Flower, Mango leaf and Papaya leaf Detection

Iftekhar Jamil, Ibrahim Khalil, Tulshi Chandra Das

Institute of Information Technology, University of Dhaka

Abstract- Classification of objects by image processing is an important field of machine learning. Our objective is to identify correct class of flower, papaya leaf and mango leaf of a given input image. We took about 100 images of each class to prepare our model. We used Convolutional Neural Network learning approach to create our model. Our model gives 60% accuracy at testing set.

Index Terms- classification, flower, mango leaf, papaya, CNN.

I. INTRODUCTION

Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. [1] They have applications in image and video recognition, recommendation systems, [2] image classifications, medical image analysis and natural language processing. [3] In our image processing classification, we use CNN approach to reduce the processing cost. Convolutional networks were inspired by biological processes [4] [5][6] [7] in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The input and a output layer, as well as several protected layers, are part of a convolutionary neural network. The hidden layers of a CNN typically include a number of convolutionary layers, which are paired with a multiplication or another line. The activation function is normally a RELU-layer, then followed by additional convolutions including pooling layers, fully connected layers and layer normalization, known as hidden layers, since the activation function and final convolution cover their inputs and outputs. [8]That neuron processes information for its receptive field only. While neural networks that are completely linked to the feedforwards can be used to learn and classify features, applying this architecture to images is not feasible. Due to the very large output sizes associated with images, a very large number of neurons are required even in a shallow structure (opposite to the deep), where each pixel is a relevant variable. For example, for a (small) picture of size 100 x 100, a totally connected layer has 10,000 weights in the second layer for each neuron. The convolution method solves this problem by reducing the number of free parameters, enabling a larger network with less parameters. [9]

II. RELATED WORKS

Though the exact neural network project of flower, papaya leaf and mango leaf does not exist, there are many image-processing classification studies performed the previous years. Oluwafemi Tairu built a neural network model using cnn approach to detect the diseases of plant by image of leaf. [10]

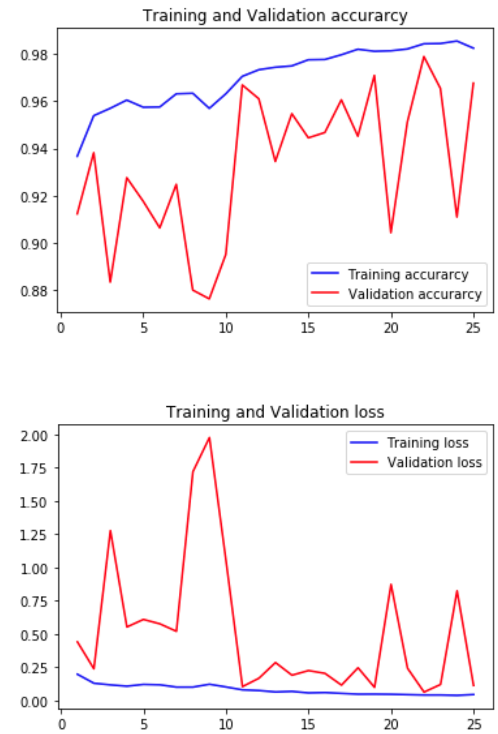

Figure 1:Training validation accuracy and loss of Oluwafemi Tairu

He used about 200 images for each class and got model accuracy of 96.77 %.

There was a study on currency recognition system [11] in Department of Computer Science and Engineering at National Institute of Technology Karnataka of Surathkal, India. The used Identification using empty regions (in the center, left, or right portions of the note) approach to classify the items. Harald Scheidl built a model to build a Handwritten Text Recognition System using TensorFlow. [12] In this neural network project, he created The NN consists of 5 CNN and 2 RNN layers and outputs a character-probability matrix in this project. There was a study on Using Deep Learning and CNNs to make a Hand Gesture recognition model [13]. He used 1000 different image of 10 hand gestures. He got 99.98% test accuracy.

III. DATA DESCRIPTION

Our project is working on three classes, flower, mango leaf and papaya leaf.

Figure 2:Sample Image of mangoo leaf

Figure 3:Sample flower image

Figure 4:sample papaya leaf image

We have collected our all necessary images from "Mokarram Bhaban" area. Specifically, we captured flower images behind ISRT building, papaya leaves besides CSE building and Pharmacy building, mango leaves from Jagannath hall. We have collected approximately 575 images. Image distribution is mentioned below.

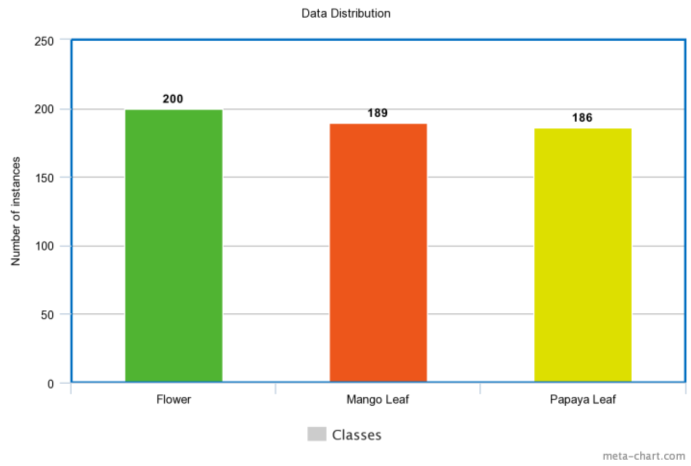

Figure 5:Training data distribution

We have found comparatively better accuracy for flower images, as they are comparatively distinguishable from other 2 classes. We have found some difficulties in predicting papaya and mango because they belong to the same colors and not easily differentiable when the image number is around 200.

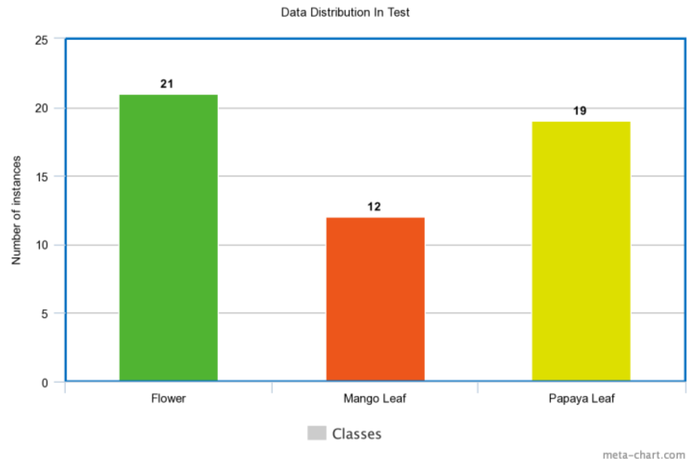

Figure 6:Test data distribution

IV. Methodology

Artificial Neural Network(ANN) uses the processing of the brain as a basis to develop algorithms that can be used to model complex patterns and prediction problems.[i-1] The network architecture has an input layer, one or several hidden layer(s) and one output layer. In our project, we applied Convolutional Neural Network to detect image. In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do image recognition, image classifications. Objects detection, recognition faces etc are some of the areas where

CNNs are widely used.[i-2] It ultimately works like ANN. But prior to ANN, it competes 3 additional steps. All four steps are:

i. Convolve

ii. Max Pool

iii. Flatten

Iv: ANN

Convlove: Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernal. Let consider a 5 x 5 image and filter matrix 3 x 3 as shown in below:

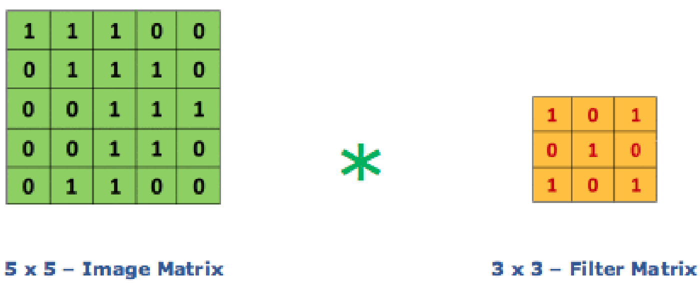

Figure 7:Convolve layer

For this image and filter matrix, the output featured map will be,

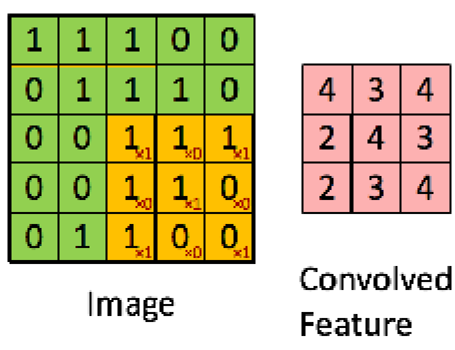

Figure 8:Convolved layer 2

Different types of filter matrix can generate different convoled feature. In our project, our image size is 80x80, filter size is 5x5 and stride is 1.

Max Pooling: Pooling layers section would reduce the number of parameters when the images are too large. We have used max pooling instead of average pooling and sum polling. In max pooling technique, we have taken the largest element from rectified feature map.

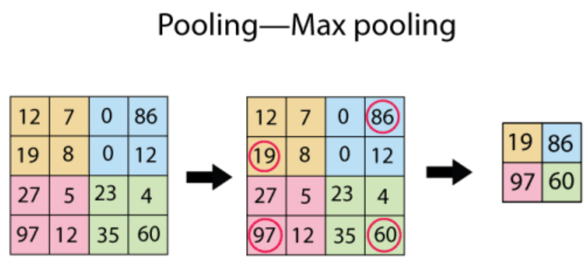

Figure 9: Max pooling

In our project, we have used 2x2 grid for max pooling.

Flatten: In between the convolutional layer and the fully connected layer, there is a 'Flatten' layer. Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.

ANN: after Flatten step, we find a vector which is basically our input features. Now we are ready to pass the vector to ANN. and we find an output array of probabilities of all classes.

V. RESULTS AND FINDINGS

In our project, we have used 3 different CNN models. These are the following:

Model 1: The first model that we have tried in our project is the one that is a traditional one. This model is tune-based and a shallow CNN model which involves 2 hidden layers only. So, along with an input and an output layer the total number of layers is 4.

For this model we have used a total number of 16 filters/kernels, each of which are of size 5*5. We have resized the input image into 80*80 pixels what may have caused the loss of some image information, but also caused reduction of possible number of inputs and extensive computation and processing. In the first hidden layer we have used Relu activation function. We have used other activation functions as well but Relu provided a better result for us.

In this model, we have trained the whole dataset in total of 5 epochs.

We have achieved the following set of values:

Table 1:Accuracy measure of model 1

Epoch No	**Accuracy**	**Loss**	**Value_Loss**	**Value_Accuracy**
1	0.44	1.0369	0.9947	0.33

2	0.75	0.7161	0.3643	0.80
3	0.86	0.3815	0.1166	0.99
4	0.89	0.3215	0.0856	0.99
5	0.92	0.2513	0.0621	1.00

We have saved the intermediate weights after each epoch and we have found that after 3 epochs, the dataset is seemed to be overfitted and thus we have decided to use the model which contains the weights that are obtained after running 3 epochs. A comparative graph that shows the changes of aforementioned 4 values after every epoch is shown below:

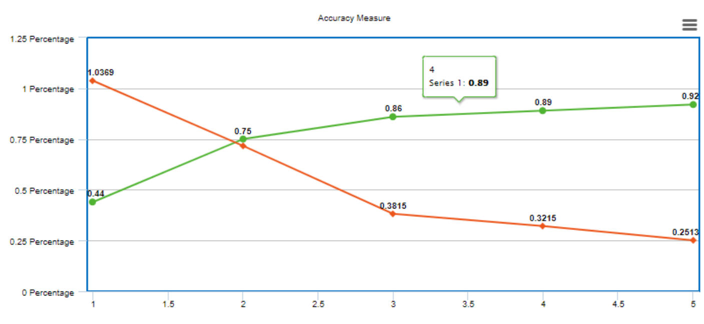

Figure 10: epoch values

We found that this model works well in the test set as well. But when we try to test the model with an external image, apart from the dataset, it doesn't show very satisfactory result. For example, we have tested 5 images of each class, which were collected externally and tested our model with those images. We found that our model had successfully predicted 11 out of 15 images in total. So, we can roughly say that the model accuracy is 73% for test data.

The relatively low value of accuracy despite fairly distinguishable images is because of very low amount of data. We believe that a more generalization can be achieved if we increase the data size. In that case, we may achieve better accuracy score. Another approach which we considered is to increase the number of layers. That also doesn't work probably because our dataset is not large enough to store information in the hidden layers that are too large in size. One more technique that we applied in our model is the use of dropout. As our model was frequently falling in the trap of overfitting, a dropout value of 0.2 is used and it helped greatly to reduce overfitting problem.

Model 2: The second approach that we have tried in our image is a well-defined CNN architecture named ResNet50. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer visions tasks. This model was the winner of ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients.

As we have very little dataset, a network of this depth is not going to work well, so we have made little customization in this model. This model yielded better accuracy for us. The training accuracy, validation accuracy and loss information after each epoch is provided in the following table:

Table 2: Accuracy measures for model 1

Epoch No	Accuracy	Loss
01	0.5758	4.33524
02	0.7273	0.75274
03	0.3333	3.14122
04	0.4848	2.37797
05	1.0000	0.00083

So, we can see that overfitting has started as soon as we have crossed the 5th epoch, so we have used the weights that we have obtained after 5th epoch. This model has, however had better accuracy for the 15 image that we have downloaded from the web and trained with. In this case, the image successfully predicted 13 images out of 15, causing a better accuracy of 86%. A comparative graph that shows the change of tabular values is given value:

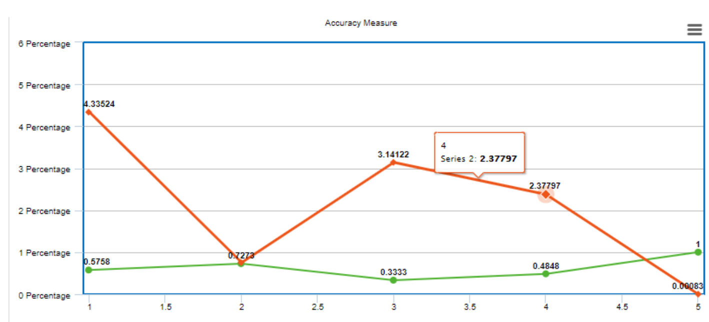

Figure 11:Accuracy measure for model 2

VI. CONCLUSION

This project was done for some experimental purpose. We have found some satisfactory result though our dataset is not large enough for training deep learning model. The scope of our project can be extended in future by introducing different other leaves.

References

[1] I. Goodfellow, Y. Bengio and A. Courville, " Deep Learning," *MIT Press*, p. 326, 2016.

[2] V. d. Oord, Aaron, Dieleman, Sander, Schrauwen, Benjamin, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Q. Weinberger, "Deep content-based music recommendation Curran Associates,," p. 2643–2651., 2013.

[3] R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. Proceedings of the 25th International Conference on Machine Learning. .," *ICML 08*, Vols. New York, NY, USA: ACM., 2008.

[4] K. Fukushima, ""Neocognitron". Scholarpedia. .," *doi:10.4249/scholarpedia..*, vol. 2(1), p. 1717, 2007.

[5] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195 (1), p. 215–243.

[6] Fukushima and Kunihiko, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics.*, vol. 36 (4), pp. 193-202, 2013.

[7] M. Matusugu, K. Mori, Y. Mitari and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network" (PDF). Neural Networks. : .," *doi:10.1016/S0893-6080(03)0*, vol. 16(5), p. 555–559.

[8] "CS231n Convolutional Neural Networks for Visual Recognition. .," *cs231n.github.io*, Retrieved 2018-12-13..

[9] Habibi, Aghdam, Hamed and E. J. Heravi, Guide to convolutional neural networks : a practical application to traffic-sign detection and classification., Cham, Switzerland..

[10] Oluwafemi Tairu, "plant-ai-plant-disease-detection-using-convolutional-neural-network-," 2018.

[11] V. Abburu, S. Gupta, S. R. Rimitha, M. Mulimani and S. G. Koolagudi, "Currency Recognition System Using Image," *IEEE*, 2017.

[12] H. Scheidl, "Build a Handwritten Text Recognition System using TensorFlow," Jun 15, 2018.

[13] F. Borba, "Using Deep Learning and CNNs to make a Hand Gesture recognition model," 2019.

AUTHORS

First Author – Iftekhar Jamil, mail: bsse0802@iit.du.ac.bd
Second Author – Ibrahim Khalil, mail: bsse0804@iit.du.ac.bd.
Third Author – Tulshi Chandra Das, mail: bsse0811@iit.du.ac.bd