Fruit Classification Using Machine Learning Techniques

Pranav Babulkar
North Carolina State University
Raleigh, NC
pbabulk@ncsu.edu

Yoonchul Ro
North Carolina State University
Raleigh, NC
yro@ncsu.edu

Anmol Desai
North Carolina State University
Raleigh, NC
adesai5@ncsu.edu

Raj Kumar Shrivastava
North Carolina State University
Raleigh, NC
rkshriva@ncsu.edu

ABSTRACT

Image classification using machine learning has been an ever-evolving domain. In this work, we use an dataset of fruits images. We presents a technique which uses a combination of several classifiers to predict the type of a fruit from a given image. In our work, we primarily explore the techniques of Random Forest, Convolution Neural Networks, Support Vector Machines and K-Nearest Neighbors. We analyze these models for their performance in the fruit classification task and further describe a method in which these techniques can be combined together to produce an enhanced efficiency.

1 INTRODUCTION

Machine learning is being increasingly used in a variety of applications. In this paper, we study various machine learning algorithms that can be used in image classification. We explore a data-set of images of various fruits in this work and use machine learning techniques to classify different fruits. We present an analysis of the performance of various techniques for this particular data-set. The data-set consists of a total of 82213 different images of fruits belonging to 120 different classes. To begin with, we chose the topic of classifying fruits since it has multiple applications. One application could be in a grocery store. Fruits are found in almost all grocery stores. So, for a grocery store, if we have a fruits identifier, an AI robot can directly identify which fruits are under-stocked and thus reduce the need of human labor. Secondly, considering the fact that sometimes two different fruits can have very similar appearance, the problem of fruit classification becomes challenging. It would require an efficient classifier for this classification. In this work, we talk about Random Forests and deep learning techniques like convolution neural network(CNN) and ResNet which identify the type of fruit from images. Random forests are an ensemble learning method consisting of many decision trees used for classification and uses mean prediction or mode prediction of the individual trees.

Talking about deep learning[7], it is a class of machine learning algorithms that use multiple layers containing processing units which are non linear. Each level learns to transform its input data into a marginally more abstract and composite representation. Deep neural networks have managed to perform better than other machine learning algorithms in several problems. Convolution Neural Networks (CNN)[7] are a part of deep learning models. A typical CNN architecture consists of each convolution layer followed by a Rectified Linear Units (ReLU) layer, then a Pooling layer, then one or more convolution layers, and lastly, one or more fully connected layers. Compared to a regular neural network, a CNN can be more effective in some problems since instead of transforming the input into a one dimensional array which makes the classifier less sensitive to positional changes, it considers the structure of the images while processing them. A convolution layer[7] consists of groups of neurons that make up kernels. The kernels extract high level features and produce a two dimensional activation map by sliding across the width and height of the input. Applying a convolution layer over an image of size 32 x 32 results in an activation map of size 28 x 28. If we apply more convolution layers, the size is further reduced. We use padding to take care of loss of information and the vanishing gradient problem. Padding increases the input data size by filling constants around the input data.

A Pooling layer progressively reduces the amount of parameters and computation in the network. The most common approach is the max pooling. We can also use average pooling or minimum pooling. These are the layers from a regular neural network. Each neuron from a fully connected layer is linked to each output of the previous layer. Loss Layers are used to manage the adjustments of weights across the network. The motivation for using CNN comes from its efficient use in the past to obtain great results in the field of image classification. A residual neural network (ResNet)[9] is an artificial neural network (ANN) that builds on constructs known from pyramidal neurons that are present in the cerebral cortex of our brain. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double- or triple- layer skips that contain non-linearities (ReLU) and batch normalization in between.

The remaining sections of the paper have been organized as following. Section 2 describes various related words done in the past. In section 3, we talk about method being used in our work. Experiments performed in our work has been described in section 4. We further present our finding in section 5. In the final section, the paper presents a conclusion of our work.

2 BACKGROUND

The problem of fruit classification has been addressed in the past in various ways. In [7], Horea et al.(2018) have approached the problem of fruit classification using convolutional neural network to achieve a maximum accuracy of 97.04%.

In another approach[5], Jena et al.(2017) make use of Support Vector Machine to perform this classification. The authors have shown the use of texture and color of the images to classify different fruits with an accuracy of 83.33%.

Zeng(2017) in [10] uses VGG model, a specific configuration of convolutional neural network, to approach the problem of fruits and vegetables classification simulataneously and achieved an appreciable accuracy of 92.5%.

Rocha et al.(2010) in [8] present an approach which combines several classifiers which uses concatenated features for classifying fruits and vegetables from their images.

Bargoti et al.(2017) in their work[3] have used Faster R-CNN along with data augmentation techniques to classify fruits even when there are a number of fruits in the same image. Faster R-CNN is an efficient architecture for object-detection introduced in 2015. It makes use of convolutional neural networks including YOLO and SSD algorithms.

For a specific fruit, pineapple, Moonrinta et al.(2010) [6] present an application of fruit detection in a pineapple farm. For in-field pineapple detection, the paper used feature extraction techniques like SIFT, SURF and Harris algorithms. The features obtained from these algorithms have been used for 3D construction of pineapples for yield tracking. Support Vector machine algorithm has been employed for the classification using these features.

3 METHOD

3.1 Approach

We plan to use total of 5 models: ResNet-18, k-NN, SVM, Random Forest, and merged classifier of (k-NN, SVM, Random Forest). According to our research, we thought it was best to use ResNet-18 model as the baseline for comparison, as it would presumably have the highest accuracy. The three other models that are mentioned are going to be merged to create an additional model. The comparison of results of all individuals models and the merged ones will be discussed later.

3.1.1 Random Forest. This is a novel classifying technique which we have not covered in class. Random forest classifier is an ensemble tree-based learning algorithm, where it is composed of a set of multiple (usually a large number) relatively uncorrelated decision trees. Each tree is constructed from a randomly selected subset of training data. When tested using the test data, it aggregates the votes from different decision trees in order to decide on the final single class of the corresponding input. The test data will be classified into the class with the highest number of votes. Recall that ensemble algorithm is an algorithm which combines more than one algorithms of same or different kind in order to classify (in this particular task) data. Figure 1 illustrates how random forest works.

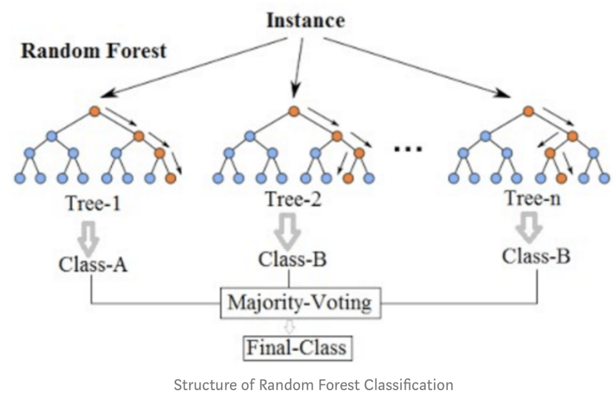

Figure 1: Random Forest

3.1.2 ResNet. In order to tackle the problem mentioned above, we are also using deep residual network (ResNet) because of its ability to achieve compelling performance even with hundreds of layers[4]. A residual neural network is an artificial neural network of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. We also considered using residual networks as deeper neural networks are difficult and more costly to train when compared to the Residual Networks. Residual Networks reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. There is empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. Figure 2 shows the example of skip connections, or shortcuts to jump over some layers.

3.1.3 K-Nearest Neighbour. We also use the k-Nearest Neighbours (KNN) as one of the classification model for fruits classification. The positive aspect of this model is that it does not require any actual training. It only requires some preprocessing which involves calculating closeness between the data points in an n-dimensional coordinate system. K-nearest neighbours are determined for the test data point. The K-nearest neighbors are simply the closest k points to the test data point. The closeness is generally determined in terms of Euclidean distance but we can also use other distance metrics as well. To determine the class using these nearest neighbours list, KNN uses either majority vote wherein we take the most common class label or weighted vote wherein the votes could be weighted according to the distance.

3.1.4 Support Vector Machines. The last model we evaluate for our fruit classification task is Support vector machines (SVM). SVMs are an efficient classification models which make use of a hyperplane

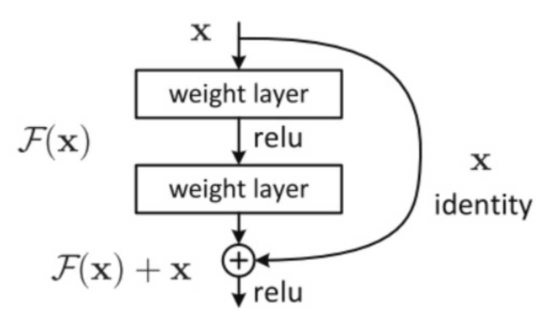

Figure 2: Residual Learning

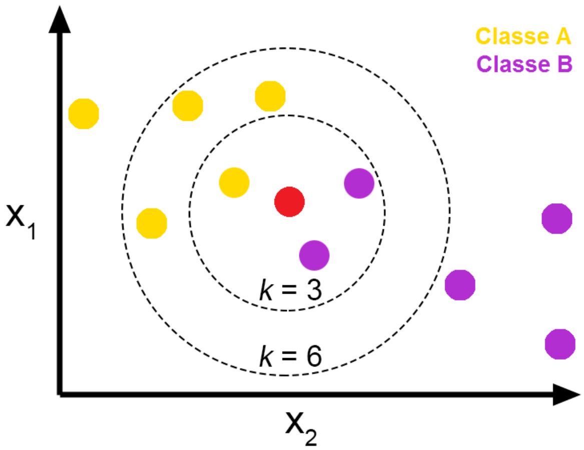

Figure 3: K-nearest neighbour

to separate two classes. Hyperplanes act as the decision boundaries such that the data point falling on different sides of this plane belong to different classes. SVM work by maximizing the margin between the data points of different classes. This is done by identifying support vectors. Support vectors are data points which are nearest to the hyperplane from both the sides of the plane. Using these support vectors, we define the hyperplane for the particular SVM model.

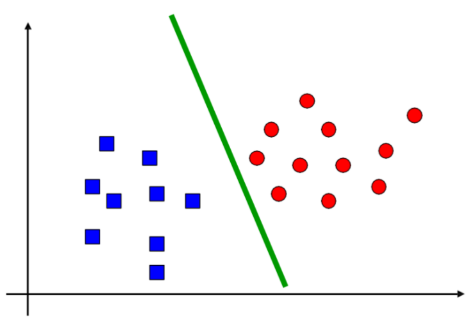

Figure 4: Support Vector Machines

3.1.5 Combining Classifiers. For combining ensemble classifiers - SVM, KNN and Random forest, we used majority voting rule using the Voting Classifier. A Voting Classifier[2] is a Machine Learning model that trains on a group of models and predicts an output class based on the highest probability. It aggregates the predictions of each classifier passed into it as arguments and predicts the output based on the highest majority of voting. Voting Classifier supports two types of voting.

- Hard Voting: In hard voting, the output class which is predicted by the voting classifier is the class with the highest majority of votes, which is the class with the highest probability of being predicted by each of the classifiers.
- Soft Voting: In soft voting, the predicted output class is based on the average of probability given to that class.

We can also add weights to each classifier based on the accuracy of different classes. With the hard voting, we can weight the occurrences of predicted class labels and with soft voting we can weight the class probabilities before averaging. If we do not assign weights, the voting classifier uses uniform weights.

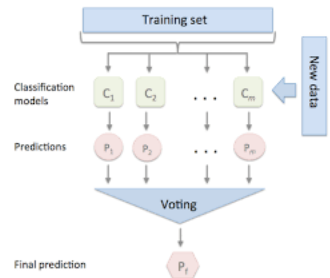

Figure 5: Voting Classifier

3.2 Rationale

We described four methods to classify fruits in the previous subsection. To reiterate, they are random forest, CNN, SVM, and k-NN. These four methods are suitable for this task due to following reasons:

- The problem is classification task
- The dataset are images
- Image data are high in dimension

Furthermore, CNNs are experimentally proven to be effective in image classification throughout multiple research. It is a non-parametric approach, makes no assumption on the data. It is able to efficiently learn the given environment and store its knowledge. CNNs are fully connected feed forward neural networks which are effective in reducing the parameters without loosing on the quality of the models. This quality of CNN makes them apt for the problem we are dealing with. This helps to classify the images accurately.

Random forest is also a non-parametric technique. It is appropriate in this specific task as it is not sensitive to over-fitting, easy to parametrize, and good at dealing with outliers in the training data.

K-NN is by far one of the simplest classification method. There are only two things that needs to be considered when performing k-NN: distance metric and the parameter k.

Support vector machines have been proven to be quite effective for classification problems. Using appropriate kernel based on the distribution of data ensures good-quality results. Upon visualization of this particular dataset, we get a good intuition that Support Vector Machines would result in a great accuracy in classifying the fruits.

We use Random Forest, SVM, and K-NN to perform majority voting. Majority voting outputs a class of fruit using three different independent predictions from their respective models. It chooses the class which has been predicted by the most number of models. By maintaining a total of three different methods, which is an odd number, we overcome the problem of tie in the majority voting in cases where there is lack of consistency in the models' respective predictions.

We use the merged classifier to get an idea of how the accuracy of all the models combined works as compared to their accuracy individually. Further, we compare this accuracy with the CNN model to get a better idea of the performance of the merged classifier.

4 EXPERIMENT

4.1 Dataset

4.1.1 Description. We are using the Fruits-360 dataset from Kaggle. The dataset is obtained through the following URL: https://www.kaggle.com/moltean/fruits/data. This dataset is 730MB in size, which consists of 82213 images of fruits and vegetables of 120 classes. Our primary focus is on classification of fruits only. After pre-processing the data and getting rid of all the vegetable data, it resulted in total of 114 classes, with 57,389 training data and 19,587 test data.

The 114 fruit classes are the following: Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot, Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambula, Cauliflower, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Dates, Eggplant, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine (Regular, Flat), Nut (Forest, Pecan), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Williams), Pepper (Red, Green, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow), Walnut.

4.1.2 Properties.

(1) Total number of images: 82213 (68,976 after removal of vegetables)
(2) Training set size: 61488 images (57,389 after removal of vegetables).
(3) Test set size: 20622 images (19587 after removal of vegetables).
(4) Multi-fruits set size: 103 images (more than one fruit per image).
(5) Number of classes: 120 (114 after removal of vegetables).
(6) Image size: 100x100 pixels.
(7) Filename format: imageindex100.jpg (e.g. 32100.jpg) or rimageindex100.jpg (e.g. r32100.jpg) or r2imageindex100.jpg or r3imageindex100.jpg. "r" stands for rotated fruit. "r2" means that the fruit was rotated around the 3rd axis. "100" comes from image size (100x100 pixels).

4.1.3 Pre-processing of the Data. Before the data is trained onto the random forest model, each image in the training data-set is converted into a 100x100 numpy array for each RGB (Red, Green, and Blue respectively) dimension. It is then flattened into a single vector which is further scaled by subtracting the mean of the data set. This is shown in the Figure 6. For feeding the data to the ResNet-18 model the data is normalized based on the stats of the RGB channels from the ImageNet data-set so as to be in accordance with the pre-trained model of ResNet-18.

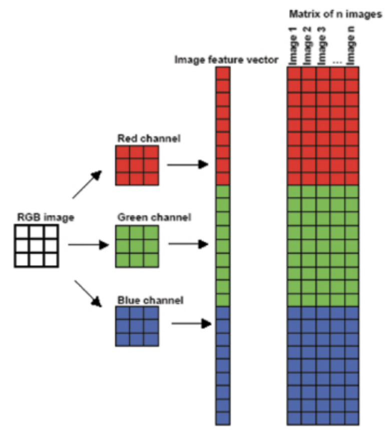

Figure 6: Image Matrix

Since image data have large number of dimensions, it is important to decrease the number of dimensions in the dataset. Although this may lead to lower test accuracy, it has a great advantage of speeding up the learning process, as lower dimension indicates fewer data. We plan to explore and experiment on the number of dimensions for each different methods. This is further explained in next subsection.

4.2 Hypotheses

The following are the questions which were investigated by us before the experiments were conducted:

(1) Since the data points of a given class appear to be clustered together when visualized on a two-dimensional plane, lower values of k would result in a better accuracy than the higher values of k in the K-NN model.
(2) Since the classes are not linearly separable, a linear kernel would result in a poor performance when compared with non-linear kernels like radial base function (RBF) in an SVM model.
(3) The performance of the merged classifier will have better test accuracy than one single individual classifier. Specifically, merged classifier of (Random Forest + SVM + K-NN) will out-perform the individual classifiers of (Random Forest), (SVM), and (k-NN).
(4) SVM will have the highest accuracy, followed by Random Forest and K-NN.
(5) Majority Voting with higher weights assigned more accurate classifiers give better results than uniform weighted majority voting.
(6) The merged classifier would have lower or equal performance to that of ResNet-18.

4.3 Experimental Design

4.3.1 Random Forest. After each data in the training set is flattened into a single vector and is further scaled (as mentioned in section 3.1.3), the data is further processed before it is used to train random forest model. The data is first explored and visualized in lower dimensions using PCA. Figure 7 shows the results of dimension reduction performed on a single data. More details are covered in section 5. Once this is done, the data is split into 0.75:0.25 ratio of training and validation set.

The four sub-figures in Figure 7 represent the original data (image), reduced to 50-D, reduced to 10-D, and reduced to 1-D from left to right respectively. For now the data has been reduced to 50-D, and the results are shown in the section 4. We plan to further experiment on the number of dimensions so that the processing time is reduced but the accuracy is not decreased significantly.

4.3.2 ResNet-18. We have also implemented ResNet-18 on our data. ResNet-18 is an 18 layer deep convolution neural network that can classify images into 1000 different objects. The architecture handles the images as illumination at each pixel of the picture as a 2D matrix which are passed on to various convolution layers and based on the loss on the prediction the weights are adjusted based on the learning rate at each mini-batch cycle.

The model is trained on 80% of the data and the model is then validated on the rest 20%. The model is trained with the pre-built `cnn_learner` function, using the provided data, ResNet-18 residual network architecture, and accuracy metric as a parameter.

The labels for the data are directly taken from the folders as the data set is provided in such a format that the folder names are the labels for the images. Learning Rate is the crucial parameter to be decided while training a deep neural network. It determines the amplitude of the jump to be made by the optimization technique in each iteration. It cannot be too low or too high. We are using `fit_one_cycle`() function from `fast.ai`[1] library which randomly decides on the learning rate based on the loss in the previous layers of the network.

We relied on the `fit_one_cycle`() function from `fast.ai`[1] library which randomly decides on the learning rate on our first iteration but in the second iteration we researched on the best approaches to fit our deep neural network model. We found that recording the learning rate with respect to the loss at each batch using `lr_find`() function can give us a track of the optimum learning rate which can lead to better accuracy scores and improvement in the loss. Figure 8 shows the plot of the loss with respect to learning rate with a suggestion on optimum learning rate. In this iteration we used this and gave the accuracy of the model a substantial bump. We also tried to train the model better with respect to our data set by unfreezing the initial layers of the model and training the model with a very low learning rate which in our case was a slice between 1e-6 and 1e-4. This again gave a push to our accuracy and also reduced the error rate in the validation set.

4.3.3 K-Nearest Neighbors. The dataset consists of 114 different classes of fruits. We processed the images by transforming them into vectors. We used K-Nearest Neighbors for different values of k to evaluate the variation in results with the change in number of neighbors for a test data point considered to predict its class. In our experiment, we predicted the classes for the dataset with 14 different values of k ranging from 1 to 111 (both inclusive) differing at an interval of 10. The results derived from this experiment has described in Section 5.

4.3.4 Support Vector Machine. We also tried to use Support Vector machines to see how they work with the given dataset for fruit classification. We used the same pre-processed dataset obtained by using PCA in the previous models. We trained SVM with different kernels to observe the variation in accuracy. We experimented with linear kernel, radial basis function (RBF) kernel and polynomial kernel in our work. Since, the classes are not linearly seperable, linear kernel is not expected to perform well. In contrast, the RBF kernel produces good results for the given preprocessed dataset.

4.3.5 Combined classifier. We implemented the majority voting rule using the Voting Classifier. For this, we considered the 3 independent classifiers, namely, random forest, KNN and SVM classifiers. We used hard voting, soft voting and weighted classifiers while assigning weights to some models according to the accuracy achieved in them. While implementing this, we used random forest with 100 estimators as we felt this was the right balance between the accuracy and the being computationally expensive, the SVM with rbf kernel and KNN with k=1 as this had the highest accuracy.

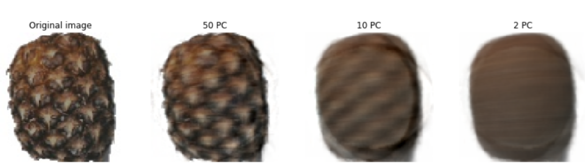

Figure 7: Images for Different Dimension

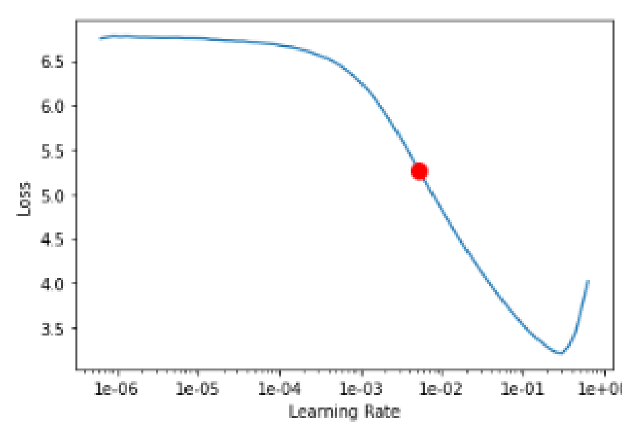

Figure 8: Plot for Learning Rate Finder

No. of Estimators	Training Accuracy (%)	Test Accuracy (%)
1	91.02	64.35
5	98.65	77.95
10	99.61	84.15
15	99.74	86.66
20	99.80	87.87
25	99.89	88.66
40	99.88	89.79
70	99.91	91.00
100	99.93	91.40
200	99.92	91.85
300	99.93	91.76
400	99.94	92.12
700	99.94	92.19
1000	99.94	92.20

Table 1: Training and testing accuracy for Different Numbers of Estimators in Random Forest

5 RESULTS

5.1 Random Forest

The table shows the training and test accuracy (rounded up to second decimal digit) for different number of estimators in random forest. The table suggests that as the number of estimators increases, training accuracy and test accuracy increases. For now, we were only able to test the number of estimators up to 1000. We plan to further conduct experiments on this number as well as reducing the original data into different dimension in order to find the optimal number of estimators so that this single model have the highest test accuracy.

5.2 ResNet-18

We have created an image data bunch from the data set and trained our model on the training data with 80% as training and 20% as validation data. We have got a validation accuracy of 99.9%. As per the `fast.ai`[1] docs we have implemented the learning rate finder to find the optimum learning rate for the model. We also have made some tuning on the hyper-parameters of the model which gave us an increment in the accuracy of the predictions made by the model.

In the final model we reached the accuracy of 99.3% on the test data set which was never seen by the model while training. Figure 9 shows the images with top losses.

5.3 K-Nearest Neighbors

The K-Nearest neighbor model produced good results for the fruits classification. We used the preprocessed data in the KNN model with varying value of k, i.e., the number of neighbors to be considered while making prediction for a given input. Table 2 shows the training and testing accuracy for various values of k.

We can see in the graph in Figure 10 that both the training as well as the testing accuracy decreases with the increasing value of k.

k	Training accuracy(%)	Testing accuracy(%)
1	99.97	94.01
11	99.40	88.73
21	97.86	85.07
31	96.55	82.47
41	95.16	80.07
51	93.73	78.03
61	92.39	76.34
71	91.08	75.04
81	89.68	73.72
91	88.58	72.30
101	88.58	70.98
111	88.38	69.99

Table 2: Training and testing accuracy of K-NN

Kernel	Training Accuracy (%)	Test Accuracy (%)
Linear	99.83	93.78
Polynomial	97.89	90.4
RBF	99.95	95.68

Table 3: Accuracy of SVM over different kernels

5.4 Support Vector machine

We got good results in SVM as can be seen in Table 3. Table 3 shows the accuracy of SVM with the preprocessed fruits dataset with different kernels. It can be observed that rbf kernel resulted in the best accuracy. This was an expected result lookoing the distribution of the data. The data needs to be raised to multiple dimensions which supports why rbf kernel accuracy is better than the linear kernel accuracy in the SVM model.

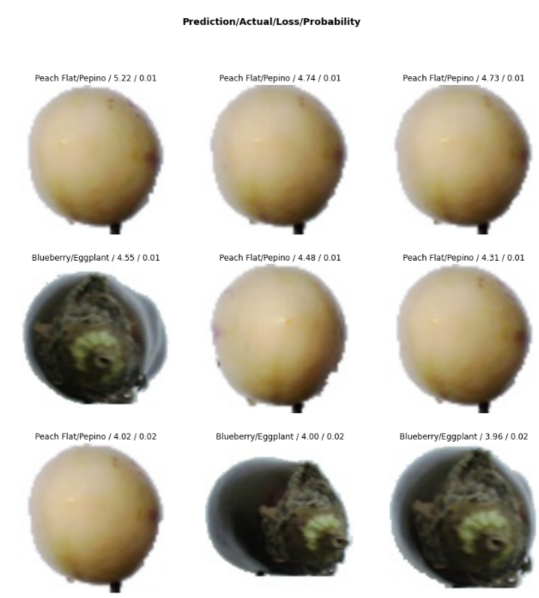

Figure 9: Top Losses on the Test data images

Voting Classifier (Majority Voting)	Training Accuracy	Test Accuracy
Soft	99.92%	94.32%
Hard	99.87%	95.01%
Soft (Weighted)	99.91%	95.3%
Hard (Weighted)	99.94%	95.98%

Table 4: Voting Classifier

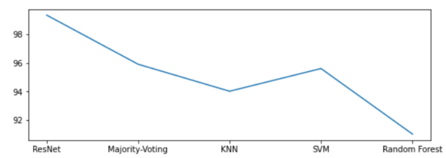

Figure 11: Accuracy of Classifiers

5.5 Combination of Classifiers

For combining the classifiers, we used the voting classifier for majority voting. We performed hard majority voting, soft majority voting, hard majority voting with weights and soft majority voting with weights. For the weighted classifier, since the accuracy of SVM was the highest, we assigned a weight of 2 to SVM and 1 to the rest. Table 4 shows the accuracy of the majority voting classifier with the different methods. Figure 11 shows the output of all the classifiers used and ResNet as well.

6 DISCUSSION

We initially planned to just use one of either SVM or KNN and combine with ResNet-18 and Random Forest. However, considering the high accuracy of ResNet-18, we considered using it as our baseline for comparison.

To summarize the results for each of the classifiers, we got the following: k-nn: 94.01% (k=1), SVM: 95.68% (RBF), Random Forest: 92.2% (1000 Estimators).

After merging these three classifiers, specifically with k=1 for k-NN, RBF SVM, and 100 estimators for Random Forest, the model resulted in 95.98% when using hard (weighted) majority voting.

These results were true for some of the hypotheses and false for others. Specifically, the merged classifier out-performed all the three individual models as we have hypothesized. Moreover, when looking at the three models separately, SVM had the best performance, followed by k-NN and Random Forest. This is in contrary to our hypotheses, where we assumed that SVM will have the highest accuracy, followed by Random Forest and k-NN. For k-NN, lower values of k did indeed lead to a higher performance, and for SVM, the RBF model had the best performance as well, which supports our hypotheses. Lastly, even though we merged the three classifiers, it still had a lower performance than ResNet-18, even though we assigned higher weights to the models that performed better.

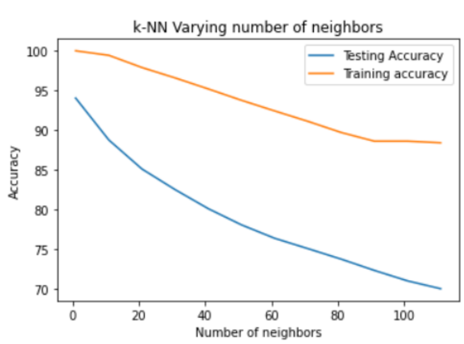

Figure 10: Accuracy vs number of neighbors (k)

7 CONCLUSION

In this work, we have evaluated various classifiers by considering the problem of fruit classification. The models can only be trained using vectors as inputs. Since, the original dataset involves images as inputs, data pre-processing forms an important aspect in such problems. Further, the color images contains high dimensions. Training the models without dimension reduction involved huge computational cost. To save computation power and time, dimension reduction can be performed by using methods like PCA and t-SNE.

After developing the models, its essential to tune them by choosing the most optimum parameters by experimenting several times with different parameters combination. The best values of these parameters cannot be generalized as it is mostly specific to the problem being considered and the dataset being used.

As a future scope, we can experiment with more classifiers for this problem. It would be interesting to see how different neural networks would behave with this dataset. In the future, we would also want to try more combinations of classifiers to evaluate how effective choosing the right combination could be even when the individual models are weak.

REFERENCES

[1] 2020. *fast.ai, "Vision" (Online).* https://docs.fast.ai/vision.html

[2] ankushkuwar05. 2020 (accessed April, 2020). *ML | Voting Classifier using Sklearn.* https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/

[3] Suchet Bargoti and James Underwood. 2017. Deep fruit detection in orchards. In *2017 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 3626–3633.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[5] Susovan Jana, Saikat Basak, and Ranjan Parekh. 2017. Automatic fruit recognition from natural images using color and texture features. In *2017 Devices for Integrated Circuit (DevIC).* IEEE, 620–624.

[6] Jednipat Moonrinta, Supawadee Chaivivatrakul, Matthew N Dailey, and Mongkol Ekpanyapong. 2010. Fruit detection, tracking, and 3D reconstruction for crop mapping and yield estimation. In *2010 11th International Conference on Control Automation Robotics & Vision.* IEEE, 1181–1186.

[7] Horea Mureşan and Mihai Oltean. 2018. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica* 10, 1 (2018), 26–42.

[8] Anderson Rocha, Daniel C Hauagge, Jacques Wainer, and Siome Goldenstein. 2010. Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture* 70, 1 (2010), 96–104.

[9] Wikipedia. 2020. Residual neural network — Wikipedia, The Free Encyclopedia. Online; accessed 2-April-2020.

[10] Guoxiang Zeng. 2017. Fruit and vegetables classification system using image saliency and convolutional neural network. In *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC).* IEEE, 613–617.

A MEETING ATTENDANCE

We have conducted the following virtual meetings for this project:

(1) March 28th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj
(2) March 30th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj
(3) April 4th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj
(4) April 8th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj
(5) April 11th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj
(6) April 18th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj
(7) April 21th, 2020 7:00 PM to 9:00 PM,
Attendees: Pranav, Yoonchul, Anmol, Raj

B CODE

Please visit the following GitHub repository for the codes of this project: https://github.com/rajshrivastava/Fruit-Classification