e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

COIN COUNTER FOR THE VISUALLY CHALLENGED USING YOLOV5 OBJECT DETECTION

Abishiek Srinivasan*1, Saairam Venkatesan*2, Nikhil Kumar Arumugam Pillai Palani Aram*3, Gokkul Anbazhagan*4

*1,2,3,4UG Scholar, Department Of Computer Science And Engineering, College Of Engineering Guindy Anna University, Chennai, India.

DOI : https://www.doi.org/10.56726/IRJMETS30601

ABSTRACT

There are around 31.6 million people in India in 2020 who are visually challenged. One of the most common challenges they face in their daily life is identifying coins, be it in their homes, in a shopping mall, supermarket and so on. Furthermore, there are different sizes of coins for the same denomination and vice versa. This makes distinguishing the coins difficult for the visually impaired. There is no app or software which helps people with visual impairment to recognise the different types of coins which are available in the country currently. Coin Counter is a mobile application which detects the coins in hand, classifies the nomination of the coins, adds up the total value of coins and outputs the total value as speech.

Keywords: Image Recognizer, Object Detection, Mobile Application Development, Artificial Intelligence, Machine Learning, Computer Vision, Cloud Deployment.).

I. INTRODUCTION

A. Existing problem

The statistics say that there are around 285 million people around the world with visual impairments. One of the most common challenges they face in their daily life is identifying objects in their surroundings, be it in their homes, in a shopping mall, supermarket and so on. There is no app or software which helps people with visual impairment to recognise the different types of coins which are available in the country currently. There is an app from the Reserve Bank of India but it detects only the currency notes.

B. Solution

Coin Counter is a mobile application which detects the coins in hand, classifies the nomination of the coins, adds up the total value of coins and outputs the total value as speech. User opens the application installed on the phone. Application has access to the camera and the user can take a picture of the coins in the palm. Since the end users are visually impaired, gesture control using physical volume buttons are also added to enable easy usage of the app. Coin detection and classifier model, deployed in the cloud, processes the image and sends the output back to the app. The application finally outputs the result in the form of speech..

II. METHODOLOGY

Overview:

This application is specifically designed for the visually challenged to aid them in identifying and calculating the total value of the coins they have in hand. This application is built by keeping visually impaired people in mind. Sound is one of the primary stimulus for them. Consequently, having a UI which is ergonomic is more important than having a visually pleasing one. Keeping this in mind, we have provided multiple language support for people to use this application. The users have the liberty to pick the language they prefer the application to provide the information. The users can use the hardware buttons to choose the language which will be very useful for the visually challenged. For the implementation of this project, we have added support for two languages - English and Tamil. Users are asked to choose between the two when they open the application. Once chosen, all the instructions are provided via voice and text on the screen in their choice of language. Hardware controls are used to facilitate the interaction of visually-impaired people with the application. Although the instructions can be given via voice, taking voice inputs from the user is not secure as it requires microphone access in the background. Interaction using the on-screen controls can hinder the ease of use of application. To overcome this difficulty, this application has the support to get inputs via volume control

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

buttons. Since volume buttons are real world objects whose position does not change with respect to the mobile device, it is easier for users to control the flow of application. Volume hardware buttons provision the targeted users to do the following: Select the language between English and Tamil Take the picture of the coins and retake if necessary Send the picture of coins to the cloud model Navigate throughout the application.

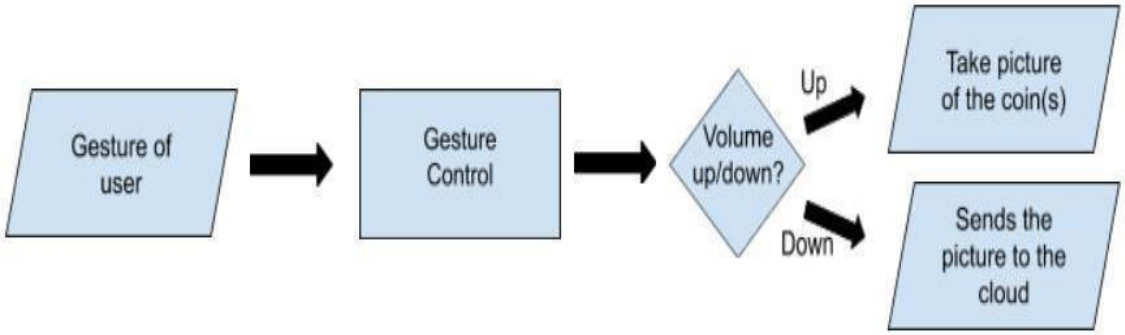

Figure 1: Block Diagram for Hardware Control

Visually impaired people have an ordeal interacting with on-screen controls. As a result, the communication between the device and them can be made possible through either haptic feedback or voice instruction. This application incor- porates voice instruction as one of the media of interaction with the users, which will serve as the bridge between the application's outputs and the user. Users are given the choice to choose the language in which they need the instructions to be given and are guided by voice accordingly. Once the user selects the preferred language, further instructions follow in the same language throughout the application. The voice instructions allow the user to navigate throughout the application. The number and value of the coins are provided to the user also through voice. The user clicks the picture of the coins in hand or on any other surface and sends it to the server, where the ML model is deployed. This model then localizes the coins in the image if any. Once localized, bounding boxes are drawn enclosing the coins. Then, the model classifies the coins inside the boxes into predefined denominations. Calculating value The Classifier model writes the output containing the class names along with the coordinates of the coins in a text file. This module opens up the file and reads the values of the coin and sums up the total. It then sends the number of coins belonging to each class and total sum of the coins present in the image back to the application. The application voices out the information it receives from the server.

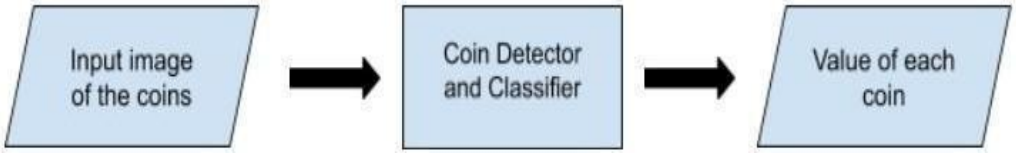

Figure 2: Block Diagram for Coin detection and classification

A. Flutter :

Flutter [1] is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.

- Features of Flutter: Fast development - Paint your app to life in mil- liseconds with Stateful Hot Reload. Use a rich set of fully-customizable widgets to build native interfaces in minutes.
- Expressive and flexible UI - Quickly ship features with a focus on native end-user experiences. Layered architecture allows for full customization, which results in incredibly fast rendering and expressive and flexible designs.
- Native performance - Flutter's widgets incorporate all critical platform differences such as scrolling, naviga- tion, icons and fonts, and your Flutter code is compiled to native ARM machine code using Dart's native com- pilers.

B. Flask :

Flask [2] is a micro web framework written in Python. It is classified as a micro-frame work because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Ex- tensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

C. YOLOv5 :

You Only Look Once (YOLO) [3] is a network that uses Deep Learning (DL) algorithms for object detection. YOLO performs object detection by classifying certain objects within the image and determining where they are located on it. Previous object detection methods like Region- Convolutional Neural Networks(R-CNN), including other variations of it like fast R-CNN, performed object detection tasks in a pipeline of multi-step series. R-CNN focuses on a specific region within the image and trains each individual component separately. This process requires the R-CNN to classify 2000 regions per image, which makes it very time-consuming (47 seconds per individual test image). Thus it cannot be implemented in real-time. Additionally, R-CNN uses a fixed selective algorithm, which means no learning process occurs during this stage so the network might generate an inferior region proposal. This makes object detection networks such as R-CNN harder to optimize and slower compared to YOLO. YOLO is much faster (45 frames per second) and easier to optimize than previous algorithms, as it is based on an algorithm that uses only one neural network to run all components of the task. YOLOv5 is an improvement of YOLOv4. On June 25th 2020, the first official version of YOLOv5 was released by Ultralytics.

III. MODELING AND ANALYSIS

A. Mobile Application - Flutter

Language Selection : There are two languages available in the application as of now - English and Tamil. The users can choose the language they are comfortable with and the same language will follow throughout the application. There are two ways in which the user can pick the language - UI and Hardware control. In UI, there are two raised buttons with each having the corresponding language labeled on them. Clicking on the raised button sets the language which the user can change later if needed. While using hardware control, Volume Up sets the language to English and Volume Down sets the language to Tamil.

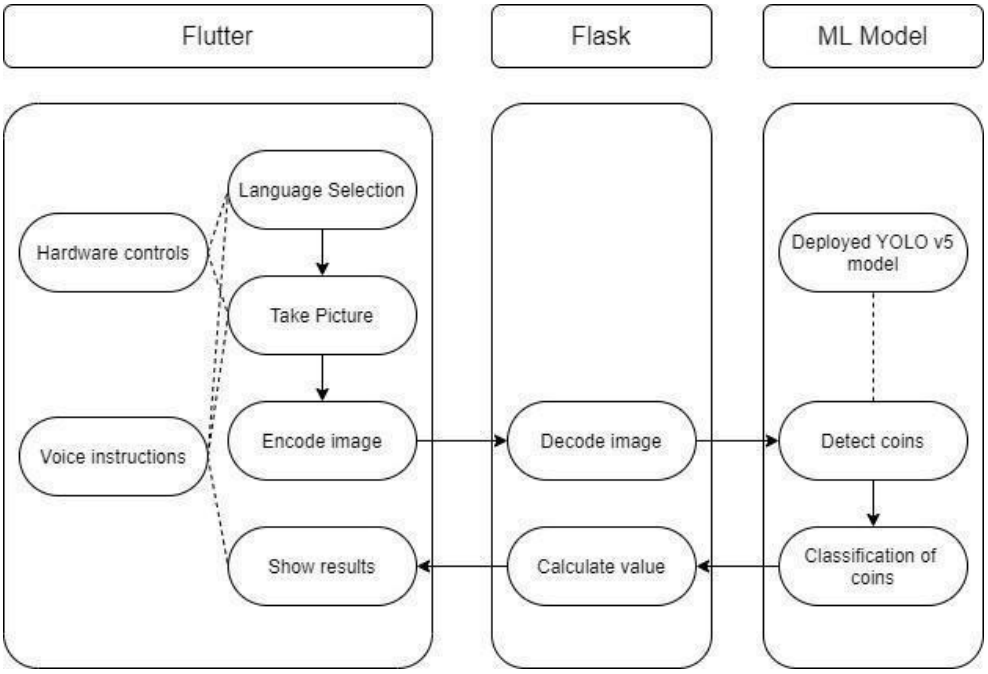

Figure 3: Overall Architecture

Camera Module: In order to take pictures, the device's camera access is required. This requirement is satisfied by the camera [4] plugin from pub.dev. This module allows the user to have a live preview of the camera in a widget. It also allows capturing snapshots and saves it to a file. The camera plugin is first imported as a dependency. The plugin lists out the available cameras and the primary camera is selected. The takePicture() function is used to capture the snapshots and save the image in a temporary directory. This path is used for

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

further manipulation of the image. In terms of hardware buttons, the Volume down button is used to capture the image and the volume up button navigates to the language screen.

Voice Instructions : Sound being the primary input to the targeted audience, voice instructions play a crucial role in guiding the users as well as speaking out the value of the coins to the users. To implement this logic, the flutter tts [5] plugin from pub.dev is used. This module speaks out the string value given to it. It has support for more than 20 lan- guages. The flutter tts plugin allows to set the language, pitch rate, speech rate, speech volume, thus, enabling to customize how each language is spoken. In this application, this plugin talks out the different languages present, instructing the users how to take pictures of the coins, navigating through the application, and speaks out the number and value of the coins if present.

Hardware Controls : For this module to function, hard- ware buttons [6] plugin from pub.dev is used. This plugin is used to create a subscription channel listening to the hardware button events. It can detect three hardware events namely, volume, home screen and lock screen. The vol- ume event alone is employed for implementing the nec- essary functionality. This plugin works for both iOS and Android platforms. In Android, the created subscription channel listens for the KEYCODE events. Each hardware button has a particular KEYCODE constant value, which when clicked gets registered to the subscription channel. KEYCODE VOLUME DOWN (Constant value : 25) gets registered when the user clicks the volume down button. Similarly, KEYCODE VOLUME UP (Constant value : 24) gets registered to the channel when the user clicks the volume up button. Listening to this channel constantly helps in detecting the volume control events and is used to implement the logic.

A. Machine Learning Model

Dataset Preparation: The data for the ML model is sourced from Kaggle [7] as well as locally. After sourcing the data, annotations were given to each image in the form of bounding boxes. Annotations were given in YOLO (.txt) for- mat using LabelIng [8] software. Then, data preprocessing was carried out using Roboflow [9]. The images were auto- oriented, resized to 416x416, and subjected to augmentation steps. Augmentation steps include vertical flip, 90o rotation and brightness adjustments. The final dataset contained 2556 images belonging to 4 different classes. The dataset was then split into training, validation and testing directories each containing 70%, 20% and 10% of total images respectively.

Building And Training Model: The model was built from YOLOv5 with changes in its configuration. YOLOv5 is a ML model developed by Ultralytics and it is open source. The source code is obtained by cloning the GitHub repository. The model is trained on Google Colaboratory

[10] platform, which provides python notebooks with GPU support for faster training. Colaboratory also provides linux-like command line processing. The preprocessed data is downloaded from Roboflow.

After downloading the dataset, the configuration variables for training the ML model are set, which includes the number of classes, anchors, backbone and head. Then the model is trained for 300 epochs.

B. Deployment

The trained model is deployed on Heroku platform using flask, a microframework used for hosting web applications. To get the predicted output from the ML model in the cloud, a POST request is sent to the URL of the cloud server along with the route '/predict', which is the route for the prediction of the sent image. The image is encoded in base64 format string, and is sent to the server. The flask application then decodes the image and runs the model to get the prediction. The predictions are written in a text file

The resultant image is encoded into base64 format string and the different classes of the coins are sent to the flutter application using a JSON response.

IV. RESULTS AND DISCUSSION

Precision and recall are two metrics which together are used to evaluate the performance of classification or infor- mation retrieval systems. Precision : Precision is defined as the fraction of relevant instances among all retrieved instances.

Precision is given by the formula in Figure 4, Where True Positive is an outcome where the model correctly predicts the positive class and False Positive is an outcome where the model wrongly predicts the positive class. Our model attained a precision of 82.50%.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:10/October-2022 Impact Factor- 6.752 www.irjmets.com

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figure 4: Precision

Recall, sometimes referred to as sensitivity, is the fraction of retrieved instances among all relevant instances. Recall is given by the formula in Figure 5,

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 5: Recall

True Positive is an outcome where the model correctly predicts the positive class and False Negative is an outcome where the model correctly predicts the negative class. Our model attained a recall of 95.5%.

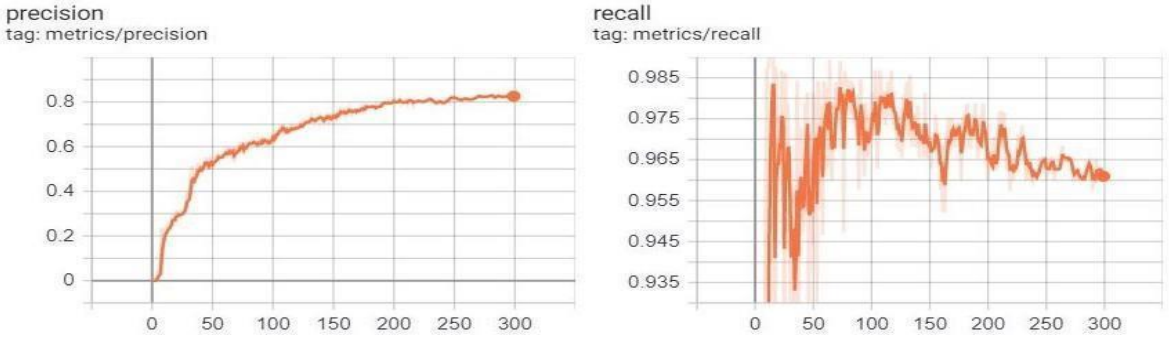

Figure 6: Precision and Recall curve of the model for 300 epochs

V. CONCLUSION

A Coin Counter application is successfully built using Flutter. It sends the images to a web server hosted on Heroku. The web server uses a ML model which was created using YOLOv5 and sends the prediction to the mobile application. The mobile application outputs the result to the user via both voice and text.

ACKNOWLEDGEMENTS

First and foremost, we would like to express our deep sense of gratitude to Mr. Thiyagu , subject instructor for the creative and innovative project lab, under whom we did the project. We would like to thank our faculties for their excellent guidance and constant motivation. We would also like to thank the Dept. of CSE, CEG for sowing the idea for this project.

VI. REFERENCES

[1] Flutter (framework), for documentation visit https://flutter.dev/docs
[2] Flask (framework), for documentation visit https://flask.palletsprojects.com/en/1.1.x/
[3] YOLOv5 (Machine Learning model), for repository and documentation visit https://github.com/ultralytics/yolov5
[4] Camera: 0.5.8+11 (module), for documentation visit https://pub.dev/packages/camera
[5] Flutter tts: 1.3.0 (module), for documentation visit https://pub.dev/packages/flutter tts H
[6] Hardware buttons: 1.0.0 (module) for documentation visit https://pub.dev/packages/hardware buttons
[7] Dataset for training the Machine Learning model, for the dataset visit: https://www.kaggle.com/vishu53/indian-coins-dataset
[8] LabelIing (Python Library), for repository and documentation visit: https://github.com/tzutalin/labelImg
[9] Roboflow (website), https://roboflow.com/
[10] Google Colaboratory - online platform for Jupyter notebooks (plat- form), https://colab.research.google.com/