MEDICINALPLANTS IDENTIFICATIONUSINGDEEPLEARNING

Ms. S. AGNES JOSHY [S1], Mr. ANBU SIRANJEEVI.S [S2],

[S1] Assistant Professor, Department of Information Technology ,Francis Xavier Engineering College, Tirunelveli, India,
[S2] PG Scholar, Department of IT ,Francis Xavier Engineering College, Tirunelveli, India

Abstract; -Ayurveda is the traditional medicine system of India. The ingredients from which Ayurvedic medicines are made are mostly herbal and mineral in nature. Also, there are many herbal home remedies in India for general ailments. This knowledge has been passed down from generation to generation in large joint families. This knowledge is slowly fading away in the current generation of nuclear families. The current generation is unable to identify even locally available plants. The total number of identified plant species on earth is nearing four hundred thousand as of date. With such a huge number of plant species, there is a need for an intelligent system for plant species recognition. The leaf is one of the most important and prominent parts of a plant and is available throughout the year. Leaf plays a major role in the identification of plants. Plant Leaf Classification (PLC) is the process of automatically recognizing the plant species based on the image of the plant leaf. This area of research is found to be emerging and has its association with computer science and botany. Many researchers have worked in this area of PLC using image processing, feature extraction and machine learning techniques. In this thesis work, the focus area is mainly to classify leaf images of few Ayurvedic plants which are locally available in and around Visakhapatnam. Visakhapatnam is a popular city in the state of Andhra Pradesh, India. In this thesis, the main objective is to identify a few leaves which have their importance in Ayurveda (a very old traditional medical practice). It is only to identify leaves that can be extended further to any plant leaf. Convolutional Neural Networks (CNN) have been used to solve this problem of Ayurvedic Plant Leaf Classification (APLC) in this thesis work. An APLC was built for recognizing the seven Ayurvedic plants/trees available locally. The seven Ayurvedic plants/trees are Ocimum Tenuiflorum (Tulasi), Ficus Religiosa (Raavi or Peepul), Syzygium Cumini (Neredu), Hibiscus Rosa Sinensis (Mandara), Tabernaemontana Divaricata (Nadhivardanam), Eucalyptus and Catharanthus Roseus (Billa Ganneru). Since there are seven different Ayurvedic plants/trees to be classified based on leaf images, this is a multiclass classification problem. A CNN Architecture consisting of an input layer, three convolution layers, three max-pooling layers, one flattening layer, one dense layer, one Rectified Linear Unit (ReLU) activation layer, one dropout layer, an vi output dense layer and a softmax activation layer, was built. This CNN architecture is named 'AYURNet'. The name 'AYURNet' has been created by picking the first four letters 'AYUR' from Ayurveda and the last three letters 'Net' from Convolutional Neural Networks. This CNN Architecture needed to be trained with the respective Ayurvedic leaf images. The Ayurvedic plant leaf image dataset was not readily available. Therefore, the leaf image dataset collection and creation had to be done as a part of this thesis work. The Ayurvedic leaf image dataset creation was a tedious and time-consuming task. Due to the hot and humid environment in the Visakhapatnam location, there was the problem of the quick withering of leaves upon plucking them. So the leaves had to be photographed immediately in the outdoor location lest they withered and wrapped. It is common knowledge that CNN requires a large training image dataset for the creation of the CNN model which can predict with high accuracy. In absence of any guidelines about the size of the plant leaf image dataset required for building the CNN model which could classify the ayurvedic plant leaves with high accuracy, experiments to find out the optimum dataset size were done as a part of this thesis work. Keras deep learning framework with TensorFlow as backend was used for building the CNN model. The platform used for training this model was Google Colab available free of cost. The leaf images were stored on Google Drive in a specific folder structure. Graphics processing unit (GPU) hardware acceleration was employed while training the CNN model. In addition to GPU, more powerful and advanced hardware called Tensor Processing Unit (TPU) is available on the Google Colab platform. Training can be accelerated much more using TPU. There is an

issue with training using TPU. The existing TensorFlow documented method for training using TPU requires the image dataset to be stored in TFRecord format in Google cloud storage buckets. Google cloud storage buckets are proprietary and entail dollar amount for its provisioning. As a part of this thesis work, this problem was addressed. Method and program were developed to load data from normal storage like Google drive to TPU. After that, the TPU hardware acceleration was used for training the CNN model. The results obtained during several phases of the experimentation process were found to be satisfactory and this proposed methodology can be extended or implemented for the classification of every plant leaves

1 . INTRODUCTION

1.1 Significance of Ayurvedic Plant Leaf Classification

Ayurveda is an ancient Indian system of medicine that is used for treating various diseases. Extracts of medicinal plants that are locally available are used for the preparation of various ayurvedic medicines. Various minerals are also used for the preparation of ayurvedic medicines. Traditional Indian herbs and plants such as Neem and Turmeric were also patented. There were prolonged legal patent disputes about the same and finally it was acknowledged that the medicinal properties of these herbs and plants were already known and were a part of the Indian traditional system of medicine. With the proper scientific study of various other ayurvedic medicinal plants, more and more proper scientific assessments and documentation of their medicinal properties are being generated. There is a need for the application of modern scientific methods to this ancient medicinal system of Ayurveda. One such application of modern technology is in recognition of ayurvedic plants. Ayurvedic plants and trees are available locally as well as in forests. It is difficult for the untrained eye to recognize the plant species just by looking at them. Leaves of plants are one of its prominent and distinguishing features and are available throughout the year. Computer vision and artificial intelligence techniques can be used to recognize the plant species based on its leaf image. Till recently, image processing and Machine Learning (ML) techniques were mainly used for plant leaf image classification. Image processing techniques were used for extracting the features of the leaves. These extracted features were fed as input to the various ML algorithms. These ML algorithms would then classify the leaves upon training.

It is not always possible for developers to come up with explicit patterns and features that can be fed as input to computer programs for image classification of nongeometric figures like Leaves. Convolutional Neural Networks (CNN) are a good fit for solving this problem since they can come up automatically with models that can identify some patterns and features which may not be understandable and interpretable by humans but can do a very good job of image classification.

1.2 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. 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. The information is passed among the neurons of the network. Each neuron in the network is responsible for taking input and processing the same. Weights are assigned to the links between nodes. The weighted sum of the input is calculated at each node and an activation function is applied to it, to generate an output. Due to the activation function, ANN will be able to process complex patterns. The output generated from an individual neuron is passed as input to the next neuron. These layers of neurons are connected to form neural networks. Neural networks are of two types named feedforward neural networks and feedback neural networks. In feedforward neural networks, the connections do not form a cycle among the networks. In feedback neural networks, the connections form cycles among the network. A single neuron in a neural network is called a perceptron and acts as the fundamental building unit of DL. Multiple layers of perceptrons are combined to generate a DNN. The DL models are trained using an algorithm called Backpropagation.

1.3 Motivation and Problem Statement

There is a need to make the Indian ayurvedic system of medicine more relevant to the current era by the usage of the latest available technologies. Classification of ayurvedic plants is normally a task that can be performed by experts in the field of Ayurveda and Botany only. Leaves are one of the most prominent parts of plants with distinct features such as shape, size, color and texture. Identifying a Leaf will help in identifying the plant species to which it belongs. Plant leaf identification based on the image of the leaf is a multiclass classification problem.

The current research aims at designing a custom CNN model for the multiclass classification of ayurvedic plant leaves images. The goal is to design a CNN model that is lightweight and which can classify with very high accuracy of above 99%. The goal also is to find the optimum dataset size, which is enough for training the CNN model to achieve very high accuracy of above 99%. Another goal is to use Graphic Processing Units and TPUs for hardware acceleration of the training of the CNN model. This is to leverage pre-trained image recognition models like DenseNet201,

1.4 Research Objectives and Contributions

The research objectives (RO) and research contributions (RC) of the research work are presented as follows:

RO1- To do an in-depth study of various available architectures for multiclass classification of plant leaf images.

RC1:- A critical study of literature on various concepts related to plant leaf feature extraction, ML methods and CNN methods related to PLC is presented. An in-depth concentration is laid on the multiclass classification of images using CNN.

RO2- To create a custom dataset of locally available ayurvedic plant leaf images of ayurvedic plants available in and around Visakhapatnam.

RC2:- No Ayurvedic plant leaf image dataset with a sufficient number of plant leaf images required for CNN training was available in our region. A custom dataset of locally available ayurvedic plant leaf images for seven different ayurvedic plant leaf species was created. One thousand distinct leaf images per each one of the Ayurvedic plant leaf species were photographed. On a whole, a custom ayurvedic plant image dataset of 7000 distinct plant leaf images was created.

RO3- To create a custom CNN architecture and model which can classify the ayurvedic plant leaf images with high accuracy of above 99%

1.5 Proposed Methodology

This dissertation aims to automatically classify and predict the plant species of the ayurvedic plant leaf images of seven different Ayurvedic plants. The custom ayurvedic plant leaf images dataset consisting of leaf images of seven different Ayurvedic plant species is used for the design, training and development of a new CNN model named AYURNet. The Ayurvedic plant leaf image dataset could not be found in any of the publicly available open dataset repositories. Outsourcing this task of leaf image dataset acquisition is a costly affair. Therefore, the creation of the leaf image dataset is done as a part of this research work. The motivation behind this work is to develop an easily usable CNN application that can classify images of ayurvedic leaves photographed using a normal smartphone camera by an average user who does not possess any special photography skills. Therefore, the leaf image dataset is created by photographing the plant leaves using a personal smartphone camera. The entire work done in this research work is depicted in figure 1.1

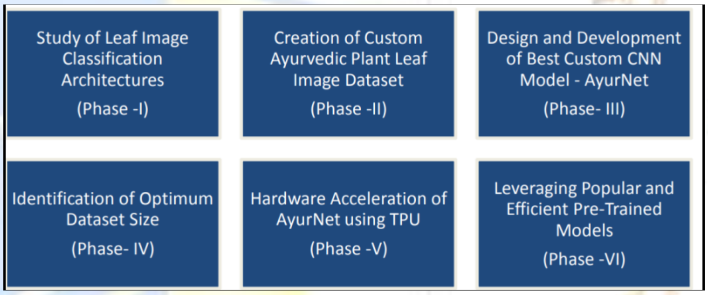

Fig. 1.1 : Six Phases of this Research Work

2.RELATED WORK

This section reviews some of the PLC literature published between the year 2016 and the year 2020

Nisar Ahmed et al. in 2016 [69], proposed an automatic leaf-based plant identification system. For this purpose, they used the Flavia dataset consisting of 1900 leaf image samples of 32 plant species. Some of the important steps in their approach where segmentation (conversion to grayscale, binarization using otsu‟s method, convolution using a 3X3 kernel), extraction of 15 different shape features, feature normalization, dimensionality reduction using principal component analysis (PCA) and classification using Multiclass support vector machine (MSVM). An aggregate accuracy of 87.40% was achieved using this method.

Pushpa BR et al. in 2016 [70], proposed a methodology for Ayurvedic plant species recognition based on various statistical parameters extracted from leaf images. They used 208 sample leaf images of 26 different species for this purpose.The leaf factor was calculated using statistical parameters and stored it in the database. For new images to be classified, the leaf factor was calculated and compared with the leaf factors of various species already stored in the database. The new image was classified as belonging to the species with which the leaf factor matched most closely. Leaf detection accuracy of 93.7% was achieved using this method.

C H Arun et al. in 2017 [72], presented a medicinal plant identification system. The stages in the development of the system are color transform, texture feature extraction, and classification. Texture features of leaves were calculated using statistical, the Grey Tone Spatial Dependency Matrix (GTSDM), and the Local Binary Pattern (LBP). The authors have used 5 different classifiers among which the Quadratic Discriminant Analysis(QDA) classifier gave the best performance. A dataset of 250 images was formed from 5 different species with 50 images per species. An identification rate of 98.7% was achieved using this method.

Jiachun Liu et al in 2018 [73], used Convolutional Neural Network (CNN) for the classification of plants based on leaf images. The initial steps in their work are image dataset collection and image dataset preparation. Flavia leaf database was used to perform experiments. The image dataset preparation included image preprocessing and data augmentation. Data augmentation was done using the rotation of leaf images by 90 degrees and 180 degrees anticlockwise to increase data. For classification, a multilayered CNN model consisting of 10 layers was designed. An accuracy of 87.92% was achieved using this CNN model. The results of the proposed model were compared with traditional methods in this work.

Anh H. Vo et al. in 2019 [75], proposed to use the convolutional neural network (CNN) for image feature extraction for Vietnamese herbal plants. A modified VGG16 network was used for this purpose. They then did a comparative analysis of the recognition rate achieved by applying seven different classifiers over the extracted features. The classifiers used were Random forests, Support Vector Machine (SVM), Logistic Regression, Extreme gradient boosting, Adaboost, K-nearest neighbors, and Light gradient boosting machine (LightGBM). It was observed that the LightGBM classifier outperformed all the other classifiers by providing a recognition rate of 93.6%

Mohammad Aminul Islam in 2019 [78], proposed an automatic plant detection based on the identification of plant leaf images. For their work, they used the Flavia leaf dataset consisting of 1907 images of 32 plant species. The steps used in their approach were image segmentation, image normalization, feature extraction, and classification. As a part of feature extraction, they combined the features extracted using two methods named Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP). Then they did the classification using the Multiclass support vector machine (MSVM). Overall detection accuracy of 91.25% was achieved

Muammer Turkoglu et al in 2019 [79], proposed a leaf recognition system. The steps involved in this method are image Pre-processing, leaf divide and rotation, feature extraction, data normalization, and classification. The features were separately acquired from each piece of the leaf after dividing the leaf image into two and four pieces. A feature vector that describes the entire leaf image was generated by combining these features. Feature extraction was done using Color Features, Fourier Descriptors, Vein Features, and the GLCM method. Extreme Learning Machines 79 (ELM) classifier was used for classification. Flavia leaf dataset was used for the experiment. An accuracy of 99.10% was achieved using this method.

Oktaviani Oktaviani et al in 2019 [80], developed a plant identification system based on leaf shapes. The steps involved in this implementation are image Pre-processing, feature extraction, and leaf image identification. The authors used 16 basic leaf shapes based on 5 geometric features and 7 digital morphological features of leaves in this work. Multiclass Support vector machine (SVM) was used for classification. The authors used three publicly available leaf datasets named Flavia, Folio, and Swedish leaf datasets for research. 2540 leaf images belonging to 56 plant species were used for this experiment. An accuracy of 98.48% was achieved using this method.

Raja Naga Lochan et al in 2020 [82], used Regional Convolution Neural Network (RCNN) for plant identification based on leaf features. Fast RCNN using convolution networks was used for feature extraction and a support vector machine was used for classification. An accuracy of 96.2% was obtained using this method. A custom leaf image dataset consisting of 10 classes and 10000 leaf images was used. The number of training images per class was 850 and the number of testing images per class was 150.

2.2 SUMMARY

Many researchers have worked in this area of Plant Leaf Recognition(PLR) using image processing, feature extraction, machine learning, and convolution neural network techniques. This section gives an overview of gaps in the literature with an emphasis on the how the problem is tackled. In multiclass classification problems in machine learning, classifiers are trained on the known relationships between the input data and respective classes. The trained classifiers are then used to predict the class for new observations or inputs. In the machine learning approach to PLR, the training dataset consists of plant leaf images and the respective plant species names. The classifiers are trained using the features extracted from the processed plant leaf images in the training dataset and the respective plant leaf species names. Some of the classifiers used in the machine learning based approach to PLR are Artificial Neural Network (ANN), Decision Tree Classifier, Moving Median Center Hypersphere Classifier, Naive Bayes Classifier, Nearest Neighbor Classifier, Probabilistic Neural Network (PNN), Random Forest Classifier and Support Vector Machine (SVM). The proposed solution uses Convolutional Neural Network(CNN). It is a more automated approach to plant leaf image classification which requires very less pre-processing and does not require any explicit handcrafted feature extraction. The plant leaf image pixel information is fed through the input layer. Convolution layers act as filters that extract features from the input images and generate feature maps. To summarize the features and reduce the dimension of the feature maps, pooling layers are used. Though Max Pooling operations are frequently used in CNN, there are other pooling operations like Average Pooling and Global Pooling too. The flattening layer is added between the convolution layers and the fully connected dense layers. It converts the output data from the convolution layer into a 1-dimesional array so that it can be input 81 to a fully connected dense layer. The building blocks for dense layers are artificial neurons which are mathematical functions that calculate the weighted aggregates of inputs. Since neural networks are used to solve problems which involve data which is not linearly separable, non-linearity is introduced using activation functions like Sigmoid, Relu and Tanh. Dropout layers too may be introduced for the purpose of preventing overfitting. They prevent the neural network from memorizing the training data. After one or more hidden dense layers, the final layer is the output layer. This layer is used to predict the plant leaf class. The functions used in the output layer are sigmoid or softmax function. For a binary classification problem, the sigmoid function is sufficient. Since PLR is a multiclass classification problem, softmax function is used. During the training process, the training leaf image data is processed by the chosen CNN model and the leaf species name is predicted. This is compared with the actual leaf species name. The difference between the true value and the predicted value is calculated using the loss function. This information is used to refine the network for better prediction. One full round of processing of complete training set by the CNN during training is known as an epoch. Due to the concepts of Gradient Descent and Backpropagation used during the training of CNN, there is a progressive decrease in loss and a progressive increase in accuracy with the number of epochs. During the training process, the CNN itself internally learns the features which help it in distinguishing images of one leaf species from another. That is why these features are known as learned features. The training is stopped when sufficient accuracy of prediction is achieved and no further significant improvement in accuracy with further epochs is observed. The CNN architecture described in this section is a simple sequential model. Much more complex and deep CNN architectures may also be designed and used as required.

3.METHODOLOGY

In this chapter, a custom CNN architecture for APLC and an innovative AYUR-Best model approach to achieve the highest classification accuracy have been proposed. For experimentation, a comprehensive results analysis takes place using the custom leaf dataset.

Classification of plant species is important to be able to take full advantage of the benefits provided by the respective species. Given the huge number of plant species, the classification of plant species requires knowledge and expertise. An expert botanist has the skill to classify plant species based on morphological characteristics. Manual techniques to classify plants are time-consuming and demand expert knowledge. Classification of plant species based on leaf images has become an active area of research. Due to advances in image processing and artificial intelligence techniques, it is possible to solve the complex problem of APLC.

CNN's have gained popularity for the last 10 years with the availability of supporting hardware and software platforms. The problem chosen in this work is medicinal plant species identification by the classification of respective 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 explicit patterns and features that can be fed as input to computer programs for image classification of nongeometric figures like leaves. CNN's are a good fit for solving this problem since they can come up with models that can identify some patterns and features which may not 101 be understandable and interpretable by humans but can do a good job of image classification. A large data set of images of leaves of various plants to be classified was first collected. A CNN architecture was designed which resulted in a model that was able to classify the chosen leaves with targeted accuracy.

3.2 Experiment Platform

Google Colaboratory (Colab in short)[104] was used as the computing environment for the development of the CNN model for APLC problem described in this work. Colab is a ready-made free Jupyter notebook cloud computing environment that does not require any special configuration. The main reason for choosing Colab is the free GPU[105] access that comes with Colab. At the time of executing these experiments, the Colab environment provided the following hardware:

CPU: Intel(R) Xeon(R)CPU@2.30GHz GPU: NVIDIA Tesla T4
RAM: 12 GB

Colab does not provide long term persistent data storage facility. Data uploaded to colab cannot be expected to persist across sessions. For persistent storage of training 102 and test leaf image datasets, google drive was used. Google drive provides free storage of 15GB. Google provides the facility to mount google drive locally to google colab using an authorization code.

5.3 Pre-Processing

The only preprocessing step required in this work was to rescale the leaf images from 4000x3000x3 pixel size to 150x150x3 pixel size. This image rescaling could have been done as a part of the main Keras code itself. The only reason for implementing it separately in the local system using python was to reduce the size of images before uploading them to google drive. This is to increase the speed of upload from the local system to google drive. This is also keeping in view of the upper limit of 15GB free storage availability in google drive.

3.4 Building the CNN Model

The ML techniques of leaf image classification rely on data from hand-crafted features. The leaf images are run through several pre-processing steps. Hand-crafted features are a set of features extracted and derived from leaf images by researchers, to help the machine differentiate one leaf class from another. This feature data is collected from several leaves belonging to various plants. The leaf feature data and the corresponding leaf labels are fed to relevant ML classification algorithms and training is done. So, the usual steps involved are pre-processing, feature extraction, and classification. Some of the classification algorithms used are the Multiclass support vector machine (MSVM) [106]and the Random Forest classifier[107]. Upon training, the classification algorithms can predict the leaf label based on the input feature data of the new leaves which were previously not part of the training datase

After the training of the model is completed and the desired training accuracy is achieved, the evaluation of the model against previously unseen test data needs to be done. Only when the test accuracy similar to training accuracy is achieved, the model is considered as successful. Sometimes, much less accuracy may be achieved during testing when compared to training. This is due to overfitting. Overfitting means that the model has memorized the training data instead of coming up with a proper relationship that will achieve similar accuracy on any test data. In such cases, changes 106 to the model will again be required. CNN modeling aims to come up with the simplest possible network design which will give the highest possible accuracy for the chosen image classification task. For the APLC task, we designed the CNN model as shown in figure 3.1.

3.5 Choice of Loss Function

The whole idea behind CNN modeling is to find the approximate relationship which is as near as possible to the true complex relationship between input and output. In our case of training the CNN model for leaf classification of multiple classes, the inputs are images of leaves of different known classes and the outputs are the leaf class 109 labels of the respective leaf images. During the training process, we get the predicted output. For the training dataset, we know the true output. To assess the correctness of the model, we find the summation of the difference between the true output and the predicted output. This difference is calculated using a loss function. During the training, the loss is calculated using the loss function for all the data in the training dataset and it is summed up. It is this loss value that helps us choose the model which better approximates the relationship between the input data and output. It is obvious that the lower is the loss value, the better is the model. Some of the loss functions used in neural networks are squared error loss, cross-entropy loss, and KL divergence. For this case of multi-class classification problem of leaves where we are using softmax function as the activation function in the fully connected output layer, the loss function which we chose is the categorical cross-entropy. The formula for calculation of categorical cross-entropy is given in equation 3-3

3.6 Learning

Artificial neurons in the neural network consist of inputs, weights, bais, and activation functions. Once we chose a neural network model to tackle a problem, we need to find the weights and bais which are the parameters that define the model. The values of these parameters are not known at the time of choosing the model. The best possible parameter values need to be found through the process of learning. To start with, the parameters for the chosen model will be initialized with some values. The inputs are then processed by the model and the outputs are predicted. During the training process, the inputs are the training dataset for which the outputs are known. The predicted output would not match with the known output after the first pass 110 through the network. The loss function is used to find the difference between the actual known output and the predicted output. The aim of the learning process to minimize the loss such that we arrive at the parameter values which results in predicted output to be as near to the known output as possible. This will result in the required model which will best describe the relationship between the input and the outpu

3.7 Evaluation

In the leaf classification problem described in this work, the dataset is perfectly balanced. Since the dataset was created exclusively for this work, it was made sure that the count of images in the training, validation, and test dataset of different classes of leaves is exactly equal. Since the counts of images of different classes are exactly equal, the dataset is perfectly balanced. The advantage of a balanced dataset is that accuracy can be used as the metric to evaluate this model. The advantage of accuracy as a metric is that it is very easy and intuitive to understand. Both the overall accuracy as well as per class accuracy have been calculated for this model. The formula for calculation of accuracy is given in equation 3-4.

3.8 Methodology

In this work, the self-created data set of leaf images of six different species of locally available ayurvedic plants are used. For each of the six plant species, 1000 leaf images are taken. The total dataset size is hence 6000 leaf images. For each of the 1000 plant species images, the data set is split into training, validation, and test dataset in the ratio 70:15:15. The training and validation data set are used by our Keras CNN program while building and training the model. The leaf images are 112 rescaled to 150x150x3 pixel size. The rescaled leaf images are uploaded to google drive in the standard folder format as expected by the designed Keras CNN program. The Keras CNN program is coded in the Google colab environment. The google drive is mounted locally to the google colab environment so that the leaf image dataset on the google drive is accessible to the Keras CNN Program. Keras CNN program is trained using the training dataset for 200 epochs. For every epoch, the training accuracy of the model is calculated. With the parameter weights calculated for each epoch, the model is evaluated against the validation dataset and the validation accuracy is calculated. Using the checkpointing capability provided by Keras, the model with parameter weights is saved at the end of each epoch. While the training is in progress, the training loss, training accuracy, validation loss, and validation accuracy for each epoch is monitored. This is to see if the training is progressing in the right direction and to be sure that there is an overall improvement in accuracy with the increase in epochs. The plots for accuracy and losses for the training and validation dataset over the epochs are given in figure 5.3 and figure 5.4. This gives an idea about how the trained model after each epoch, is performing on the validation dataset. After all the 200 epochs have completed execution, we have the 200 saved models. We use these saved models to arrive at the best CNN model among them. We define the AYUR-Best model as the one whose training accuracy as well as the validation accuracy is high and the difference between training accuracy and validation accuracy is low. This helps in identifying the model which gives high accuracy and is also not prone to overfitting. Overfitting is the scenario where the model gives high accuracy for the training data set but fails to provide similar accuracy for the validation data set. We created one load model program for calculating the training accuracy and validation accuracy. The training dataset and the 113 validation dataset is loaded into memory. The saved models are loaded iteratively one after the other. In each iteration, the respective model does the class prediction for the leaf images in both the training dataset and the validation set. The accuracy for each model is then calculated based on the actual known class and predicted class of the leaf images. Thus we have the 200 actual training accuracies and 200 actual validation accuracies corresponding to the 200 saved models. These accuracy details are plotted in figure 5.5. We created another python program that finds out the AYUR-Best model using the earlier mentioned logic of high training accuracy and high validation accuracy and the lowest difference between the training accuracy and validation accuracy. This identified AYUR-Best model is then used to find the final accuracy of the CNN model designed in this research work. The test dataset is used to find this accuracy. The test data set is kept separate and is previously unseen by the Keras CNN program during the training process. The identified AYUR-Best model is used for leaf class prediction of the test data set and the final accuracy is calculated. This accuracy gives a fair representation of how the CNN model will perform in the task of ayurvedic leaf class prediction in the real-world.

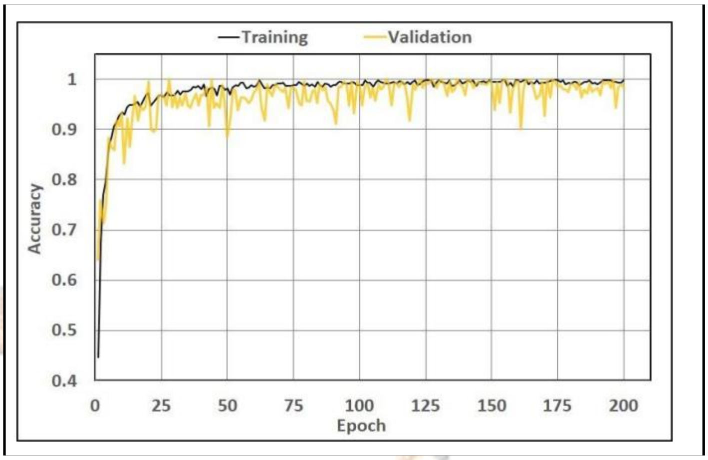

Fig. 3.1: Model Accuracy

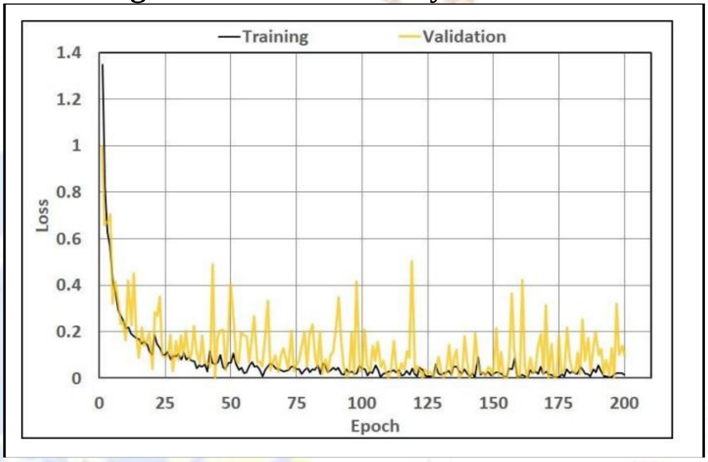

Fig. 3.2: Model Loss

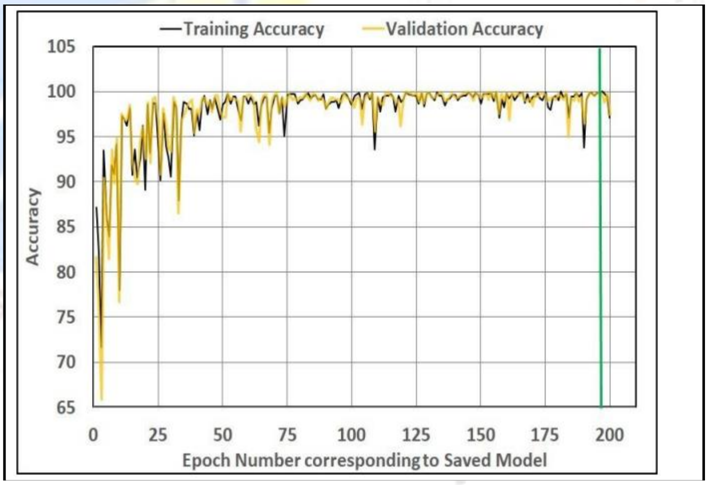

Fig. 3.3: Training Validation Accuracy - Checkpoint Saved Models

3.9 Summary

This chapter has successfully presented the custom CNN model and AYURNet approach for APLC. The motivation behind the work was to come up with an automatic computer vision based system to identify locally available Ayurvedic plants. In this work, the part of the plant chosen for plant identification is the leaf. The leaves of various plants are distinguishable from each other due to morphological differences. Leaves are the most easily available part of the plant and are available throughout the year. Several previous works done about plant leaf classification were reviewed. We used a convolution neural network (CNN) to solve this problem of Multi-Class Classification of Plant Leaves of some Andhra Ayurvedic Plants. The convolution layer part of CNN was used for feature extraction and the fully connected dense layer part of CNN was used for Multiclass Classification. Using the known best practices, a very simple and elegant CNN model was designed and built using Keras to solve the Plant leaf Multi-Class Classification problem. The model was trained on the training dataset for 200 epochs. Using the weight parameters obtained in each epoch, the model was tested against the validation dataset. Training accuracy and 120 Validation accuracy were compared at each epoch and the model with the best weight parameters was chosen. The logic used to choose the AYUR-Best model was high accuracy above the chosen threshold of 99% and the least possible difference between training accuracy and validation accuracy. This was to ensure that the accuracy of the model was very high while ensuring that there was no overfitting. The model chosen using this method performed very well on the test dataset too and it resulted in an accuracy of 99.88%. Similar high accuracies were achieved by leveraging popular pre-trained models like DenseNet169, EfficientNetB6, InceptionResNetV2, ResNet152V2, VGG16 and Xception, but it is seen that the respective models are heavy with a large number of parameters when compared to the custom CNN model described in this work. Due to the small size of the Custom CNN model, it is suitable for the development of the mobile application for Ayurvedic plant species identification based on the respective leaf images.

4.CONCLUSION AND FUTURE WORK

As a part of this research, we proposed dataset of Indian Medicinal Plant organ images along with IMPINet a network for the Indian Medicinal Plant Identification and a novel multi-organ-based approach for the Indian Medicinal Plant Identification. We tested our approach on the newly create dataset of Indian Ayurvedic Medicinal Plant Species field images and obtained 97.5 % accuracy. The researchers compared the newly created IMPINet network with xstate-of-the-art networks in image classification namely VGG16, VGG19, MobileNet, MobileNet-V2 and Xception and obtained 94.36%, 92.56%, 93.96%, 97.63%, 98.19% accuracies respectively. The newly created network IMPINet was also evaluated on state-of-the-art dataset in plant identification known as Flavia dataset, the sample dataset has been created and IMPINet achieved accuracy of 99.5% on sample Flavia dataset.

Still, we have to achieve expert level accuracy and need to include more species and there are many opportunities for improvements and future work for researchers in the field of Indian Ayurvedic plant identification. A model could be improved for the higher accuracy. Apart from the medicinal plants the model’s usability could be explored for the non-angiosperm species like algae and mosses as they also play vital role in maintaining the environment. There are various other plant organs like stem, fruits and roots, which could aid in to better identification accuracy, the dataset could be extended to incorporate such organs. The work can be extended for the plant family identification. The family level identification could be done from the feature similarity and differences within the species of the same family. More network architectures and topologies along with different model ensembling could be explored. The techniques other than the averaging for the fusion of the scores and different threshold values could be explored. Plant organ identification and generation from occulted (or incomplete) image could be experimented. The species identification based on dried or infected leaves or flowers could also be carried out.

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