International Journal of Information Retrieval Research
Volume 12 • Issue 3

Flutter-Based Real-Time Mobile App for Fruit Shelf-Life Prediction (FSP) Using Multi-Modality Imaging

Varsha Yogesh Bhole, Sir Padampat Singhania University, India*
https://orcid.org/0000-0002-6673-669X

Arun Kumar, Sir Padampat Singhania University, India

ABSTRACT

The present work uses flutter and transfer learning-based techniques for predicting the shelf-life of fruits in real-time by reducing the demand for huge number of samples and longer training durations. The work uses mango fruit as a case study by capturing two types of images axiomatically through smartphone rear camera and a Seek thermal camera. A set of models based on transfer learning were created, then trained, and post-training quantization for optimizing the model size and low latency was applied. The models were subsequently converted into TFLite format with TF-Select Ops and deployed in the Flutter app code, thereby generating the APK files for real-time execution on mobile devices. The performance metrics of the models were evaluated on accuracy, model size, and latency. With a minor level of degradation in accuracy, the model with quantization outperforms and achieves better latency and 12x times reduction in size. The overall accuracy achieved for both the cameras (normal and thermal) on mobile device was 90% and 93.33%, respectively, for predicting the fruit shelf life.

KEYWORDS

Deep Learning, GPU, Mobile Application, Quantization, Shelf-Life Prediction, Thermal Imaging, Transfer Learning

1. INTRODUCTION

Deep learning (DL) and flutter are the high-tech solutions in computer science. Deep learning methods have significant strides in visual identification applications. Detecting and categorizing image datasets with deep learning is progressive research (Abdel-Hamid et al., 2014; Alzu'bi et al., 2020; Zhang et al., 2019). DL offers a solution to accomplish artificial intelligence enabling machines to solve problems in such a way as the human brain does. It has been implemented in several areas of people's day-to-day lives. For example, Siri and Alexa, the voice-based apps are utilized to recognize the speech to help and enable people to communicate with devices by speaking instead of keyboard based inputs.

An ordinary work in image processing is recognizing the same kinds of items using machine learning techniques for classifying and clustering the fruits (Bhole et al, 2020b). Previously, a thermal imaging idea in Naik and Patel (2017), machine learning with a color feature extraction model (Bhole et al, 2020a) and deep neural network (Bhole et al, 2020c) have been suggested and manifested for evaluation of quality and categorizing maturity of the mango (Behera et al., 2020; Sahu and Potdar,

DOI: 10.4018/IJIRR.298023 *Corresponding Author

2017; Sultana et al., 2017; Bhole and Kumar, 2020d). Numerous techniques have been employed for identifying the quality of fruits with maturity levels through RGB imaging, either under-controlled or real situations (Behera et al., 2020; Sahu and Potdar, 2017; Sultana et al., 2017; Intaravanne et al, 2012). Particularly, these non-destructive methods have been dependent on the examination of visible and spectral imaging using pattern analysis. Moreover, very little work has been performed based on thermal imaging for fruit quality evaluation. The benefit of the thermal imaging approach is that it is contactless, non-destructive, faster, and also works well in dark environs to lighting environments as well as helpful in addressing the social problems and creating the applications seamlessly. It offers an opportunity for creating new ways to inspect processes that have not been even possible previously.

Furthermore, the number of published works, particularly on the evaluation of shelf-life of fruits, is yet restricted to particular cases. Fan and Zhou (2011) utilized Near-Infrared Spectroscopy to inspect the trait of apple at different periods of shelf life and obtained the accuracy up till 100%. But, the use of Near Infrared Spectroscopy is expensive and not handy technique. However, this is a destructive approach which was used at the laboratory level and can't be useful for real-time evaluation. Zhang et al. (2013) employed VIS/NIR spectrometer to assess the different stages of Malus Asiatica Nakai fruit by observing the variation in pectin of the fruit. However, this is a destructive approach which was used at the laboratory level and can't be useful for real-time evaluation. Nandan Thor (2017) predicted the shell life of banana by extracting color features along with 7 machine learning methods and claimed the maximum accuracy as 52%. This approach is non-destructive but the features are extracted manually through image processing techniques by using external surface parameters with very little accuracy.

The automatic shelf-life prediction system has become essential, mainly in the agriculture and food business to reduce the human-intensive work processes and this can be done on mobile devices (Karar et al., 2021) to enhance efficiency. But, this would need sophisticated feature identification process to evaluate fruit shelf-life using similar features that have a high color resemblance. Therefore, customized machine learning models are required to evaluate fruit's shelf-life and address the intricate features (Bhole et al., 2021) of images for particular applications.

DL centered classifiers, like Convolutional Neural Network (CNN), enhance the classification results for numerous objects (Krizhevsky et al., 2012; Lecun et al., 2015). Amongst several methods of deploying DL models for production, the simplest solution is to install it on mobile phones. Besides the published applications, very few mobile device applications have been designed particularly for fruit quality monitoring activities. In the field of grapevine, smartphone application has been utilized for automatic counting of berries (Grossetete et al., 2012). Another study developed an android application to analyze the quality of apple (Karwa et al., 2016). By (Intaravanne et al., 2012; Intaravanne et al., 2015), two applications have been designed to evaluate the ripeness of banana fruits and to find out the nitrogen content in the rice leaves based on the assessment of color features. Watermelon Classify (Zeng et al., 2014), a mobile app that examines acoustic signals to determine the ripeness of watermelon. Mobile applications like Fruit-Checker, Color Detector, Colorimeter, and Catch Color are available to roughly measure fruit quality under constrained settings on the smart mobile phone (Google Play store) which is due to enhanced growth of smart mobiles (Global mobile statistics). But, till now all the existing applications are based on the android platform and none of them have been deployed on any other platform for evaluating the shelf-life of fruits.

With mobile devices becoming more universal, new technology is required which influences mobile businesses with the fastest lap for development of the market. For a couple of years back, mobile developers have been struggling in developing separate code bases for each of the platforms like iOS, Android, Blackberry, and Windows Phone. But, in late 2018, Google made an announcement of a cross-platform development framework i.e. flutter. Flutter is a free, open-source, and completely customer-centric platform. It gives a solution with a single code base for both the worlds (Android and iOS) for hardware-accelerated graphics and user-interface where the users could receive an answer within a few touches. While deploying DL models on a mobile, three aspects have had to

consider: accuracy, model size, and latency. Accuracy is the major factor for any of the systems and if the size of the model is large, it is difficult to deploy it on the device. Also, high latency would cause inconvenience to the users. Presently, companies like Ali Baba, IBM, Hamilton Musical and of course Google themselves are using flutter in their production apps.

Currently, most of the existing fruit shelf-life prediction work is based on human judgment or through RGB imaging systems. The information received from these approaches such as through human speculation which is subjective in nature and RGB imaging system provides inference based on external features only. This information is not useful to find the internal features of the fruits which become necessary for the fruits like kiwi, avocado, or the Langra mango whose outer surface color remains same after it ripens. This is not adequate for fruit's shelf-life prediction through RGB imaging in a real-time production system which causes post-harvest loss due to lack of technological applications. Almost all existing work has limited their use to explore quantizing models during training; none of them have been focused on the post-training optimization previously in real-time applications.

Inspired by the hurdles and intricacies cited in the literature, the proposed system addresses these challenges by identifying internal features of the fruits specifically in a non-destructive manner through thermal imaging technique and post-training optimization through floating-point and integer quantization based on GPU or TPU. To harness the power of deep learning and open-source flutter framework for solving the problem of real-time evaluation of shelf-life, the proposed work is to develop a real-time mobile app based on RGB and thermal imaging by using pre-trained models through transfer learning. This application shall be beneficial for consumers who can buy the right product and use it in order to pay the correct price for it, as well as in mass production units.

The significant contributions in the paper are manifold:

- A robust mobile thermal imaging system is proposed with transfer learning model which uses mobile camera to find internal characteristics of fruits through temperature values with a non-destructive approach and also creates the proposed algorithm much faster because of the combination of thermal imaging with a transfer learning model.
- The proposed algorithm generates the compressed flat buffer model by saving the trained model into .h5 format and then converting it into .tflite with the TFLite converter, which makes the model up to three times smaller and that will be easy to deploy it in the designed app code.
- In the best scenario, the purposed work has achieved the quantized model even after the training through optimization techniques as well as with the use of TF-select ops to overcome unsupported ops resulting in reduced model size up to 10x to 12x along with low latency.
- A real-time application using cross-platform framework has been developed i.e. flutter which provides a robust solution with a single codebase for mobile (Android and iOS), web, and desktop app with the use of GPU delegates resulting in lower latency and improve power efficiency.
- Finally, the APK has been generated for the deployment of app on mobile devices which runs without an Internet connection. The robustness of the proposed system is it is generic and has been verified in a real-time on the created mobile app. The proposed system is implemented using python and dart language.

The remainder of the paper is structured as follows: Section 2 briefly explains the overall process and implementation details for real-time prediction of fruit's shelf-life. The Section 3, describes the experiment performed to estimate and discuss the experimental outcomes and finally the, Section 4 with conclusion with a shade on future directions.

2. Materials and Methods

The patented proposed system (Bhole and Kumar, 2020e) is based on the implementation of real-time mobile app to predict fruit's shelf-life. It consists of four phases, as mentioned below and depicted in Figure 1.

1. Preparing the dataset (preprocessing it and splitting into train and validation set).
2. Building a model using transfer learning for shelf-life prediction.
3. Saving, converting, and optimizing a trained model.
4. Running/deploying the models on mobile devices.

2.1. Preparing the Dataset

The deep learning analysis requires good quality data which must have high-resolution images. But, no specific dataset exists for shelf-life evaluation which can be used directly. So, first it is created. As creating a high-quality dataset is a very lengthy and tedious task, so from an experimental point of view the present work concentrates only on the mango fruit and that balances the tradeoff between size vs. quality dataset (i.e. big but low-quality as well as small but good-quality dataset). Figure 2(a) presents the layout of the proposed mobile-based stand-alone automatic image capturing system through mobile RGB and low-cost thermal camera and Figure 2(b) represents a portable arrangement for shelf-life prediction through mobile app.

Figure 1. Complete procedure of proposed shelf-life prediction system

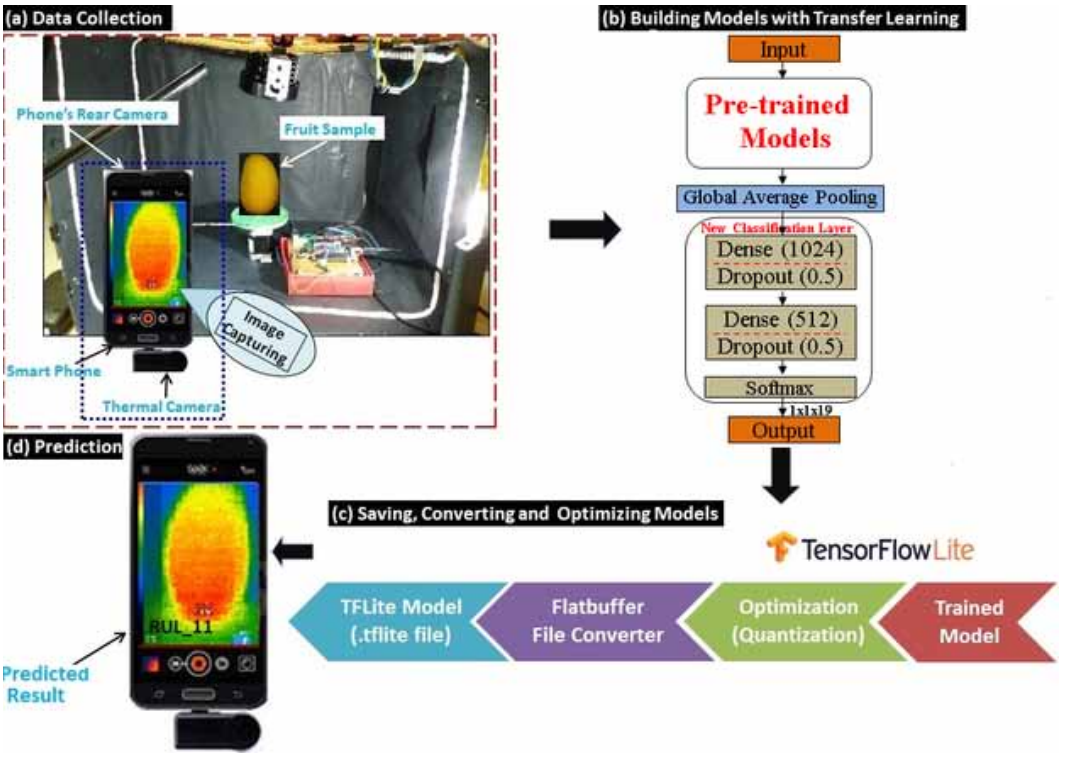
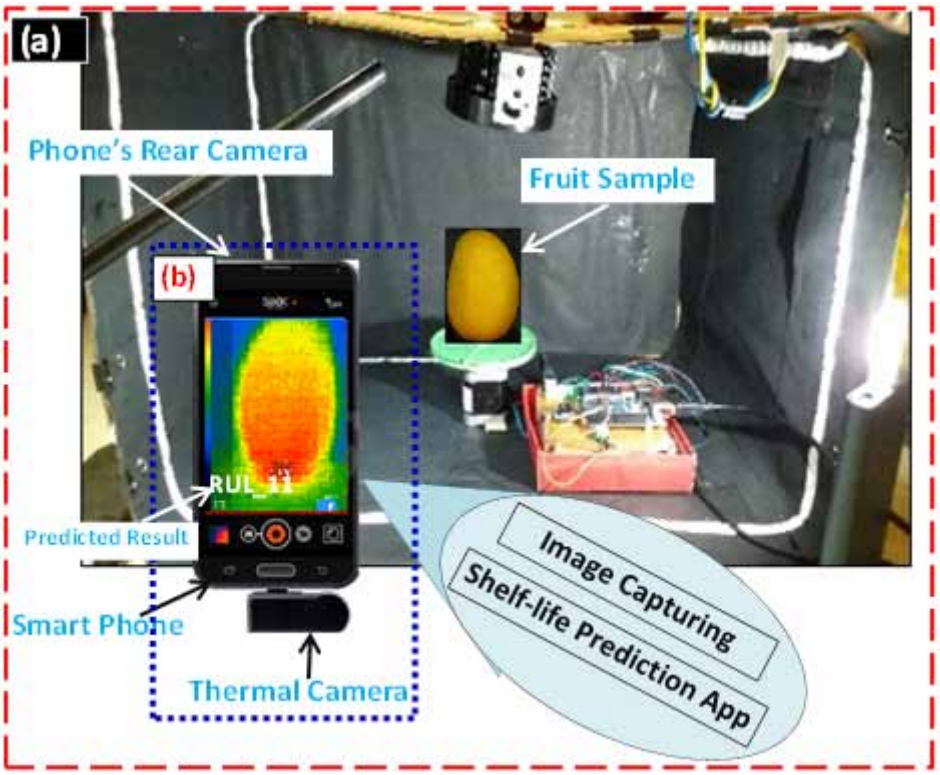


Figure 2. Prototype of mobile-based (a) stand-alone automatic data collection system (b) portable structure for shelf-life prediction through the mobile app



The stand-alone system would be helpful at the mass production or at export houses to pinpoint the remaining useful life of the fruits. The portable system seems a boon for people to purchase the best quality fruits without getting fooled by the seller as well as it could be an assisting tool for novices in the horticultural field. Here, RGB and thermal images have been recorded automatically from all the directions as proposed in (Bhole et al., 2020b) because the surface color (i.e. maturity) of entire mango fruit is not similar as well as the surface color of mango fruit varies from green to yellow from its immature to ripe states.

However, evaluating shelf-life via RGB imaging absolutely fails in case of 'Langra' mango whose surface color remains green in all its states. To overcome this limitation, we suggest and experimentally prove for the first time the concept of thermal imaging for the prediction of the shelf-life. We have formed two datasets, one is RGB and another one is thermal. We have taken the images for 19 days and defined the 19 classes' viz. RUL-1 (Remaining Useful Life-1) to RUL-18 (Remaining Useful Life-18) and No-Life. The classes reflect how much life is remaining for the particular fruit. Complete details of the sample collection and preparation, as well as image acquisition have been reported in the previous work (Bhole et al., 2020d).

Data splitting: Overall, more than 10,000 images have been obtained for both RGB and thermal datasets, but each class does not have the same number of images. So, for even distribution of images, 240 images in every class have been taken. Then dataset has been separated into training and validation set with the ratio 75:25. It is necessary to have a separate test set that can be utilized after the training has been finished to show the generalization power of the model. However, as a

mobile application has been designed, so there is no idea that in reality users will use what type of images, and hence, actual utilization of it is not beneficial to verify the accuracy.

Pre-processing: Before training, two pre-processing steps have been performed that are important together for training and validation set i.e. resizing and normalization. Usually, the images should be down-sampled into the size of a pre-trained model, herewith 224 x 224 or 227 x 227 according to the model. Furthermore, to increase the robustness of the network and to enhance the generalization ability on unseen samples, normalization has been done by dividing the channels by 255 to overcome the true color of images. Then, batch sizes of 16 have been created, divided the data into batches, and performed shuffling for the rearrangement of the images randomly.

2.2. Building a Model Using Transfer Learning for Shelf-Life Prediction

A model for training a CNN have been created to predict the shelf-life of mangoes by categorizing them according to the remaining edibility days of fruit based on the fruit's shelf life. Training of CNN has been adopted through transfer learning mechanism (Pan and Yang, 2010). In this work, the core models have been adopted such as SqueezeNet (Iandola et al., 2016), DenseNet121 (Gao et al., 2017), MobileNet (Howard et al., 2017), NasNetMobile (Zoph et al., 2018), ShuffleNet (Zhang et al., 2018), and MobileNetV2 (Sandler et al., 2018) that are trained before-hand on the ImageNet dataset (Krizhevsky et al., 2012) which consists of 1000 image categories and more than 1.2 million images (Figure 3). This is useful for expanding the dataset which has a small number of images.

In this study, the proposed model wherein only the classification layer has been learned by taking the core pre-trained models as the feature extractor and after that added new dense layers with the number of classes, which is just 19 (Figure 1b). The performance has been validated by comparing it with the weights which are trained before-hand on the ImageNet model. The blueprint of the last dense layers embedded in the proposed network are presented in Figure 4 that consists of global average pooling (GAP) followed by two dense layers with dropout = 0.5 and finally output classification layer with softmax activation. GAP determines the average value from each feature map and generates the vector (GAP_v) which leads to reduction in parameters and computation substantially with better generalization performance. Considering that GAP does not consist of any parameters for optimization which makes possible to overcome with overfitting problem as well as it summed up the spatial information, so more robust for spatial transformations of the input. The mathematical representation of GAP is illustrated in Equation (1) (Lin et al., 2013) where r-rows and c-columns of the feature map respectively and GAP_v implies the item of the vector.

$$GAP_v = \frac{1}{r*c}\left(\sum_{i}^{r}\sum_{j}^{c} p_{ij}\right) \tag{1}$$

$$Feature\_map_{i,j,k} = max\left(\left(weight_k\right)^T * p_{ij}, 0\right) \tag{2}$$

ReLU activation function is utilized for accelerating the training operation (Kingma and Ba, 2014) and specified in Equation (2) where p_{ij} is the element at index (i,j) and k represents the channel index. Then additional technique i.e. dropout has been used to control the overfitting problem which may cause by the small dataset (Zhao et al., 2019). Then after building the model, the authors trained the model with TensorFlow by setting some hyper-parameters as categorical cross-entropy as the loss for RGB as well as thermal dataset. During training, selecting the number epochs is challenging with respect to generalization performance on unseen data. For handling this, the loss performance

Figure 3. Designed fine-tuned architectures (a) DenseNet121 (b) MobileNet (c) SqueezeNet (d) MobileNetV2 (e) NasNetMobile (f) ShuffleNet

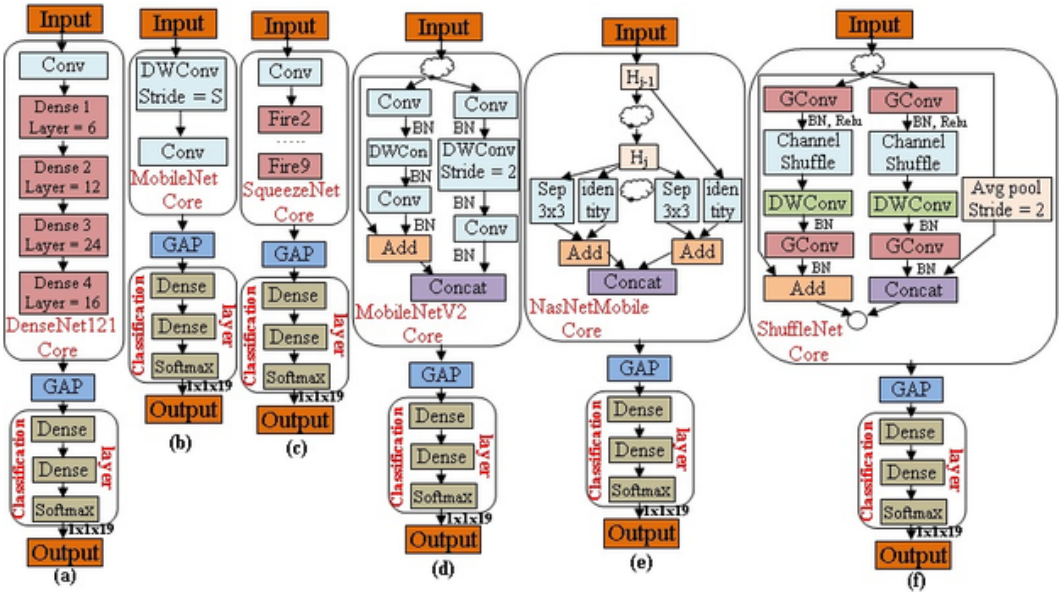


Figure 4 Blueprint of last dense layer

```
# Add some new fully connected layers
x = GlobalAveragePooling2D()(x)
# Add a fully connected layer with 1024 hidden units and ReLU activation
x = Dense(1024,activation='relu')(x)
# Add a dropout rate of 0.5
x = Dropout(0.5)(x)
# Add a fully connected layer with 512 hidden units and ReLU activation
x = Dense(512,activation='relu')(x)
# Add a dropout rate of 0.5
x = Dropout(0.5)(x)
# Add a final layer with softmax activation for classification
preds = Dense(19, activation='softmax')(x)
model = Model(base_model.input,preds)
```

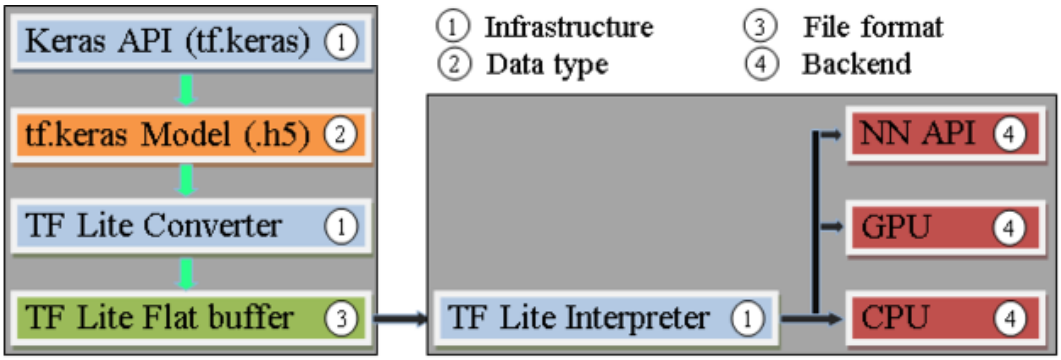
on validation images has been observed and finalized the number of epochs which contributed in the best accuracy while even continuing the generalization on unseen images. Model tuning is a simple procedure which gives optimized performance with respect to inference time and use of memory. However, the authors fine-tuned the network with predefined options of adam optimizer (Kingma and Ba, 2014) to which the 0.0001 learning rate has been passed.

2.3. Saving, Converting and Optimizing Trained Model

The steps to flatten the model for easy deployment on mobile devices are depicted in Figure 5. First, the authors exported the trained model which has been trained using TensorFlow with the Keras APIs to save it into model format. These models save in three ways as a Keras model, Saved model, or a set of concrete in-memory functions for later use (TensorFlow). In current study, as Keras is used, so it is H5 format with the very simple way model.save('filename.h5'). When the SavedModel is exported, it generates MetaGraph which is a data flow graph that comprises a list of classes along with classification signatures for prediction as this is nothing but the image classification problem. Moreover, the authors converted these Keras model to produce the TFLite model using the TFLite converter which is saved into .tflite format. Then, finally this TFLite file has been saved by writing it to the file system. This .tflite format is a flat buffer which is used on mobile device along with the backend i.e. GPU. So, this model can be instantiated from the saved Keras model based on how the model is represented in order to get an output.

Optimization: However, these models are ended up with 32-bit floating-point models. Considering that these models are going to deploy on mobile devices where resources (i.e. processing and power) are limited and efficiency of computation can become a major concern, optimization is necessary for their better working on mobile devices. There are several techniques to achieve these types of optimization. However, the authors focus on two optimization techniques to achieve acceleration: 1) post-training quantization techniques which converts model into fixed point model and 2) by using TF Lite delegate i.e GPU delegate with device in-built GPU.

Figure 5. TensorFlow Lite conversion process graph



2.3.1. Conversion of Fixed-point Model Which Supports CPUs

Here, rather than quantizing a model during training and effectively changing the training code, the authors quantize it as a part of the process of converting the model to the TFLite formats which

converts all the floats in the weights of the model into integers. The authors set the default optimization mode where it figures out the best balance between size and latency. The subsequent model optimizes reduced precision representations of weights for both storage and computation. Sometimes in the model conversion process, the framework supports a limited number of TensorFlow operations and to overcome this problem, the authors have used built-in TF select ops.

2.3.2. Use of GPU Delegate

One way of accelerating the execution of inference on mobile devices with compute-heavy deep learning models is the use of TF Lite delegate i.e GPU delegate which accelerates models on devices that have an available GPU backend on device. This backend leverages "OpenGL 3.0 ES" compute shader on android and for iOS uses "Metal". The framework automatically makes the balance between delegation of operations to the GPU or the CPU accordingly. But the cost of switching leads to higher latency as well as the size of the model is bigger. Authors deployed this part in the flutter app code.

2.4. Running/Deploying the Models on Mobile Devices

While creating a solution with machine learning or deep learning, building models are just a small part of what is required to create for production systems i.e. developing a real-time application. By sequentially following the previous steps, models have been built, converted, optimized, and saved them through TFLite technology. Then, the converted .tflite model has been deployed into the mobile app code for which the authors have used the flutter cross-platform framework. Cross-platform means developing an app for both Android and iOS using just one codebase. Flutter creates iOS and Android apps in the background and updates them as the Flutter app is created. The steps for creating an app and deploying it on mobile devices are as follows:

1. Installation and configuration of Flutter SDK.
2. Adding TFLite file and class label file in the asset folder in the app code.
3. Adding three important functions for loading tflite model, capturing the image and classification of captured image in main.dart file to work the app.
4. Accessing TFLite interpreter for running inference and improving the inference performance by setting GPUDelegate as backend.
5. Generate the APK to deploy the app on mobile devices, once the Shelf-Life application is ready.

3. Results and Discussion

3.1 Experimental Evaluation

In the current study, the experimental evaluation has been carried out by new own generated dataset, as there is no benchmark dataset available related to the study undertaken for the shelf-life of fruits. The authors make use of smart phone and more cost-effective SEEK thermal camera for the creation of the datasets as explained in Section 2.1. The experimental set-up for the same is represented in Figure 1(a). The use of double cameras provides a blending of traditional and thermal camera techniques. Thermal image confers more details whenever fruits have remarkable changes in the temperature; various shades of color in the fruit image signify distinct temperature conversely. The dataset consists of multi-modal images i.e. RGB and thermal with 19 categories viz. RUL_1 to RUL_18 and No_Life based on the remaining edibility days of the fruit. In the earlier work, authors have studied the maturity and grading of fruits with the SqueezeNet model (Bhole and Kumar, 2020d). Despite that, in the current work, authors hypothecate that the pre-trained framework through transfer learning via thermal imaging mechanism can predict fruit's shelf-life well even with the same colors and patterns. Here, the robustness of the thermal technology compared to RGB has been shown as well as test the generalization of the proposed approach in terms of other varieties of mangoes (e.g. Alphonso).

Authors mainly concerned on the pre-trained models (DenseNet121, MobileNet, SqueezeNet, MobileNetV2, NasNetMobile, and ShuffleNet) as explained in Section 2.2 for transfer learning with the weights is set as "imagenet". So, first imported them and added new dense layers with the number of classes' as 19 as demonstrated in Figure 3. After the training has been finished; the performance has been measured on validation data by importing the metrics from sklearn library. Then, the trained model has been flattened to ".tflite" file for easy deployment on mobile devices (Figure 1c) and results have been displayed on the device (Figure 1d).

Implementation Details: The experiments have been performed using Keras API over TensorFlow in Python 3.7.2 and MATLAB 2019a, Natick, USA for training the models and analyze the results with the help of GPU: NVIDIA GEFORCE GTX 730M with 8 GB RAM. Moreover, to develop the mobile app, the experimentation has been performed through Dart with Flutter.

Optimization parameters: The hyper-parameters have been standardized for enabling the fair comparison within the outcomes (with respect to validation accuracy) of all experiments and making it best suited to the small datasets as mentioned in Table 1. The values have been defined by trial and error as these are closely based on the problem, the datasets, and the models which have used.

Table 1. Empirical optimization parameters

	DenseNet121	MobileNet	SqueezeNet	MobileNetV2	NasNetMobile	ShuffleNet
RGB Dataset						
Batch	16	16	16	16	16	16
Epochs	15	15	20	10	15	5
Thermal Dataset						
Batch	16	16	32	16	16	32
Epochs	15	15	25	15	15	10

Table 2. Common empirical optimization parameters

Parameter	Optimizer	Loss Function	Learning Rate	Momentum	Activation Function
Value	Adam	categorical cross-entropy	0.0001	0.9	ReLU

As recorded in Table 1, the best choice for batch size is 16 for training all models except the SqueezeNet and ShuffleNet model on thermal dataset which is set to 32 which specifies the number of training inputs for the optimizer. EarlyStopping() function has been employed to handle the over-fitting issue which means that the models significantly performed well on training samples despite that, not on validation samples. The training gets stopped when there is no significant improvement in the validation accuracy after 5 epochs. Furthermore, the common optimization parameters for RGB and thermal datasets have been tabulated in Table 2. The classification accuracy has been judged on validation data with three optimizers' viz. Adam, Rmsprop, and Sgdm. However, the best accuracy amongst the proven optimizers has been recorded for Adam, so is chosen with the loss function and learning rate are set as categorical_cross-entropy and 0.0001 respectively. The small learning rate

causes more training time, but results in achieving more accuracy. The default value of the momentum i.e. 0.9 has been selected which identifies how speedily the adam optimizer convergences to get the final results.

3.2. Experimental Results

3.2.1. Experiment1: Training Results

Here, Figures 6 and 7 exhibits the rate of learning in terms of validation accuracy (Figure a) and validation loss (Figure b) for all RGB and thermal models/architectures - DenseNet121, MobileNet, SqueezeNet, MobileNetV2, NasNetMobile, and ShuffleNet respectively which have been assessed on the validation dataset. As results could find from Figures 6 and 7, the curves are not smooth which is because this data is naïve to the models. Considering that the EarlyStopping approach has been employed, the epoch count varies for each model as recorded in Table 1. In every model, the performance has been inferior at the first epoch but after some epochs, accuracy has been enhanced and error has been dropped. Then, near about after eight to nine epochs the accuracy converges to almost small error except the NasNetMobile model. As demonstrated in Figures 6 and 7, with

Figure 6 Learning curves of validation accuracy and validation loss over various RGB frameworks

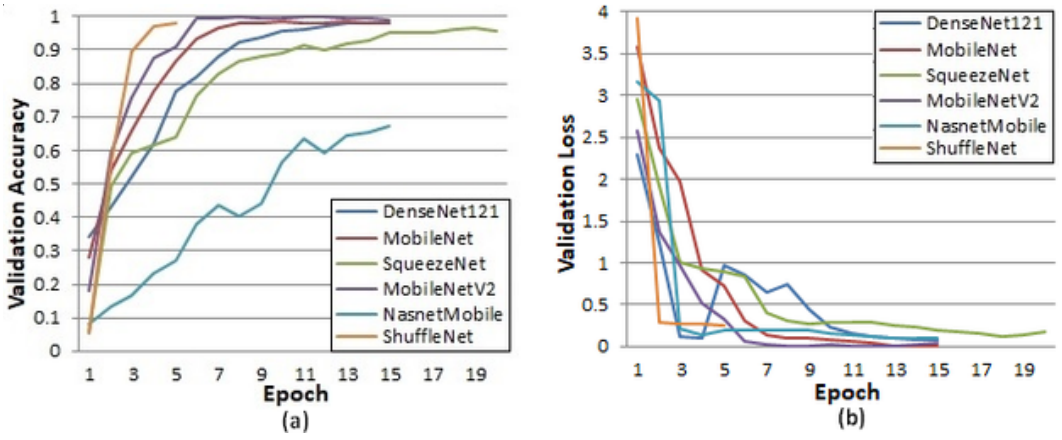


Figure 7. Learning curves of validation accuracy and validation loss over various Thermal frameworks

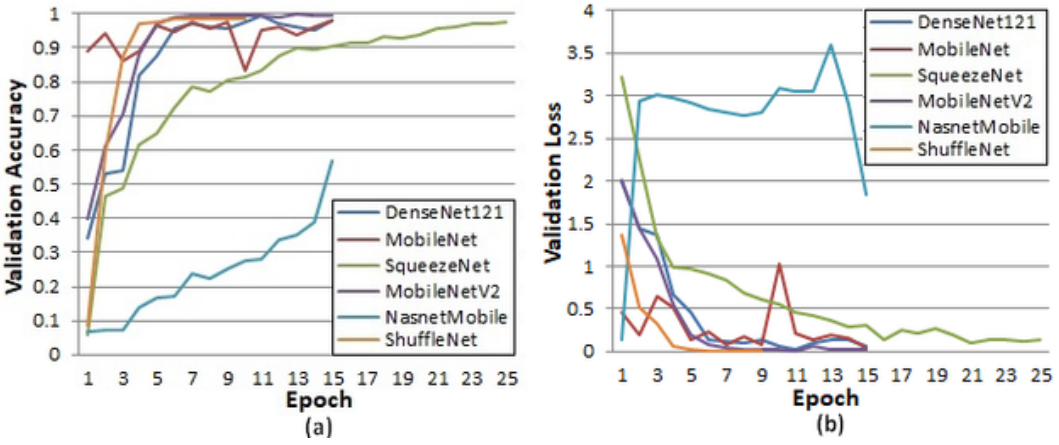
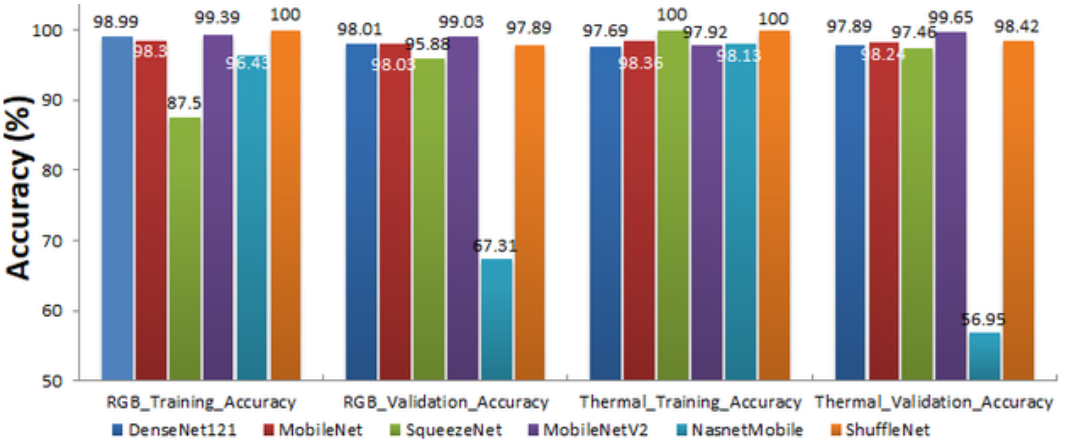


Figure 8. Comparison of each model's average accuracy over RGB and Thermal datasets



increasing numbers of epochs the accuracy consistently improves in MobileNetV2 which reached up to 99%, without degradation in performance. Contrarily, the loss curve of MobileNetV2 quickly gets converged and has huge rift with respect to error rate as compared to other models. According to Figure 8, it became apparent that the MobileNetV2 model has effectively good accuracy over the validation dataset respectively.

Table 3. Evaluation of performance metrics (size and accuracy) on various proposed models

Metrics	Model size (MB)					Accuracy (%)		Training time per image(sec)		Speed up
Framework	Trained (.h5)	Unquantized (.tflite)	Reduction in unquantized	Quantized (.tflite)	Reduction in quantized	RGB	Thermal	RGB	Thermal	
DenseNet121	100	33	3.03x	8	12.5x	98.01	97.89	22	10	2.2x
MobileNet	55	18	3.05x	5	11x	98.03	98.24	6	3	2x
SqueezeNet	16	5	3.2x	2	8x	95.88	97.46	2	1	2x
MobileNetV2	**48**	**16**	**3x**	**4**	**12x**	**99.03**	**99.65**	**6**	**3**	**2x**
NasNetMobile	71	22	3.22x	6	11.83x	67.31	56.95	11	6	1.83x
ShuffleNet	20	6	3.33x	2	10x	97.89	98.42	2	1	2x

3.2.2. Experiment2: Post-training Quantization Results

Table 3 represents evaluation of size, accuracy, and inference time measures on designed architectures where model size denotes actual occupied space of model on the hard disk and inference time has been computed on NVIDIA GEFORCE GTX 730M 8 GB. According to Table 3, the results the MobileNetV2 model is relatively superior with respect to other models collectively in accuracy (RGB - 99.03% and Thermal - 99.67%), reduction in model size (unquantized - 3x and quantized - 12x) and inference time. Additionally, from the table, it has been observed that thermal imaging models perform almost double in speed compared to RGB models. Considering that we have to deploy the trained model on the mobile device for which we need the lightweight but more accurate model together with low processing time, so considering all the above observations the MobileNetV2

model has been included in the mobile application, then generated the APK of it, and lastly deployed it into mobile devices.

3.3. Mobile Application

Figure 9(a) shows the GUI of the real-time mobile app for Fruit Shelf-life prediction where three buttons are displayed on the screen and Figure 9(b) represents the installed application "fruitshelflife" on the mobile. If the user clicks on the first button, the RGB images have been captured by smartphone rear camera and application shows the results on the basis of the outer features of the fruit. The text

Figure 9: (a) GUI of real-time fruit shelf-life prediction mobile app (b) The mobile application

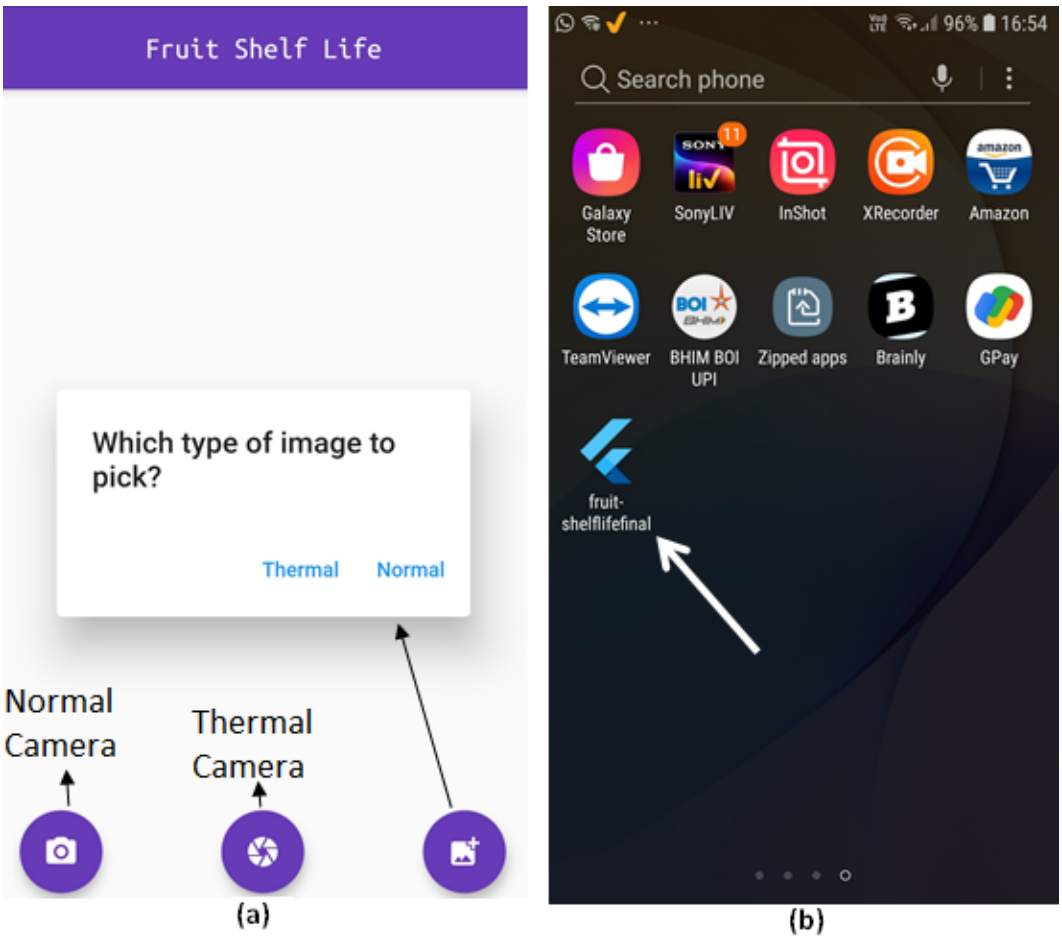


Table 4. Comparative results of un-quantized and quantized models on RGB and Thermal dataset

Model	Un-quantized					Quantized				
Dataset	Minimum Confidence	Maximum Confidence	Minimum Latency (ms)	Maximum Latency (ms)	App Size (MB)	Minimum Confidence	Maximum Confidence	Minimum Latency (ms)	Maximum Latency (ms)	App Size (MB)
RGB	0.7712	0.9999	378	465	61.3	0.8148	0.9923	231	369	34.1
Thermal	0.5864	1.0	238	288		0.5617	0.9963	204	225	

line on the result shows the confidence value (i.e. the percentage of correctness of the predicted results) and latency i.e. processing time required to predict the results. The threshold value for confidence level has been set as (0.75- un-quantized and 0.8- quantized RGB models; 0.55- for both un-quantized and quantized thermal models) to decide the correct predicted results. If the user clicks on the second button, thermal images have been captured by the Seek thermal camera and application shows the results based on the internal features of the fruit. After pressing the third button, again two buttons will get appeared on the screen, Thermal and Normal respectively, to select the images from the gallery for the study purpose to learners.

3.3.1. Quantitative Results

Table 4 illustrates the comparison with an un-quantized model, significant improvement in the average latency from 421ms (minimum = 378ms and maximum = 465ms) to 300ms (minimum = 231ms and maximum = 369ms). Over the thermal dataset, the average latency has been enhanced from 263ms (minimum = 238ms and maximum = 288ms) to 214ms (minimum = 204ms and maximum = 225ms), measured for 30 runs. Also, compared to an un-quantized model app, the app size has been improved from 61.3MB to 34.1MB. The substantial improvement verifies the superior performance of a quantized model on the thermal dataset. In the cross-device assessment, authors closely observed that a more prominent difference is in the latency but less than one second with any kind of device with a GPU or CPU. The results shown in the current study have been tested on the device: Samsung M31S smartphone with 8-core 2.3 GHz CPU.

3.3.2. Qualitative Results

For complete evaluation of the performance, the results of un-quantized and quantized model for RGB and thermal images have been visualized in Figure 10. The results in Figure 10(a and b) displays the preview of RGB image and Figure 10(c and d) represents the preview of a thermal image for un-quantized and quantized models, respectively. For example, a text line in Figure 10(b) represents that this mango has “No life” as 91.76% and took 364ms to predict the result. By observing the figure it is proved that the quantized model attains lower latency with the same recognition results.

Figure 11 demonstrates the error case where the example shows the wrong prediction in Figure 11(b) for quantized model and this can be verified by an analysis of the confidence value which is less than the set threshold.

Cross-scenario and generalization performance: From the observation of Figure 12(a), the shelf-life result is as “RUL_9” for RGB image whereas for thermal image the shelf-life is predicted as “RUL_1” which validates that thermal images shows correct remaining life of the fruits/mangoes. For cross-scenario experiment, the results of the hot (i.e. at room temperature) and cold environment are visualized in Figure 12(b and c) which also proves that the model predicts the same result with shelf life as “RUL_1” for the same mango. Here, the goal is to experiment with the models which are trained in restricted scenarios can be generalized to new situations as if the system needs to be used, it is difficult for the people to utilize it in a similar environment and this is the most noticeable aspect in the analysis of shelf life.

Moreover, the ability of generalization has been evaluated for the proposed model which is shown in Figure 13. Even if shelf-life prediction has been tackled with the ‘Kesar’ mango variety in room temperature, its qualitative and quantitative analysis proves the generalization with other varieties of mangoes (here for example as Alphonso) in a cold environment for shelf life prediction, enhancing contemporary results. The main idea behind this is to demonstrate that the shelf life prediction problem could be resolved by training the model once whereas validating it with the other test set.

Figure 10. Exemplary results of (a) RGB_un-quantized (b) RGB_quantized (c) Thermal_un-quantized and (d) Thermal_quantized model

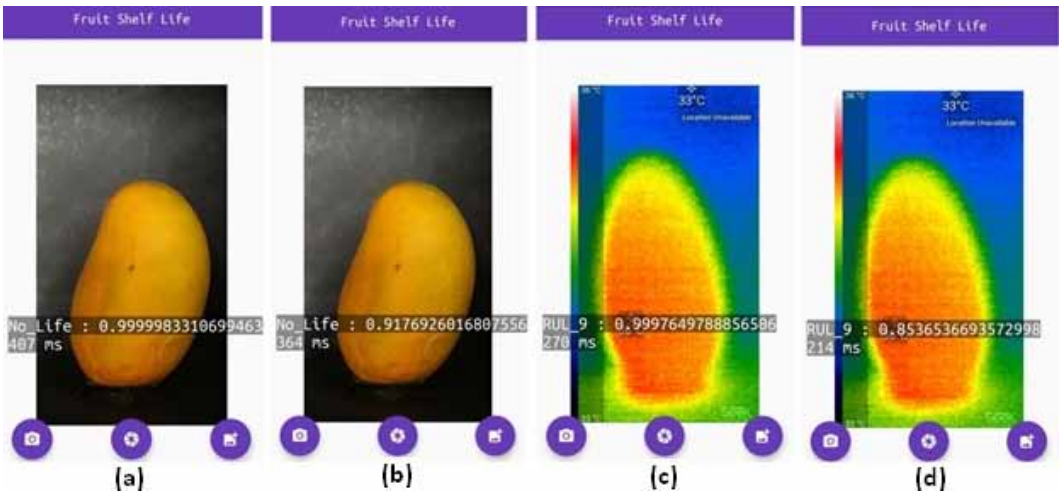


Figure 11. Example results of error case in comparison of (a) RGB_un-quantized and (b) RGB_quantized model

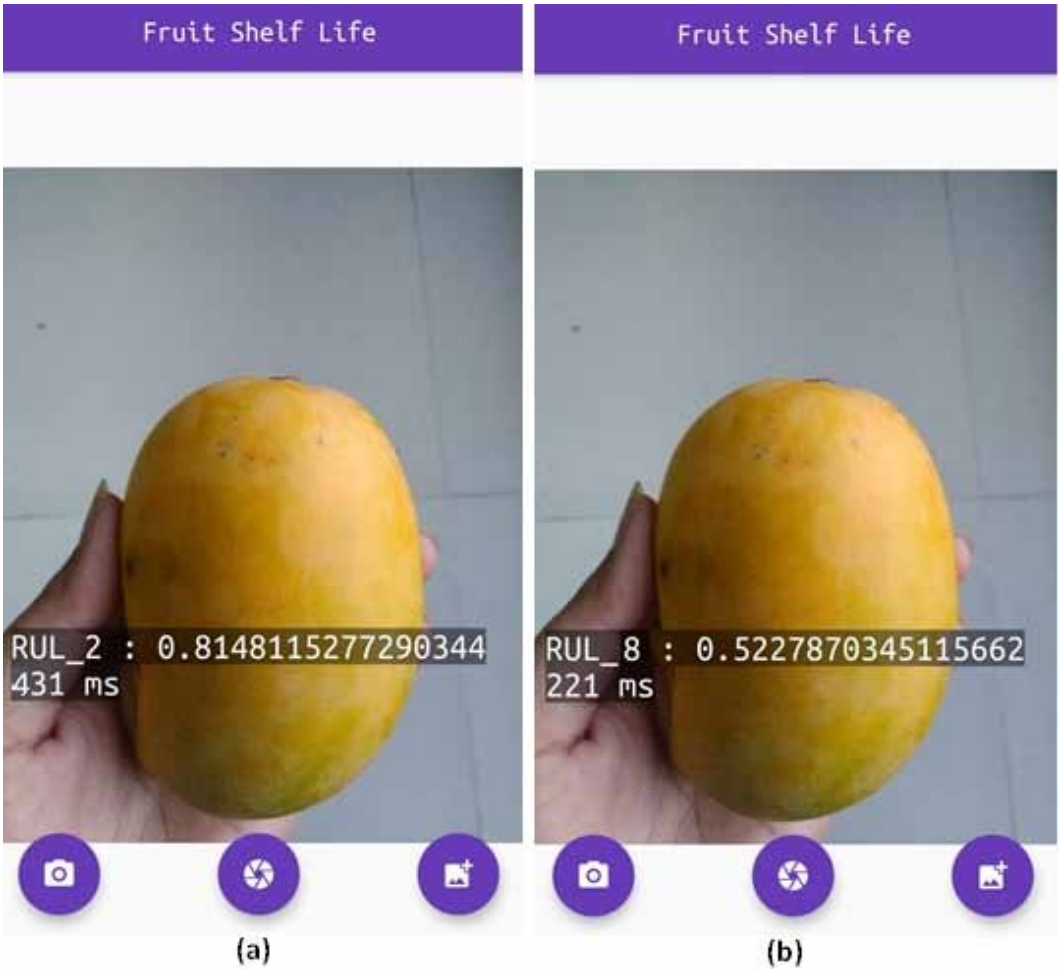


Figure 12. Example results of error case in comparison of (a) RGB and (b), (c) Thermal images as well as example of cross-scenario experiment for (b) hot and (c) cold environment

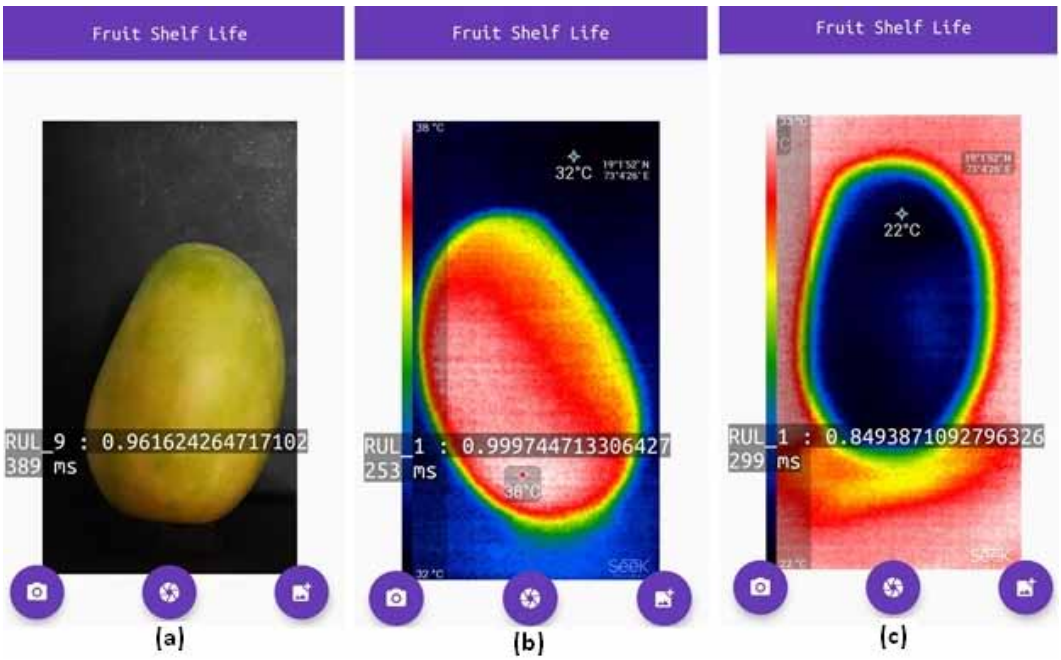
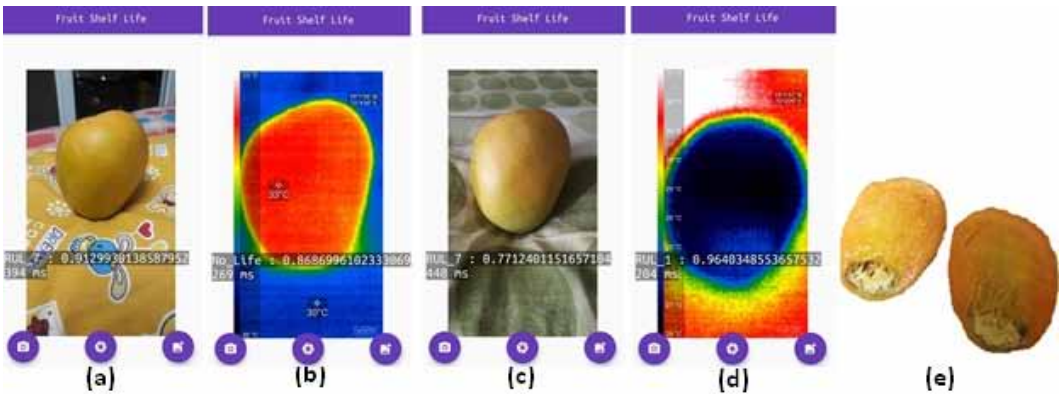


Figure 13. Example results of generalization performance of the proposed model for which error case in comparison of (a) RGB1 and (b)Thermal1_hot, (c) RGB2 and (d) Thermal2_cold images as well as example of cross-scenario experiment for (b) hot and (d) cold environment and (e) verified proof of Thermal2_cold image

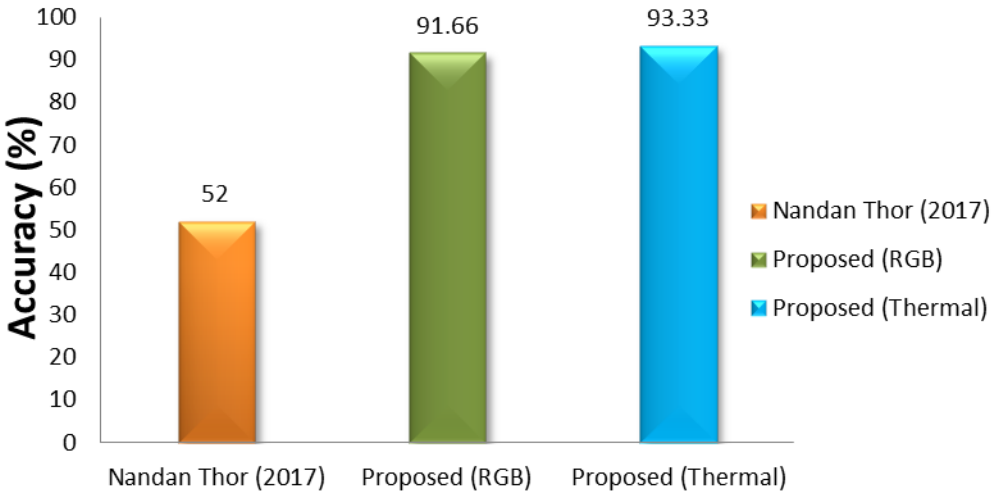


3.4. Comparative Work

The proposed work has been compared with the work of other researchers (Nandan Thor, 2017). Even though it is not reliable to compare the results of current work with the other researchers because of dissimilar datasets but the results have been compared on the basis of accuracy. The researcher has introduced a technique which is not destructive, to forecast the shelf-life of bananas based on the external features like color features and different machine learning algorithms were applied over

these color features. A highest predictive accuracy value of 52% has been gained which is smaller compared to the achieved from the RGB and thermal datasets of the present work-91.66% and 93.33% respectively and shown in Figure 14.

Figure 14. Comparative analysis of current work with Nandan Thor, 2017



4. Conclusion

In the present study, authors have developed an emerging shelf-life prediction model based on a case-study of mango fruit by deep transfer learning and deliver it in the form of mobile app designed through Flutter. With the extended goal of building a shelf-life prediction system using mobiles for a routine usage, the present work considers the benefit of mobile-based imaging abilities viz. enhanced RGB and thermal imaging. The conventional understanding of the fruit shelf-life is just limited to RGB based color and internal damage, but the use of this novel conceptualization strives to improve the human understanding to greater extent.

In a nutshell, the proposed system works as follows: to show the effectiveness of the system, six lightweight models have been built with the use of transfer learning mechanism and compared them. Accordingly, based on the consideration of accuracy, time, and computing power required on mobile devices, the final choice is MobileNetV2. The tool utilized for training the model was Tensorflow and after training quantization technique has been applied for optimizing the model, which makes it achieve greater performance in terms of the optimal model size and low latency. Thenceforth, the models have been converted into TensorFlow Lite format to deploy it in the Flutter app code, and generated the APK files for real-time execution on mobile phones. The experiential outcomes exhibit that the proposed model found to be superior with quantization outperforms for latency as well as 10x to 12x reduction in model size along with good recognition accuracy. The model deployed on the mobile device achieves the validation accuracy as 99.03% and 99.65% whereas test accuracy as 91.66% and 93.33% for RGB and Thermal datasets, respectively which proves that thermal imaging approach outperforming than RGB. So, from the results, it can be inferred that with the use of lightweight but robust framework as the essence of our proposed methodology, for running the inference on mobile it imparts the results within a fraction of a second and also resulting in good generalization ability. In

short, this paper aims to demonstrate real-time application through mobile cameras, featuring quick reply, portability, and easy to implement which helps society in beneficial ways.

In future research, authors intend to investigate several varieties of fruits to a greater extent. Also, authors tend to investigate the tradeoff between mobile processing time and accuracy through federated learning. Also, the proposed work can be further extended to release the other app formats for iPhones and hardware devices like Raspberry Pi with its single code base functionality.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Abdel-Hamid, O., Mohamed, H., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(10), 1533–1545. doi:10.1109/TASLP.2014.2339736

Alzu'bi, A., & Abuarqoub, A. (2020). Deep learning model with low-dimensional random projection for large-scale image search. *Eng. Sci. Technol. International Journal (Toronto, Ont.)*, *23*(4), 911–920.

Behera, S. K., Rath, A. K., & Sethy, P. K. (2020). (in press). Maturity status classification of papaya fruits based on machine learning and transfer learning approach. *Information Processing in Agriculture*, 2214–3173.

Bhole, V., Kumar, A., & Bhatnagar, D. (2020a). Fusion of color-texture features based classification of fruits using digital and thermal images: A step towards improvement. *Grenze Int. J. Eng. and Technol.*, *6*(1), 133–141.

Bhole, V., Kumar, A., & Bhatnagar, D. (2020b). A texture-based analysis and classification of fruits using digital and thermal images. *Lect. Notes Net. Syst.*, *93*, 333–343. doi:10.1007/978-981-15-0630-7_33

Bhole, V., & Kumar, A. (2020c). Analysis of convolutional neural network using pre-trained squeezenet model for classification of thermal fruit images. *ICT for Competitive Strategies*, 759-768.

Bhole, V., & Kumar, A. (2020d). Mango Quality Grading using Deep Learning Technique: Perspectives from Agriculture and Food Industry. In *Proceedings of 21st International Conference on Inf. Technol. Educ. (SIGITE' 20)* (pp. 180-186). ACM. doi:10.1145/3368308.3415370

Bhole, V., & Kumar, A. (2020e). *System and method for identifying fruit shell life*. Indian Patent 352727.

Bhole, V., & Kumar, A. (2021). A transfer learning-based approach to predict the shelf life of fruit. *Inteligencia Artificial*, *24*(67), 102–120. doi:10.4114/intartif.vol24iss67pp102-120

Fan, J. C., & Zhou, G. M. (2011). Near infrared spectroscopy qualitative analysis of apple shelf life. *Food and Nutrition in China*, *17*, 47–49.

Gao, H., Zhuang, L., Laurens, M., & Weinberger, K. Q. (2017). *Densely Connected Convolutional Networks*. arXiv:1608.06993.

Global Mobile Statistics. (2020). Retrieved from http://mobithinking.com/mobile-marketingtools/latest-mobile-stats

Google Play. (2020). Retrieved from https://play.google.com/store

Grossetete, M., Berthoumieu, Y., Da Costa, J. P., Germain, C., Lavialle, O., & Grenier, G. Early estimation of vineyard yield: site specific counting of berries by using a smartphone. In *Proceedings of International Conference on Agric. Eng. (AgEng 2012)* (pp. C-1915). Academic Press.

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.* arXiv preprint arXiv:1704.04861.

Iandola, F. N., Han, S., Moskewicz, M., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size.* arXiv preprint arXiv:1602.07360v4.

Intaravanne, Y., Sumriddetchkajorn, S., & Nukeaw, J. (2012). Cell phone-based two dimensional spectral analysis for banana ripeness estimation. *Sensors and Actuators. B, Chemical*, *168*, 390–394. doi:10.1016/j.snb.2012.04.042

Intaravanne, Y., & Sumriddetchkajorn, S. (2015). Android-based rice leaf color analyzer for estimating the needed amount of nitrogen fertilizer. *Computers and Electronics in Agriculture*, *116*, 228–233. doi:10.1016/j.compag.2015.07.005

Karar, M. E., Alsunaydi, F., Albusaymi, S., & Alotaibi, S. (2021). A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alexandria Engineering Journal*, *60*(5), 4423–4432. doi:10.1016/j.aej.2021.03.009

Karwa, A., Soni, N., Gavali, A., & Patil, M. B. (2016). Android based application for fruit quality analysis. *Int. J. Res. Sci. Eng. Technol.*, *12*(5), 20480–20487.

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Academic Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. doi:10.1038/nature14539 PMID:26017442

Lin, M., Chen, Q., & Yan, S. (2013). *Network in network*. arXiv preprint arXiv:1312.4400.

Naik, S., & Patel, B. (2017). Thermal imaging with fuzzy classifier for maturity and size based non-destructive mango (Mangifera Indica L.) grading. In *Proceedings of the Conference on Emerging Trends and Innovation in ICT* (pp. 16-20). IEEE. doi:10.1109/ETIICT.2017.7977003

Thor, N. (2017). Applying machine learning clustering and classification to predict banana ripeness states and shelf life. *Int. J. Adv. Food Sci. Technol.*, *2*(1), 20–25.

Pan, S., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. doi:10.1109/TKDE.2009.191

Sahu, D., & Potdar, R. (2017). Defect identification and maturity detection of mango fruits using image analysis. *American J. Artif. Intell.*, *1*(1), 5–14.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of CVPR*, 4510-4520.

Sultana, F., Galib, S., Hasan, F., & Jerin, A. (2017). Digital measurement of maturity indices of mangoes using selected image features. *Journal of Food Processing & Technology*, *8*(11), 1–7.

TensorFlow. (2020). *TensorFlow lite converter*. Retrieved from https://www.tensorflow.org/lite/convert

Zeng, W., Huang, X., Arisona, S. M., & McLoughlin, I. V. (2014). Classifying watermelon ripeness by analysing acoustic signals using mobile devices. *Personal and Ubiquitous Computing*, *18*(7), 1753–1762. doi:10.1007/s00779-013-0706-7

Zhao, D., Yu, G., Xu, P., & Luo, M. (2019). Equivalence between dropout and data augmentation: A mathematical check. *Neural Networks*, *115*, 82–89. doi:10.1016/j.neunet.2019.03.013 PMID:30978610

Zhang, S., Xue, J., Sun, H., & Zhou, J. (2013). Study of malus asiatica nakai's firmness during different shelf lives based on visible/near-infrared spectroscopy. *Mathematical and Computer Modelling*, *58*(11-12), 1829–1836. doi:10.1016/j.mcm.2012.12.021

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey and New Perspectives. *Computer Survey*, *52*(1), 1–38. doi:10.1145/3285029

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of CVPR*, 6848-6856. doi:10.1109/CVPR.2018.00716

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). *Learning Transferable Architectures for Scalable Image Recognition*. arXiv:1707.07012.

Varsha Bhole is an Assistant Professor at A.C. Patil College of Engineering, Navi Mumbai, India, and is a Ph.D. candidate at the Sir Padampat Singhania University, Udaipur, India.