IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

EMPOWERING HERBAL KNOWLEDGE: A DEEP LEARNING APPROACH FOR ACCURATE AND EFFICIENT IDENTIFICATION OF MEDICINAL PLANTS

Ms. S. AGNES JOSHY S1
S1 Assistant Professor, Department of Information Technology
Francis Xavier Engineering College
Tirunelveli, India

Mr. ANBU SIRANJEEVI.S S2
S2 PG Scholar, Department of Information Technology
Francis Xavier Engineering College
Tirunelveli, India

Ms..D.Aveline Sarah S3
S3 Assitant Professor,Department of Electronics and Commuication Engineering
Government college of Engineering
Tirunelveli,India

Abstract

Traditional Indian Ayurvedic medicine. The raw materials from which Ayurvedic medicines are made in nature are mostly herbs and minerals. India also has many homemade herbal remedies for common ailments. This knowledge was passed down from generation to generation in large joint families. This knowledge is slowly fading away from the current generation of nuclear families. The current generation cannot identify even locally available plants. To date, the total number of plant species on Earth has been identified as almost four hundred thousand. With so many plant species, an intelligent system is needed to identify the plant species. The leaf is one of the most important and visible parts of the plant and is available throughout the year. Leaves are important in plant identification. Plant Leaf Classification (PLC) is the process of automatically identifying plant species based on an image of a plant leaf. This field of research was found to be emerging and has connections with computer science and plant science. Many researchers have worked in this area of PLC using image processing, feature extraction and machine learning techniques. This thesis mainly focuses on the classification of leaf images of Ayurvedic plants available in and around Visakhapatnam. Visakhapatnam is a popular city in the Indian state of Andhra Pradesh. The main purpose of this thesis is to identify some leaves that are important in Ayurveda (a very old traditional medicine). Only the leaf tag can be extended to any leaf of the plant. Convolutional Neural Networks (CNN) were used to solve the Ayurvedic Plant Leaf Classification (APLC) problem of this thesis. APLC is designed to identify seven locally available Ayurvedic plants/trees. The seven Ayurvedic plants/trees are Hibiscus Rosa Sinensis (Mandara),

Tabernaemontana Divaricata (Nadhivardanam), Ocimum Tenuiflorum (Tulasi), Ficus Religiosa (Raavi or Peepul), Syzygium Cumini (Neredu) Eucalyptus and Catharanthus Roseu (B). Since seven different Ayurvedic plants/trees can be classified based on leaf images, it is a multi-class classification problem. A CNN architecture consisting of an input layer, three convolutional layers, three max-connection layers, one smoothing layer, one dense layer, one rectified linear unit (ReLU) activation layer, one output layer, dense vi output layer and Softmax activation. layer was built This CNN architecture called "AYURNet". The name "AYURNet" is created by extracting the first four letters "AYUR" from Ayurveda and the last three letters "Network" from Convolutional Neural Networks. This CNN architecture had to be trained based on images from the respective Ayurvedic journals. Images of Ayurvedic plant leaves were not readily available. Therefore, the collection and creation of magazine photo material had to be carried out as part of this thesis. Creating an image database for an Ayurvedic journal was a labor-intensive and time-consuming task. Due to the hot and humid environment of Visakhapatnam, the problem was that the leaves wilted quickly when picked. That is why the leaves had to be photographed from the outside, so that they would not wither and fall off. It is well known that CNN needs a large training image dataset to build a predictive CNN model with high accuracy. Since the amount of plant leaf data needed to build a CNN model did not provide guidance for highly accurate classification of Ayurvedic plant leaves, experiments were conducted as part of this thesis to determine the optimal dataset. Specified size. The Keras deep learning framework was used as the Tensor Flow backend to build the CNN model. The freely available Google Colab platform was used to train this model. The images of the magazine were saved in a specific folder in Google Drive. Graphics processing unit (GPU) hardware acceleration was used to train the CNN model. In addition to the GPU, the Google Colab platform has a more powerful and advanced hardware called the Tensor Processing Unit (TPU). You can train much faster with TPU. There is a problem with TPU training. The current Tensor Flow documented method for training with TPU requires storing image data in TF Record format to Google's cloud storage devices. Google's cloud storage packages are proprietary and charge a dollar for delivery. This thesis addressed this topic. A method and program have been developed to download data from shared storage such as Google Drive to TPU. TPU hardware acceleration was then used to train the CNN model. The results obtained in several stages of the experimental process were found to be satisfactory, and the proposed method can be extended or used for the classification of each plant leaf.

I. INTRODUCTION

A. Importance of Leaf Classification of Ayurvedic Plants

Ayurveda is an ancient Indian medicine used to treat various diseases. Extracts of medicinal plants available locally are used to prepare various Ayurvedic medicines. Various minerals are also used in the preparation of Ayurvedic medicines. Traditional Indian herbs and plants such as neem and turmeric are also patented. The patent disputes were long and finally it was concluded that the medicinal properties of these herbs and plants were already known and they belong to traditional Indian medicine. With proper scientific research of several other Ayurvedic herbs, more and more proper scientific reviews and documentation of their medicinal properties are appearing. It is imperative to apply modern scientific methods to this ancient Ayurvedic system of medicine. One such application of modern technology is the identification of Ayurvedic plants. Ayurvedic plants and trees are available both locally and in forests. To the untrained eye, it is difficult to identify plant species just by looking at them. The leaves of the plant are one of its most important and distinctive features and are available throughout the year. Using computer vision and artificial intelligence techniques, plant species can be identified based on their leaf image. Until recently, image processing and machine learning techniques were mainly used to classify plant leaf images. Image processing techniques were used to extract foliage features. These extracted features were provided as input to several ML algorithms. These ML algorithms then classify the pages after training. Programmers are not always able to come up with clear patterns and functions that can be fed into computer programs to classify non-geometric patterns such as sheets. Convolutional Neural Networks (CNNs) fit this problem because they can automatically come up with models that recognize certain patterns and features that humans may not understand or interpret, but are very good at classifying images..

B. A Brief Introduction to Deep Learning

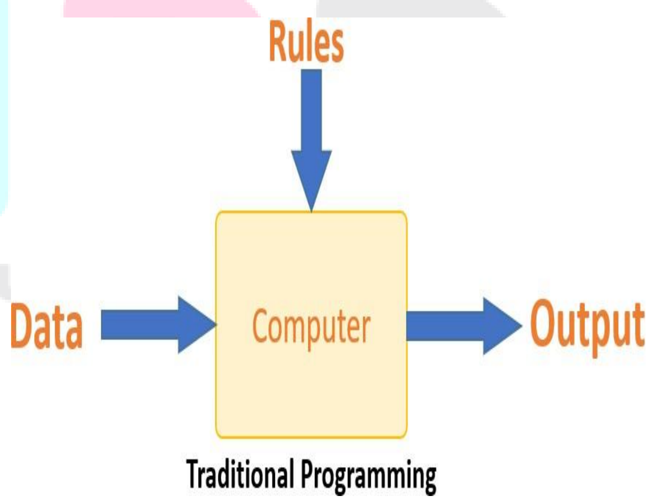
DL is a subfield of ML, which in turn is a subfield of artificial intelligence (AI). DL consists of algorithms that mimic the behavior of the human brain. An artificial neural network (ANN) is used to implement DL. Several artificial neurons are connected to form an ANN. Each neuron in the network acts as an information processing unit. Information is passed between neurons in the network. Each neuron in the network is responsible for receiving and processing inputs. Weight is assigned to links between nodes. At each node, a weighted sum of the input is calculated and an activation function is applied to it to form the output. Thanks to the activation function, ANN can handle complex patterns. The result produced by one neuron is passed as input to the next neuron. These layers of neurons are connected to form neural networks. There are two types of neural networks called feedforward neural networks and feedback neural networks. In feedforward neural networks, the connections between networks do not form a chain. In feedforward neural networks, the connections between networks form loops. A single neuron in a neural network is called a perceptron and is the basic building block of DL. Multiple perceptron layers are combined to create a DNN. DL models are trained using an algorithm called Backpropagation.

C. WHAT IS MACHINE LEARNING

Machine learning is a system of computer algorithms that can learn from an example through self-development without being separately coded by a programmer. Machine learning is a branch of artificial intelligence that combines data with statistical tools to predict results that can be used to generate useful knowledge. Success is based on the idea that a machine can learn from data (ie example) to produce accurate results. . Machine learning is closely related to data mining and Bayesian predictive modeling. The machine takes data as input and uses an algorithm to formulate answers. A typical machine learning task is to make recommendations. For those with a Netflix account, all movie or series recommendations are based on the user's historical data. Tech companies are using unsupervised learning to improve user experience with personalization recommendations. Machine learning is also used for various tasks such as fraud detection, predictive maintenance, portfolio optimization, task automation, etc.

D. MAXINALLERNO vs. TRADITIONAL PROGRAMMING

Traditional programming is different from machine learning. In traditional programming, the programmer codes all the rules in collaboration with an expert in the field for which the software is being developed. Each rule is based on a logical basis; the machine executes the output after the logic statement. As the system becomes more complex, more rules must be written. Maintaining this can quickly become unsustainable.

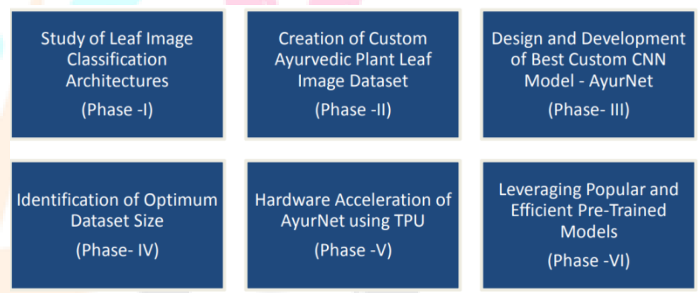


E. Motive and Problem Statement

The Ayurvedic system of medicine in India needs to be more relevant to the present era using the latest available technology. Classification of Ayurvedic plants is usually a task that only experts in Ayurveda and botany can do. Leaves are one of the most visible parts of a plant and have different characteristics such as shape, size, color and texture. Identifying a leaf helps identify the plant species it belongs to. Plant leaf recognition based on a leaf image is a multi-class classification problem. The aim of this study is to design a custom CNN model for multi-class classification of Ayurvedic plant leaf images. The goal is to design a CNN model that is lightweight and can classify with a very high accuracy of more than 99%. The goal is also to find an optimal data set that is sufficient to train a CNN model to achieve a very high accuracy of more than 99%. Another goal is to use GPU and TPUs for hardware acceleration of CNN model training. It aims to use pre-trained image recognition models such as DenseNet201,

F. Research

Research Objectives (RO) and Research Contributions (RC) are stated as follows:\ nRO1- Conduct thorough research. an overview of the different architectures available for multi-class classification of plant leaf images.RC1: - A critical literature review of different plant leaf extraction concepts, ML methods and PLC-related CNN methods is presented. Focus closely on multi-class classification of images using CNN.RO2 - Create a custom dataset based on locally available leaf images of Ayurvedic plants in and around Visakhapatnam.RC2: - There was not enough dataset of Ayurvedic plant leaf images in our area. Information needed for CNN training.number of plant leaves. A modified dataset for seven different Ayurvedic plant leaf species was created from locally available images of Ayurvedic plant leaves. A thousand separate leaf images were photographed for each leaf type of the Ayurvedic plant. In total, a custom Ayurvedic plant image dataset containing 7000 different plant leaf images was created.RO3- Create a custom CNN architecture and model that can classify Ayurvedic plant leaf images with a high accuracy of more than 99%1.7 Proposed Method This thesis aims to automatically. classify plant types and divination in Ayurvedic language. tongue plant leaf images of seven different Ayurvedic plants. A custom Ayurvedic plant image dataset consisting of seven leaf images of different Ayurvedic plant species is used in the design, training and development of the new CNN model AYURNet. Ayurvedic plant leaf image dataset was not found in any publicly available open dataset repository. Outsourcing this magazine image data acquisition is expensive. Therefore, the creation of the image material of the magazine takes place within the framework of this research. The motivation behind this work is to develop an easy-to-use CNN application capable of classifying Ayurvedic journal images taken with a standard smartphone camera by a common user without special photography skills. Therefore, a leaf image dataset is created by photographing plant leaves with a personal smartphone camera. All the work done in this study is illustrated in Fig.



The six stages of this research

II. RELATED WORKS

This section reviews the PLC literature published between 2016 and 2020 by Nisar Ahmed et al. In 2016, [69] proposed an automatic leaf-based plant detection system. For this purpose, they used the Flavia dataset, which consisted of a sample of 1,900 leaf images of 32 plant species. Some important steps of the approach including segmentation (grayscale conversion, binaries using Otsu's method, convolution with a 3x3 kernel), feature extraction of 15 different shapes, feature normalization, dimensionality reduction using Principal Component Analysis (PCA), and classification using multiple. . classes Support vector machine (MSVM). This method achieved an overall accuracy of 87.40%. Pushpa BR et al. In 2016, [70] proposed a method to identify Ayurvedic plant species based on several statistical parameters taken from leaf images. For this, they used 208 sample leaf images of 26 different species. The leaf factor was calculated using statistical parameters and stored in the database. To classify the new images, the leaf factor was calculated and compared with the leaf factors of different species already stored in the database. The new image was classified among the species to which the author of the magazine reacted the most. This method achieved a leaf detection accuracy of 93.7%. CH Arun et al. In 2017, [72] presented a medicinal plant identification system. The stages of system development are color conversion, texture feature extraction and classification. Leaf textural features were calculated using grayscale statistical spatial dependence matrix (GTSDM) and local binary pattern (LBP). The authors used five different classifications, of which the Quadratic Discriminant Analysis (QDA) classification gave the best result. A dataset of 250 images was created from 5 different species and 50 images per species. This method achieved a detection rate of 98.7%. In 2018, Jiachun Liu et al [73] used a convolutional neural network (CNN) to classify plants based on leaf images. The initial stage of their work involves the collection of footage and the preparation of footage. The Flavia database was used to conduct the experiments. Image data preparation included image processing and data addition. The images in the magazine were rotated 90 degrees and 180 degrees counterclockwise to increase the information. A multilayer CNN model with 10 layers was designed for classification. This CNN model achieved 87.92 percent accuracy. In this work, the results of the proposed model were compared with traditional methods. Anh H. Wo et al. In 2019 [75] proposed the use of a convolutional neural network (CNN) for image extraction.2020. in Raja Naga Lochan et al [82] used a Regional Convolutional Neural Network (RCNN) for plant identification based on leaves. RCNN using fast convolutional networks was used for feature extraction and support vector machine for classification. This method gave an accuracy of 96.2%. A custom magazine image dataset consisting of 10 classes and 10,000 magazine images was used. The course had 850 training images and 150 test images per class.

A. SUMMARY

Many researchers have worked in this field of plant leaf recognition (PLR) using image processing, feature extraction. , machine learning and convolutional neural network techniques. This section provides an overview of the gaps in the literature and describes how to address the problem. For multi-class classification problems in machine learning, classifiers are trained based on known input data and the relationships between the corresponding classes. The trained classifiers are then used to predict the class for new observations or inputs. In the PLR machine learning model, the learning material consists of leaf images of plants and similar plant species. In the training of classifiers, features extracted from plant leaf images discussed in the study material and the corresponding species names of plant leaves are used. Some of the classifiers used in the PLR machine learning approach are artificial neural network (ANN), decision tree classifier, moving average centroid hypersphere classifier, naive Bayesian classifier, nearest neighbor classifier, probabilistic neural network (PNN), random forest classifier, and support. Vector Machine (SVM). The proposed solution uses a Convolutional Neural Network (CNN). It is a more automated approach to plant leaf image classification that requires much less pre-processing and does not require special manual feature extraction. The pixel data of the plant leaf image is input through the input layer. Convolutional levels act as filters that extract features from input images and create feature maps. Blending layers are used to reduce the size of object summary and map feature maps. Although Max Pooling functions are often used in CNN, there are other common functions such as Average Pooling and Global Pooling. A smoothing layer is added between the convolutional layers and the fully connected density layers. It converts the output data of the convolution layer into a 1-dimensional array which is fed to 81 fully connected density layers. The building blocks of dense layers are artificial neurons, which are mathematical functions that compute weighted aggregates of inputs. Since neural networks are used to solve problems involving data that are not linearly separable, nonlinearity is used in activation functions such as Sigmoid, Relu, and Tanh. Drop layers can also be applied to avoid overfitting. They prevent the neural network from remembering the training information. After one or more hidden dense layers, the final layer is the printed layer. This layer is used to predict the leaf class of the plant. The output layer uses sigmoid or softmax functions. The sigmoid function is sufficient to solve the binary classification problem. Since PLR is a multi-class classification problem, the softmax function is used. The CNN model selected during training processes the image data of the study magazine and predicts the name of the magazine. This is compared to the actual name of the leaf species. The difference between the actual value and the predicted value is calculated using a loss function. This information is used to refine the network to get a better prediction. One complete round of processing the training sets determined by the CNN during training is called an epoch. Due to the gradient descent and backpropagation concepts used in CNN training, the loss gradually decreases and the accuracy increases gradually with the number of epochs. During the training process, the CNN itself internally learns features that help distinguish magazine-type images from others. That is why these traits are called learned traits. Training is stopped when sufficient prediction accuracy is achieved and no significant improvement in accuracy is observed with new time steps. The CNN architecture described in that section is a simple sequential model. Much more complex and deep CNN architectures can also be designed and used if necessary.

III. METHODS

In this chapter, a custom CNN architecture for APLC and an innovative AYUR-Best model are proposed to achieve the highest classification accuracy. For the exam, a thorough analysis of the results is done based on the information obtained from the personal journals. Classification of plant species is important so that the benefits offered by each species can be fully exploited. Due to the large number of plant species, the classification of plant species requires knowledge and expertise. An expert botanist knows how to classify plant species based on morphological characteristics. Manual plant classification methods are time-consuming and require special knowledge. Classification of plant species in magazine photos has become an active area of research. Advances in image processing and artificial intelligence technology allow solving the complex APLC problem. CNNs have gained popularity over the past 10 years due to the availability of supporting hardware and software platforms. The problem chosen in this work is the identification of medicinal plant species by classifying their corresponding leaf images. The plants and trees chosen for the work presented in this work are Catharanthus Roseus, Eucalyptus, Ficus Religiosa, Hibiscus Rosa Sinensis, Syzygium Cumini, Tabernaemontana Divaricata. It is not always possible for humans to come up with clear patterns and features that can be fed into computer programs to classify non-geometric shapes such as magazine images. CNNs are well suited to solve this problem because they can come up with models that recognize some patterns and features that humans may not understand or interpret, but can classify images well. First, a large dataset of leaf images of various classified plants was collected. Designed a CNN architecture that resulted in a model capable of classifying selected pages with targeted accuracy.

A. Experimental Platform

In the development of the CNN model, the experimental platform Google Collabory (abbreviated Colab) was used as a computing environment[104]. described in this paper for the APLC problem. Colab is a free cloud computing environment for Jupyter notebooks that does not require special configuration. The main reason to choose Colab is the free graphics card that comes with Colab[105]. For these experiments, the Colab environment provided the following hardware: CPU: Intel(R) Xeon(R)CPU@2.30 GHz GPU: NVIDIA Tesla T4RAM: 12 GtColab does not provide a long-term persistent dataset. Data uploaded by Colab is not expected to persist across sessions. Google Drive was used for permanent storage of 102 visual materials and test sheets. Google Drive offers 15 GB of free storage. Google offers the ability to link Google Drive locally with Google Colab using an authorization code.

B. Pre-processing

The only pre-processing required for this job was to scale the magazines from 4000x3000x3 pixels to 150x150x3 pixels. . size This image scaling can be done as part of the Keras code itself. The only reason to implement this separately in Python on the local system was to reduce the size of images before uploading them to Google Drive. This will increase the upload speed from your local system to Google Drive. It also includes 15GB of free Google Drive storage.

C. Building CNN Models

Newspaper image classification techniques to create CNN ModelsML are based on hand-crafted features. Magazine photos go through several pre-processing steps. Handcrafted features are features that researchers have extracted and derived from magazine images that help the machine distinguish between magazines. These properties were collected from the leaves of several different plants. The page feature data and the corresponding page tags are fed into the corresponding ML classification algorithms and the exercise is performed. So the common steps are preprocessing, feature extraction and classification. Some of the classification algorithms used are MSVM (Multiclass support vector machine) [106] and Random Forest classifier[107]. During training, classification algorithms can predict page labeling based on input data for new page types that were not previously part of the training data. After the model training has been completed and the desired training accuracy has been achieved, the model must be evaluated against the previous one. unseen test data. A model is considered successful only if it achieves a test accuracy equal to the training accuracy. Sometimes much less accuracy can be achieved during the test than during training. This is due to over specification. Overfitting means that the model memorized the training data instead of finding the correct relationship that achieves the same accuracy for all test data. In such cases, it is again necessary to modify the model 106. The goal of CNN modeling is to find the simplest possible network structure that provides the highest possible accuracy for the chosen image classification task. For the APLC task, we designed a CNN model as shown in the figure.

D. Choice of loss function

The idea behind CNN modeling is to find an approximate relationship that is as close to the truth as possible. A complex relationship between input and output. In our case, training the CNN model to classify pages of several categories, the inputs are page images of different known categories and the outputs are the titles of the corresponding page images Page Categories 109. During training, we get the predicted income. For the training dataset, we know the actual result. To evaluate the accuracy of the model, we find the sum of the difference between the actual and predicted performance. This difference is calculated using a loss function. During training, the loss is calculated using the loss function of all the data in the training dataset and summed. It is this loss value that helps us choose a model that better approximates the relationship between input data and output. It is clear that the lower the value of the damage.

E. Learning

Artificial neurons in a neural network consist of inputs, weights, weights and activation functions. After choosing a neural network model to solve the problem, we need to find the weights and bais, which are the defining parameters of the model. The values of these parameters are not known when choosing the model. The best possible parameter values must be found through a learning process. Initially, the selected model parameters are initialized with certain values. The model then processes the inputs and predicts the outputs. During the training process, the inputs are a set of training data whose outputs are known. The predicted output does not match the known output after the first pass through network 110 . The loss function is used to find the difference between the actual known output and the predicted output.

F. Unsupervised learning

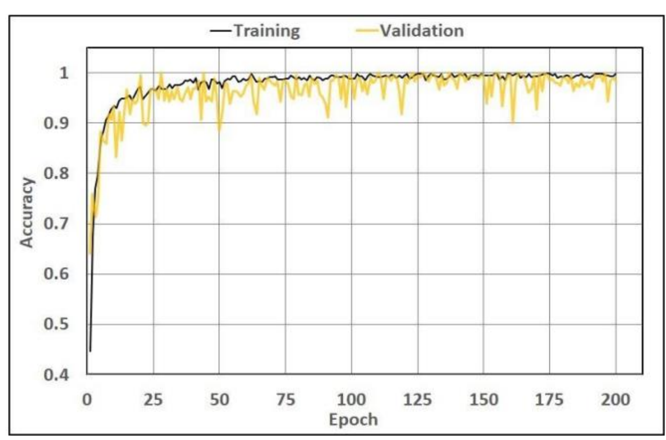
The goal of learning is to minimize the loss so that the parameter values are reached so that the predicted output is as close as possible to the known output. This provides the necessary model that best describes the relationships between input and output.F. Unsupervised Learning In unsupervised learning, an algorithm examines input data without providing an explicit output variable (eg examines customer demographics to identify patterns). You can use it when you don't know how to classify data and you want an algorithm to find patterns and classify data itselfFor learning Algorithm name arrayAlgorithmDescriptionTypeK characters for grouping Assign data to specific groups (k), each of which contains similar data. features (model-defined, not human-specified) Clusters Gaussian mixture model A generalization of Kmeans grouping that provides more flexibility in the size and shape of groups (clusters). rating system Can be used to form a cluster for a cluster loyal customer Recommendation system Help define the information needed to make a recommendation. PCA/T-SNEM clustering is most commonly used to reduce data dimensionality. Algorithms reduce the number of features to 3 or 4 vectors with the highest variance. Dimensional Reduction The table above shows the names of supervised learning algorithms.

G. Evaluation

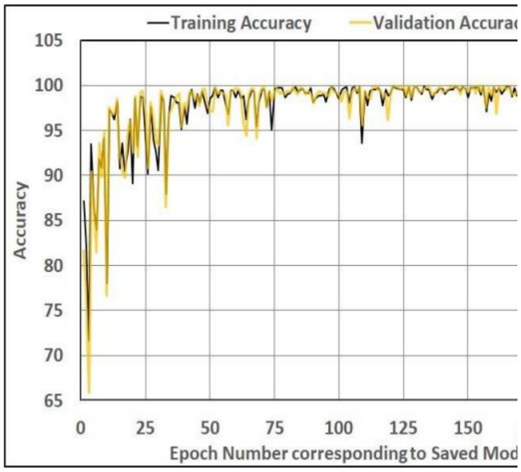
For the paper classification problem described in this paper, the data set is perfectly balanced. Since the dataset was created exclusively for this work, it was ensured that the number of images in the training, validation and test datasets for different journal classes was exactly the same. Since the number of images in different categories is exactly equal, the dataset is perfectly balanced. An advantage of a balanced dataset is that accuracy can be used as a metric when analyzing that model. The advantage of precision as a metric is that it is very easy and intuitive to understand. Both overall accuracy and class-specific accuracy were calculated for that model. The precision calculation formula is given in equation 3-4.

H. Methodology

In this work material from leaf forms of six different locally available Ayurvedic plants was used. 1000 leaf photographs were taken of each of the six plant species. The total volume of the database is therefore 6,000 magazine images. For each image of 1000 plant species, the dataset is divided into training, validation and test data in the ratio of 70:15:15. Keras CNN uses training and validation datasets to build and train models. Magazine images have been resized to 150 x 150 x 3 pixels. Updated magazine images are uploaded to Google Drive in a standard folder, as expected from the proposed Keras CNN program. Keras CNN program is coded in Google colab environment. Google drive is installed locally in the google colab environment so that the image material from the google drive magazine is available to the Keras CNN program. The Keras CNN program is trained with 200 epochs of training data. The training accuracy of the model is calculated for each epoch. After calculating the parameter weights for each epoch, the model is evaluated on the validation dataset and the validation accuracy is calculated. Using the checkpoint provided by Keras, the model parameter weights are saved at the end of each epoch. During training, training loss, training accuracy, validation loss and validation accuracy are monitored at each time step. This way we can see if the training is progressing in the right direction and the overall accuracy improves as the epochs increase. Accuracy and loss curves of training and validation data for different time periods are shown in Figures 5.3 and 5.4. It provides insight into how the trained model performs after each data validation cycle. After running all 200 epochs, we have 200 saved models. We use these saved models to find the best CNN model. We define the AYUR-Best model as a model with high training and validation accuracy and a small difference between training accuracy and validation accuracy. This helps to identify a model that provides high accuracy and does not overfit. Superfitting is a scenario where a model provides high accuracy on the training data set, but does not provide the same accuracy on the validation data set. We created a single bootstrap model program to calculate training accuracy and validation accuracy. The training dataset and the 113 validation dataset are loaded into memory. Saved models are repeatedly loaded one after the other. In each iteration, the corresponding model predicts the class of magazine images in both the training dataset and the validation set. The accuracy of each model is then calculated based on the actual known class and predicted class of the magazine images. So we have 200 true training accuracies and 200 true validation accuracies corresponding to the 200 stored models. These accuracy data are shown in Figure 5.5. We created another python program to determine the AYUR-Best model using the previously mentioned logic of high training accuracy and high validation accuracy and the smallest difference between training accuracy and validation accuracy. This identified AYUR-Best model is then used to find the final accuracy of the proposed CNN model in this study. Test data is used to determine its accuracy. The test material is stored separately and the Keras CNN program has not seen it before during the training. The identified AYUR-Best model is used to predict the leaf class based on the experimental data and calculate the final accuracy. This accuracy provides a good indication of how the CNN model performs in real-world Ayurvedic journal category prediction.



Model Loss



Training Validation Accuracy - Checkpoint Saved Models

I. Summary

In this chapter, a CNN model adapted with APLC and AYURNet approach has been successfully presented. The work was motivated to develop an automated system based on computer vision to identify locally available Ayurvedic plants. In this work, the leaf of the plant part is chosen to identify the plant. The leaves of different plants are distinguished by morphological differences. The leaves are the most readily available parts of the plant and are available throughout the year. Some previous works on the classification of plant leaves are reviewed. We used a Convolutional Neural Network (CNN) to solve the multi-class classification problem of plant leaves of Andhra Ayurvedic plants. The convolutional layer part of the CNN was used for feature extraction, and the fully connected dense layer of the CNN was used for multi-class classification. Using known best practices, a very simple and elegant CNN model was designed and built using Keras to solve the multi-class facility classification problem. The model was trained with a training dataset of 200 epochs. The model was tested on the validation dataset using the weight parameters obtained at each time step. At each epoch, the training accuracy and 120 validation accuracies were compared and the model with the best weight parameters was selected. The logic used to select the AYUR-Best model was high accuracy above the selected 99% threshold and the smallest possible difference between training accuracy and validation accuracy. This ensured a very high accuracy of the model, which ensured that there was no overfitting. The model selected by this method also performed very well with the experimental data and had an accuracy of 99.88%. Similar high accuracy was achieved with popular pre-trained models such as DenseNet169, EfficientNetB6, InceptionResNetV2, ResNet152V2, VGG16, and Exception, but it can be seen that these models are heavy and have many parameters compared to the standard CNN model described in Sect.

IV. CONCLUSIONS AND FUTURE WORK

As part of this study, we proposed a life history dataset of Indian medicinal plants using the IMPINEt network to identify Indian medicinal plants and a new multi-organ approach to Indian medicinal plant research. Identification of plants. We tested our approach on a newly created dataset of field images of Indian Ayurvedic medicinal species and achieved 97.5 percent accuracy. The researchers compared the newly created IMPINet network with the state-of-the-art image classification networks VGG16, VGG19, MobileNet, MobileNet-V2 and Xception and obtained accuracies of 94.36%, 92.56%, 93.96%, 97.63%, 98, 19%. . The newly created IMPINet network was also evaluated against the latest known plant identification data from the Flavia dataset, a sample dataset was created, and IMPINet achieved 99.5 percent accuracy on the Flavia sample dataset. We have not yet achieved expert-level accuracy, and the need to involve more species and researchers in the field of Indian Ayurvedic plant identification offers many opportunities for improvement and further work. The model could be improved to increase accuracy. In addition to medicinal herbs, the applicability of the model could also be explored for non-angiosperm species such as algae and mosses, as they also play an important role in environmental protection. There are several other plant organs such as stem, fruit, and root that can help improve recognition accuracy, and the dataset can be expanded to include such organs. The identification of plant families can be continued. Family-level identification can be made based on similarities and differences in characteristics between species of the same family. More network architectures and topologies and different model configurations could be explored. Methods other than averaging could be explored to combine prices and different thresholds. Based on the hidden (or incomplete) image, one can try to identify and create plant organs. The species can also be determined by dried or infected leaves or flowers..

5. REFERENCE

[1]https://courses.lumenlearning.com/boundless-biology/chapter/leaves/

[2] Neilsen,M.A.(2015).”Neural Networks and Deep Learning”.Determination Press

[3] Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. IEEE Potentials, 13

(4), 27-31. [4] Kak, S.C.(1993). Feedback neural networks: new characteristics and a generalization. Circuits Systems and Signal Process 12, 263–278. https://doi.org/10.1007/BF01189877

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

[6] Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. IEEE transactions on Systems, Man, and Cybernetics, (4), 364-378.

[7] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning Representations by Back-propagating Errors. Nature, 323, 533--536. doi: 10.1038/323533a0

[8] https://owlcation.com/stem/Structure-of-a-Neuron

[9] Turing, A. M. (1950). Computing Machinery and Intelligence. Mind, 59, 433-- 460.

[10] Luger, G. F. (2005). Artificial intelligence: structures and strategies for complex problem solving. Pearson education.

[11] Fitch, F. (1944). Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. Journal of Symbolic Logic, 9(2), 49-50. doi:10.2307/2268029 150

[12] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386-408.

[13] Svozil, D., Kvasnicka, V., & Pospichal, J. (1997). Introduction to multi-layer feed forward neural networks. Chemometrics and intelligent laboratory systems, 39(1), 43-62.

[14] Haykin, S. S. (2009). Neural networks and learning machines/Simon Haykin.

[15] Cybenko, G. (1989) "Approximation by superpositions of a sigmoidal function", Mathematics of Control, Signals, and Systems, 2(4), 303–314. doi:10.1007/BF02551274

[16] P. J. Werbos. (1974) “Beyond regression: new tools for prediction and analysis in the behavioral science”. In: Ph. D. Thesis, Harvard University

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. (1988) “Learning representations by back-propagating errors”. In: Cognitive modeling 5.3, p. 1

[18] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2). Cambridge: MIT press.

[19] Michelucci, U. (2018). Applied Deep Learning. A Case-Based Approach to Understanding Deep Neural Networks.

[20] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[21] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. Neural networks, 12(1), 145-151.

[22] Y. Nesterov, “A method of solving a convex programming problem with convergence rate O(1/k2),” in Soviet Mathematics Doklady, vol. 27, no. 2, 1983, pp. 372–376.

[23] Bottou, L. Online learning and stochastic approximations. On-line Learning in Neural Networks, 17(9):142, 1998 151

[24] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of Machine Learning Research 12.Jul (2011): 2121-2159.

[25] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent

[26] Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. 2014. arXiv:1412.6980v9