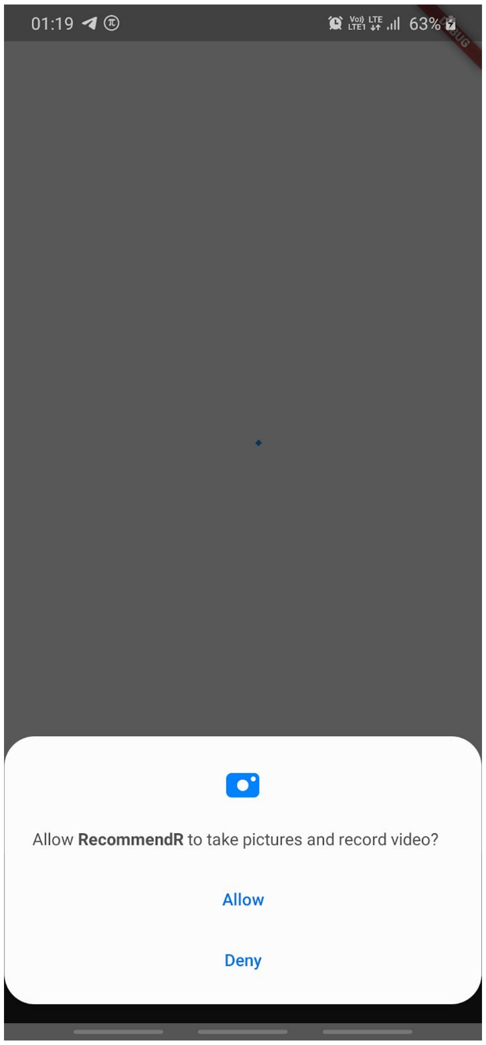
Software Implementation

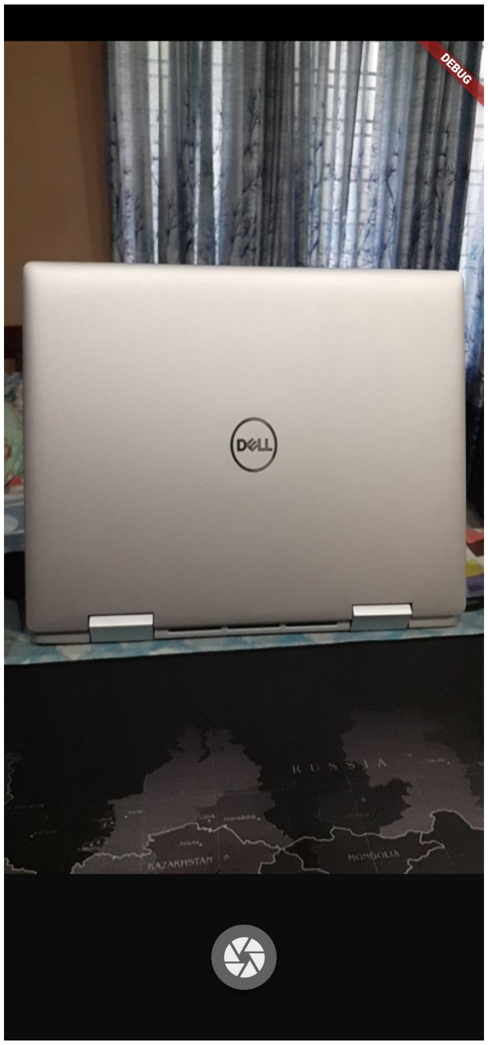
The cross-platform mobile application has been developed using flutter and dart with the tensorflow model added using the tflite package. The screenshots depicting the working of the application have been attached below.

1. <u>Camera Access:</u> The application requires access to the camera for the proper functioning. So, during installation, permission to use the camera is explicitly asked.

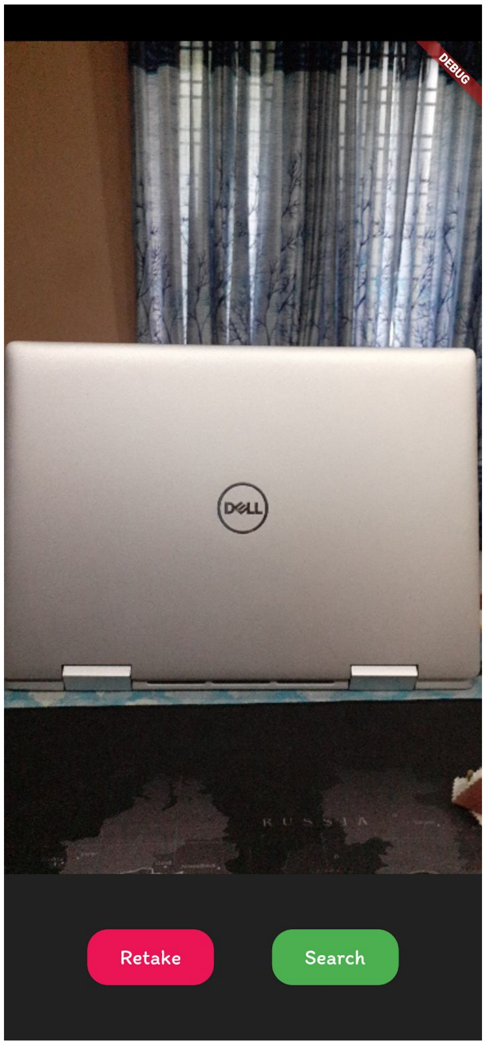


Allow **RecommendR** to take pictures and record video?

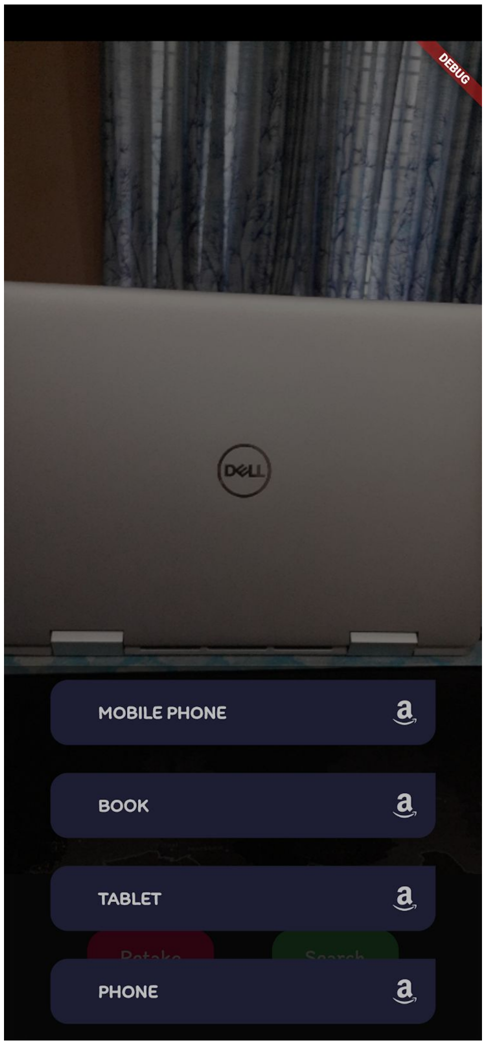
2. Image Capture Screen: This is the home screen where the app shows the camera view and provides a button, which when pressed, will capture the current camera view and create an image file and store it in a temporary location so that it can be used later whenever necessary.



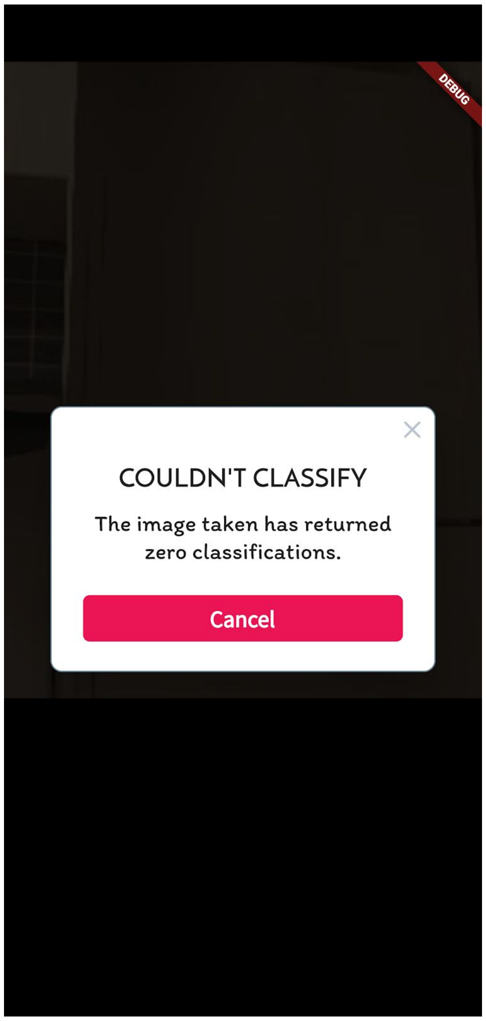
3. Captured Image Preview: This screen depicts the captured image and offers the user two options. If the image is not satisfactory or if it has not correctly captured the image, then an option is provided to *retake* the image and return to the previous image capture screen. Otherwise, the user can tap on the *search* button, and the app will redirect to the recommendations listing.



4. <u>Generated Recommendations List:</u> The recommendations generated from the captured image after it has been passed through the image processing unit (IPU) is shown in a list view for the user. The user can then select the classification that depicts the closest match. (In this example, we have selected '*tablet*' as the recommendation.)

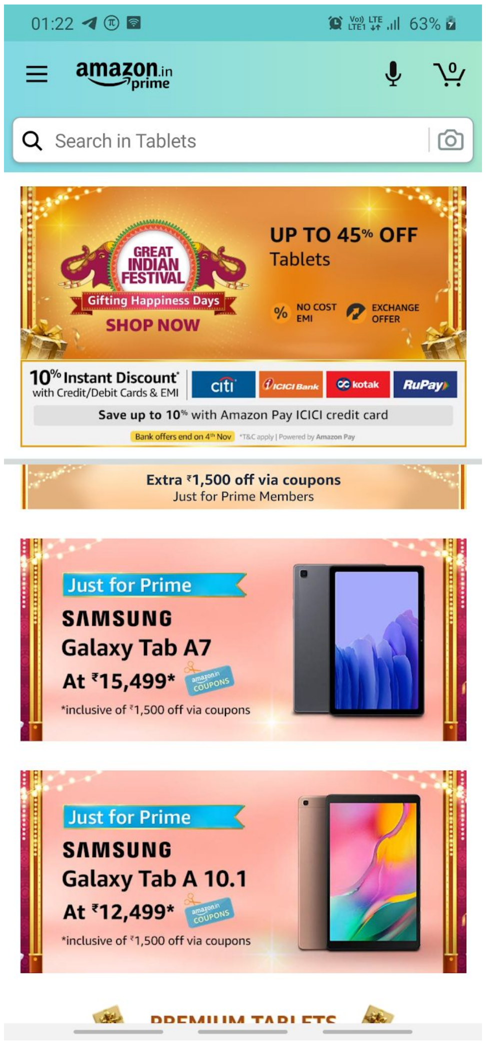


5. Unable to Classify: The implemented tensorflow image processing unit ensures that any classification that the model returns only those classifications that have a probability of at least 0.05. For this reason, there is a possibility that the model isn't able to identify any classification that has a sufficiently high probability. In this case, an alert is displayed that tells the user that the model wasn't able to classify. Upon pressing the Cancel button in the alert, the user is redirected to the home screen.

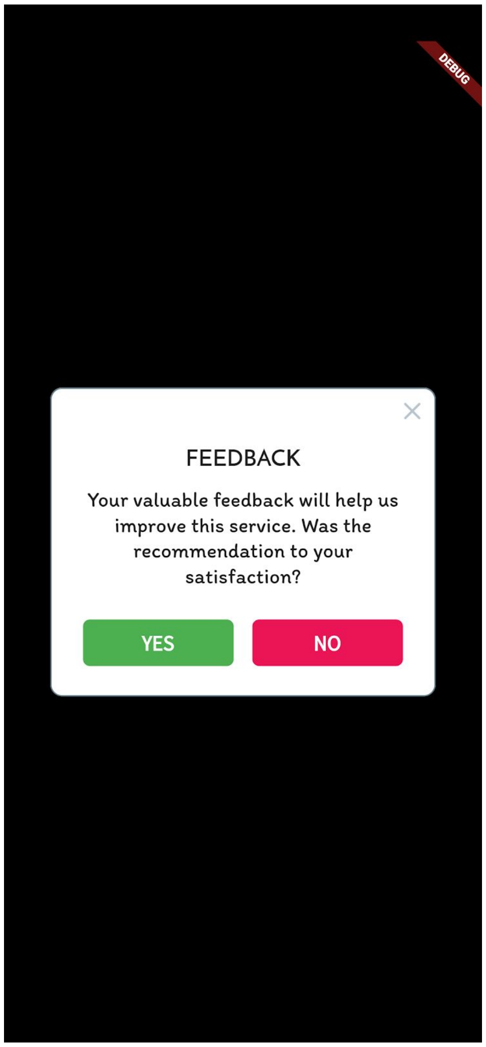

COULDN'T CLASSIFY

The image taken has returned zero classifications.

6. <u>Recommendation Display:</u> The text-based classification that the user selects is used to perform an API call to the google CustomSearch API with site-restrictions enabled so that only amazon.in is used for the search results. The URL thus obtained will then be used to open up a webview within the app itself. If for some reason, this is not possible, the URL is opened in the device's default browser.

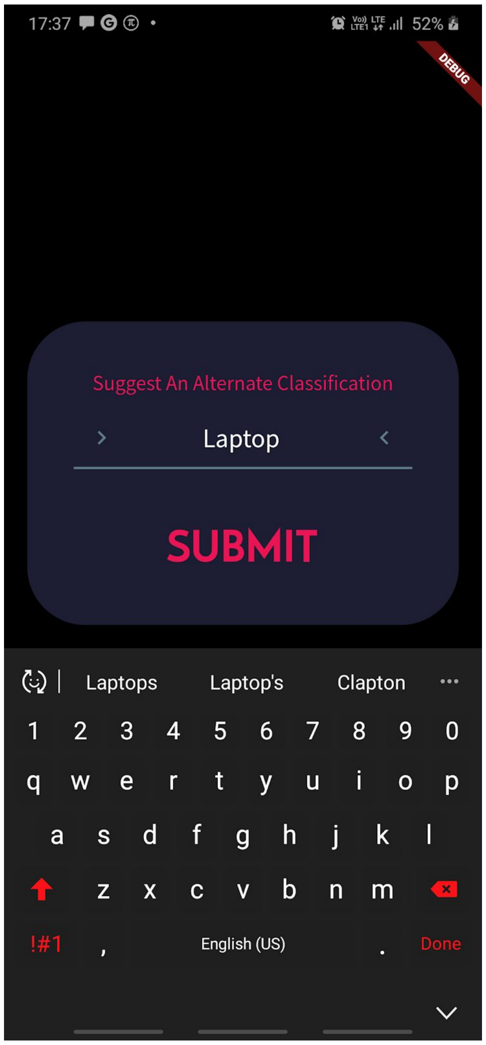


7. <u>Asking for User's Feedback:</u> When the user closes the webview and returns to the app, an alert immediately pops up asking the user for his feedback on whether the app has satisfied its purpose or if it can be improved. If the user is satisfied with the results, then he/she will be redirected to the home screen. If not, then the user is allowed to give his feedback.

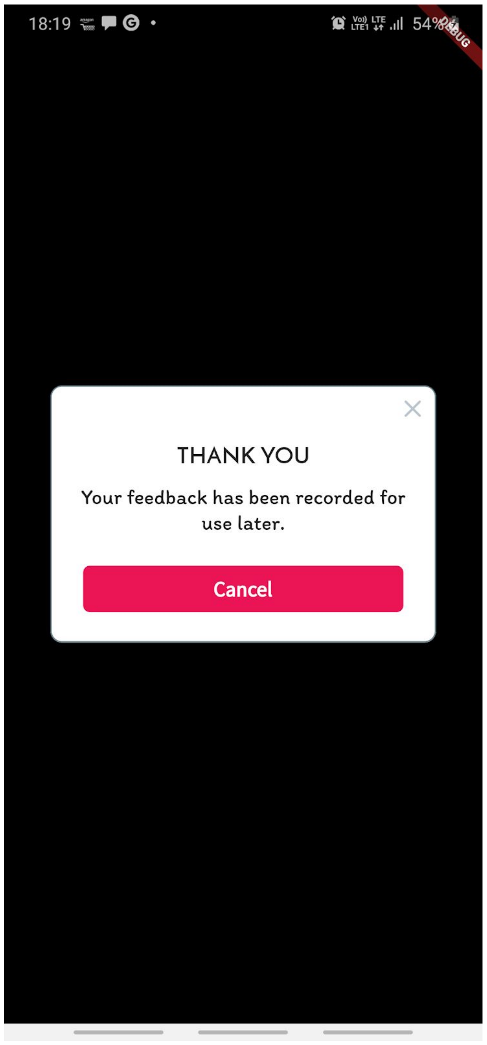


Your valuable feedback will help us improve this service. Was the recommendation to your satisfaction?

8. Feedback Screen: In this section of the app, the user is allowed to enter in a keyword / group of keywords that, along with the image and the app recommendation, gets stored in the app's firebase database, which will later on be used to modify / improve the user experience. The suggested recommendation string also goes through a cross checking mechanism to make sure that the suggested keyword is indeed valid.



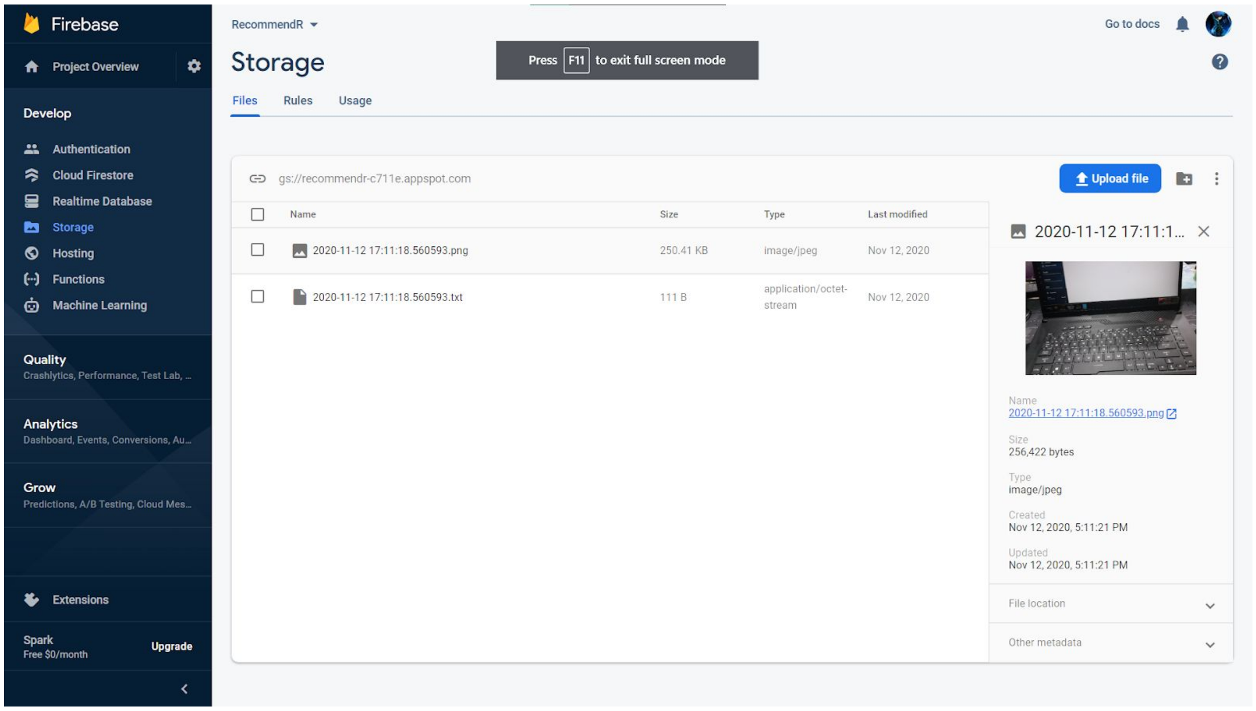
9. <u>Thank You for the Feedback:</u> Once the user submits the recommendation, the app redirects to a screen where the user is thanked for taking the time to provide the feedback necessary to maintain and improve our app and its user experience. The alert that pops up is automatically dismissed after a short duration, or the user can dismiss it simply by tapping on the *Cancel* button. Dismissing this screen redirects the user to the home screen.



THANK YOU

Your feedback has been recorded for use later.

10. <u>User-Feedback gets stored in Firebase Storage:</u> Once the user provides feedback, the image as well as a text file (named using the current time) containing the app recommendation and user-suggested recommendation is uploaded to the Firebase Storage instance. This happens in the background and so, there is no delay in the working of the app.



Press F11 to exit full screen mode

gs://recommendr-c711e.appspot.com

Name	Size	Type	Last modified
2020-11-12 17:11:18.560593.png	250.41 KB	image/jpeg	Nov 12, 2020
2020-11-12 17:11:18.560593.txt	111 B	application/octet-stream	Nov 12, 2020

2020-11-12 17:11:1...

Name
2020-11-12 17:11:18.560593.png

Size
256,422 bytes

Type
image/jpeg

Created
Nov 12, 2020, 5:11:21 PM

Updated
Nov 12, 2020, 5:11:21 PM

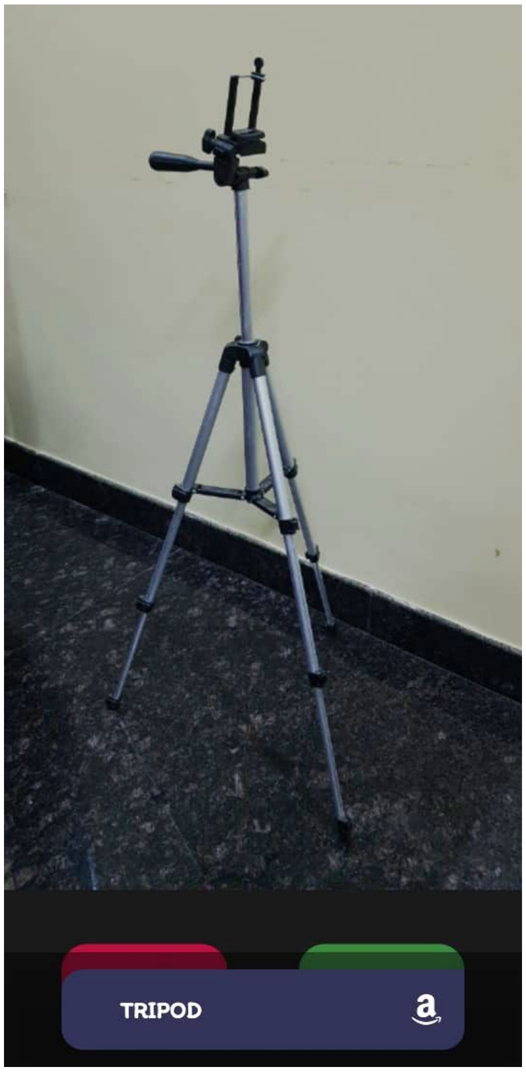
File location

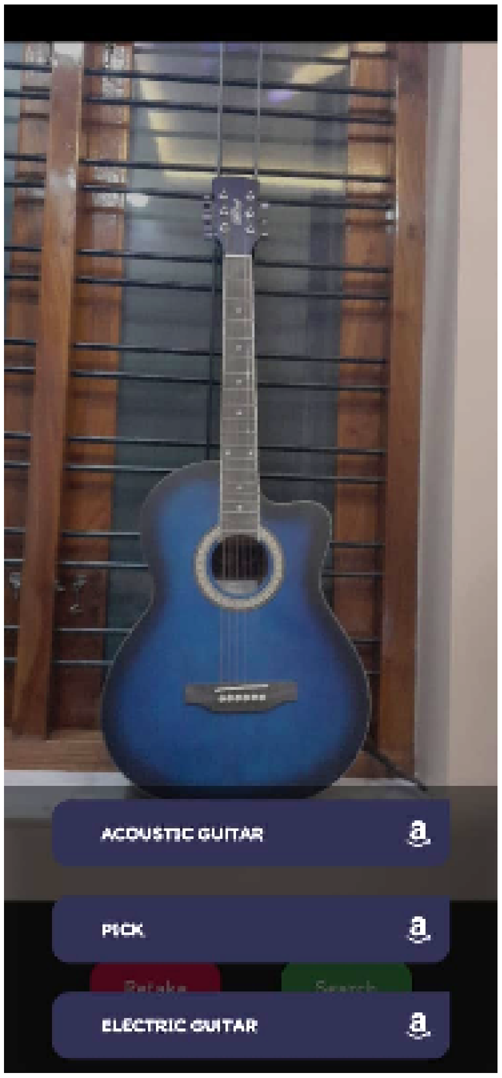
Other metadata

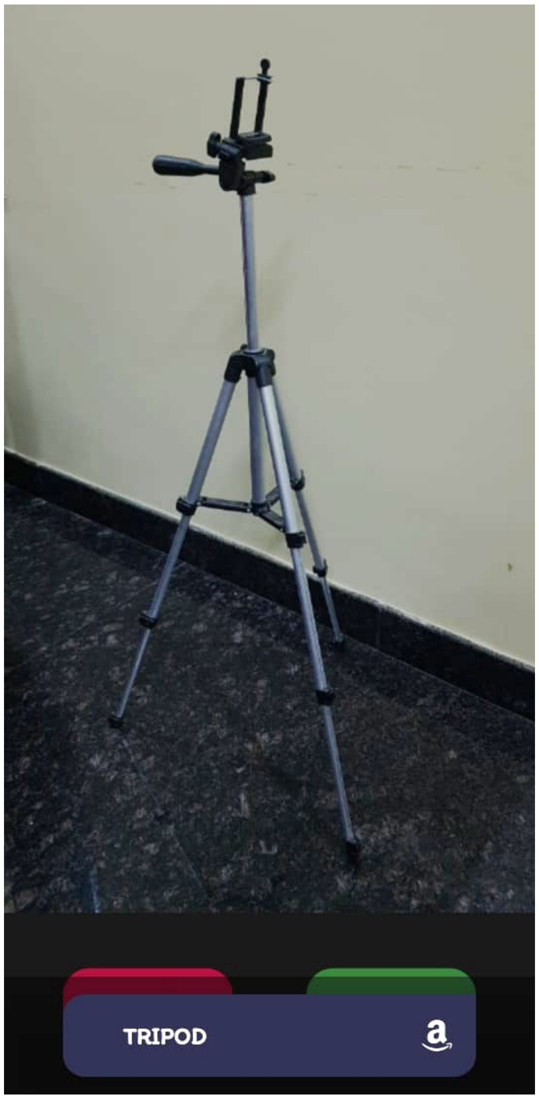
Software Testing

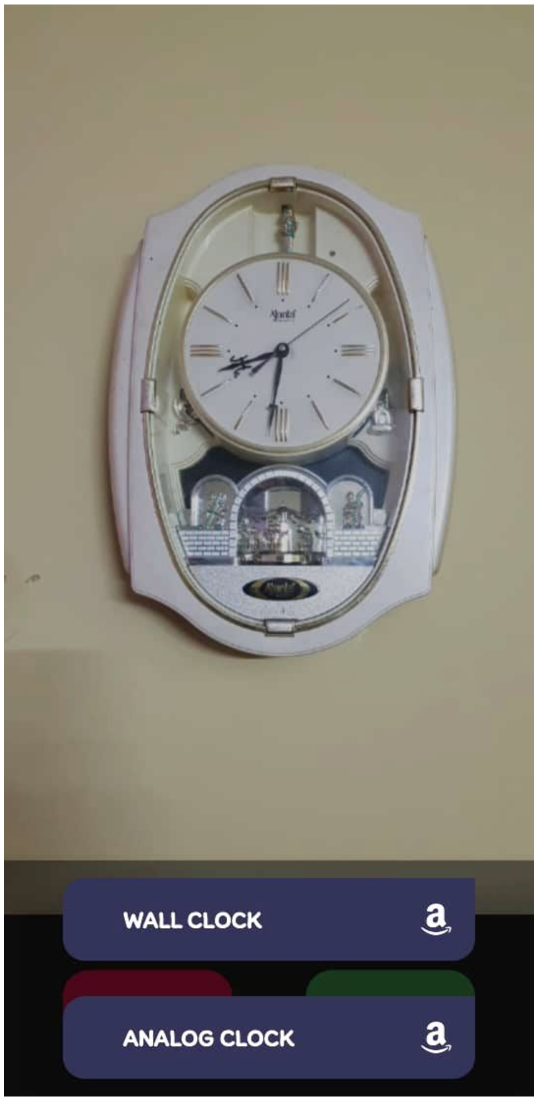
In order to perform testing of the mobile application, we first decided to run it on various different devices. It has no problems while being run on Android 10 devices but we have been unable to check on actual IOS devices although it runs smoothly on the IOS emulator.

Shown below are some of the cases where the object has been classified correctly by the application.

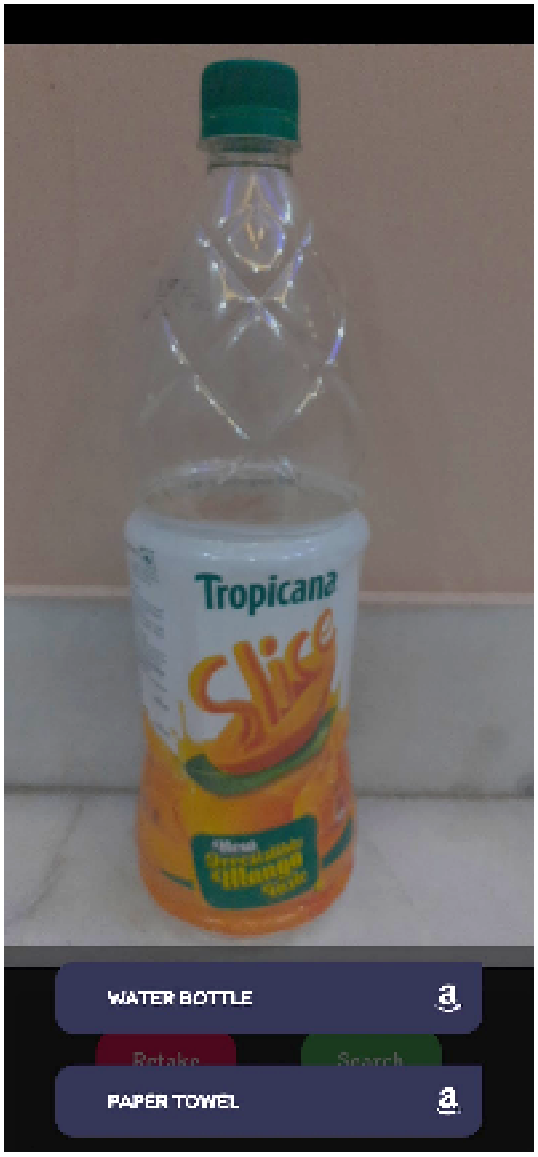


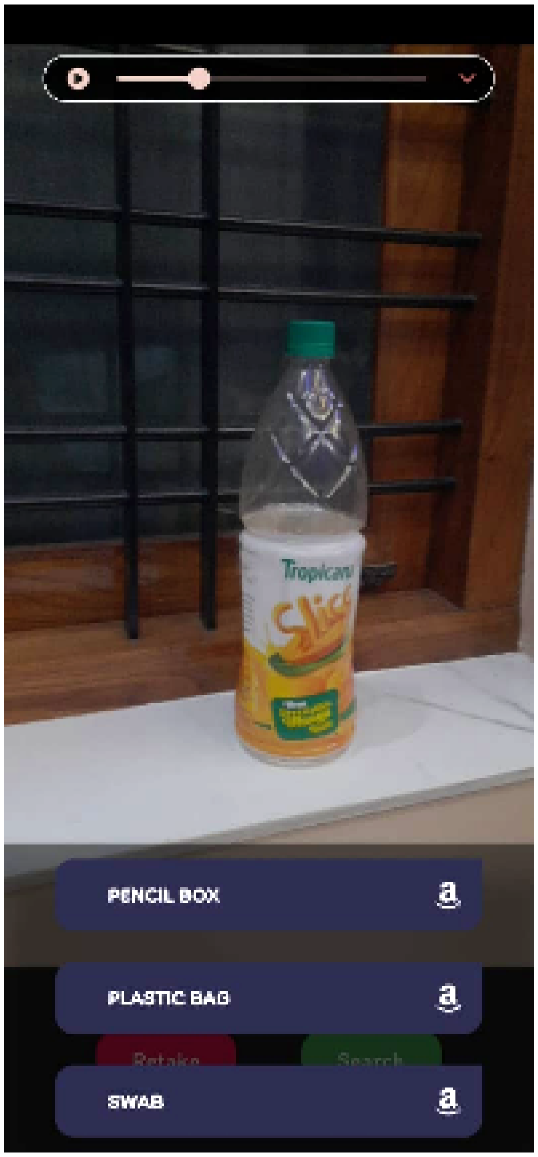



TRIPOD


WALL CLOCK
ANALOG CLOCK

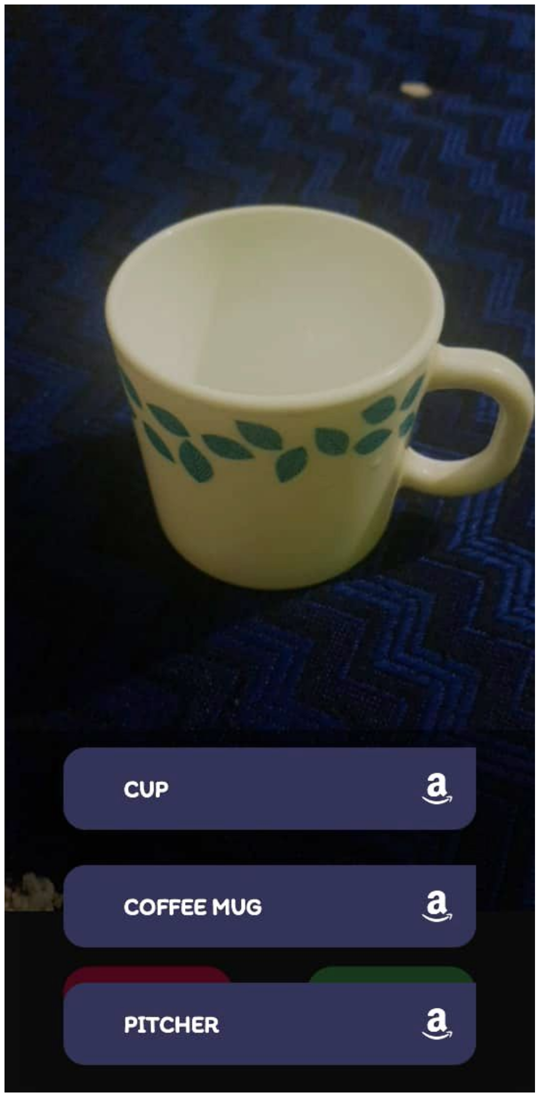
While testing the app using a wide variety of objects, we could see that as long as the object in question is placed under a plain background, the classifications that were produced were more accurate. On the other hand, if the object has a very cluttered background, the accuracy of the application declines.

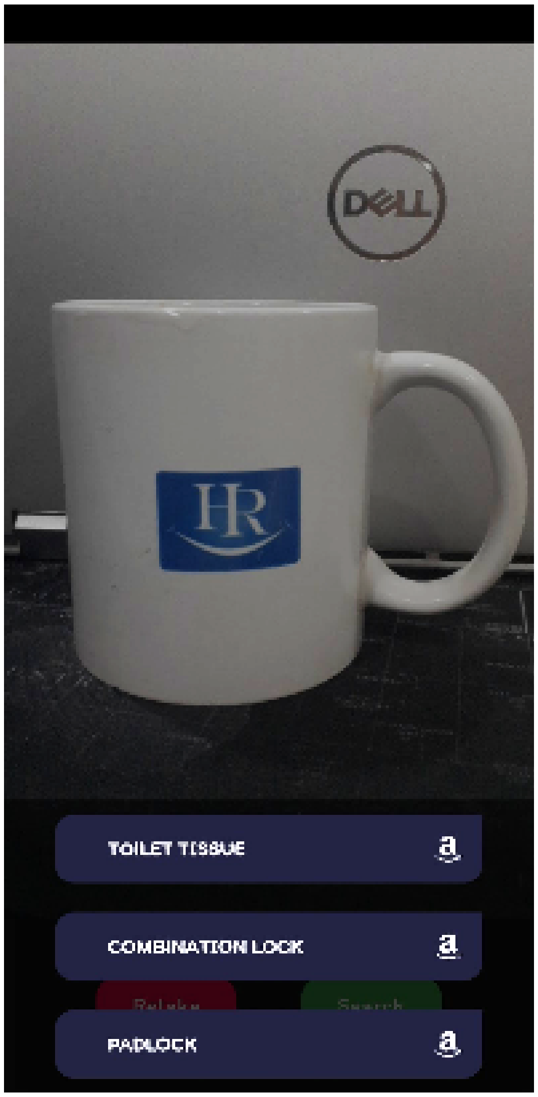




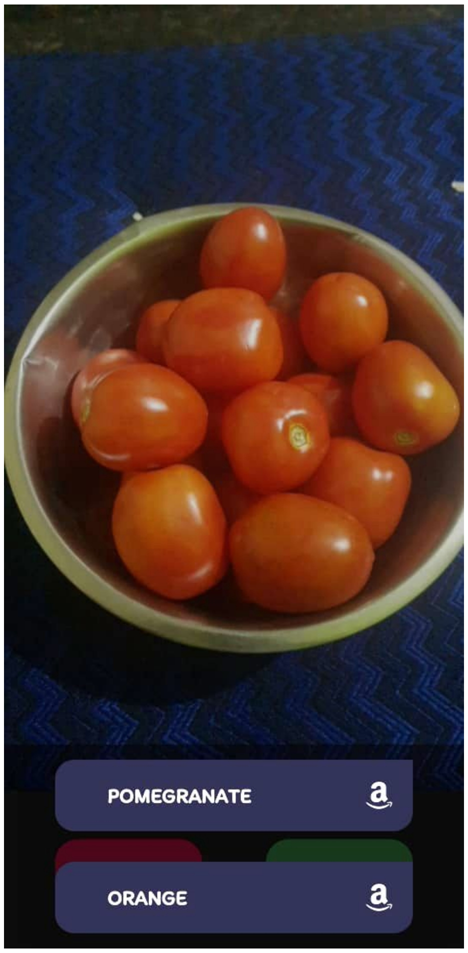
As you can see above, two images of the same object, one in a plain background and one in a slightly cluttered background have returned different classifications.

Two different images of coffee mugs both produced different recommendations. While one produced the correct recommendations, the other was wrong and confusing the object for a padlock, combination lock and toilet tissue. This is again probably due to the reason mentioned above as we see that the mug with the clear background is classified correctly while the one with the cluttered background isn't.





We also took an image of tomatoes which is not a supported class to examine what classification is made. The recommendations made were for pomegranates, oranges which are supported classes. These objects are similar to tomatoes and the classification that is made is reasonable.


POMEGRANATE
ORANGE