Proposed model

We have trained the 4 models- CNN, VGG16, InceptionV3 and DenseNet which are described in methodology chapter to detect the weeds on a unique dataset of our own which contains more than 4,300 digital images of different 11 weeds. The algorithms will detect the images using image classification technique. To train the algorithms, we first preprocessed the data or images so that the models can be trained fast and accurately. Then we trained the models.

3.1 Data Preprocessing

As per number of weeds we created 11 folders and named them after the scientific name of the weeds. Images are categorized according to the name of the weeds using categorical variable (e.g., Commelina benghalensis folder contains pictures of Commelina benghalensis weed).

Since the size of each image is not same- every image has different size, we stated the image size: 150×150. Again every image's pixel range is from 0 to 255, this range is too large for numerical computation and the system is also unable to be trained well. So, we stored the image data into an array and the array was reshaped into 150×150×3 array of floats with each value divided by 255 so that all values are between 0 and 1. Then the images were divided into training data and testing data, 80% images were used as training data and 20% images were used as testing data.

3.2 Model Training

We designed the system in such a way that each model is trained in the same way. As we mentioned before, the models detected the weeds in image classification method. Firstly, the data were augmented so that the models are trained well. Then the augmented data sent to the model as input. We used Softmax activation function as the models would classify multiclass classification. ADAM optimizer was used as optimization algorithm and Categorical Cross Entropy function was used as loss function. ADAM optimization algorithm is used to update network weights based on training data. This algorithm works better than classical stochastic gradient descent algorithm. On the other hand, Categorical Cross Entropy is a type of loss function used in supervised learning problems with categorical labels. This loss function is used in classification tasks.

We set callback function to monitor and to avoid over fit. **acc_calback** was used so that the system stops learning when the training accuracy reaches 99.99%. **EarlyStopping** callback was used in such a way that the model stopped learning if the validation loss did not decrease until 15 epochs. The **ReduceLROnPlateau** callback was used such that the learning rate was reduced by a factor of 0.1 if the model ran for 5 epochs without losing validation. The '**Modelcheckpoint**' callback was used so that the model that achieved the lowest validation loss would be stored in a defined folder.

Then we ran K-fold cross validation on our training data, where the value of K-fold was set to 5. From the 80% of training data, 70% of data was used for training data and the rest 10% data was used for validation. So the models ran the training process 5 times, each time using a different subset as the validation dataset. The epoch and batch size was set to 100 and 128 respectively. So the model was trained 5 times and run for 100 epochs each time with a batch size of 128.

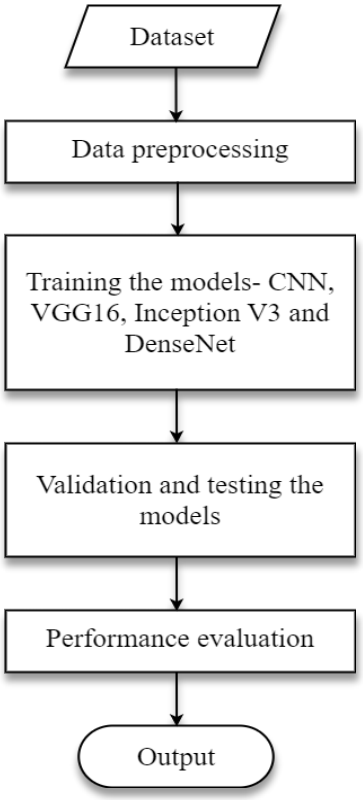

Figure- 14: Architecture of proposed model

3.3 Android App Development

Based on the system an android mobile app was developed using Android Studio to detect the weeds in real time