Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, 34, S. 344-353, Mart 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 34, pp. 344-353, March 2022

Research Article

The Effectiveness of Transfer Learning and Fine-Tuning Approach for Automated Mango Variety Classification

Nagham Alhawas[1*], Zekeriya Tüfekci[2]

[1*] Çukurova University, Faculty of Engineering, Departmant of Computer Engineering, Adana, Turkey, (ORCID: 0000-0002-7407-1392),nagham_hs@hotmail.com
[2] Çukurova University, Faculty of Engineering, Departmant of Computer Engineering, Adana, Turkey, (ORCID: 0000-0001-7835-2741), ztufekci@cu.edu.tr

(2nd International Conference on Applied Engineering and Natural Sciences ICAENS 2022, March 10-13, 2022)

(**DOI:** 10.31590/ejosat.1082217)

ATIF/REFERENCE: Alhawas, N. & Tüfekci, Z. (2022). The Effectiveness of Transfer Learning and Fine-Tuning Approach for Automated Mango Variety Classification. *European Journal of Science and Technology*, (34), 344-353.

Abstract

For the mango-producing sectors, an accurate vision system for classifying distinct mango types is essential. The process is mostly done using manpower labor and it is cost-inefficient. However, actual success in this field is still narrow, and there is a significant shortage of studies on this issue. This paper presents the effectiveness of applying transfer learning and fine-tuning on the identification of mango types. An imagery dataset of eight Pakistani mango varieties is used to fulfill the study purpose. Based on the experiments different image preprocessing and data augmentation techniques are applied. Two main experiments are conducted on MobileNet and ResNet50. For MobileNet, the performance behavior was compared between loading only the modal's architecture with random weights, then with the use of transfer learning, and finally by cooperating fine-tuning. Different hyperparameter tuning was studied to improve the model's performance. For the ResNet50, a hybrid ResNet50 with machine learning models is built. The ResNet50 with transfer learning is used as a feature extractor and number of 2084 features have resulted. Principal component analysis PCA is applied to reduce the dimensionality of features. The 187 resulted feature is scaled, then fed to Naïve Bayes, Logistic Regression, SVM with different kernels all are tested with 10 stratified repeated kfold. Different performance evaluation metrics were used to assist the models' behaviors. We showed that transfer learning and fine-tuning is the best practice in terms of performance and execution time for the mango varieties identification. The best testing accuracy, recall, F1, and precision is 100%.

Keywords: computer vision, Resnet50, MobileNet, Transfer learning, fine-tune, hybrid CNN-ML models.

Otomatik Mango Çeşitlerinin Sınıflandırması İçin Transfer Öğrenme ve İnce Ayarlama Yaklaşımının Etkinliğinin Değerlendirilmesi

Öz

Mango üreten sektörler için, farklı mango türlerini sınıflandırmak için doğru bir görüş sistemi esastır. İşlem çoğunlukla insan gücü emeği kullanılarak yapılır ve maliyet etkin değildir. Ancak, bu alandaki gerçek başarı hala sınırlı ve bu konuda önemli bir çalışma eksikliği var. Bu makale, mango türlerinin tanımlanmasında transfer öğrenimi ve ince ayarın uygulanmasının etkinliğini sunar. Çalışma amacını yerine getirmek için sekiz Pakistan mango çeşidinden oluşan bir görüntü veri seti kullanılmıştır. Deneylere dayalı olarak farklı görüntü ön işleme ve veri büyütme teknikleri uygulanmaktadır. MobileNet ve ResNet50 üzerinde iki ana deney gerçekleştirildi. MobileNet için, performans davranışı, yalnızca modul mimarisini rastgele ağırlıklarla yükleme, ardından transfer öğrenme kullanımı ve son olarak ince ayar işbirliğiyle karşılaştırıldı. Modelin performansını iyileştirmek için farklı hiperparametre ayarlaması çalışıldı. ResNet50 için, makine öğrenimi modellerine sahip hibrit bir ResNet50 oluşturuldu. Transfer öğrenmeli ResNet50 öznitelik çıkarıcı olarak kullanıldı ve 2084 adet öznitelik elde edildi. Temel bileşen analizi PCA, özelliklerin boyutunu azaltmak için uygulanır. Elde edilen187 öznitelik ölçeklendi, daha sonra Naïve Bayes'e, Lojistik Regresyon vw farklı çekirdeklere sahip SVM'e girdi olarak verilip 10 tabakalı tekrarlanan kfold ile test edildi. Modellerin davranışlarına yardımcı olmak için farklı performans değerlendirme metrikleri kullanıldı. Mango çeşitlerinin tanımlanması için performans ve uygulama süresi açısından transfer öğrenimi ve ince ayarın en iyi uygulama olduğunu gösterdik. En iyi test doğruluğu, geri çağırma, F1 ve kesinlik oranı %100'dür..
Anahtar Kelimeler: bilgisayarla görme, Resnet50, MobileNet, hibrit, Transfer öğrenme, ince ayar, CNN-ML modeller

* Corresponding Author: nagham_hs@hotmail.com

1. Introduction

There are many intelligent systems to automate the process of sorting, grading, and classifying vegetables and fruits. We see multiple examples in this field, whether to classify dry grains, coffee beans, avocado, and other types. These studies use computer vision, machine learning techniques, and deep learning to classify the various types of these crops for goals related to either sorting, grading, or harvesting systems. The agricultural system can use such systems to increase quality and production while minimizing the manpower and time cost. The effectiveness and significance of these methods and research in the agricultural area are still being debated. One of the trends that still need to be addressed is the mango classification. The mango classification system is still needed on the market and in the agricultural field and there is a lack of this kind of study in literature. Mango is a delicate agricultural crop that can readily develop brown stains during postharvest processing, shipping, and marketing. Mango is a popular type of fruit because of its sweet taste. Most of the countries that produce it are India, China, and Pakistan Figure 1 shows the production of each country. The procedure of classifying mangos across the world is primarily carried out by farmers' manual work. The development of automatic system can decrease human labor and the cost while keeping the high-quality standards. Not much study is available in literature classifying mangos varieties compared to studies done for other fruit. This might be due to the lack of such available datasets. Because of the wide variety of geometric shapes, forms, and features, distinguishing the various mango types is a challenging task. Computer vision by involving machine learning and deep learning techniques makes the recognition of mango types and quality easier than ever before. The physical and geometrical appearance is the key factor to implementing useful models.

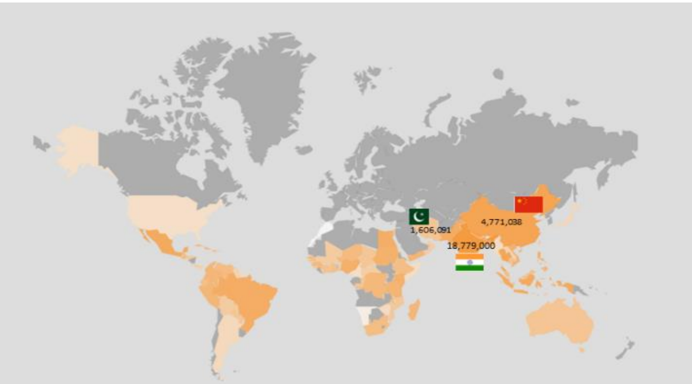


Figure 1 *The World Top Mango Production by Country*

Different research in literature used shallow approaches to extract shape, size, area, texture, and color features from mango images. (Yossy et al., 2017) extracted the physical characteristic of 4 types of mangos for sorting purposes. Color conversion is done to convert RGB images into HSV color models, then convert it to grayscale image for segmenting ROI. Finally, they extracted size, area, and color information. These features are fed into artificial neural network model with a different number of hidden layers. With a 0.000001 learning rate and 40 hidden layers their accuracy is 94%.

Five mango varieties were classified by the Naïve Bayes model in (Win & Misigo, 2019) image processing techniques were utilized to detect the mango features in a better way. All the RGB images were resized into 300 × 400 then the median filter is applied to remove the noise from the images. To segment the region of interest first all a color conversion is considered to convert RGB images to HSV color model then the images are converted to binary images. To detect the edges even better Sobel mask operator is applied. Then the shape and texture features were extracted as Area, Perimeter, Major Axis length, and Minor Axis for the former. All features of 50 images of each five-mango class are fed to the Naïve Bayes classifier to distinguish. 20 images of each class were used for testing and the average accuracy is 94%.

In (Naik & Shah, n.d.) data sets of 1082 images of seven mango varieties were prepared to build a deep learning model that is able to distinguish between the different types. All RGB images were resized int 512 × 512 then fed to Inception v3, Xception, DenseNet, and MobileNet achieved 90, 92.42, 91,42, and 88.57 % accuracy respectively. These pre-trained models are used without further fine-tuning steps. 60 images of each type were used for training and 10 for testing.

This study (Pandey et al., 2021) aims to classify 15 varieties of mango collected from markets using transfer learning. 1850 images were collected so, each category contains 100-200 samples. All images resized to 277× 277 and fed to AlexNet, GoogLeNet, ResNet50, and VGG16 pre-trained models. The mean, minimum, and maximum values of the F1 score and FPR were measured to evaluate models' performance.

(Abbas et al., 2018) use the texture and shape features to identify 7 types of mangos via a model designed mainly to process medical images which is B11 program. 140 features were obtained after applying PCA, the total testing accuracy is 83 %.

(Behera et al., 2019) proposed a method for automatic classification of 10 mango types using statistical features and SVM. The RGB input image goes under contrast enhancement and then transfers the color space from RGB to L*a*b color space to be able to use k-mean to detect the mango and extract 13 features. Then multiclass SVM was trained to classify 30 mango images. The accuracy 90%.

(Rizwan Iqbal & Hakim, 2022) enhance Inceptionv3, ResNet15, and VGG16 pre-trained models to classify eight mango types of 2400 samples. The images were resized into 280×260. Horizontal and vertical Sobel filters were applied to RGB images to detect the mangos edges so, all the RGB images are segmented and contain only mango. Rotation, translation, zooming, shearing, and horizontal flipping were used as augmentation techniques. The dataset was split into 85% for training and 15% for testing. The accuracy range from 96 to 99% and the highest result is 99.16% test accuracy and 0.1 test loss with 50 epochs and 32 batch sizes.

This study contributes to the classification of several mango varieties utilizing transfer learning and fine-tuning techniques. The study aims to find a model capable of differentiating different types of mangoes, even if they are of great and complex similarity, using advanced techniques. This study is the first of its kind in terms of building a hybrid CNN model with machine learning models to classify mango types. This study also makes a significant contribution to a detailed study of the dataset and provides various experiments of transfer learning techniques and their effectiveness, as well as a detailed study of each model's performance. Unlike previous studies in mango classification, this study gives detailed stages for improving images, preparing data, building models, tuning the hyperparameters, and demonstrating the impact of this on performance. Also, previous studies do not show clear matrices related to the test process and the cost of time, but in this study, we present and analyze these results deeply and analytically in which other resarchers could benefit. (Naik & Shah, n.d.) stated that there are no studies in the classification of mango types to compare his work with, and here we give an in-depth study that we hope helps researchers in the disciplines of classification, transfer learning, and fine-tuning to find this study serve them in their objectives, fields, and research area.

This paper is organized as follows in section 2 we provide a detailed explanation of the material and methods used to develop the proposed framework. The results and decision is presented in section 3 and section 4 respectively. The final words of this study are stated in section 5. At the end of the paper, all the cited references are shown.

2. Material and Method

In this section, the dataset and methodology that used in this study is illustrated in the following section.

2.1. Dataset

The imagery dataset (Hakim, 2021) is acquired from a public repository. This dataset is published in July 2021, and it's not been studied well before and this motivates us to work with it. The dataset contains Classification_dataset folder that contains eight folders for eight varieties of Pakistani mangoes which are: Anwar Ratool, Chaunsa (Black), Chaunsa (Summer Bahisht), Chaunsa (White), Dosehri, Fajri, Langra, and Sindhri. Each folder contains 200 RGB images which are quite little to train deep learning model. Figure 2 Shows samples from the dataset with the appearance of each type.

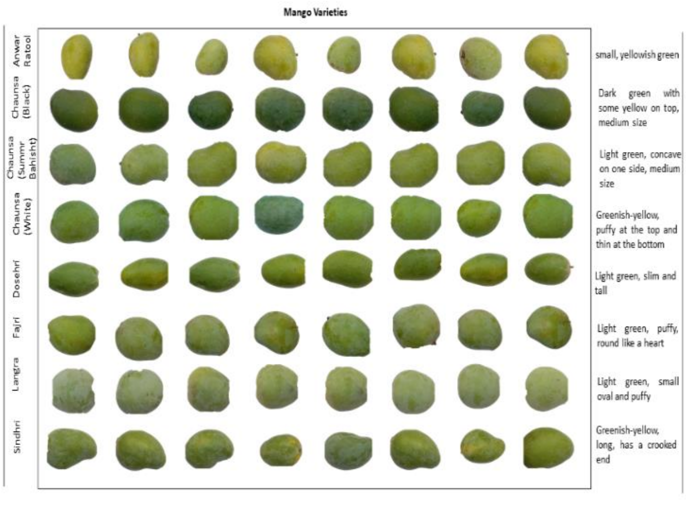


Figure 2 *Mango Types Dataset*

2.2. *Method*

The proposed methodology to build a robust multi-class classifier that can determine the different mango types take the advantage of transfer learning with fine-tuning based on pre-trained models. Two efficient CNN architectures were inspected, Resnet50 (He et al., 2016) and MobileNet (Howard et al., 2017). MobileNet has a lightweight and small-size architecture, Resnet 50, on the other hand, is deeper in-depth and size. The input to the proposed models is RGB images and the output is one of the different eight mango types. The framework that we propose consists of two parts. The first is to use the MobileNet to carry out the multi-classification process to determine the types of mangos after preparing the images to be compatible with the model. The second part of the suggested framework is to build a hybrid deep learning model that uses the state-of-art deep learning model as a feature extractor then fed it to the state-of-art machine learning models after applying PCA to carry out the multi-classification task. For this purpose, Resnet50 is used as the feature extractor and Naïve Bayes, logistic regression, and SVM with different kernel functions were used as classification models. The Resnet50 is characterized by being very deep and has residual connection nature which makes it one of the best feature extractor choices (Sharma et al., 2018). In this framework, the effectiveness of applying transfer learning and fine-tuning will be studied after applying it to two parts then comparing performance and results. Figure 3 shows the proposed methodology to perform multi-classes mango classification then study the effectiveness of transfer learning and fine-tuning approach.The hardware and software configuration are shown in Table 1.

Table 1 *Hardware and Software Configuration*

Hardware / software	Parameter
Operating System	Windows 11 pro× 64
CPU	11th Gen Intel® Core ™ 3.30 GH
GPU	NIVIDIA GeForce RTX 3060 l
Programming language	Python
IDE	Jupyter
Deep learning library	TensorFlow
Computer Vision library	OpenCV

2.2.1 MobileNet

MobileNet was the first deep learning model designed for mobile and embedded vision applications (Benjamin Planche & Eliot Andres, 2019). It is also known as Depth wise Separable Convolution Network as the network uses it as the core building block for the MobileNet layers apart from the first layer, which is a complete convolutional layer. By the mean of depthwise and pointwise convolutions, the network performs a single convolution on each RGB input channel individually. Unlike the standard convolution operation that is used in other models which rely on filtering and merging data based on convolutional kernels to create a new representation in one step, the MobileNet makes the filtering and combination in two steps. That is achieved by using depthwise convolution followed by pointwise convolution. The depth-wise applies a separate filter to each of the M input channels. It is a channel-wise of size Dk×Dk spatial convolution. Then it is followed by pointwise convolution it applies a 1×1 filter to combine the output from the depthwise convolution into a linear combination so it modifies the channel

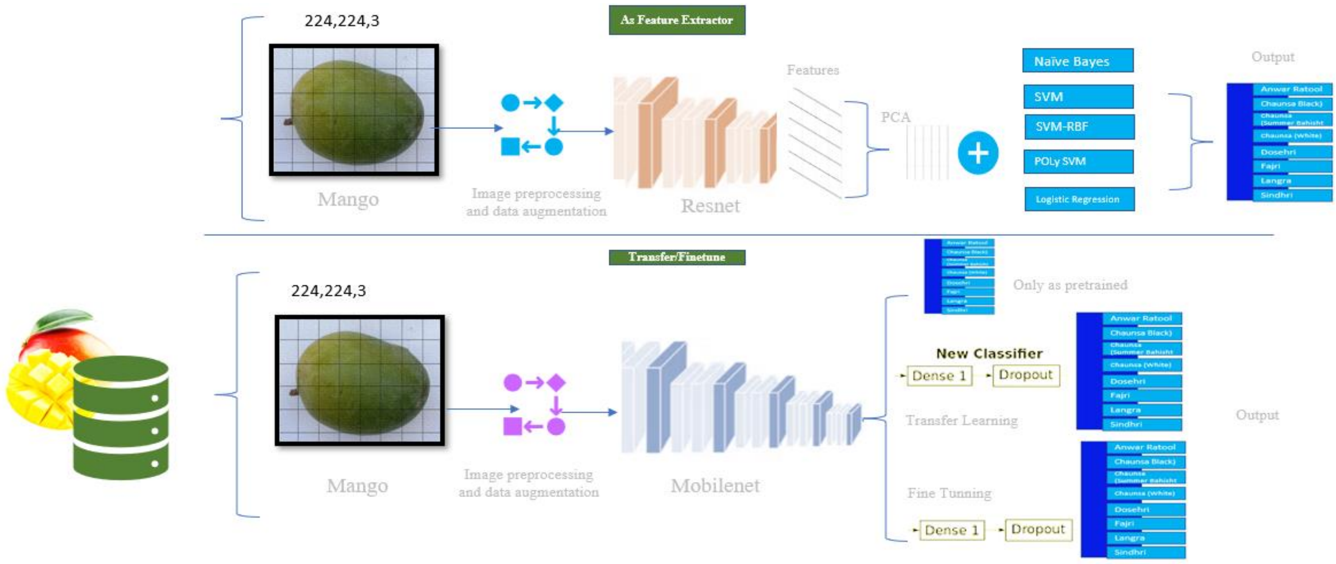


***Figure 3** The Overall Diagram of the Proposed Multi-Classification Mango Framework*

dimensions (Howard et al., 2017). The size of 3 × 3 depthwise separable convolutions is used by MobileNet. MobileNet has a total of 28 layers. Apart from the last fully connected layer, which feeds into a softmax layer for classification, all layers are followed by a batchnorm and ReLU nonlinearity. It is preferable architecture in the restricted recourses environment, as stated by (Howard et al., 2017) the depth-wise separable convolutions done in two stages is significantly decreased model size and though reduce the required computation.

The summary of the MobileNet architecture is shown in Figure 4.

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
5× Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

***Figure 4** MobileNet Architecture* (Howard et al., 2017)

2.2.2 Deep Residual Networks ResNet50

The deep residual network was introduced by (Sik-Ho Tsang, 2018) in 2015 and it was the winner of ILSVRC. It has a deep model, and it can be 152 layers in depth. Unlike the earlier deep learning models, ResNet50 (50 refers to number of layers) demonstrated skip connection to overcome vanishing gradients problem (Turgut, n.d.). Its architecture consists of 48 Convolution layers along with 1 Max Pooling and 1 Average Pooling layer. On top of each other, five convolution blocks are stacked with different filters size. The ResNet50 architecture is shown in Figure 5, with the skip/ shortcut connection.

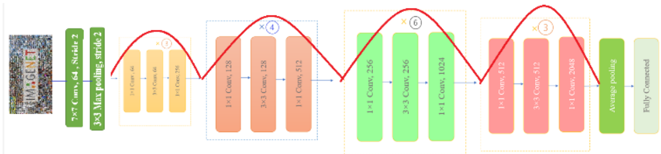


***Figure 5** ResNet50 Architecture*

2.2.3 Transfer Learning and Fine-Tuning

The larger the dataset the better the deep learning model's performance can be scored. As we have relatively not large enough dataset, we go for the transfer learning approach which is the best-trending of deep learning. Transfer learning is an approach used to enhance the performance efficiency with the limited learning images by utilizing existing knowledge from the source learner in the target task (Zhuang et al., 2019). For instance, ResNet50 takes 14 days in 90 epochs on ImageNet which consist of 1.4 M images and 1000 classes (You et al., 2017). These models were trained very well on a huge amount of data and training them from scratch is a resource and time-consuming process without a guarantee of the performance due to the limited dataset that we have. So, it is better practice to load those pre-trained models and leverage the knowledge that already the two models gain in their original task to enhance performance efficacy in our situation. Because transfer learning and fine-tuning are generally part of the same process, these two terms are frequently mixed up and used interchangeably. As we usually fine-tune after transfer learning, many call the whole process fine-tuning. However, fine-tuning is more than just using the pre-trained models and their weights. It is also utilizing the pre- knowledge but freezing some layers while training the last layers with a small learning rate to adapt the model to the current task. Visualizing the result of the convolution deep learning model gives us better insight into the whole process. Figure 6 shows the result of applying a 9 features filter on different convolution blocks, the YlGnBu color map is used to plot colored images. The right side of the images shows their feature map with 9 activations. We can notice that the filters on the earlier layers detect general features pattern, intensity, edges, corners, etc. This low-level feature can be shared among a wide range of computer vision tasks. While in the last layers as clear in same figure it contains a detailed mango object. So, the use of the pre-trained model to detect these low-level features in the earlier layer appears and further fine-tune the new add layers to address the compact mango object.

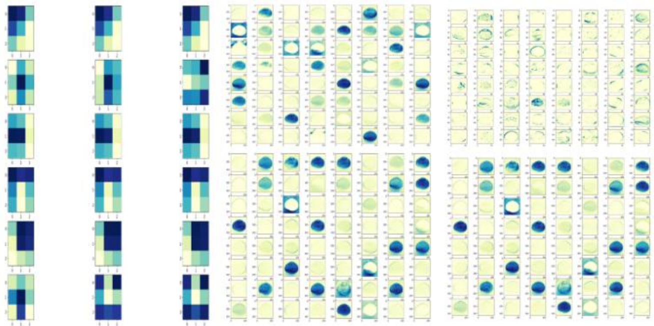


Figure 6 *The Feature map for Convolutions Block with mango image*

The following subsection demonstrates the specific stages that we follow for each part of developing the proposed framework based on the discussed methodology.

2.2.4 Experiments

To compare the effectiveness of TL and fine-tuning, we will do the following experiments on MobileNet architecture:

2.2.4.1 MobileNet Experiments

i. Use its elegant architecture and initialize the weights randomly.

Here we use the MobileNet architecture without including the top layers and without using its weights. All the layers were set to be trainable thus we have 3,206,976 trainable parameters which is the total number of parameters in the whole model. Adam is used as optimizers and the loss is set to 'categorical_crossentropy', and the accuracy is chosen as evaluation metrics. The model is fit with the training images and validation ones with different 10, 32 batch size and with 5, 50, and 100 epochs.

ii. Use the pre-trained model with the pre-trained weights (Transfer Learning)

Load the MobileNet model and load its pre-trained weights on ImageNet. Cut off the last 4 layers which is a task-specific layer and instead of it we add a Dense layer with SoftMax as an activation function to classify the 8 classes that we have in the dataset. The earlier layers were frozen, so the weights won't be updated throughout the learning stage as we want to get benefit from the model architecture as well as from the knowledge it has. This experiment was held with different batch sizes and a different number of epochs.

iii. Fine Tune The transfer pre-trained MobileNet

Load the MobileNet model and load its pre-trained weights on ImageNet. Cut off the last 4 layers which is a task-specific layer and instead of it we add a Dense layer with SoftMax as an activation function to classify the 8 classes. Freeze the weights in the first 22 layers, so the weights won't be updated as that layer contributes more to detecting general features pattern. Then the rest layers are set to be trainable, so the total number of parameters is 3,237,064 and 1,870,856 trainable parameters while 1,366,208 non-trainable parameters. Batch size 10, and 32, and the step per epoch is 132 and 42. learning rate 0.001.

2.2.4.2 ResNet50 Experiments

As discussed, before we intend here to create a hybrid deep learning and machine learning model. The ResNet50 is used as a feature extractor then the extracted features will be fed to ML models. To do this we load the pre-trained Resnet50 model with ImageNet weights without including the top layers. We extracted the features that equal to 2048 features from the last convolution 5 block and Global Average Pooling layer. The extracted features are saved in a data frame. Figure 7 presents the portion of the first row of data frame.

feature4	feature5	feature6	...	feature2039	feature2040	feature2041	feature2042	feature2043	feature2044	feature2045	feature2046	feature2047	feature2048
0.029922	0.00199	0.345766	...	0.036568	0.0	1.988224	0.142801	0.05966	0.0	0.09371	0.384578	0.035017	0.0

Figure 7 *Example of the extracted Features From ResNet50*

As shown in Figure 7 there are some zero features, and 2048 features is a huge number of features to feed the ML model and probably it degrades performance and causes overfitting. To avoid such problems, we apply the principal component analysis PCA to make dimensionality reduction (Shlens, n.d.).We study the 2048 PCA components and we found 187 principal components explain 80% of the total variance. Thus 187 features are kept in a data frame as shown in Figure 8.

PCA n°6	PCA n°7	PCA n°8	PCA n°9	PCA n°10	...	PCA n°179	PCA n°180	PCA n°181	PCA n°182	PCA n°183	PCA n°184	PCA n°185	PCA n°186	PCA n°187	Label
1.002570	-1.820766	-4.868489	-4.207813	0.677669	...	0.192824	0.484566	0.010494	0.685720	-0.239070	-0.035710	-0.002076	-0.484373	-0.842516	Langra
5.776774	-4.477458	1.651155	1.501931	-3.044599	...	-0.347239	-0.742401	-1.114967	-1.095126	0.077886	1.893518	1.089296	-0.038459	0.536149	Langra
0.089784	3.665230	-2.504724	-9.822440	-13.583184	...	-0.081180	-0.493386	-3.808086	-3.011707	0.740941	-1.613160	-1.552638	0.007485	-0.116579	Sindhri
2.385564	-7.028305	6.273039	3.614757	-3.582744	...	0.845319	1.538602	-0.568837	-0.681806	-0.618619	0.676393	1.636233	-0.214293	-0.144761	Langra
2.848694	-5.976341	-0.560865	8.953789	-5.719344	...	3.181737	0.540028	0.997123	-0.026570	-2.069035	0.338076	0.021442	4.315318	0.303050	Dosehri

Figure 8 *Features with PCA*

These features fed to Naïve Bayes, logistic regression, and SVM with radial basis function, polynomial, linear kernel function. The models are examined with 10-fold cross-validation.

2.2.4.2.1 Image preparation for ResNet50

This stage is very crucial to enhance the model's performance by further improving the data from which models will learn from.

A. Data Augmentation

To prepare images for the ResNet50 Model different image augmentation techniques were used. This step is extremely significant and has a strong impact on the DL models. Data Augmentation is a simple and effective way to add extra data to prevent over-fitting. However, for the proposed methodology horizontal flipping, rotating, and zooming techniques were used. Hence, we created new augmented data on the training database. For this purpose, 4000 random samples were generated for the ResNet50 model as we want to use it with ML models which need quite a large number of learning samples.

B. Images Enhancements

Images of mangoes in the dataset are very similar and the differences that can be seen with the naked eye are confusing. And

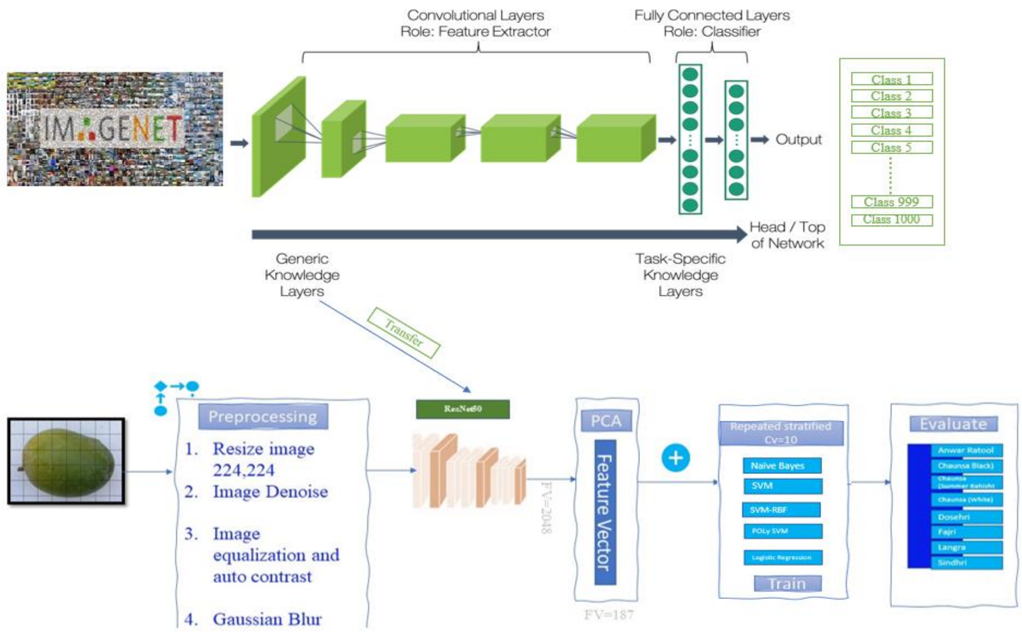


Figure 9 *ResNet50 overall experiment*

because we want to use these features in ML models, so we want to improve the images before feeding them into the ResNet50 model so that we improve the edges, colors, and clarity, and for this purpose, we have made the following enhancement:

i. Resize

Each image in each 8 types has a different sizes for example size of images in the type Chaunsa (White) folder has different values as 798 ×562 , 741× image resolution with large model can give better performance according to a study done by Wu et al(Wu et al., 2015). For this, all the images were resized into 224×224×3.

ii. General preprocessing

To make Images Denoising Equalization and Guessing Blur that remove the high frequency content. We use available functions with the mean of openCV library.

iii. ***Using the ResNet50 preprocessing***

After converting the photos to BGR, each color channel is zero-centered with respect to the ImageNet dataset, with no scaling (TensorFlow, 2021).

Figure 10 shows all the image enhancements that were applied to all images in the dataset.

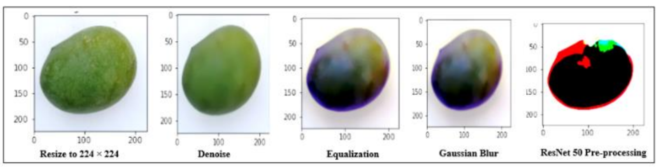


Figure 10 *Images Enhancement process*

The overall strategy that we follow for the ResNet50 Experiment is shown in Figure 9.

2.3. Evaluation Matrices

To evaluate the performance of each model under two main experiments, different performance evaluation matrices are used. The number of classifications a model successfully predicts divided by the total number of predictions is known as model accuracy. The accuracy of the training, validation, and testing data is measured which is defined as the number of classifications a model correctly predicts divided by the total number of predictions made. The precision is a fraction of relevant examples (true positives) among all the examples which were predicted to belong in a certain class. The recall defined as being a fraction of examples that were predicted to belong to a class with respect to all the examples that truly belong to that class. Then the precision, recall, and F1 are calculated for each class. Equations for accuracy, precision, recall, and F1 are given in equations 1,2,3,4 respectively. The confusion matrix for the testing data is plotted. The Confusion Matrix is a statistic for describing the machine learning model's prediction performance. It provides a comparison between the values and the classifier's estimated value. Understanding true positives, false positives, true negatives, and false negatives might aid in determining where the model went wrong. Rather of depending on the Accuracy, which is sometimes misinterpreted, this increases comprehension of the model's performance. To get a better insight of the model performance the run time of the model is measured, and the accuracy and loss plot is plotted with different numbers of epochs and batch size.

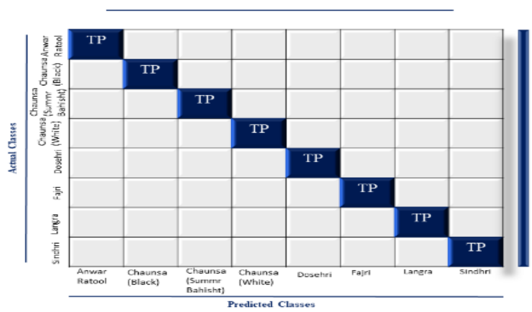


Figure 11 *Confusion Matrix of Eight Classes Analogy*

$$Accuracy = \frac{\text{correct predictions}}{\text{all predictions}} \text{(Eq. 1)}$$

$$Precision = \frac{\text{true positives}}{\text{true positives +false positives}} \quad \text{(Eq. 1)}$$

$$Recall = \frac{True\ positives}{true\ positives + false\ positives} \quad \text{(Eq. 2)}$$

$$F1 = 2 \times \frac{Percision \times recall}{precision + recall} \quad \text{(Eq. 3)}$$

3. Results

Two leading experiments were applied to classify the mango varieties using advanced techniques. In all experiment the same software and hardware configurations were used as presented in Table 1. Two experiments heavily depend on MobileNet and ResNet50 models.

3.1. *MobileNet Experimental Result*

In the first experiment, fine-tuning and transfer learning TL were studied on the MobileNet architecture with different batch sizes 32 and 10. The epochs examine with different values as well 5, 50, and 100. For all experiments with the MobileNet the dataset is split into 1320 training images, 240 samples for validation, and 40 images for testing. The experiments of the three different trails on MobileNet model without transfer learning, with transfer learning, and with fine-tune achieved different testing accuracy range from very low testing accuracy to full accuracy. We keep improving and tuning the hyperparameter of the models in every three trails to improve the testing accuracy. This is done to show how these different setting could affect the overall performance. The best testing accuracy achieved in every three trails on the MobileNet was 85%, 100% , and 100% testing accuracy respectively. The minimum run time required for good accuracy is 2:17 min. However, 17:20 min is time to train the model on more epochs and higher batch size, the model achieves 100% testing accuracy. Table 2 shows the complete results of each of the three trials of the first experiment.

Table 2 *MobileNet Experimental results*

A= Experiment on MobileNet	A (i)	10	5	132	64 %	13%	17%	0:06:38
		32	50	42	100%	85%	85%	1:03:20
		32	100	42	100%	72%	78%	2:53:54
	A (ii)	10	5	132	91%	95%	95%	0:02:14
		32	50	42	100%	99%	100%	0:19:13
		32	100	42	100%	99%	100%	0:34:09
	A (iii)	10	5	132	100%	100 %	93%	0:2:17
		32	50	42	100%	98%	100%	0:17:20
		32	100	42	100%	99.7 %	100%	0:34:49

i: MobileNet only as architecture, ii: MobileNet with transfer learning, iii: MobileNet with transfer learning and fine-tuning

The plot of every trial is shown in Figure 12.

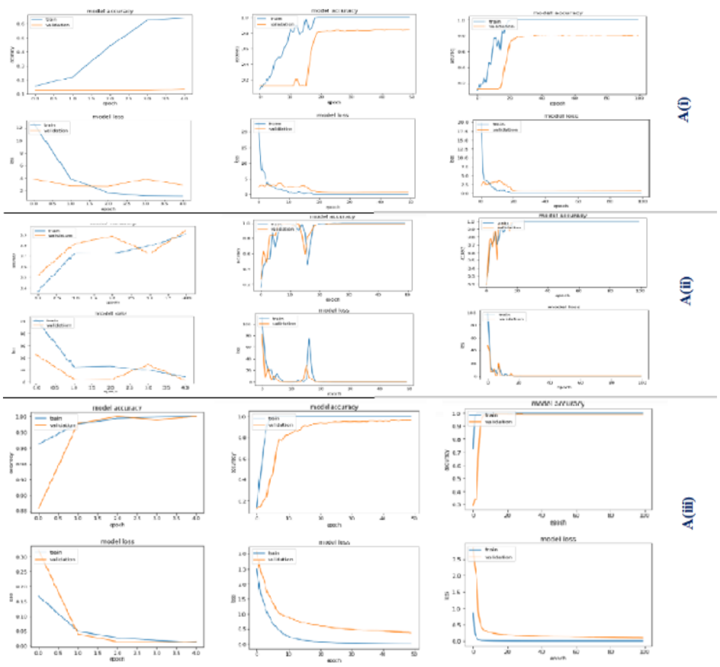


Figure 12 *Accuracy and Loss Plot of MobileNet Model trails with 5, 50, and 100 epochs respectively*

The confusion matrix for the testing images of this experiment on MobileNet is shown in Figure 13 .

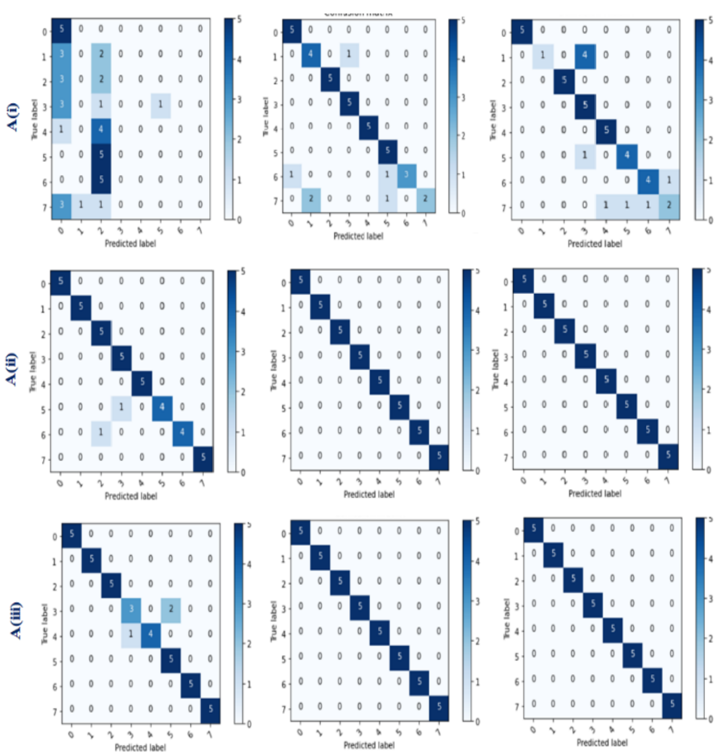
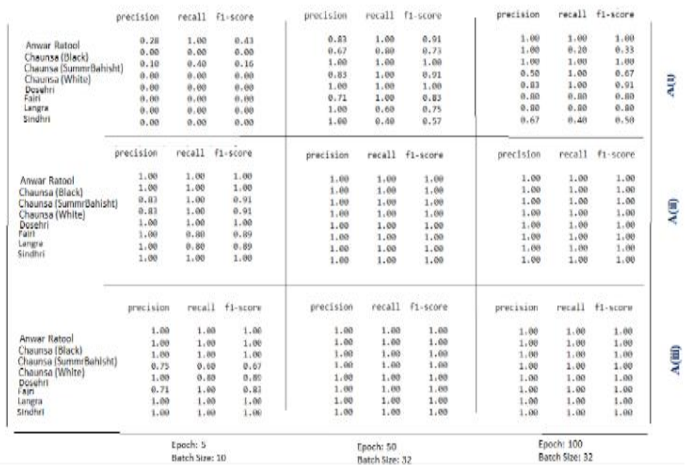


Figure 13 *Confusion Matrix for MobileNet Experiments*

Precision and recall and F1 for the testing samples are shown in Figure 14.

Table 3 *ResNet50 Performance Results*

MODEL	NO. SAMPLES	NO. CLASSES	IMAGES PREPROCESSING		TRAINABLE PARAMETER	EXTRACTED FEATURES		PCA	
ReNet50	4000	8	**APPLIED**	**TIME**	23,534,592	**NO.**	**TIME**	**APPLIED**	**N-COMPONENTS**
			Yes	20:44 m		2084	7:46	Yes	185



A(I)

	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Anwar Ratool	0.28	1.00	0.43	0.83	1.00	0.91	1.00	1.00	1.00
Chaunsa (Black)	0.00	0.00	0.00	0.67	0.80	0.73	1.00	0.20	0.33
Chaunsa (SummrBahisht)	0.10	0.40	0.16	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (White)	0.00	0.00	0.00	0.83	1.00	0.91	0.50	1.00	0.67
Dosehri	0.00	0.00	0.00	1.00	1.00	1.00	0.83	1.00	0.91
Fajri	0.00	0.00	0.00	0.71	1.00	0.83	0.80	0.80	0.80
Langra	0.00	0.00	0.00	1.00	0.60	0.75	0.80	0.80	0.80
Sindhri	0.00	0.00	0.00	1.00	0.40	0.57	0.67	0.40	0.50

A(II)

	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Anwar Ratool	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (Black)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (SummrBahisht)	0.83	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (White)	0.83	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00
Dosehri	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Fajri	1.00	0.80	0.89	1.00	1.00	1.00	1.00	1.00	1.00
Langra	1.00	0.80	0.89	1.00	1.00	1.00	1.00	1.00	1.00
Sindhri	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

A(III)

	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
Anwar Ratool	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (Black)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (SummrBahisht)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Chaunsa (White)	0.75	0.60	0.67	1.00	1.00	1.00	1.00	1.00	1.00
Dosehri	1.00	0.80	0.89	1.00	1.00	1.00	1.00	1.00	1.00
Fajri	0.71	1.00	0.83	1.00	1.00	1.00	1.00	1.00	1.00
Langra	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sindhri	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Epoch: 5 Batch Size: 10 | Epoch: 50 Batch Size: 32 | Epoch: 100 Batch Size: 32

Figure 14 *The complete classification report*

3.2. ResNet50 Experimental Results

Panda sample () is used to generate 4000 random samples to extract the features. All the images are resized and go under the proposed pre-processing discussed in the methodology section. The ResNet50 pre-trained model was loaded with ImageNet weights without including the top layer. All layers are set to be trainable, 23,534,592 is the total number of trainable parameters. The 2048 extracted features are reduced to 185 by applying PCA. The features are scaled using standard scalar then fed to ML classifiers and tested with 10 Repeated Stratified K-Fold. However, the total time to apply pre-processing with the OpenCV library is 20:44 minutes. The total execution time of the features extraction is 7:46 minutes. The total time required to train the 5 ML classifiers with 10 kfold RepeatedStratifiedKFold is 11 s. Table 3 Shows these performance results. The best-scored ML classifiers are logistic regression, linear, and RBF SVM. However, Naïve Bayes and polynomial SVM show slightly comparable performance with 10-Fold, Figure 15 shows the performance distribution of these models in the 10-fold.

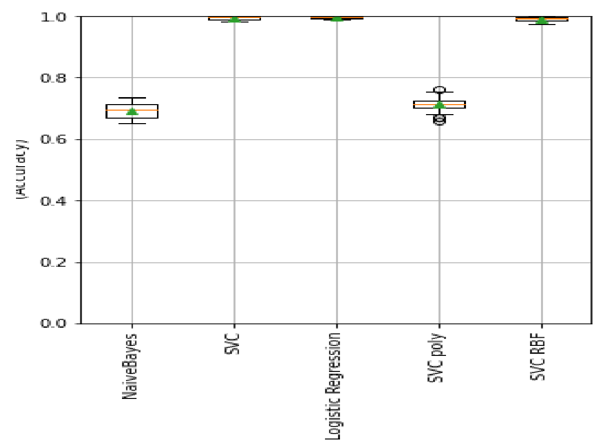


Figure 15 *Models accuracy with 10-Fold*

4. Discussion

Both Experiments that were conducted to classify 8 mango varieties gave good results. The result of the three trails on MobileNet led us even to improve the performance of the second experiment to build a hybrid ResNet50-ML classification framework. MobileNet, as discussed before, has elegant and lightweight architecture, however, using the model and initializing the weights randomly, without enhancing the transfer learning or fine-tuning techniques gave low performance compared to the same architecture with such consideration. Even the trial to improve the performance by tuning the hyperparameter doesn't contribute significant improvement. Without any doubt using a very small batch size 10 with a small number of epochs degrades the model performance. The accuracies were 64%, 13% and 17% for training, validation, and testing which reflects underfitting, as the model cannot capture the relation between input and output. The small batch size is noisy and led to generalization error due to the huge number of parameters and the with the given dataset. But as expected the running time for this model configuration is faster than other models that have different configurations, the model run time is about 6 minutes. Nevertheless, we know that the number of trainable parameters is 3,608,392 using the same configurations but with transfer learning decline run time to 2:14 m and with fine-tune to 2:17 however, the accuracy differs according to the used approach. The run time with fine-tuning is a little higher than TL and that is due to freezing and unfreezing some layers. Increasing the batch size to 32 which is pretty default in literature improves the performance with 50 epochs for the validation and testing accuracy improved from 13-17% to 85% which is quite good. However, the run time take 1:03:20 hour which is relatively costly compared to the same batch size and epochs configuration but with transfer learning and fine-tuning. With transfer learning, the testing accuracy is 100% with 19:13 m for execution time. While it takes 17:20 m with fine-tuning to score 100% accuracy. On the other hand, increasing the epochs to 100 affects the mobileNet stability when it is trained with random weights. The model is overfitted and fails to generalize to unseen data and it took about 3 h to train the model as seen in Figure 12 the gap between the accuracy and validation accuracy is high and the loss is increasing.

However, this does not appear as a problem while using transfer learning and fine-tune and the execution time is about 35 m. In most of the different experiments, the best-classified mango class is Anwar Ratool, and this was expected as it has very distinguishable features than other types.

***Table 4** Comparison Between our study and Previous Work, [1](Yossy et al., 2017),[2](Win & Misigo, 2019), [3](Naik & Shah, n.d.),[4](Pandey et al., 2021),[5](Abbas et al., 2018),[6](Behera et al., 2019), [7](Rizwan Iqbal & Hakim, 2022)*

	Year	study	Mango Types	No. Training	No. validation	No. testing	Model	Testing accuracy	Time
Referenced Study	2017	[1]	4	52	-	-	ANN	94%	40 s
	2019	[2]	5	250	-	20	Naïve Bayes	94%	-
	2022	[3] With TL	10	600	-	70	Inception v3, Xception, DenseNet and MobileNet	90% 92.42% 91,42% 88.57 %	-
	2021	[4] With TL	15	100-200 for each class	-	-	AlexNet, GoogLeNet, ResNet50, VGG16	F1 Score were only measured	-
	2018	[5]	7	140 Features extracted after applying PCA			B11 program for process medical images	83%	-
	2019	[6]	10	13 Features from 30 images			SVM	90%	-
	2022	[7]	8	2040	-	240	Inceptionv3, ResNet15 and VGG16	Range from 96% to 99%	
Our study	2022	Mobile Net	8	1320	240	40	MobileNet Transfer learning	100%	19:13m
				1320	240	40	MobileNet Fine-Tune	100%	17:20
		Hybrid ResNet50		4000 sample, 185 extracted features with 10Fold validation			Resnt50-Naïve Bayes Resnet50-LinerSVM Resnt50PolySVM ResNet50LogisticRegression	Range from 70 -100 in 10-Fold	For all 11s

For the ResNet50 results, we can see how the application of transfer learning and fine-tuning improve the extracted features and lower the run time to 7:46 m which is quite good. The pre-processing step was cost in time but contribute to results improvements. The use of ResNet50 as a feature extractor is led to better performance with less effort and time compared to the traditional methods to extract the most significant features. This method is outperformed all the results of the traditional way that used to extract features manually to classify mango types in the literature comparing the test accuracy. Compared our results with (Win & Misigo, 2019), (Abbas et al., 2018), (Behera et al., 2019) in which preprocessing, segmentation, and feature extraction were performed in the traditional way to fed features to ML models; we can see that our hybrid ResNet50 with ML models give better performance with less required effort and steps. In addition, the use of transfer learning and fine-tune shows that this approach is more effective when it is used in the proper way. In (Naik & Shah, n.d.), using the MobileNet pre-trained model the accuracy was 88.47% compared to 100 testing accuracy in our transfer learning and fine-tuning proposed work. we have comparable results with (Rizwan Iqbal & Hakim, 2022); however, we use more lightweight architecture which is MobileNet also, they use transfer learning, and the best accuracy of 99.16% was achieved with 50 epochs with 32 batch sizes. However, in our study we have achieved 100% testing accuracy by using transfer learning with fine tuning with even 50 epochs and 32 batch sizes and the plot of our accuracy and model loss compared to their models shows that our model with the same parameter configuration is more satiable due to fine-tuning. As it is clear from Table 4 most of the previous work did not take into account the validation data samples. It is true that the validation data samples are considered optional for some, but it is very useful to tune the parameter and to avoid the overfitting of the model. As discussed before our dataset was split into training, validation, and testing. In addition, we provide a complete measurement of the required training time, hyperparameter configuration, and full evaluation of the models' performance.

5. Conclusion

The applications of artificial intelligence in various fields of life have moved the world to a stage that cannot be compared with the past. Thanks, for computer vision with deep learning techniques that made sorting, grading, and harvesting tasks for different fruits and crops easier than it was before. There are many studies in the field but very few on mango varieties classification. There are various reasons, and the lack of a suitable dataset may be one of the most prominent reasons. We have looked at all the previous studies that we were able to access related to the classification of different types of mangoes and found that most of them were not completed as required. As discussed earlier there is no sufficient information about the performance evaluation process. Most of the studies ignore the effect of the learning execution time on the performance. Also, the clearness of the split of a dataset for training, validation, and testing is confusing. As a result, we didn't come across a study that shows in deep the effect of tuning the models' parameters. Therefore, we conducted a study that shows the impact of using deep learning with the application of transfer learning and fine-tuning in the automatic classification of 8 mangoes varieties. We applied two experiments, the first on MobileNet with, and without transfer learning and fine-tuning. The second is done to build a hybrid ResNet50 with logistic regression, liner SVM, Poly SVM and Naïve bayes ML classifiers. For both experiments, we have shown successfully how transfer learning and fine-tuning improve the learning and generalization of the models with low learning time, small batch sizes, and epochs numbers. Compared to the use of MobileNet architecture without transfer learning or fine-tuning which makes the model overfit easily on the training data even with different hyperparameter tuning considerations. And trying to improve its performance made the learning time jump to 2 hours without a real improvement in the testing Accuracy. On the other hand, applying transfer learning by using the pre-trained weights with the same architecture improves the training and validation accuracy and gives 95% testing accuracy while minimizing the learning time to 2:05m with only 5 epochs and 10 batch sizes. The best testing accuracy with transfer learning and fine-tuning is 100%. The fine-tuned MobileNet model is more stable and requires about 17m for the learning phase to score 100% testing accuracy. For the second experiment, the hybrid ResNet50 with ML classifiers gives very good results for all classifiers with 10 stratified repeated kfold. The execution time required to extract 2048 features with pre-trained ResNet50 from 4000 pre-processed and enhanced random samples of the 8 mango classes is 7:46m. Even though both experiments give good and comparable results. Therefore, the fine-tuned MobileNet and the hybrid ResNet50-ML can be deployed in the real-world application. However, the choice between the two models is a trade-off between the execution time and the required stages. While that fine-tuned MobileNet requires no significant preprocessing and post-processing techniques other than the resizing of the images, the pre-trained ResNet50 requires important preprocessing for image enhancement and post processing to apply PCA to reduce the dimensionality of the features. Fine-tuned MobileNet is faster and requires minor effort to acquire a 100% testing accuracy to classify the 8 mango varieties, compared the other models presented in this study. The study that we have presented in this paper can be used in the pre-sorting, grading, and classification processes, and can also be used for purely research purposes. We believe that we have been able to present a new and good study in the process of mangoes identification with the most advanced trend in deep learning techniques. Our proposed fine-tuned MobileNet is outperformed better than all the previous models in the literature in terms of testing accuracy and time.

References

Abbas, Q., Iqbal, M., Niazi, S., Noureen, M., Muddasir Iqbal, M., Saeed Ahmad, M., & Nisa, M. (2018). Mango Classification Using Texture & Shape Features. In IJCSNS International Journal of Computer Science and Network Security (Vol. 18, Issue 8). http://www.eletel.p.lodz.pl/programy/mazda/index.php?action=mazda

Behera, S. K., Sangita, S., Rath, A. K., & Sethy, P. K. (2019). Automatic Classification of Mango Using Statistical Feature and SVM. In Lecture Notes in Networks and Systems (Vol. 41, pp. 469–475). Springer. https://doi.org/10.1007/978-981-13-3122-0_47

Benjamin Planche, & Eliot Andres. (2019). Hands-On Computer Vision with TensorFlow 2 (1 st). Packt Publishing.

Hakim, A. R. I. H. M. (2021, July 6). Mango Variety and Grading Dataset. Https://Data.Mendeley.Com/Datasets/5mc3s86982/1.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. https://doi.org/10.1109/CVPR.2016.90

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. http://arxiv.org/abs/1704.04861

Naik, S., & Shah, H. (n.d.). Classification of Mango (Mangifera Indica L.) fruit varieties using Convolutional Neural Network.

Pandey, C., Sethy, P. K., Behera, S. K., Rajpoot, S. C., Pandey, B., Biswas, P., & Panigrahi, M. (2021). Evaluation of Transfer Learning Model for Mango Recognition. Smart Innovation, Systems and Technologies, 213 SIST, 467–474. https://doi.org/10.1007/978-981-33-4443-3_45

Rizwan Iqbal, H. M., & Hakim, A. (2022). Classification and Grading of Harvested Mangoes Using Convolutional Neural Network. International Journal of Fruit Science, 22(1), 95–109. https://doi.org/10.1080/15538362.2021.2023069

Sharma, J., Granmo, O.-C., & Olsen, M. G. (2018). Deep CNN-ELM Hybrid Models for Fire Detection in Images. ICANN.

Shlens, J. (n.d.). A Tutorial on Principal Component Analysis.

Sik-Ho Tsang. (2018, September 15). Review: ResNet — Winner of ILSVRC 2015 (Image Classification, Localization, Detection). Https://Towardsdatascience.Com/Review-Resnet-Winner-of-Ilsvrc-2015-Image-Classification-Localization-Detection-E39402bfa5d8.

TensorFlow. (2021). tf.keras.applications.resnet50.preprocess_input. https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet50/preprocess_input

Turgut, Z. (n.d.). A THEORETICAL COMPARISON OF RESNET AND DENSENET ARCHITECTURES ON THE SUBJECT OF SHORELINE EXTRACTION.

Win, O., & Misigo, R. (2019). Classification of Mango Fruit Varieties using Naive Bayes Algorithm. Published in International Journal of Trend in Scientific Research and Development (Ijtsrd), 5, 1475–1478. https://doi.org/10.31142/ijtsrd26677

Wu, R., Yan, S., Shan, Y., Dang, Q., & Sun, G. (2015). Deep Image: Scaling up Image Recognition. http://arxiv.org/abs/1501.02876

Yossy, E. H., Pranata, J., Wijaya, T., Hermawan, H., & Budiharto, W. (2017). Mango Fruit Sortation System using Neural Network and Computer Vision. Procedia Computer Science, 116, 596–603. https://doi.org/10.1016/j.procs.2017.10.013

You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., & Keutzer, K. (2017). ImageNet Training in Minutes. http://arxiv.org/abs/1709.05011

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2019). A Comprehensive Survey on Transfer Learning. http://arxiv.org/abs/1911.02685