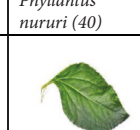


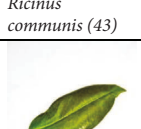
TABLE 1: Handcrafted features with supervised classifiers.

Reference	Features	Dataset	Algorithm	Accuracy (%)
[55]	Texture and shape features	Medicinal plant specimen library of anhui university of traditional Chinese medicine	SVM classifier	93.3%
[56]	Perimeter, a number of vertices, length, width, perimeter and area of hull, colour	Dataset of 24 different plant species having 30 images each from the tropical island of Mauritius	Random forest classifier	90.1%
[57]	Leaf shape and venation structure features	Philippine herbal medicine plants using leaf features	Logistic regression, naïve bayes, K-nearest neighbor (KNN), linear discriminant analysis, classification and regression trees, SVM, and neural networks (NN)	98.6%
[58]	Texture, colour, and shape	Herbal medicinal plants on a dataset containing 50 different species having 500 leaves.	Neural networks	93.3%
[59]	Color, texture and shape feature	Ayurvedic medicinal plant	SVM	96.66%
[60]	Centroid contour curve form signature, a fine-scale margin feature histogram and an interior texture feature histogram	Fisher's iris plant, wheat seed kernels, and 100 plant leaves	Extreme learning machine (ELM) algorithm with K-nearest neighbor, decision tree classifier, support vector machine, naïve bayes classifier, and a multilayer perceptron trained with backpropagation algorithm	Iris data set (97%) Seed data set (96%).
[61]	Shape, texture, and colour	A total of 3,150 leaf photos from 25 different herbal, fruit, and vegetable species	Support vector machine, K-nearest neighbors, multilayer perceptron, random forest, and decision tree algorithms	85.82
[62]	14 features were selected using a chi-square feature selection strategy	Six varieties of medicinal plant leaves	Multilayer perceptron, random forest, logit-boost, basic logistic, and bagging	99.01%
[63]	Morpho-colourimetric parameters Visible (VIS)/Near infrared (NIR) spectral analysis	20 different Chinese medicinal plants	ANN model	98.3%
[64]	Texture and colour features	Swedish leaf dataset	Multiclass-support vector machine	93.26%.

TABLE 2: Deep learning models.

Reference	Algorithm	Dataset	Accuracy (%)
[66]	GoogLeNet + linear SVM		87.34%.
[67]	Convolution neural network		86%
[68]	Five-layered convolutional neural network (CNN)	Flavia leaf dataset Swedish leaf dataset	98.22%.
[69]	CNN-LSTM network with 20 layers		95.06%.
[70]	MobileNetV2		98.97
[71]	Dual-path CNN (DP-CNN)		95.67%
[72]	Dual-path CNN model	14 species of Taiwan's most prevalent trees	77.1%
[73]	AlexNet, GoogLeNet, VGG-19, ResNet50, and MobileNetV2	Leafsnap image dataset	92.3%
[74]	5-Layer CNN architecture	Flavia leaf dataset Swedish leaf dataset	95.5 98.2
[75]	GoogleNet, VGGNet, and AlexNet	LIFECLEF 2015 dataset	80%
[76]	Two AlexNets pretrained models		99.3%
[77]	ResNet152 and Inception-ResNetv2 architectures with LBP	Swedish leaf dataset	99%
[78]	Seven-layer CNN	Flavia dataset Flavia Folio Swedish leaf dataset	94% 94% 98% 99%
[79]	AlexNet and GoogLeNet		97.9%
[80]	17-Layer CNN architecture	Folio Flavia	96% 96%
[81]	VGG19 architecture with a logistic regression classifier	Swedish leaf datasets	99%,
[82]	AousethNet	Mendeley dataset (MD2020)	99%

TABLE 3: Description of dataset.

					
<i>Adenia cissampeloides</i> (1)	<i>Ageratum conyzoides</i> (2)	<i>Alchornea cordifolia</i> (3)	<i>Aframomum melegueta</i> (4)	<i>Amaranthus spinosus</i> (5)	<i>Bridelia ferruginea</i> (6)
					
<i>Baphia nitida</i> (7)	<i>Bidens pilosa</i> (8)	<i>Blighia sapida</i> (9)	<i>Cassia alata</i> (10)	<i>Clausena anisata</i> (11)	<i>Citrus aurantifolia</i> (12)
					
<i>Capparis erythrocarpus</i> (13)	<i>Cnestis ferruginea</i> (14)	<i>Cassia occidentalis</i> (15)	<i>Chromolaena odorata</i> (16)	<i>Carapa procera</i> (17)	<i>Cryptolepis sanguinolenta</i> (18)
					
<i>Desmodium adscendens</i> (19)	<i>Dialium guineense</i> (20)	<i>Datura metel</i> (21)	<i>Ficus asperifolia</i> (22)	<i>Fleurya aestuans</i> (23)	<i>Griffonia simplicifolia</i> (24)
					
<i>Hoslundia opposita</i> (25)	<i>Kigelia africana</i> (26)	<i>Khaya senegalensis</i> (27)	<i>Lantana camara</i> (28)	<i>Momordica charantia</i> (29)	<i>Mangifera indica</i> (30)
					
<i>Morinda lucida</i> (31)	<i>Monodora myristica</i> (32)	<i>Mondia whitei</i> (33)	<i>Nauclea latifolia</i> (34)	<i>Newbouldia laevis</i> (35)	<i>Ocimum gratissimum</i> (36)
					
<i>Physalis angulata</i> (37)	<i>Palisota hirsuta</i> (38)	<i>Parquetina nigrescens</i> (39)	<i>Phyllanthus nururi</i> (40)	<i>Plumbago zeylanica</i> (41)	<i>Passiflora foetida</i> (42)
					
<i>Ricinus communis</i> (43)	<i>Rauwolfia vomitoria</i> (44)	<i>Sida acuta</i> (45)	<i>Synedrella nodiflora</i> (46)	<i>Trema orientalis</i> (47)	<i>Vernonia amygdalina</i> (48)
					
<i>Xylopiya aethiopica</i> (49)					

dataset (MD2020), the proposed model gave an accuracy of 99% and a precision of 98% with the MVC.

A summary of the reviewed papers is presented in Table 1 and Table 2.

3. Methodology

3.1. Dataset. A medicinal plant leaf dataset (MyDataset) [83] has been developed for this study from the Centre for Plant Medicine Research (CPMR) in Akuapem Akropong, Ghana and is presented in Table 3. The digital images are acquired using the NIKON D3500 camera on the abaxial portions of the medicinal plant leaves, in the uncompressed JPEG format in YCbCr colour format with the dimension $6000 \times 4000 \times 3$, in a closed environment to maintain constant illumination. The benchmark standards followed to create the dataset includes relevance, representativeness, nonredundancy, experimentally verified cases, scalability, and reusability [84, 85]. The working dataset consists of 2450 images, 50 images each from 49 medicinal plant leaves. The dataset created (MyDataset) is compared with four benchmark datasets which are: Flavia dataset, Swedish Leaf dataset, Mendeley Dataset (MD2020), and Folio dataset. Flavia dataset contains 1907 samples of 33 species of common native plants in Yangtze Delta, China [86, 87]. All the leaf images in this dataset have no petioles. The Swedish dataset contains 75 samples from each of the 15 Species of Swedish plants or trees [88, 89]. The Folio dataset contains 576 images with 18 samples each from 32 species [90] while the Mendeley dataset contains 1835 images from 30 species [91].

3.2. Feature Extraction. Pretrained CNN networks which includes AlexNet, inceptionv3, DenseNet201, GoogLeNet, resnet101, resnet18, resnet50, mobilenetv2, vgg16, and vgg19 are used as feature extractors. The process of using features from a pretrained network is known as transfer learning [92]. Transfer learning aims to increase target learners' performance on target domains by transferring data from a variety of related source domains. The dependency on a big amount of target domain data for developing target learners can be decreased using this strategy. The last layer before the fully connected layer for classification commonly called the bottleneck layers are used in this study. A description of the pretrained network and the layers used are presented in Table 4.

3.3. Supervised Classifiers. Six (6) well-known supervised learning algorithms are chosen for this study. These classifiers are chosen based on the following groups: Bayesian, lazy classifiers, trees, and functions [93]. Bayesian classifiers assign membership probability to new objects in order to categorize them, and they are known to be quick and accurate even when dealing with massive amounts of data [94]. From a training set of objects represented by various

qualities, trees derive rules. Because the derived rules may be represented as a treelike graph, they can be easily understood [95]. Lazy (or instance-based) classifiers save all of the training samples and wait until a new instance needs to be classified before building a classifier. During the training phase, lazy-learning algorithms require less calculation time, but during the classification phase, they require greater computation time [96]. Functions or non-probabilistic classifiers in this category strive to generalize the training data before accepting queries. The majority of the methods in this family can be thought of as simple applications of optimization theory and statistical estimation [97]. The classifiers used are Naive Bayes, Support Vector Machine (SVM), K-nearest Neighbour (kNN), Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF).

3.4. Proposed Model. The proposed CNN model (OTAM-Net) created is based on the DenseNet architecture since it has the following advantages: since error signals can be easily transferred to older levels more directly, there is a significant gradient flow; an implicit deep supervision is provided because the final classification layer provide strict supervision to earlier levels; more diversified features can be extracted because each layer in DenseNet receives all preceding layers as input as opposed to standard CNN models where the classifier uses the most complex features [98]. In DenseNet, the classifier uses features of all complexity levels, giving more smooth decision boundaries [98]. The model fused handcrafted features i.e., Log-Gabor features into each dense block of the DenseNet model and propagated features to each next dense block as shown in Figure 1. The minimum and maximum frequencies, the filter bandwidth to employ, the scaling between center frequencies of subsequent filters, the number of orientations, the number of scales, and the angular spread of each filter are the parameters used in designing a Log-Gabor filter [99]. The parameters used are presented in Table 5 [100].

Features from each dense block are fused with the log Gabor features before being sent to the next block allowing for rich complex features at each stage.

3.5. Classification Metrics. The confusion matrix is the most often used assessment measure in predictive analysis, owing to its simplicity and ability to compute other important metrics including accuracy, recall, and precision. It is an $N \times N$ matrix that depicts a model's overall performance when applied to a dataset, where N represents the number of class labels in the classification task. These metrics are as follows [101]:

True Positive (TP) refers to a situation in which the actual value was positive and the predicted value was positive as well

TABLE 4: Description of pretrained networks.

No	Network	Image input size	Depth	No of features	Layer
1	Alexnet	227-By-227	8	4096	fc7
2	DenseNet201	224-By-224	201	1920	avg_pool
3	Googlenet	224-By-224	22	1024	pool5-7x7_s1
4	inceptionv3	299-By-299	48	2048	avg_pool
5	mobilenetv2	224-By-224	53	1280	global_average_pooling2d_1
6	resnet18	224-By-224	18	512	pool5
7	resnet50	224-By-224	50	2048	avg_pool
8	resnet101	224-By-224	101	2048	pool5
9	vgg16	224-By-224	16	4096	fc7
10	vgg19	224-By-224	19	4096	fc7

False Positive (FP) refers to a situation in which the actual value is negative but the predicted value is positive

True Negative (TN) refers to a situation in which the actual value was negative and the predicted value was negative as well

False Negative (FN) refers to a situation in which the actual value is positive but the predicted value is negative

Accuracy (ACC) is the fraction of appropriately classified connections (true positives and true negatives) over the total number of connections in the dataset

$$ACC = \frac{TP + TN}{N}, \quad (3)$$

where N = number of instances. The fraction of true positives to actual positive cases is known as Recall, also known as sensitivity or True Positive Rate (TPR). Simply said, recall is the number of true positives discovered (recalled) out of all true positive cases.

$$TPR = \frac{TP}{TP + FN}. \quad (4)$$

Precision, also known as Positive Predictive Value (PPV), is the ratio of true positives to false positives. Simply put, precision refers to how many of the cases discovered were true positives.

$$PPV = \frac{TP}{TP + FP}. \quad (5)$$

False Positive Rate (FPR) or “Fall-Out”: this is the percentage of negative cases in the data that are mistakenly recognized as positive (i.e., the probability that false alerts will be raised).

$$FPR = \frac{FP}{FP + TN}. \quad (6)$$

The harmonic mean of recall and precision is the F1 score, often known as the F score or F-measure. Its value varies from 0 to 1, with 0 being the worst and 1

being the greatest. It can be calculated in the following way.

$$F_1 = \frac{2 * (PPV * TPR)}{PPV + TPR}. \quad (7)$$

4. Analysis

Ten pretrained networks were used to extract complex features from the leaf images. The number of features extracted is directly connected to the layer at which the feature was extracted. The layers used in each pretrained network are the last layer before the fully connected layer for classification. The number of features extracted ranges from 512 to 4096. The AlexNet, vgg16, and vgg19 extracted the highest number of features of 4096 while the resnet18 extracted the lowest number of features of 512. The time taken for each pretrained network is not directly proportional to the number of features but the complexity of the pretrained network. From Figure 2, it can be seen that the vgg19 performed worse in terms of time taken and AlexNet performed the best taking a time of about 54 seconds to complete as compared to 1498 seconds for vgg19. The DenseNet201 model also used about 956 seconds to complete which was the third worst performance in terms of time. To ascertain why the DenseNet201 model was chosen as the base model for OTAMNet, the features derived from the pretrained network were used for the classification process using the six supervised learning algorithms mentioned earlier. The average result from the six classifiers was used as a benchmark to check the performance across the various metrics. The results are presented in Figures 3–7.

As shown in Figure 3 to Figure 7, the DenseNet201 performed the best in terms of accuracy, F1 score, False Positive Rate, True Positive Rate, and Positive Predictive Value. In terms of accuracy, DenseNet201 scored 87% with resnet50 following with 85% and GoogLeNet performing the worst with 79%. With the F1 score, DenseNet201 also performed the best with 87%, resnet18 followed with 86% and GoogLeNet scored the lowest with 79%. DenseNet201 also had the lowest False Positive Rate with 0.26%, resnet18 following with 0.28% and the highest being 0.42% by GoogLeNet. DenseNet201 outperformed all other networks

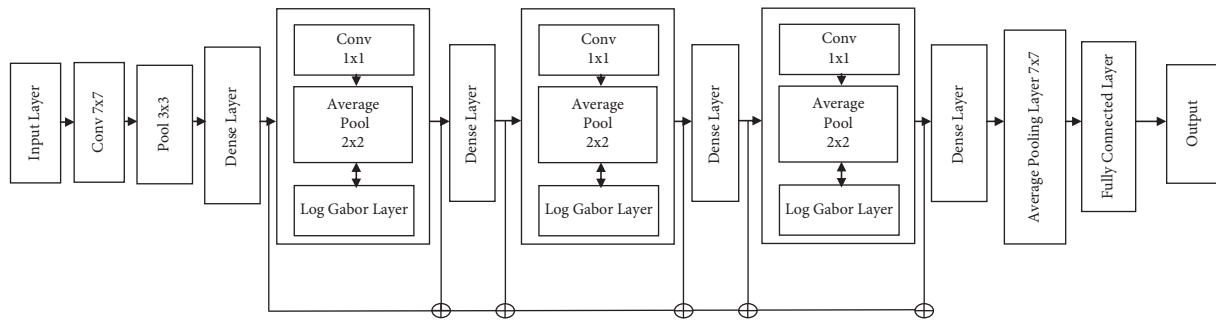


FIGURE 1: Proposed CNN model.

TABLE 5: Log-gabor parameters.

Parameter	Value
Number of filter scales	8
Number of filter orientations	10
Minimum frequency	3
Scaling between centre frequencies	2
Filter bandwidth	0.65
Angular spread of each filter	1.5

also in terms of the True Positive Rate and Positive Predictive Value scoring 88% and 87% respectively. The resnet50, resnet18, and resnet101 follow with 86%, 85%, and 83%, respectively for the True Positive rate and 86%, 86%, and 85% in terms of the Positive Predictive Value.

The time taken for the classifier to complete its operations was also taken into consideration as presented in Figure 8 vgg19, vgg16, and AlexNet had the highest running time of 23, 22, and 21 seconds as compared to resnet18, GoogLeNet, and mobilenetv2 having the lowest time of 3, 4, and 5 seconds, respectively. DenseNet201 performed fairly well with 10 seconds as shown in Figure 8. Overall, the denset201 model proved to be the best pre-trained model to be used as the base model for developing OTAMNet.

OTAMNet was implemented in the python environment and tested on the dataset “MyDataset” created by the researcher and four benchmark datasets. The metrics used during the training of the model are accuracy and loss. A 70 : 30 ratio was used for splitting the dataset into training and testing sets. The model was also validated using a 33% validation split on the training data. The Adam optimizer was used during the training of the model. For training the model, the Google colab server was used to boost the running time of the model. To check for overfitting, an EarlyStopping callback was used to check the minimum validation loss with a waiting time of 5 epochs to stooped the model if the validation loss starts to increase after the waiting period. Early stopping is a method of training that allows a set of an arbitrary large number of training epochs and then terminate when the improvement of the model’s performance on the validation dataset stops. A ModelCheckpoint was also used to only save and use the best weights which are derived from the model. To reduce the running time and

improve performance, a batch size of 8 was used for both training and testing of the model.

The results obtained after running the OTAMNet model on MyDataset is presented in Figure 9 and 10.

An accuracy metric is used to calculate the algorithm’s processing ability in an interpretable way. It is a metric that measures how closely your model’s predictions match the actual facts. A loss function is used to fine-tune a machine-learning algorithm. Training and validation are used to estimate the loss, and the model’s performance in these two sets determines the significance of the loss. The sum of all errors made during each training or validation set is calculated for each example. After each optimization cycle, the loss value shows how well or poorly a model performs. The training accuracy reached 100% and the training loss also decreased from 2 to 0.2 after the 9th epoch. The validation accuracy increased from the 1st to the 5th epoch, then the 7th, 8th, 15th, 35th, and 51st epoch. The EarlyStopping model kicked in at the 53rd epoch and the training stopped because the model performance stopped improving on the validation dataset. The validation accuracy for the model was 98% with a loss of 0.08.

The model was saved and used to predict a new set of data and the results are presented in Table 6.

75% of the Plant leaf tested had 100% accuracy and 25% had a 99% accuracy. OTAMNet also performed well in terms of the False Positive Rate with 87% of the dataset scoring 0% with the highest FPR being 0.04%. 87% of the dataset had a TPR of 1 showing a high rate of predicting positive classes.

The model was trained on two other optimizers, Stochastic Gradient Descent (SGD) and the Root Mean Square Propagation (RMSProp) to determine its performance which is presented in Figures 11–14.

It could be seen that the Adam and RMSProp optimizer both had a sharp curve after the 10th epoch as compared to the SGD. Overfitting occurred earlier in the RMSProp optimizer i.e., at the 34th epoch as compared to the Adam optimizer which occurred at the 53rd epoch whiles the SGD run on the maximum epoch allocated which was 100. When the model was tested on the test data, the Adam optimizer produced the highest accuracy of 98% and the lowest loss of 0.08 as compared to the SGD and RMSProp. This is presented in Table 7 and a summary

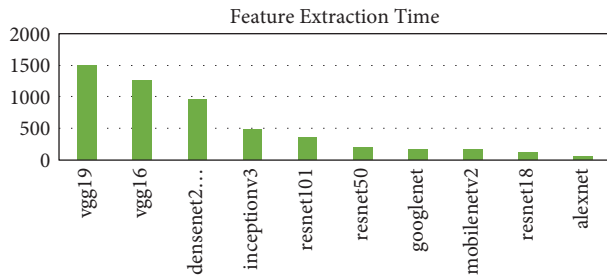


FIGURE 2: Feature Extraction Time for Pretrained networks.

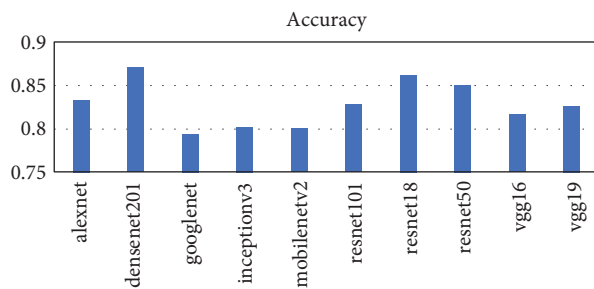


FIGURE 3: Accuracy metric for Pretrained networks.

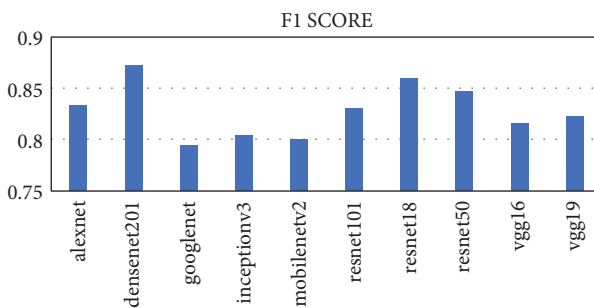


FIGURE 4: F1 score metric for Pretrained networks.

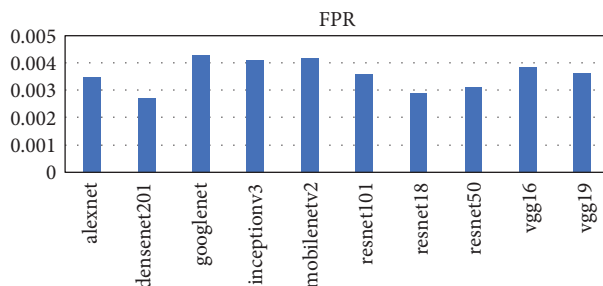


FIGURE 5: FPR score metric for Pretrained networks.

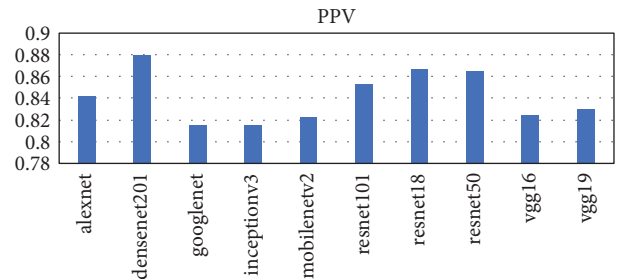


FIGURE 6: PPV score metric for Pretrained networks.

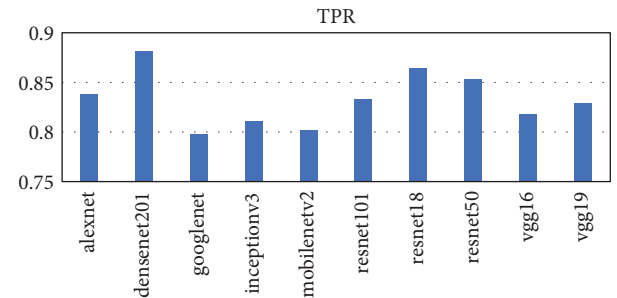


FIGURE 7: TPR score metric for Pretrained networks.

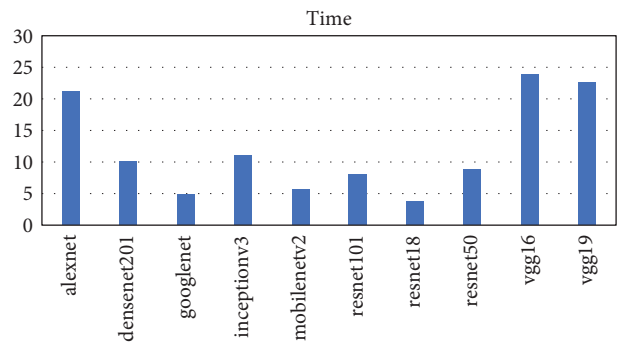


FIGURE 8: Time metric for Pretrained networks.

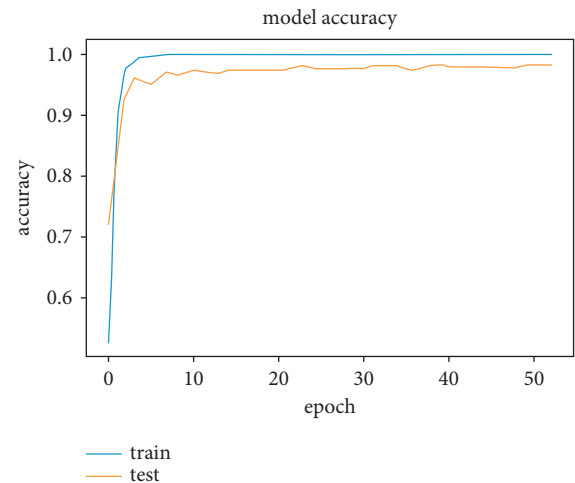


FIGURE 9: Model accuracy for MyDataset.

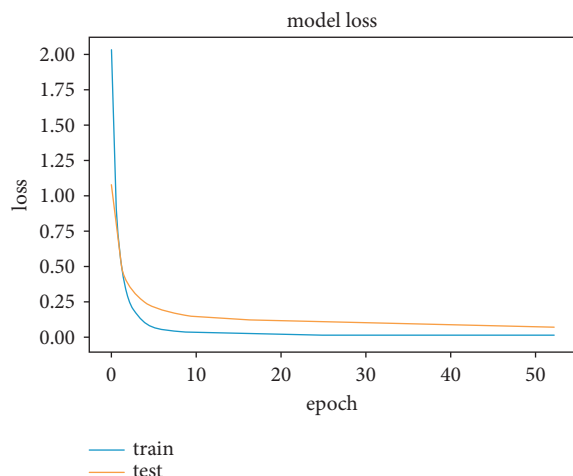


FIGURE 10: Model loss for MyDataset.

of the analysis is presented in Table 8 using the Adam optimizer.

OTAMNet was also tested on the Flavia dataset having 32 plant species. Early Stopping kicked in at the 47th epoch to prevent overfitting of the data. The model achieved 99% accuracy on the validation data set and a loss of 0.01 as shown in Figures 15 and 16.

The model, when used on the test data had an accuracy of 99.4%, F1 score of 99.4%, TPR of 99.4%, PPV of 99.5% and an FPR of 0.00017, and the full metrics are presented in Table 9.

The Swedish data produce a 100% accuracy with a loss of 0.00 both on the training and validation test data. The model also predicted correctly all test data samples and this is presented in Figures 17 and 18 and Table 10.

When tested on the Mendeley dataset having 30 plant species, OTAMNet achieved 99% accuracy on the validation data set and a loss of 0.06. Early Stopping kicked in at the 35th epoch to prevent overfitting the data as shown in Figures 19 and 20. The model when the run of the test data had an accuracy of 98.6%, F1 score of 98.4%, TPR of 98.3%, PPV of 98.6%, and an FPR of 0.00048. The performance metric for each leaf in the dataset is presented in Table 11.

OTAMNet achieved 97% accuracy on the validation data set and a loss of 0.08 when tested on the Folio dataset having 32 plant species as shown in Figures 21 and 22. The model when run on the test data had an accuracy of 96.8%, F1 score of 96.3%, TPR of 96.5%, PPV of 97.7%, and an FPR of 0.00104. The performance metric for each leaf in the Folio dataset is presented in Table 12.

OTAMNet performed extremely well on all datasets used achieving an accuracy of 98%, 99%, 100%, 99%, and 98% on MyDataset, Flavia, Swedish Leaf Dataset, MD2020, and Folio dataset, respectively. The False Positive rate achieved was also insignificant with the highest being 0.000104 on the

TABLE 6: Metrics for mydataset.

Plant leaf	ACC	F1	TPR	FPR	PPV
1	1	1	1	0	1
2	0.99592	0.90909	1	0.00417	0.83333
3	0.99796	0.94118	1	0.00207	0.88889
4	1	1	1	0	1
5	0.99796	0.96552	1	0.0021	0.93333
6	1	1	1	0	1
7	1	1	1	0	1
8	1	1	1	0	1
9	1	1	1	0	1
10	1	1	1	0	1
11	1	1	1	0	1
12	1	1	1	0	1
13	0.99388	0.89655	0.8125	0	1
14	0.99796	0.90909	0.83333	0	1
15	1	1	1	0	1
16	1	1	1	0	1
17	1	1	1	0	1
18	1	1	1	0	1
19	1	1	1	0	1
20	0.99592	0.875	1	0.00414	0.77778
21	1	1	1	0	1
22	0.99796	0.92308	0.85714	0	1
23	1	1	1	0	1
24	1	1	1	0	1
25	1	1	1	0	1
26	1	1	1	0	1
27	1	1	1	0	1
28	1	1	1	0	1
29	1	1	1	0	1
30	1	1	1	0	1
31	1	1	1	0	1
32	0.99796	0.94118	1	0.00207	0.88889
33	0.99796	0.95652	0.91667	0	1
34	1	1	1	0	1
35	1	1	1	0	1
36	0.99592	0.9	1	0.00416	0.81818
37	1	1	1	0	1
38	0.99796	0.97143	0.94444	0	1
39	1	1	1	0	1
40	1	1	1	0	1
41	1	1	1	0	1
42	1	1	1	0	1
43	1	1	1	0	1
44	1	1	1	0	1
45	0.99592	0.88889	0.8	0	1
46	1	1	1	0	1
47	1	1	1	0	1
48	1	1	1	0	1
49	1	1	1	0	1

Folio dataset. A comparative summary of the results on all the datasets is presented in Figure 23–28.

The proposed system was compared with existing systems which have been reviewed in this study and the summary is presented in Table 13.

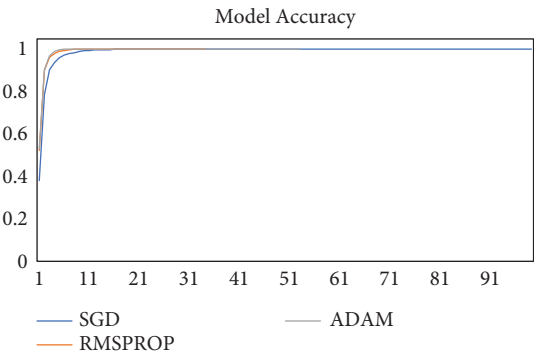


FIGURE 11: Model accuracy on optimizers.

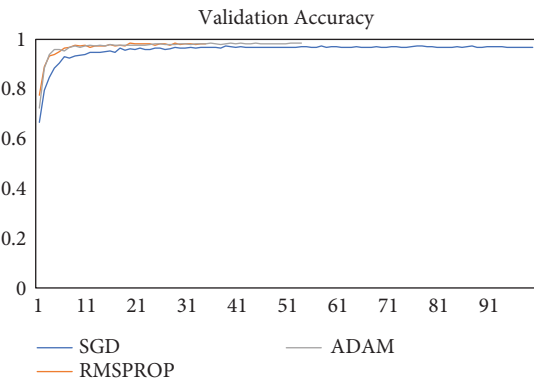


FIGURE 12: Validation accuracy on optimizers.

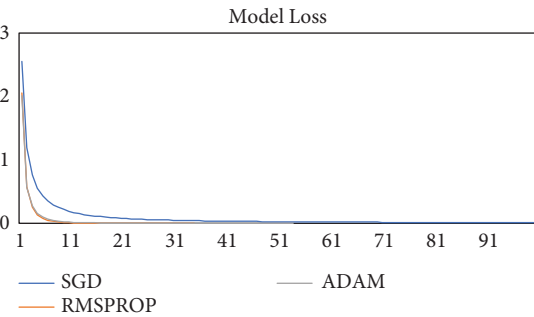


FIGURE 13: Model loss on optimizers.

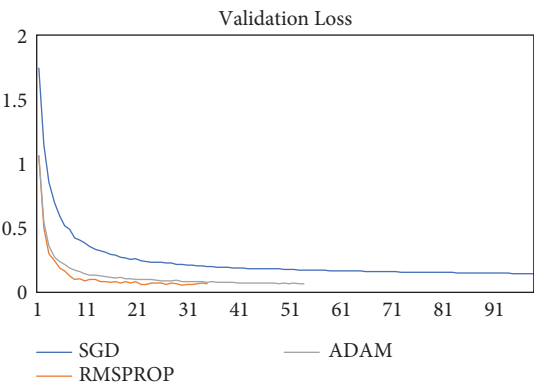


FIGURE 14: Validation loss on optimizers.

TABLE 7: Metrics based on optimizer.

Optimizer	Accuracy (%)	Loss
Adam	98	0.08
RMSProp	97	0.11
SGD	97	0.14

TABLE 8: Overall statistics.

Metric	Score
ACC	0.98163
F1	0.98117
FPR	0.00038
PPV	0.98246
TPR	0.98294

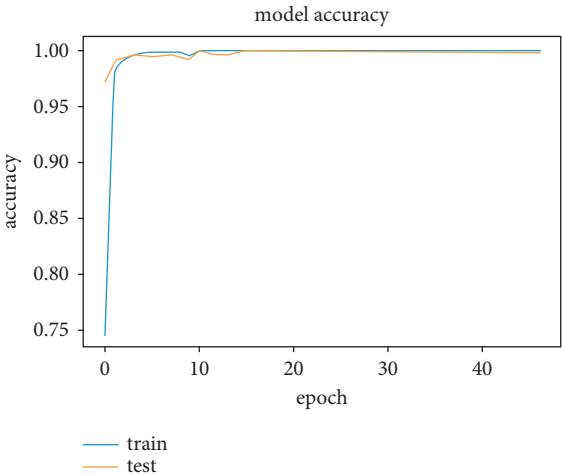


FIGURE 15: Model accuracy for flavia dataset.

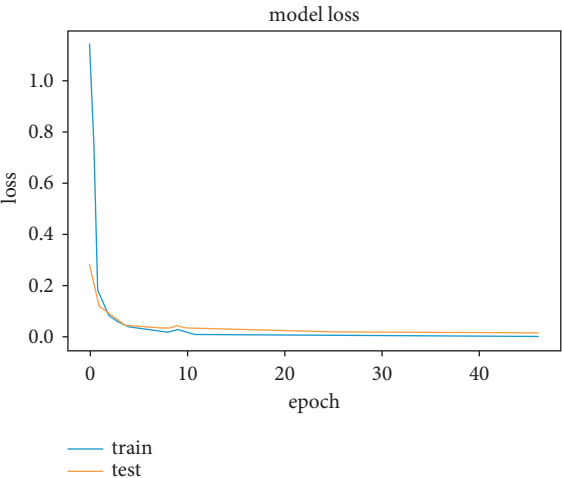


FIGURE 16: Model loss for flavia dataset.

TABLE 9: Metrics for flavia dataset.

Plant leaf	ACC	F1	TPR	FPR	PPV
1	1	1	1	0	1
2	1	1	1	0	1
3	1	1	1	0	1
4	1	1	1	0	1
5	1	1	1	0	1
6	0.99738	0.9697	1	0.00273	0.94118
7	1	1	1	0	1
8	1	1	1	0	1
9	1	1	1	0	1
10	1	1	1	0	1
11	1	1	1	0	1
12	1	1	1	0	1
13	1	1	1	0	1
14	0.99738	0.93333	0.875	0	1
15	1	1	1	0	1
16	1	1	1	0	1
17	1	1	1	0	1
18	1	1	1	0	1
19	1	1	1	0	1
20	1	1	1	0	1
21	1	1	1	0	1
22	1	1	1	0	1
23	1	1	1	0	1
24	1	1	1	0	1
25	1	1	1	0	1
26	1	1	1	0	1
27	0.99738	0.95652	1	0.0027	0.91667
28	1	1	1	0	1
29	1	1	1	0	1
30	1	1	1	0	1
31	0.99738	0.96774	0.9375	0	1
32	1	1	1	0	1

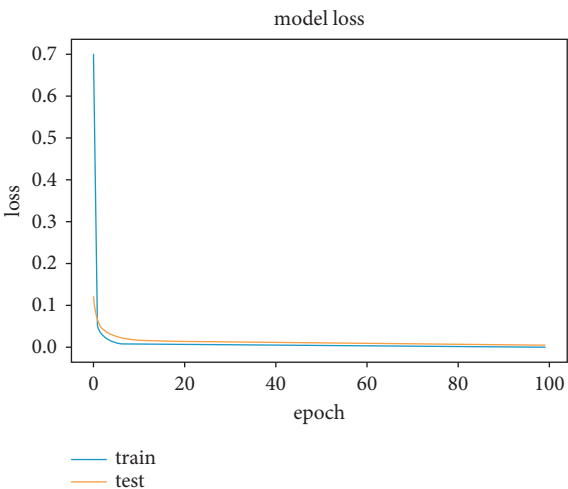


FIGURE 18: Model loss for Swedish leaf dataset.

TABLE 10: Metrics for swedish leaf dataset.

Plant leaf	ACC	F1	TPR	FPR	PPV
1	1	1	1	0	1
2	1	1	1	0	1
3	1	1	1	0	1
4	1	1	1	0	1
5	1	1	1	0	1
6	1	1	1	0	1
7	1	1	1	0	1
8	1	1	1	0	1
9	1	1	1	0	1
10	1	1	1	0	1
11	1	1	1	0	1
12	1	1	1	0	1
13	1	1	1	0	1
14	1	1	1	0	1
15	1	1	1	0	1

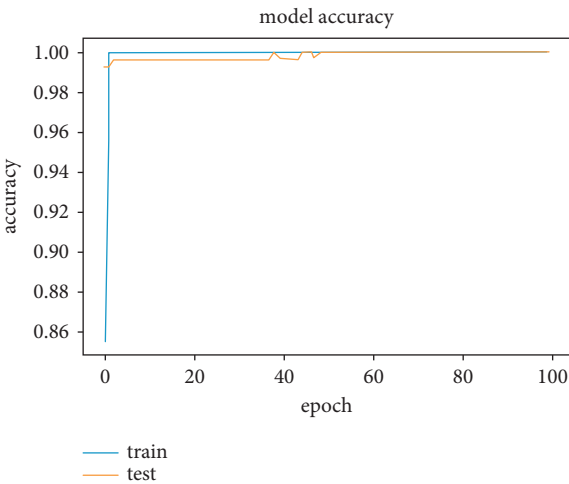


FIGURE 17: Model accuracy for Swedish leaf dataset.

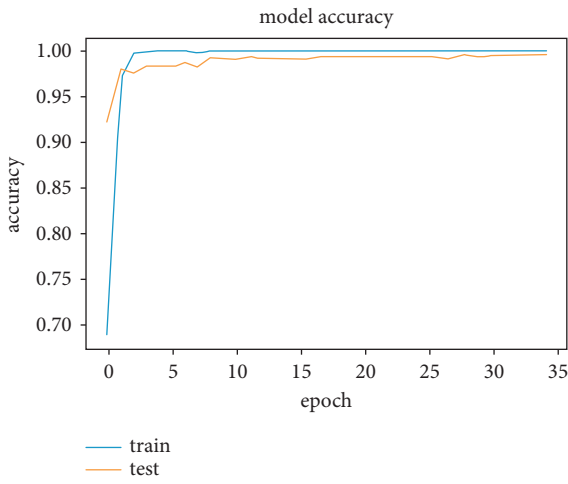


FIGURE 19: Model accuracy for mendeley dataset.

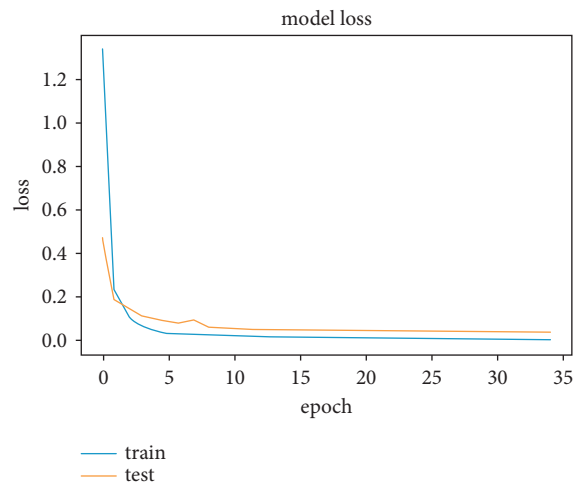


FIGURE 20: Model loss for mendeley dataset.

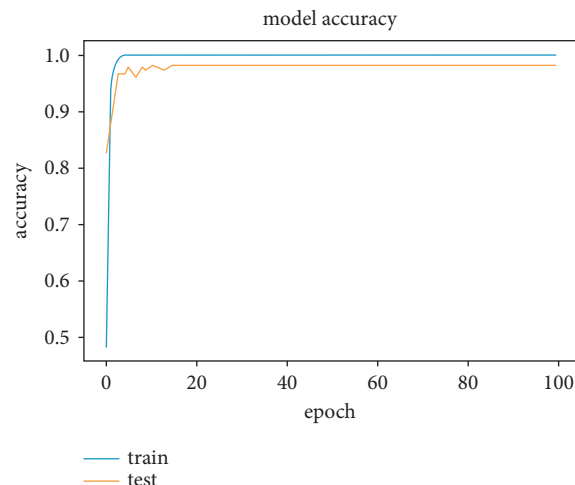


FIGURE 21: Model accuracy for folio dataset.

TABLE 11: Metrics for mendeley dataset.

Plant leaf	ACC	F1	TPR	FPR	PPV
1	1	1	1	0	1
2	1	1	1	0	1
3	1	1	1	0	1
4	1	1	1	0	1
5	0.99455	0.96154	1	0.00585	0.92593
6	1	1	1	0	1
7	1	1	1	0	1
8	1	1	1	0	1
9	0.99455	0.9	0.81818	0	1
10	1	1	1	0	1
11	1	1	1	0	1
12	1	1	1	0	1
13	1	1	1	0	1
14	1	1	1	0	1
15	0.99183	0.91429	0.88889	0.00287	0.94118
16	1	1	1	0	1
17	0.99728	0.96296	1	0.00282	0.92857
18	1	1	1	0	1
19	1	1	1	0	1
20	1	1	1	0	1
21	1	1	1	0	1
22	1	1	1	0	1
23	1	1	1	0	1
24	1	1	1	0	1
25	1	1	1	0	1
26	1	1	1	0	1
27	1	1	1	0	1
28	1	1	1	0	1
29	1	1	1	0	1
30	0.99455	0.8	0.8	0.00276	0.8

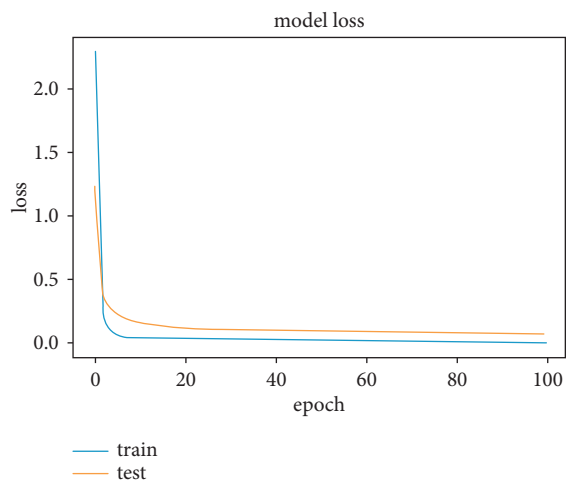


FIGURE 22: Model loss for mendeley dataset.

TABLE 12: Metrics for folio dataset.

Plant leaf	ACC	F1	TPR	FPR	PPV
1	1	1	1	0	1
2	1	1	1	0	1
3	0.98438	0.83333	1	0.01626	0.71429
4	1	1	1	0	1
5	1	1	1	0	1
6	1	1	1	0	1
7	0.99219	0.85714	0.75	0	1
8	1	1	1	0	1
9	1	1	1	0	1
10	1	1	1	0	1
11	1	1	1	0	1
12	1	1	1	0	1
13	1	1	1	0	1
14	1	1	1	0	1
15	1	1	1	0	1
16	0.98438	0.75	1	0.016	0.6
17	1	1	1	0	1
18	1	1	1	0	1
19	1	1	1	0	1
20	1	1	1	0	1
21	1	1	1	0	1
22	0.99219	0.92308	0.85714	0	1
23	1	1	1	0	1
24	1	1	1	0	1
25	1	1	1	0	1
26	0.98438	0.5	0.33333	0	1
27	1	1	1	0	1
28	1	1	1	0	1
29	1	1	1	0	1
30	1	1	1	0	1
31	1	1	1	0	1

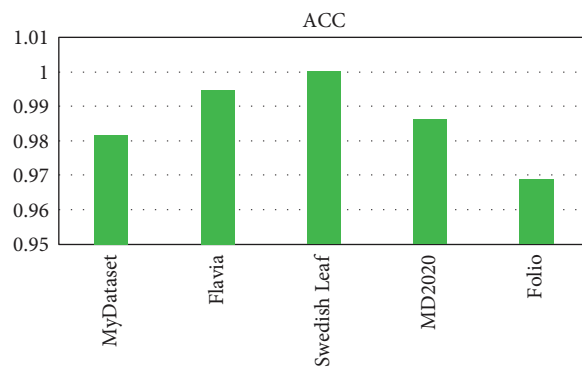


FIGURE 23: Model accuracy for all dataset.

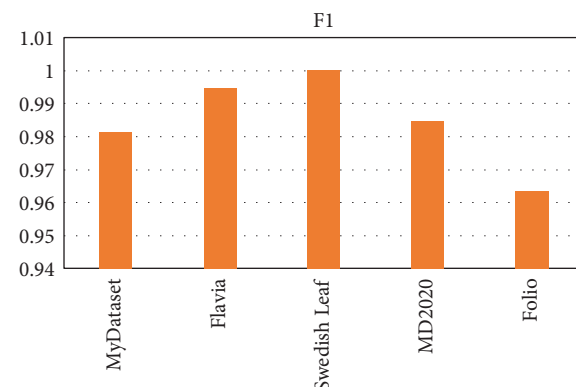


FIGURE 24: F1 score for all dataset.

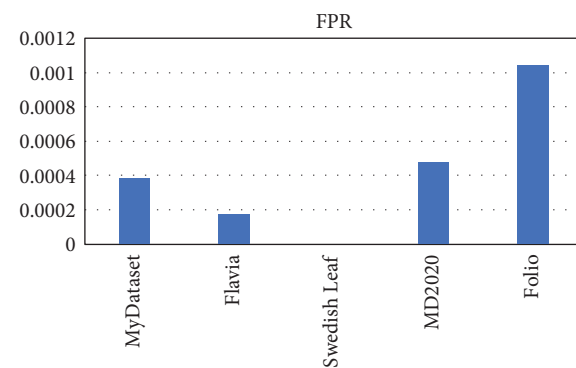


FIGURE 25: Fpr for all dataset.

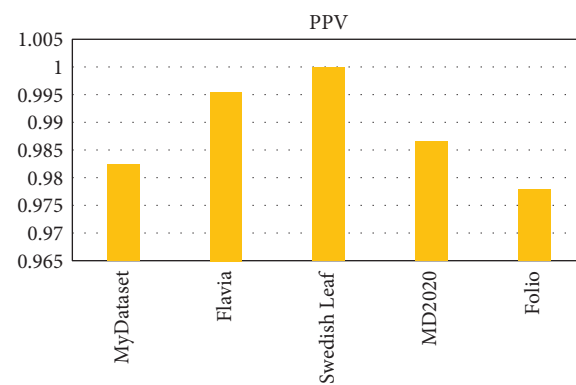


FIGURE 26: PPV for all dataset.

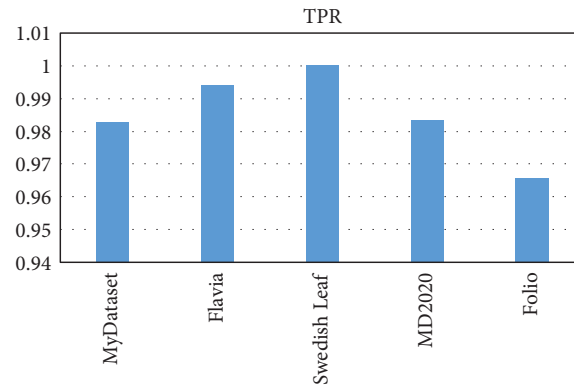


FIGURE 27: Tpr for all dataset.

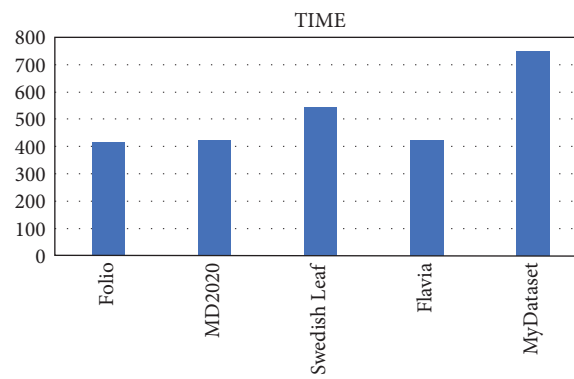


FIGURE 28: Running time for all dataset.

TABLE 13: Comparison with existing systems.

Source	Method	Dataset	Accuracy (%)
OTAMNet	Log-gabor filter and DenseNet201	MyDataset	98
		Flavia	99
		Swedish	100
		Folio	97
		MD2020	99
[82]	Modified AlexNet	MD2020	99
[73]	AlexNet, GoogLeNet, VGG-19, ResNet50, and MobileNetV2	Leafsnap	92
[65]	Binarized Neural Network (BNN)	Swedish leaf	77
[68]	CNN	Flavia	98
		Swedish leaf	
[74]	Histogram of oriented gradient (HoG) and deep convolutional neural network	Flavia	96
		Swedish leaf dataset	
[81]	VGG19 with LR	Folio	96
		Flavia	96
[102]	AlexNet and VGG16 with LDA	Swedish leaf dataset	99
		Swedish leaf dataset	99
[80]	17-Layer CNN architecture	LeafSnap	
		Flavia	97
[79]	AlexNet and GoogLeNet	Foliage datasets	
		Flavia	94
		Swedish leaf dataset	99
[75]	GoogLeNet, VGGNet, and AlexNet	LifeClef 2015 dataset	80
[103]	26-Layer CNN architecture	BJFU100 dataset	91
[78]	7-Layer CNN architecture	Flavia dataset	94
[77]	ResNet152 and Inception-ResNetv2 with LBP	Swedish leaf dataset	99

5. Conclusion

This study was carried out to identify plants based on their textural features using Log-Gabor filters and deep learning techniques. OTAMNet which was created by fusing a log Gabor layer into the transition layer of the DenseNet201 architecture achieved an accuracy of 98%. OTAMNet was tested on other benchmark datasets: the Flavia, Swedish Leaf, MD2020, and the Folio dataset. The Flavia dataset achieved 99%, Swedish Leaf 100%, MD2020 99%, and the Folio dataset 97%. A false-positive rate of less than 0.1% was achieved in all cases. For future works, the medicinal plant database can be enhanced to have more species across the nation and their variations depending upon climatic conditions so also their applications. The study can also be extended to classifying medicinal plants based on anatomical (studying of the internal structure of plants, which usually takes place at the microscopic/cellular level) and chemotaxonomical (in classifying plants using their chemical constituents) properties.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. F. A. Fernandes, J. R. R. Dórea, and G. J. d. M. Rosa, "Image Image Analysis and Computer Vision Applications in Animal Sciences: An Overview," *Frontiers in Veterinary Science*, vol. 7, Article ID 551269, 2020.
- [2] A. Sarraf, M. Azhdari, and S. Sarraf, "A comprehensive review of deep learning architectures for computer vision applications," *Technology, and Sciences (ASRJETS) American Scientific Research Journal for Engineering*, vol. 77, no. 1, pp. 1–29, 2021.
- [3] C. Yang and Z. Liu, "Application of computer vision in electronic commerce," *Journal of Physics: Conference Series*, vol. 2, p. 2021, 1992.
- [4] A. E. Welchman, "The The Human Brain in Depth: How We See in 3D," *Annual Review of Vision Science*, vol. 2, no. 1, pp. 345–376, Oct. 2016.
- [5] C. Rainey, J. McConnell, R. R. Bond, C. Hughes, and S. McFadden, "Man vs Machine: A Comparison of Computer and Human Visual," *Processes in Radiographic Image Interpretation*, 2021.
- [6] R. Singh and Geetanjali, *Chemotaxonomy of Medicinal Plants: Possibilities and Limitations*, Elsevier, Amsterdam, Netherlands, 2018.
- [7] M. G. Simpson, "Plant anatomy and physiology," in *Plant Systematics*, pp. 537–566, Elsevier, 2019.
- [8] A. Adeniyi, A. Asase, P. K. Ekpe, B. K. Asitoakor, A. Adu-Gyamfi, and P. Y. Awekor, "Ethnobotanical study of medicinal plants from Ghana; confirmation of ethnobotanical uses, and review of biological and toxicological studies on medicinal plants used in Apra Hills Sacred Grove," *Journal of Herbal Medicine*, vol. 14, pp. 76–87, October 2016.
- [9] M. B. Adinortey, R. Agbeko, D. Boison et al., "Phytomedicines Phytomedicines Used for Diabetes Mellitus in Ghana: A Systematic Search and Review of Preclinical and Clinical Evidences for diabetes mellitus in Ghana: a systematic search and review of preclinical and clinical evidence," *Evidence-based Complementary and Alternative Medicine*, vol. 2019, pp. 1–23.
- [10] A. A. Boadu and A. Asase, "Documentation of herbal medicines used for the treatment and management of human diseases by some communities in southern Ghana," *Evidence-based Complementary and Alternative Medicine*, pp. 1–12, 2017.
- [11] J. O. Ezekwesili-Ofili and A. N. C. Okaka, "Herbal Medicines in African Traditional Medicine," *Herbal Medicine*, 2019.
- [12] World Health Organization, *WHO Global Report on Traditional and Complementary Medicine*, WHO, Geneva, Switzerland, 2019.
- [13] K. Pushpanathan, M. Hanafi, S. Mashohor, and W. F. Fazlil Ilahi, "Machine Learning in Medicinal Plants Recognition: A Review," *Artificial Intelligence Review*, 2020.
- [14] K. B. Barimah and C. S. Akotia, "The promotion of traditional medicine as enactment of community psychology in Ghana," *Journal of Community Psychology*, vol. 43, no. 1, pp. 99–106, Dec 2015.
- [15] P. P. Kaur, S. Singh, and M. Pathak, "Review of machine learning herbal plant recognition system," *SSRN Electronic Journal*, 2020.
- [16] A. R. Sfar, N. Boujemaa, and D. Geman, "Vantage feature frames for fine-grained categorization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 835–842, San Juan, PR, USA, 2013.
- [17] J. Wäldchen and P. Mäder, "Plant Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review," *Archives of Computational Methods in Engineering*, vol. 25, no. 2, pp. 507–543, 2018.
- [18] M. Šulc and J. Matas, "Fine-grained recognition of plants from images," *Plant Methods*, vol. 13, no. 1, p. 115, Dec 2017.
- [19] M. Lasseck, "Image-based plant species identification with deep Convolutional Neural Networks," *CEUR Workshop Proceedings*, vol. 1866, 2017.
- [20] I. H. Sarker, "Deep Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Direction," *SN Comput Sci*, vol. 2, no. 6, p. 420, 2021.
- [21] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [22] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.
- [23] T. R. Gadekallu, DS Rajput, MPK Reddy et al., "A novel PCA-whale optimization-based deep neural network model for classification of tomato plant diseases using GPU," *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1383–1396, 2021.
- [24] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

- [25] L. Alzubaidi, J. Zhang, A.J. Humaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021.
- [26] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: Representation Learning: A Review and New Perspectives review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 1828, Aug 2013.
- [27] M. H. Farrell, T. Liang, and S. Misra, "Deep Deep Neural Networks for Estimation and Inference neural networks for estimation and inference," *Econometrica*, vol. 89, no. 1, pp. 181–213, 2021.
- [28] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: State-of-the-art in artificial neural network applications: A survey survey," *Heliyon*, vol. 4, no. 11, Article ID e00938, 2018.
- [29] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Object Detection With Deep Learning: A Reviewetection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [30] W. Xu, J. He, and Y. Shu, "Transfer learning and deep domain adaptation," *Advances and Applications in Deep Learning*, vol. 45, 2020.
- [31] L. Cheng, A. Dakkak, J. Xiong, and W. M. Hwu, "The design and implementation of a scalable deep learning benchmarking platform," *IEEE International Conference on Cloud Computing, CLOUD, DL*, vol. 2020-Octob, pp. 414–425, 2020.
- [32] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, "The The Understanding of Deep Learning: A Comprehensive Reviewunderstanding of deep learning: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–15, 2021.
- [33] J. Demšar and B. Zupan, "Hands-on training about overfitting," *PLoS Computational Biology*, vol. 17, no. 3, Article ID e1008671, pages. 2021.
- [34] J. Kukačka, V. Golkov, and D. Cremers, "Regularization for Deep Learning," *A Taxonomy*, pp. 1–24, 2017.
- [35] I. Marin, A. Kuzmanic Skelin, and T. Grujic, "Empirical evaluation of the effect of optimization and regularization techniques on the generalization performance of deep convolutional neural network," *Applied Sciences*, vol. 10, no. 21, pp. 7817–30, 2020.
- [36] S. Postalcioglu, "Performance Performance Analysis of Different Optimizers for Deep Learning-Based Image Recognitionanalysis of different optimizers for deep learning-based image recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 02, Article ID 2051003, 2019.
- [37] G. Perin and S. Picck, "On the influence of optimizers in deep learning-based side-channel analysis," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 615–636, LNCS, 12804.
- [38] R. M. Schmidt, F. Schneider, and P. Hennig, "Descending through a crowded valley-benchmarking deep learning optimizers," in *Proceedings of the International Conference on Machine Learning*, pp. 9367–9376, 2021.
- [39] C. Ma, L. Wu, and W. E, *A Qualitative Study of the Dynamic Behavior of Adaptive Gradient Algorithms*, vol. 145, pp. 1–22, 2020, <https://arxiv.org/abs/2009.06125>.
- [40] Z. Yi, "Evaluation and Evaluation and Implementation of Convolutional Neural Networks in Image Recognitionimplementation of convolutional neural networks in image recognition," *Journal of Physics: Conference Series*, vol. 1087, no. 6, Article ID 062018, 2018.
- [41] M. M. Rahman, M. S. Islam, R. Sassi, and Md. Aktaruzzaman, "Convolutional neural networks performance comparison for handwritten Bengali numerals recognition," *SN Applied Sciences*, vol. 1, no. 12, p. 1660, 2019.
- [42] M. Lewis, S. Purdy, S. Ahmad, and J. Hawkins, "Locations in the Locations in the Neocortex: A Theory of Sensorimotor Object Recognition Using Cortical Grid Cells.eocortex: a theory of sensorimotor object recognition using cortical grid cells," *Frontiers in Neural Circuits*, vol. 13, p. 22, 2019.
- [43] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [44] A. Brodzicki, J. Jaworek-Korjakowska, P. Kleczek, M. Garland, and M. Bogyo, "Pre-trained deep convolutional neural network for clostridioides difficile bacteria cytotoxicity classification based on fluorescence images," *Sensors*, vol. 20, no. 23, pp. 6713–17, 2020.
- [45] K. Wu, W. Cui, V. N. Khue, and E. Camci, "An introduction of deep learning methods for sensing applications," *Generalization With Deep Learning: For Improvement On Sensing Capability*, pp. 3–28, 2021.
- [46] Y. Lee and S. Nam, "Performance Performance Comparisons of AlexNet and GoogLeNet in Cell Growth Inhibition IC50 Predictioncomparisons of AlexNet and GoogLeNet in cell growth inhibition IC50 prediction," *International Journal of Molecular Sciences*, vol. 22, no. 14, p. 7721, Jul 2021.
- [47] J. Cao, M. Yan, Y. Jia, X. Tian, and Z. Zhang, "Application of a modified Inception-v3 model in the dynasty-based classification of ancient murals," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, p. 49, 2021.
- [48] Y. Ma, B. Niu, and Y. Qi, "Survey of image classification algorithms based on deep learning," *2nd International Conference on Computer Vision, Image, and Deep Learning*, Article ID 11911, Oct. 2021.
- [49] Z. Zhang and S. Zhang, "Towards understanding residual and dilated dense neural networks via convolutional sparse coding," *National Science Review*, vol. 8, no. 3, Article ID nwaal59, 2021.
- [50] W. Glegoła, A. Karpus, and A. Przybyłek, "ScienceDirect MobileNet MobileNet family family tailored tailored for for MobileNet family tailored for Raspberry Piaspberry raspberry pi," *Procedia Computer Science*, vol. 192, pp. 2249–2258, 2021.
- [51] J. Wu, W. Hu, Y. Wen, W. Tu, and X. Liu, "Skin Skin Lesion Classification Using Densely Connected Convolutional Networks with Attention Residual Learningesion classification using densely connected convolutional networks with attention residual learning," *Sensors*, vol. 20, no. 24, p. 7080, Dec 2020.
- [52] Y. Yang, L. Zhang, M. Du et al., "A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions," *Computers in Biology and Medicine*, vol. 139, 2021.
- [53] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America A*, vol. 4, no. 12, p. 2379.
- [54] O. Sliti, C. Gmati, F. Benzarti, and H. Amiri, "Robust object tracking using log-Gabor filters and color histogram," *ICPRAM*, in *Proceedings of the 3rd International Conference*

- on *Pattern Recognition Applications and Methods*, pp. 687–694, Loire Valley France, March 2014.
- [55] H. X. Kan, L. Jin, and F. L. Zhou, “Classification of medicinal plant leaf image based on multi-feature extraction,” *Pattern Recognition and Image Analysis*, vol. 27, no. 3, pp. 581–587, 2017.
 - [56] A. Begue, V. Kowlessur, U. Singh, F. Mahomoodally, and S. Pudaruth, “Automatic recognition of medicinal plants using machine learning techniques,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 166–175, 2017.
 - [57] R. G. De Luna, R. G. Baldovino, E. A. Cotoco et al., “Identification of philippine herbal medicine plant leaf using artificial neural network,” *HNICEM*, in *Proceedings of the 2017 - 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management*, pp. 1–8, Manila, Philippines, 2018.
 - [58] T. Vijayshree and A. Gopal, “Identification of herbal plant leaves using image processing algorithm: review,” *Research Journal of Pharmaceutical, Biological and Chemical Sciences*, vol. 9, no. 4, pp. 1221–1228, 2018.
 - [59] T. D. Dahigaonkar and R. Kalyane, “Identification of ayurvedic medicinal plants by image processing of leaf samples,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, pp. 351–355, 2018.
 - [60] L. Britto and L. Pacifico, “Plant species classification using Extreme learning machine,” *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pp. 13–24, 2019.
 - [61] D. M. C. Dissanayake and W. G. C. W. Kumara, “Plant Leaf Identification Based on Machine Learning Algorithms,” *Sri Lankan Journal of Technology*, pp. 60–66, 2021.
 - [62] S. Naem, A. Ali, C. Chesneau, M. H. Tahir, and F. Jamal, “The Classification of Medicinal Plant Leaves Based on Multispectral and Texture Feature Using Machine Learning Approach,” *Agronomy*, vol. 11, no. 2, 2021.
 - [63] J. R. Xue, S. Fuentes, C. Poblete-Echeverria et al., “Automated Chinese medicinal plants classification based on machine learning using leaf morpho-colorimetry, fractal dimension and visible/near infrared spectroscopy,” *International Journal of Agricultural and Biological Engineering*, vol. 12, no. 2, pp. 123–131, 2019.
 - [64] S. Kaur and P. Kaur, “Plant species identification based on plant leaf using computer vision and machine learning techniques,” *Journal of Multimedia Information System*, vol. 6, no. 2, pp. 49–60, 2019.
 - [65] M. M. Singh, “A Survey on Different Methods for Medicinal Plants Identification and Classification System,” *Revista Gestão Inovação e Tecnologias*, vol. 11, no. 4, pp. 3191–3202, 2021.
 - [66] T. T. N. Nguyen, T. L. Le, H. Vu, and V. S. Hoang, “Towards an automatic plant identification system without dedicated dataset,” *International Journal of Machine Learning and Computing*, vol. 9, no. 1, pp. 26–34, 2019.
 - [67] M. Jaiganesh, M. Sathyadevi, K. S. Chakravarthy, and C. Sarada, “Identification of plant species using CNN-classifier,” *Journal Of Critical Reviews*, vol. 7, no. 3, pp. 923–931, 2020.
 - [68] H. X. Huynh, B. Q. Truong, K. T. Nguyen Thanh, and D. Q. Truong, “Plant Plant Identification Using New Architecture Convolutional Neural Networks Combine with Replacing the Red of Color Channel Image by Vein Morphology Leafidentification using new architecture convolutional neural networks combine with replacing the red of color channel image by vein morphology leaf,” *Vietnam Journal of Computer Science*, vol. 07, no. 02, pp. 197–208, Feb 2020.
 - [69] J. Banzi and T. Abayo, “Plant species identification from leaf images using deep learning models (CNN-LSTM architecture),” *Tanzania Journal of Forestry and Nature Conservation*, vol. 90, no. 3, pp. 93–103, 2021.
 - [70] T. Karahan and V. Nabiyeve, “Plant identification with convolutional neural networks and transfer learning,” *Pamukkale University Journal of Engineering Sciences*, vol. 27, no. 5, pp. 638–645, 2021.
 - [71] A. Pravin and C. Deepa, “A identification of piper plant species based on deep learning networks,” *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 6740–6749, 2021.
 - [72] Y. Chung, C. A. Chou, and C. Y. Li, “Central attention and a dual path convolutional neural network in real-world tree species recognition,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 3, pp. 961–29, 2021.
 - [73] E. Adetiba, O. T. Ajayi, J. R. Kala et al., “LeafsnapNet: LeafsnapNet: An Experimentally Evolved Deep Learning Model for Recognition of Plant Species based on Leafsnap Image Dataset,” *Journal of Computer Science*, vol. 17, no. 3, pp. 349–363, 2021.
 - [74] T. Quoc Bao, N. T. Tan Kiet, T. Quoc Dinh, and H. X. Hiep, “Plant species identification from leaf patterns using histogram of oriented gradients feature space and convolution neural networks,” *Journal of Information and Telecommunication*, vol. 4, no. 2, pp. 140–150, Sep. 2020.
 - [75] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, “Plant identification using deep neural networks via optimization of transfer learning parameters,” *Neurocomputing*, vol. 235, pp. 228–235, April 2016.
 - [76] J. Krause, G. Sugita, K. Baek, and L. Lim, “WTPlant (What’s That Plant?) A Deep Learning System for Identifying Plants in Natural Images,” in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 517–520, June 2018.
 - [77] M. Sulc and J. Matas, *Texture-Based Leaf Identification*, *Computer Vision - ECCV 2014 Workshops*, Springer International Publishing, Midtown Manhattan, New York City, pp. 185–200, 2015.
 - [78] C. Zhang, P. Zhou, C. Li, and L. Liu, “A Convolutional Neural Network for Leaves Recognition Using Data Augmentation,” in *Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing Pervasive Intelligence and Computing*, Liverpool, UK, December 2015.
 - [79] P. Pawara, E. Okafor, L. Schomaker, and M. Wiering, *Data Augmentation for Plant Classification*, *Advanced Concepts For Intelligent Vision Systems*, Springer International Publishing, Midtown Manhattan, New York City, pp. 615–626, 2017.
 - [80] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, “LeafNet: A computer vision system for automatic plant species identification,” *Ecological Informatics*, vol. 40, pp. 50–56, 2017.
 - [81] S. Anubha Pearline, V. Sathiesh Kumar, and S. Harini, “A study on plant recognition using conventional image

- processing and deep learning approaches," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, p. 1997, 2004.
- [82] C. P. Blesslin Elizabeth and S. Baulkani, "Novel Network for Medicinal Leaves Identification," *IETE Journal of Research*, pp. 1–11, 2022.
- [83] S. O. Oppong, "Ghanaian Leaf Dataset," *Textural Analysis for Medicinal Plants Identification Using Log Gabor Filters*, 2022.
- [84] A. Sarkar, Y. Yang, and M. Vihinen, "Variation benchmark datasets: update, criteria, quality and applications," *Database*, p. baz117, Jan, 2020.
- [85] R. S. Olson, W. la Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "PMLB: a large benchmark suite for machine learning evaluation and comparison," *BioData Mining*, vol. 10, no. 1, p. 36, 2017.
- [86] S. G. Wu, F. S. Bao, E. Y. Xu, Y. X. Wang, Y. F. Chang, and Q. L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology*, pp. 11–16, Giza, Egypt, December 2007.
- [87] Y. Zhang, J. Cui, Z. Wang, J. Kang, and Y. Min, "Leaf image recognition based on bag of features," *Applied Sciences*, vol. 10, pp. 5177–15, 2020.
- [88] O. J. O. Söderkvist, "Computer Vision Classification of Leaves from Swedish Trees," *Computer Vision*, Department of Electrical Engineering Linköping University, 2001.
- [89] S. Kaur and P. Kaur, "Plant Species Identification based on Plant Leaf Using Computer Vision and Machine Learning Techniques," *Journal of Multimedia Information System*, vol. 6, no. 2, pp. 49–60, 2019.
- [90] T. Munisami, M. Ramsurn, S. Kishnah, and S. Pudaruth, "Plant Leaf Recognition Using Shape Features and Colour Histogram with K-nearest Neighbour Classifiers," *Procedia Computer Science*, vol. 58, pp. 740–747, 2015.
- [91] S. Roopashree and J. Anitha, *Medicinal Leaf Dataset*, Mendeley Data, 2020.
- [92] F. Zhuang, Z. Qi, K. Duan et al., "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.
- [93] R. Panigrahi, S. Borah, AK. Bhoi et al., "Performance assessment of supervised classifiers for designing intrusion detection systems: A comprehensive review and recommendations for future research," *Mathematics*, vol. 9, no. 6, p. 690, 2021.
- [94] C. R. Stephens, H. F. Huerta, and A. R. Linares, "When is the Naive Bayes approximation not so naive?" *Machine Learning*, vol. 107, no. 2, pp. 397–441, 2018.
- [95] C. Kern, T. Klausch, and F. Kreuter, "Tree-based Machine Learning Methods for Survey Research," *Surv Res Methods*, vol. 13, no. 1, pp. 73–93, Apr. 2019.
- [96] F. Herrera, S. Ventura, R. Bello et al., "Instance-Based Classification Methods," in *Multiple Instance Learning*, pp. 67–98, Springer, Berlin, Germany, 2016.
- [97] D. R. Amancio, CH. Comin, D. Casanova et al., "A systematic comparison of supervised classifiers," *PLoS One*, vol. 9, no. 4, 2014.
- [98] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger, "Convolutional Networks with Dense Connectivity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2019.
- [99] K. Kanagalakshmi and E. Chandra, "Frequency domain enhancement algorithm based on log-gabor filter in FFT domain," *European Journal of Scientific Research*, vol. 74, no. 4, pp. 563–573, 2012.
- [100] F. Twum, Y. M. Missah, S. O. Oppong, and N. Ussiph, "Textural Analysis for Medicinal Plants Identification Using Log Gabor Filters," *IEEE Access*, vol. 10, 2022.
- [101] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021.
- [102] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and Electronics in Agriculture*, vol. 158, pp. 20–29, 2019.
- [103] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep Learning for Plant Identification in Natural Environment," *Computational Intelligence and Neuroscience*, vol. 2017, 2017.