# Implementation Of A Image Classification Model On A Android Application

Siva Satyanarayana Raju Pusapati
George Mason University
sraju5@gmu.edu

## ABSTRACT

This paper describes the development of an Android app designed for efficient grocery shopping, utilizing advanced machine learning techniques for accurate item classification. The application incorporates TensorFlow Lite, a powerful framework for on-device inference, ensuring real-time processing and reducing reliance on server-side computations. Firebase, a comprehensive platform, is employed for seamless data management, facilitating tasks such as user authentication and cloud storage. To enhance the user experience, the CameraX library is integrated, enabling smooth image capture for accurate item identification. Through the combination of these technologies, the app offers a robust and user-friendly solution for efficient grocery shopping, making use of machine learning capabilities directly on the user's device.

## KEYWORDS

Android, Machine Learning, TensorFlow Lite, Firebase, CameraX

## 1 INTRODUCTION

This paper showcases the development of a grocery shopping app that capitalizes on the proliferation of smartphones equipped with robust computational capabilities and high-quality cameras. Leveraging the advancements in machine learning, the application allows users to conveniently capture images of grocery items and seamlessly add them to their shopping cart. By incorporating machine learning algorithms, the app classifies the grocery items, ensuring accurate identification. This innovative solution harnesses the potential of mobile technology and machine learning to provide users with a convenient and efficient grocery shopping experience, revolutionizing the way people shop for everyday essentials.

## 2 DATASET

The Fruit 360 dataset contains high-quality images of 131 different types of fruits and vegetables, providing a diverse range of data for image recognition tasks. The dataset contained over 90,000 images. Each fruit or vegetable is represented with several hundred images, taken from various angles and under different light conditions. All images are 100x100 pixels and are taken on a white background to minimize noise and make the fruit or vegetable the focus of each image. This uniformity helps simplify image processing tasks and makes it easier to compare different images.The dataset is open-source and is freely available for download on platforms like Kaggle and GitHub. It is maintained and regularly updated to ensure its relevance and utility in ongoing research and applications.

, ,
.

## 3 SYSTEM OVERVIEW

The grocery shopping application comprises three key components to deliver an exceptional user experience. Firstly, a TensorFlow Lite-powered classifier is integrated, enabling precise item classification through advanced machine learning algorithms. Secondly, Firebase serves as the backbone for efficient data management, facilitating seamless operations such as user authentication and cloud storage. Lastly, the app leverages the CameraX library, which empowers users to capture high-quality images of grocery items effortlessly. These three components work harmoniously to provide a robust and user-friendly platform, revolutionizing the way people shop for groceries by leveraging the power of machine learning, data management, and camera functionalities.

### 3.1 Image Classification with TensorFlow Lite

The grocery shopping app utilizes a pre-trained machine learning model to accurately classify grocery items by analyzing their images. Powered by TensorFlow Lite, a specialized framework for deploying machine learning models on mobile and embedded devices, the app employs a dedicated 'Classifier' class to encapsulate the classifier implementation. The 'Classifier' class initializes a TensorFlow Lite interpreter with the pre-trained model, allowing it to efficiently perform inference on the input image data. This seamless integration of TensorFlow Lite enables real-time and accurate classification of grocery items, enhancing the app's functionality and providing users with a reliable and convenient shopping experience.

### 3.2 Data Management with Firebase

To ensure efficient management of the shopping cart, the app leverages Firebase as a robust backend solution. The 'Cart' class encapsulates the Firebase API, offering essential functionalities for seamless item handling within the cart. By utilizing Firebase's Realtime Database, each item added to the cart is stored as a string, enabling quick and easy retrieval. The 'Cart' class efficiently handles operations such as adding, retrieving, and removing items, providing a smooth and reliable shopping experience. With Firebase as the backend, the app ensures secure and synchronized storage of the shopping cart, enhancing the user's convenience and facilitating a hassle-free grocery shopping journey.

### 3.3 Camera Functionality

The app employs the CameraX library to facilitate the image capture process for grocery items. The 'MainFragment' class takes charge of managing the camera and image analysis functionalities seamlessly. Whenever a user captures an image, it undergoes processing and is subsequently forwarded to the classifier for inference. This streamlined workflow ensures that images are efficiently analyzed and classified, empowering users with accurate information about

the grocery items they are capturing. By leveraging the CameraX library and the 'MainFragment' class, the app optimizes the image capture and analysis process, enhancing the overall user experience during grocery shopping.

## 4 MODEL TRAINING AND CONVERSION

The model training and conversion process involves the following steps:

(1) **Loading the Dataset**: The Fruit 360 dataset is loaded using the `ImageDataGenerator` from the specified directories for training and validation.
(2) **Model Definition**: The model architecture is defined using the `Sequential` model from TensorFlow. It consists of convolutional layers, max-pooling layers, dropout layers, and dense layers.
(3) **Model Compilation**: The model is compiled with the appropriate loss function, optimizer, and metrics.
(4) **Model Training**: The model is trained using the `fit` method, where the training generator and validation generator are passed. The training is performed for a specified number of epochs.
(5) **Model Conversion**: The trained model is converted to the TensorFlow Lite format using the `TFLiteConverter`.
(6) **Saving the Model**: The converted TensorFlow Lite model is saved to disk with the desired filename.
(7) **Downloading the Model**: The saved model file is downloaded using the `files.download` function.

In the updated code, data augmentation techniques are applied during training using the `ImageDataGenerator`. Dropout layers are added after each max-pooling layer to mitigate overfitting, and L2 regularization is added to the dense layer for weight decay. Additionally, an early stopping callback is included to monitor the validation loss and stop training if it doesn't improve for a certain number of epochs.

## 5 IMPLEMENTATION DETAILS

### 5.1 Classifier

The classifier in the app leverages TensorFlow Lite to load a pretrained model from the application's assets and perform inference. It utilizes a ByteBuffer to store the image data, ensuring compatibility with TensorFlow Lite. To populate the ByteBuffer, the bitmap image data is converted into a byte array and then copied into the ByteBuffer. The classifier generates an array of probabilities for each class, and the class with the highest probability is selected as the final result. This robust process enables the app to accurately classify grocery items based on the image data, providing users with reliable and informative results.

### 5.2 Cart

The 'Cart' class plays a pivotal role in the app's functionality, offering essential methods for adding, retrieving, and removing items from the shopping cart. To ensure efficient storage and synchronization, the app utilizes Firebase Realtime Database. The cart items are stored as key-value pairs, where Firebase generates a unique identifier as the key, while the item name serves as the corresponding value. This approach guarantees a seamless and organized structure for the cart items within the database. By leveraging Firebase's capabilities, the app provides users with a reliable and dynamic shopping cart experience, enabling effortless management of items and ensuring data integrity throughout the shopping journey.

### 5.3 MainFragment

The 'MainFragment' class takes charge of overseeing the camera and image processing functionalities within the app. By utilizing the CameraX library, it gains access to the device's camera, enabling the capture of high-quality images. These captured images are then converted into Bitmap format, a standard image representation in Android, before being forwarded to the classifier for inference. This seamless integration of camera access, image capture, and processing ensures a smooth and efficient workflow for analyzing grocery items. The 'MainFragment' class plays a vital role in delivering a user-friendly and reliable experience, allowing users to effortlessly capture and classify grocery items through the app's intuitive interface.

## 6 APPLICATION FLOW

The application follows the following flow:

(1) The user launches the application, and the `MainActivity` is created.
(2) The `MainActivity` inflates the layout and sets up the UI components.
(3) When the user clicks the "Scan" button, the `processImage()` function in the `MainFragment` is called.
(4) The `processImage()` function retrieves the latest captured image from the camera and converts it into a `Bitmap` format.
(5) The `Classifier` class is used to classify the captured image. The `classify()` function takes the `Bitmap` as input and returns an array of `FloatArray` representing the classification probabilities for different fruit types.
(6) The `interpretResult()` function in the `MainFragment` interprets the classification result by finding the index of the maximum probability and mapping it to the corresponding fruit label.
(7) The `showAddToCartDialog()` function displays an `AlertDialog` asking the user if they want to add the identified fruit to their cart.
(8) If the user confirms, the identified fruit item is added to the Firebase Realtime Database using the `addItem()` function in the `Cart` class.
(9) The `CartActivity` is launched, which displays the user's cart items.
(10) The `CartActivity` retrieves the cart items from the Firebase Realtime Database using the `getItems()` function in the `Cart` class. The retrieved items are displayed in a `ListView`.
(11) If the user wants to remove an item from the cart, they can click on the item in the `ListView`.
(12) The `showRemoveFromCartDialog()` function displays an `AlertDialog` asking the user if they want to remove the selected item from their cart.

(13) If the user confirms, the selected item is removed from the Firebase Realtime Database using the `removeItem()` function in the `Cart` class.
(14) The `CartActivity` updates the `ListView` to reflect the changes in the cart items.

## 7 SAMPLE IMAGES

Here are some sample images of the applicatio

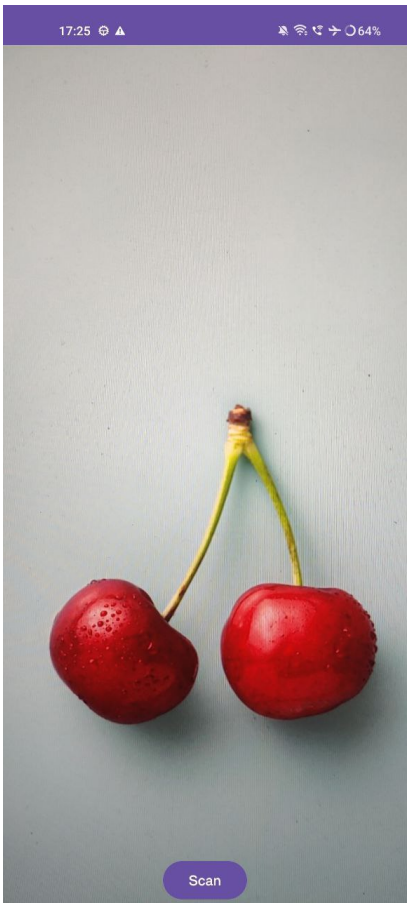


**Figure 1: scanning the image**

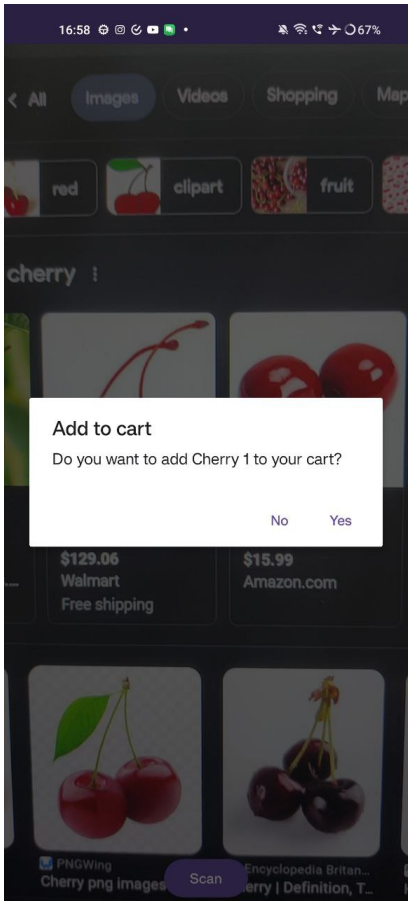


**Figure 2: Add to cart**

## 8 CONCLUSION

The developed Android application showcases the practical implementation of machine learning in a mobile environment. It exemplifies the use of advanced machine learning techniques for accurate item classification in the context of grocery shopping. The application leverages TensorFlow Lite for on-device inference, ensuring real-time processing without relying heavily on server-side computations. Furthermore, Firebase is utilized for seamless data management, facilitating tasks like user authentication and cloud storage. As future work, the application can be enhanced by expanding the range of recognizable items, refining the user interface for a more intuitive experience, and integrating with online grocery shopping services. These improvements would further enhance the functionality and usability of the application, offering users a comprehensive and efficient grocery shopping solution empowered by machine learning.