

Real Time Object Detection Android Application using Machine Learning Concepts for Visually Impaired People

Anil Thapa, Akash Bongane, Madhuri Rokade, Nilesh Wani

Computer Engineering, DY Patil School of Engineering Academy, Ambi, Pune, India

Abstract: In their daily lives, visually impaired and blind people face a variety of obstacles. The proposed project's aim is to create an Android-based application for people who are blind or visually impaired. The system's goal is to detect real-time objects and Indian currencies that are scanned through a mobile camera and notify blind people about them through audio or vocal information. The proposed framework makes use of Machine Learning and Google's TensorFlow Lite object detection API. It's a useful app that makes it much easier for blind and visually disabled people to complete daily tasks with less effort. It's a small contribution to India's Make-In-India movement.

Keywords - Object Detection, Machine Learning, Android Application, Image, SSDMOBILENET, Visually Impaired People, Computer Vision, Algorithms, Neural Networks.

INTRODUCTION

Production of images:

Image processing focuses on transforming raw images to achieve a desired result. The aim of image processing is usually to enhance images or prepare them as input for a particular task, while the goal of computer vision is to describe and illustrate images. Noise reduction, contrast, and rotation operations, for example, are common image processing operations that can be performed at the pixel level without requiring a complex understanding of what is going on in the image.

Machine Learning:

Artificial Intelligence includes Machine Learning as a subset. Machine learning is the study of giving computers the opportunity to learn and create their own programmes in order to make them more human-like in their actions and decisions. This is accomplished with the least amount of human interaction possible, i.e. no explicit programming. The learning process is automated and enhanced based on the machines' experiences along the way. Machines are fed high-quality data, and different algorithms are used to create machine-learning models based on that data. The algorithm to use is determined by the type of data available and the type of operation to be automated.

Supervised Machine learning Model:

In supervised learning, you use well-labeled data to train the computer. It indicates that some information has already been marked with the correct answer. It is comparable to learning that occurs in the presence of a supervisor or instructor. A supervised learning algorithm learns from labelled training data and aids in the prediction of unexpected data outcomes. Accurate supervised machine learning can be designed, scaled, and deployed successfully.

Unsupervised Machine learning Model:

Unsupervised learning is a machine learning methodology in which the algorithm does not require supervision. Instead, you should let the model figure out what it needs to do on its own. It is primarily concerned with unlabeled data. In comparison to supervised learning, unsupervised learning algorithms allow you to perform more complex processing tasks. Unsupervised learning, on the other hand, can be more volatile than other natural learning approaches such as deep learning and reinforcement learning.

Flutter :

Flutter is an open-source mobile SDK that allows developers to build native-looking Android and iOS apps from a single code base. Flutter has been around since Google released it in 2015, and it remained in beta until December 2018, when it was officially launched. Flutter has been generating a lot of interest since then. According to GitHub stars, Flutter is now in the top 11 app repositories.

The use of widgets is at the heart of Flutter. Developers can create the entire UI by merging different widgets. Each of these widgets describes a structural element (such as a button or menu), a stylistic element (such as a font or colour scheme), a layout attribute (such as padding), and a variety of other features. Flutter does not use OEM widgets; instead, it provides developers with ready-made widgets that appear to be native to Android or iOS apps (following Material Design or Cupertino). Developers will, of course, make their own widgets. Flutter also includes reactive-style views for developers. Flutter uses Dart as the JavaScript bridge to prevent performance problems caused by using a compiled programming language. For multiple platforms, it compiles Dart ahead of time (AOT) into native code. Flutter would be able to connect with the network without the use of a JavaScript bridge, which requires a context transition between the JavaScript and native realms. Compiling to native code, as you would expect, increases the app's startup time.

Literature Survey

1. Object Detection Using Machine Learning for

Visually Impaired People

Author Name: Venkata Naresh Mandhala¹, Debnath Bhattacharyya¹, Vamsi B.2, Thirupathi Rao N

Description: The Yolo convolution neural network algorithm divides the image into different parts and identifies closed boxes and probabilities for each part, using only a single structure neural network to detect the entire image. The image's bounding boxes are measured using probabilities that have already been determined. YOLO is used as a replacement for scale in the picture that was held under identifying the objects. Each box in the frame has a parameter (pp,bx, by,bh,bw,p), where 'pp' may be 0 or 1, indicating whether an individual 'p' is present in the picture or not, 'bx' and 'by' indicating the box's midpoint, and 'bh,' 'bw' indicating the box's height and width, respectively. Yolo V3' is a new version of Yolo. As tested, Yolo V3 was able to detect 20 objects with a 71.1 percent accuracy. It also operates with an accuracy of 80-95 percent for some of the artefacts SSD, like Faster-RCNN, uses anchor boxes in a number of aspect ratios and knows the off-set to a degree rather than the box. SSD predicts bounding boxes after several convolutional layers in order to keep the size. Since each convolutional layer operates at a different scale, it can detect artefacts of various scales.

2.Real-Time Object Detection for Visually Challenged People

Author Name: Sunit Vaidya, Naisha Shah, v, Prof. Radha Shankarmani

Description: In this paper, The datasets used in the proposed solution (Tiny YOLOv3 and YOLOv3) perform extremely well on both the Android and Web apps. Tiny YOLOv3 has a higher mAP value than YOLOv3, meaning that it detects objects in its frame with greater precision. Tiny YOLOv3 is more effective in small object detection than YOLOv3 due to its low average loss and high mean average precision. developed By pressing the same button again, the Yolo algorithm can be turned off. Using the YOLOv3-tiny algorithm, objects around blind people and their positions are detected and communicated to them through audio output. In our proposed system we have observed that YOLOv3 in real time gives less accuracy as compared to SSD.

3.An Object Detection Technique For Blind People in Real-Time Using Deep Neural Network

Author Name: Ashwani Kumar, S S Sai Satyanarayana Reddy, Vivek Kulkarni

Description: This paper proposes an object detector model for blind people that uses deep learning neural networks to detect the object. This model can recognise objects in photographs, videos, and even webcam feeds. The model is accurate to within a factor of 75. This model takes approximately 5-6 hours to practise. The SSD MobileNet FPN performs localization and classification of neural networks and is thus fast compared to other models. This model uses deep neural networks to extract feature information and then perform feature mapping. We used a single shot multi-box detector (SSD) algorithm in this study to achieve high accuracy and IOU in real time for object detection for blind people.

4. Deep Learning Based Indian Currency Detection for Visually Challenged using VGG16

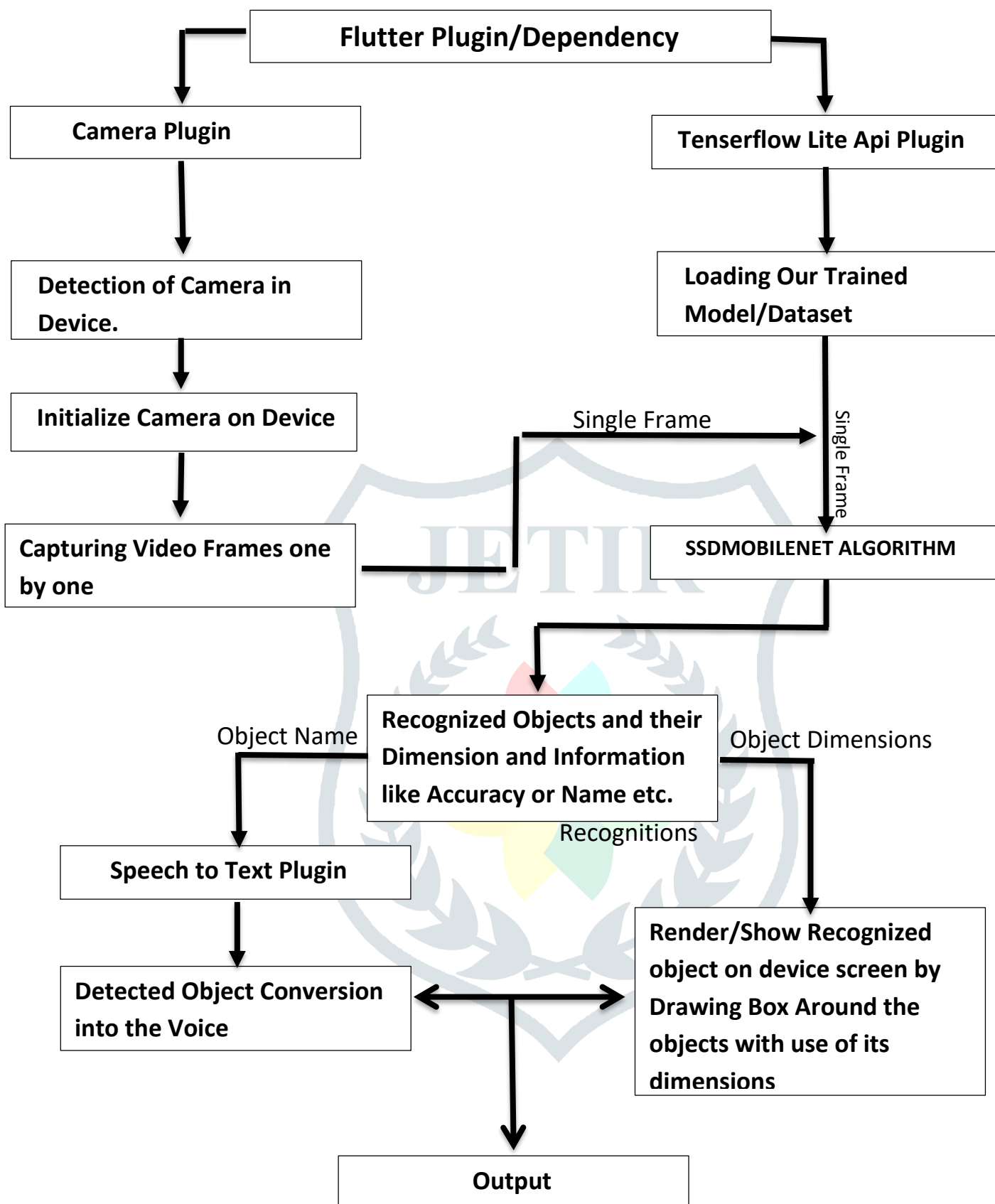
Author Name: Nijil Raj N, Anandu S Ram, Aneeta Binoo Joseph, Shabna S

Description: The method proposes a vision-based deep learning technique in this paper that can identify and classify Indian currencies. Deep features were extracted from the input image using VGG 16. The current framework employs Alexnet, which extracts fewer functions. As a result, VGG16 is employed, which is capable of extracting deeper features. The Single Shot MultiBox Detector (SSD) which we have used in our system is a refinement of the VGG16 system for object detection.

METHODOLOGY:

Flutter Architecture View





1. Flutter Architecture View:

Typically, developers interact with Flutter through the Flutter framework, which provides a modern, reactive framework written in the Dart language. It includes a rich set of platform layout, and foundational libraries, composed of a series of layers working from the bottom to the top, we have: Basic foundational classes, and building block services such as animation, painting, and gestures that offer commonly used abstractions over the underlying foundation. The rendering layer provides an abstraction for dealing with layout. With this layer, you can build a tree of renderable objects. You can manipulate these objects dynamically, with the tree automatically updating the layout to reflect your changes. The widgets layer is a composition abstraction. Each render object in the rendering layer has a corresponding class in the widgets layer. In addition, the widgets layer allows you to define combinations of classes that you can reuse. This is the layer at which the reactive programming model is introduced. The Material and Cupertino libraries offer comprehensive sets of controls that use the widget layer's composition primitives to implement the Material or iOS design languages.

1.1 Flutter/Dart Framework:

Flutter is designed as an extensible, layered system. It exists as a series of independent libraries that each depend on the underlying layer. No layer has privileged access to the layer below, and every part of the framework level is designed to be optional and replaceable.

1.2 Engine C/C++:

At the core of Flutter is the Flutter engine, which is mostly written in C++ and supports the primitives necessary to support all flutter application. The engine is responsible for rasterizing composited scenes whenever a new frame needs to be painted. It provides the low-level implementation of Flutter's core API, including graphics (through Skia), text layout, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. The engine is exposed to the Flutter framework through `dart:ui`, which wraps the underlying C++ code in Dart classes. This library exposes the lowest-level primitives, such as classes for driving input, graphics, and text rendering subsystems.

1.3 Embedder Platform Specific:

A platform-specific embedder provides an endpoint; coordinates with the underlying operating system for access to services like rendering surfaces, accessibility, and input; and manages the message event loop. The embedder is written in a language that is appropriate for the platform: currently Java and C++ for Android, Objective-C/Objective-C++ for iOS and macOS, and C++ for Windows and Linux. Using the embedder, Flutter code can be integrated into an existing application as a module, or the code may be the entire content of the application. Flutter includes a number of embedders for common target platforms, but other embedders also exist.

2. Dependencies For Flutter:

Dependencies are one of the core concepts of the pub package manager. A dependency is another package that your package needs in order to work. Dependencies are specified in your `pubspec`. You list only immediate dependencies — the software that your package uses directly. Pub handles transitive dependencies for you. For each dependency, you specify the name of the package you depend on and the range of versions of that package that you allow. You can also specify the source, which tells pub how to locate the package, and any additional description that the source needs to find the package. Here is an example of including in our app.

Syntax:

```
dependencies:
  dependencies_name: ^0.8.1

//where ^0.8.1 is version of that plugin
```

3. Camera Plugin/Package:

A Flutter Camera plugin for getting Access to camera and controlling the camera on Android and iOS, Plugin provided by official flutter authors.

Features:

- Supports previewing the camera feed
- capturing images
- capturing video
- and streaming image buffers/frames

Syntax to include in app:

```
dependencies:
  camera: ^0.8.1

//where ^0.8.1 is version of that plugin
```

4. Tensorflow lite Plugin:

TensorFlow Lite Flutter plugin provides a flexible and fast solution for accessing TensorFlow Lite interpreter and performing inference. The API is similar to the `tflite` Java and Swift APIs. It directly binds to TF `tflite` Lite C API making it efficient (low-latency). `TfLite` Flutter Helper library aims at providing an interface to easily use high-level objects involved in Image, Text, Audio processing with `TfLite` Flutter plugin, and related utility functions in addition to similar functionality as the TensorFlow Lite Android Support Library.

Syntax:

```
dependencies:
  tflite: ^1.1.2

//where ^1.1.2 is version of that plugin
```

5.Loading Our Trained Dataset/Model:

Initially when we Enter in our app we must initialize Dataset in our Flutter Code to use it Further for recognitions

Syntax:

```
await Tflite.loadModel(
  model: "assets/file_name.tflite",
  labels: "assets/file_name.txt",
);
```

6.Detecting Cameras in Phone:

By the use of Camera Plugin We can easily Have Access Of Any Android device Cameras.By the use of Function Defined in Camera Plugin named availableCameras() we get The number of Cameras Available in the Android device .

Syntax:

```
cameras = await availableCameras();
```

7.Initialize Cameras:

With the use Of Cameracontroller property of Camera Plugin we can Initialize camera on any device.

Syntax:

```
new CameraController(
  cameras[0], );
```

8.Capturing Frames Of Camera :

So After initialization of Cameras Camera Start Capturing frames one by one and Further it will Send Each Frame to ourAlgorithm/Model.

Syntax:

```
controller.startImageStream((CameraImage img) {});
```

9.SSDMobileNet:

The mobilenet-ssd model is a Single-Shot multibox Detection (SSD) network intended to perform object detection. This model is implemented using the Caffe* framework.SSD (Single Shot Detector) is reviewed. By using SSD, we only need to take one single shot to detect multiple objects within the image, while regional proposal network (RPN) based approaches such as R-CNN series that need two shots, one for generating region proposals, one for detecting the object of each proposal. Thus, SSD is much faster compared with two-shot RPN-based approaches.

10.Passing Captured Frames to Any One Algorithm:

```
Tflite.detectObjectOnFrame(
  bytesList: img.planes.map((plane) {
    return plane.bytes;
  }).toList(),
  model: "SSDMobileNet ",
  imageHeight: img.height,
  imageWidth: img.width,
```

```

        imageMean: 127.5,
        imageStd: 127.5,
        numResultsPerClass: 1,
        threshold: 0.4,
    )

```

11.It will give you the Recognitions on passed frame which will display on the screen as render boxes

12.Text to speech Plugin:

Plugin used for conversion of any text into the Desired Voice in any language .

In Our Case we have passed our recognition to this plugin so it will give you're your Desired output.

Syntax For Including Plugin In Our App:

dependencies:

flutter_tts: ^3.0.0

OBJECT NAME	PERSON	LAPTOP	MOUSE	REFRIGERATOR
SSDMOBILENET (ACCURACY)	82%	83%	83%	82 %
YOLO V2 (ACCUARCY)	72%	76%	60%	70%

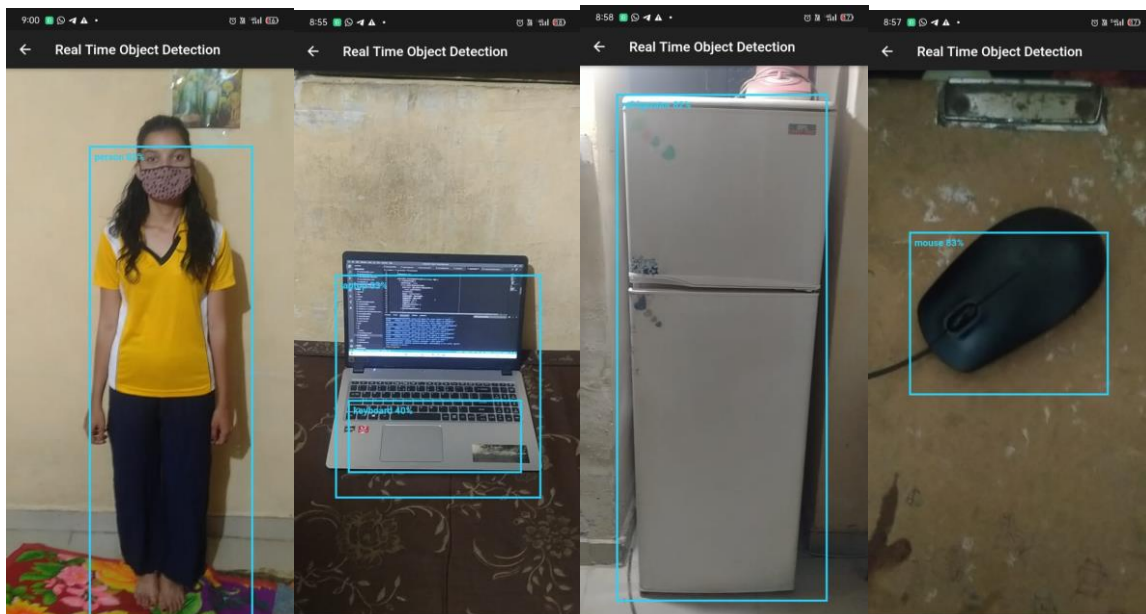
Syntax For Use:

```

Future _speak() async{
    var result = await flutterTts.speak("Hello World");
    if (result == 1) setState(() => ttsState = TtsState.playing);
}

```


Result:



SSDMOBILENET ACCURACY(ANDROID)

REFERENCE

- [1] Nasreen, W. Arif, A. A. Shaikh, Y. Muhammad and M. Abdullah, "Object Detection and Narrator for Visually Impaired People," 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Kuala Lumpur, Malaysia, 2019, pp. 1-4, doi: 10.1109/ICETAS48360.2019.9117405
- [2] S. Vaidya, N. Shah, N. Shah and R. Shankarmani, "Real-Time Object Detection for Visually Challenged People," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 311-316, doi:10.1109/ICICCS48265.2020.9121085
- [3] A. Kumar, S. S. S. S. Reddy and V. Kulkarni, "An Object Detection Technique For Blind People in Real-Time Using Deep Neural Network," 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 2019, pp. 292-297, doi: 10.1109/ICIIP47207.2019.8985965.
- [4] Venkata Naresh Mandhala¹, Debnath Bhattacharyya¹, Vamsi B.2, Thirupathi Rao N. Object Detection Using Machine Learning for Visually Impaired People. International Journal of Current Research and Review Factor: 6.1 (2018) ICV: 90.90 (2018).
- [5] Nijil Raj N, Anandu S Ram, Aneeta Binoo Joseph, Shabna S. Deep Learning Based Indian Currency Detection for Visually Challenged using VGG16 International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-9 Issue-2, July 2020.

- [6] Snehal Saraf,Vrushali Sindhikar,Ankita Sonawane,Shamali Thakare. Currency Recognition System For Visually Impaired IJARIE-ISSN(O)-2395-4396 Vol-3 Issue-2 2017.
- [7] Rakesh Chandra Joshi,Saumya Yadav,Malay Kishore Dutta,Carlos M.Travieso-Gonzalez Efficient Multi-Object Detection and Smart Navigation Using Artificial Intelligence for Visually Impaired People. MDPI Entropy 2020, 22,941.August 2020.
- [8] Rohith Pokala,Varun Teja. Indian Currency Recognition for Blind People. IRJET e-ISSN: 2395-0056 Volume: 07 Issue: 05,May 2020.
- [9] ShwetaYadav, Mr. Zulfikar Ali Ansari, KaushikiGautam Singh. Currency Detection for Visually Impaired. JETIR ISSN-2349-5162 May 2020, Volume 7, Issue 5..
- [10] Anitha.J, Subalaxmi.A, Vijayalakshmi.G. Real Time Object Detection for Visually Challenged Persons IJITEE ISSN: 2278-3075, Volume-8 Issue-8 June, 2019

