Project Report
on

DRISHTI: VISUAL ASSIST FOR THE VISUALLY CHALLENGED PEOPLE

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

by

ADITI SHUKLA (190027031009)
JAYESH SRIVASTAVA (1900270310075)
MALAY JOSHI (1900270310088)
MANYA RASTOGI (1900270310092)

Under the supervision
of

Ms. SUVARNA MUJUMDAR
Assistant Professor
Department of Electronics and Communication Engineering

AJAY KUMAR GARG ENGINEERING COLLEGE GHAZIABAD
27th Km. Stone, NH-24, Delhi-Hapur Bypass Road, Adhyatmik Nagar, Ghaziabad- 201009

DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW

Year : 2022 – 2023

AJAY KUMAR GARG ENGINEERING COLLEGE GHAZIABAD

CANDIDATES' DECLARATION

We hereby declare that the work being presented in the report entitled **"Drishti : Visual Assistance for the visually challenged"** submitted to Ajay Kumar Garg Engineering College, Ghaziabad, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering of Dr. A. P. J. Abdul Kalam Technical University, Lucknow, is an authentic record of our own work carried out during a period from July, 2022 to May, 2023 under the supervision of **Suvarna Mujumdar**, Assistant Professor, Department of Electronics and Communication Engineering. The matter presented in this report has not been submitted to any other university or institution for the award of any degree to the best of my knowledge.

ADITI SHUKLA **(1900270310009)**	**JAYESH SRIVASTAVA** **(1900270310075)**
MALAY JOSHI **(1900270310088)**	**MANYA RASTOGI** **(1900270310092)**

AJAY KUMAR GARG ENGINEERING COLLEGE GHAZIABAD

CERTIFICATE

This is to certify that the project report entitled **"Drishti : Visual Assistance for the visually challenged"** submitted to **Ajay Kumar Garg Engineering College, Ghaziabad**, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** of **Dr. A. P. J. Abdul Kalam Technical University, Lucknow**, is bonafide record of work submitted by Aditi Shukla (1900270310009), Jayesh Srivastava (1900270310075), Malay Joshi (190270310088) And Manya Rastogi (1900270310075) under my supervision. The matter presented in this report has not been submitted to any other university or institution for the award of any degree to the best of my knowledge.

Ms. SUVARNA MUJUMDAR	**Dr. P. K. CHOPRA**
Assistant Professor	**Prof & Head**
Department of ECE	**Department of ECE**
AKGEC, Ghaziabad	**AKGEC, Ghaziabad**

ACKNOWLEDGEMENTS

We take this opportunity to express our deep sense of gratitude and regard to **Ms. Suvarna Majumdar,** Asst. Professor (ECE Dept.), Ajay Kumar Garg Engineering College, Ghaziabad for his continuous encouragement and able guidance, we needed to complete this project. We would pay our sincere gratitude to the Head of the Dept. (ECE), **Prof. P. K. Chopra** for his precious and enlightening words of wisdom which motivated us throughout our project work.

ABSTRACT

“In today's society, where independent living is becoming increasingly important, it can be extremely constricting for those who are blind. Blind and visually impaired (BVI) people face challenges because they need manual support to prompt information about their environment. In this work, we took our first step towards developing an affordable and high-performing eye wearable assistive device, DRISHTI, to provide visual navigation assistance for BVI people. This system comprises a camera module, ESP32 processor, Bluetooth module, smartphone and speakers. Using artificial intelligence, this system is proposed to detect and understand the nature of the users' path and obstacles ahead of the user in that path and then inform BVI users about it via audio output to enable them to acquire directions by themselves on their journey. This first step discussed in this report involves establishing a proof-of-concept of achieving the right balance of affordability and performance by testing an initial software integration of a currency detection algorithm on a low-cost embedded arrangement. This work will lay the foundation for our upcoming works toward achieving the goal of assisting the maximum number of blind people around the globe in moving independently.”

TABLE OF CONTENTS

Chapter 5: Results

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Description
BVI	Blind and Visually Impaired
ESP	Espressif Systems
WHO	World Health Organization
AI	Artificial Intelligence
CNN	Convolutional Neural Network
OCR	Optical Character Recognition
YOLO	You Only Look Once
Open CV	Open Source Computer Vision
VGG	Visual Geometry Group
TTS	Text-To-Speech
Gtts	Google Text-to-Speech
API	Application Programming Interface
CAM	Camera Module
DIY	Do it yourself
WiFi	Wireless Fidelity

IoT	Internet of Things
BLE	Bluetooth Low Energy
CPU	Central processing unit
OV2640	OmniVision2640
microSD	Micro Secure Digital
GPIO	General Purpose Input/Output
CNN	Convolutional Neural Network
ID Block	Identity Document Block
FC	Fully Connected
CLI	Command-line Interface
IDE	Integrated Development Environment
GND	Ground
UNO	United Nations Organization
RESET	Resetting
TX & RX	Transmitted & Receiver
OCR	Optical Character Recognition

CHAPTER 1

INTRODUCTION

Drishti is a pioneering project aimed at empowering visually challenged individuals especially in case of blindness which is a daunting condition. This system integrates state-of-the-art technologies to enable visually impaired individuals to perceive and comprehend visual information, also enhancing their accessibility and independence. The core components of Drishti include a wearable device equipped with a camera and sensors, a powerful AI algorithm for real-time image recognition, and a user interface for conveying the interpreted visual information to the user through audio or tactile feedback. By utilizing these advanced technologies, Drishti has the potential to revolutionize the lives of visually challenged individuals, enabling them to navigate their surroundings, recognize objects, and access visual content more efficiently.

Visually challenged individuals often face significant challenges in their daily lives due to their impaired vision. According to a report by the WHO in 2013[1], there are an estimated 40 to 45 million people who are blind, and about 135 million have low or weak sight. According to a report by The Hindu, 62 million people in India are visually impaired, of which eight million are blind. Traditional aids such as white canes and guide dogs provide some assistance but have limitations in providing a comprehensive understanding of the visual world. The emergence of artificial intelligence, computer vision, and wearable devices has opened up new possibilities for addressing these limitations and empowering visually challenged individuals. Also, visual impairment can impact a person's quality of life and make them prone to discrimination. They face many challenges in navigating around places. There is a large number of adaptive equipment that enable visually impaired people to live their life independently. However, they are only found in nearby shops or marketplaces. Also, they are quite expensive, so only a few blind people can use such resources.

These challenges include but are not limited to:

1. Navigation: One of the most crucial aspects of independence is the ability to navigate safely and efficiently. Visually challenged individuals often face difficulties in recognizing obstacles, reading signs, and identifying landmarks, making navigation a daunting task.With the help of accessible technology or products, and their own willpower, blind people can be independent.

2. Object Recognition: Identifying and differentiating objects can be challenging without visual cues. Simple tasks such as distinguishing between similar products in a store or identifying everyday items can be time-consuming and frustrating for visually challenged individuals.

3. Text and Print Accessibility: Reading written text is a fundamental skill that enables individuals to access information independently. However, visually challenged individuals often struggle to read printed materials, such as books, signs, or labels, as they are not accessible in their standard format.

Drishti comprises a wearable device, an AI algorithm, and a user interface. The wearable device consists of a camera, ESP32, and audio mechanisms.The ESP32 camera module will capture images of the target(s) like surrounding objects, people, text document, currency, road, traffic signal,etc. This captured image will be sent to a smartphone device and will be processed in real time by multiple algorithms. Then information extracted will be converted into audio based signals which will act as a feedback for the user. This data is then processed by the AI algorithm, which performs real-time image recognition and interprets the visual information. The interpreted information is conveyed to the user through audio feedback or tactile signals via the user interface.

The AI algorithm at the core of Drishti utilizes deep learning techniques to perform real-time image recognition. It has been trained on vast datasets of diverse visual content, allowing it to identify objects, recognize text, and understand spatial relationships. The algorithm employs advanced computer vision models such as CNN model and neural networks to process the visual data captured by the wearable device, enabling it to provide accurate and reliable interpretations.

The user interface of Drishti is designed to effectively convey the interpreted visual information to the user. For individuals who are blind, audio feedback plays a crucial role. The system can convert the visual information into spoken descriptions, providing the user with a comprehensive understanding of their surroundings. Additionally, tactile feedback mechanisms such as haptic vibrations or braille displays can be employed to convey information to individuals with both visual and hearing impairments.

Drishti has the potential to revolutionize the lives of visually challenged individuals by enhancing their independence and accessibility. With the ability to navigate their surroundings, recognize objects such as pen, book, currency and access visual content, visually impaired individuals can better engage in various activities, such as navigating unfamiliar environments, shopping independently, reading printed materials, and interacting with digital interfaces. ive experience.This project is intended to help this type of people to widen their scope of independence by giving them a description of the live scenes delivered in an audio format.

Drishti represents a groundbreaking project that harnesses the power of artificial intelligence, computer vision, and wearable devices to empower visually challenged individuals. By providing real-time visual information and enhancing their perception of the world, Drishti promotes accessibility and independence. As a recommendation for future work, more optimization functions should be added, and user testing results and feedback can be included for better improvements. Text recognition features can be implemented using TesseractOCR. It is a printed text reader. It is an optical character recognition engine for various OS. OCR extracts text from images and documents without a text layer and outputs the document into a new searchable text file, PDF, or most other popular formats.

Furthermore, as it has been concluded from the experiment results, the camera quality is a very important component, therefore, as a recommendation a better camera can be added to the system to improve the results accuracy. As technology continues to advance, Drishti holds the promise of further improvements and innovations, paving the way for a more inclusive society where visually impaired individuals can thrive and participate fully.

CHAPTER 2

LITERATURE REVIEW

The visually impaired face numerous challenges in their daily lives, including navigation, object recognition, and accessing visual information. Visual assist projects aim to alleviate these difficulties by leveraging technological advancements such as computer vision, machine learning, and wearable devices.

- The camera-operated mechanism in work by Samruddhi Deshpande and Revati Shriram(2016)[2] helps them read text on things that are held in their hands easily. The proposed system uses a camera to capture the target object, and an algorithm extracts the text in the captured image from the backdrop. An algorithm extracts text from the captured image using optical character recognition (OCR) techniques. The identified text is then converted into audio output, enabling the user to access the information through a software development kit. This approach offers a portable and convenient solution for text readingAudio output is provided using a software development kit with the identified text.
- The wearable system in paper by Deepthi Jain B, Shwetha M Thakur and K V Suresh[3] is a device that receives user input and recognises things. The device has an ultrasonic sensor that assists in warning the user of objects that are in his or her path. The items are located using the Haar cascade method. It is a wearable device that can be mounted on the user's chest. The user will receive an audio of the object that was discovered.This literature review provides an overview of existing research and technologies employed in visual assist projects for visually challenged individuals.
- Traffic scenes use object detection technology in research by Jing Tao, Hongbo Wang, Xinyu Zhang, Xiaoyu Li and Huawei Yang(2017)[4] is used to find objects. Here, they have used a combination of R-FCN (Regression-based Full Convolution Network) and YOLO (optimized you only look once), which is 1.18 times faster than YOLO. It identifies and categorizes photos of vehicles, cyclists, and other objects. Location errors occur when using YOLO; to prevent these, we employ YOLO. Other possible categories of solutions

analyzed in, with promising present and future utilities, are based on the existing technologies. The system uses features like Artificial Intelligence and Machine Learning to provide a solution to the problem. A device camera captures images, objects are detected, and distance is calculated using an application.

- Utilization of computer framework advances for navigational purposes in works of Roberto Netoa, Nuno Fonsecaa(2014)[5] is proposed as a relatively recent technology. Commonly, the scene is perceived by extracting images/videos using vision sensors and a controller is used to adjust the camera view. Paper talks about a conventional method that has been used for decades for assistance of the visually impaired is the use of guide dogs that are trained to help the blind to navigate and walking cane for avoiding obstacles. In, an approach is executed to the Recognition arrangement of characters and the line of perusing. As a component of the product advancement, the Open CV (Open source Computer Vision) libraries are used to do picture catch of content, to do the character Recognition. The greater part of the entrance innovation apparatuses worked for individuals with visual impairment and restricted vision are based on the two essential structure squares of OCR programming and Text-to-Speech motors.
- The research on wearable device by Marko Lazić*Ana Perišić, Predrag Šidjanin, (2015) [6] is a A cutting-edge wearable device positioned on the user's chest that incorporates an advanced ultrasonic sensor that efficiently detects obstacles obstructing the user's path. By utilizing the highly accurate Haar cascade method for object detection, this device empowers visually impaired individuals with enhanced navigation capabilities. The device provides invaluable assistance by relaying audio feedback to the user, allowing them to identify and avoid potential obstacles, thereby significantly improving their overall safety and independence while navigating through their surroundings. With its innovative design and sophisticated technology, this wearable device proves to be an invaluable tool in facilitating obstacle avoidance for visually impaired individuals.
- Works of M. Ebling (2009)[7], R. Lima, L. Barreto, A. Amaral and S. Paiva(2023) and Edwards and Alistair D.N. (2006)[9] showed that traditional navigation aid methods such as sticks are very limited. The proposed model is to build a smart walking stick that has

the ability to take pictures and recognize and tell obstacles in the path. The photo is taken by the user using a joystick switch in the direction the user wishes. The output is read aloud as to what objects are present in that direction. In smart sticks, all the operations are done using images which reduces the infrastructure significantly. Therefore it is much cheaper than the ultrasonic ones. To assist blind users in identifying the type of room using smartphone-based acoustic fingerprinting. The proposed work offers cognitive assistance to blind people in room type recognition and awareness regarding its occupants. The approach used the audience dataset and CAFFE model. It uses image processing for counting the number of people, age group,and gender.

- In a journal by Rakesh Chandra Joshi, Saumya Yadav, Malay Kishore Dutta, and Carlos M. Travieso-Gonzalez(2020)[10] focuses on providing the helping aid for the visually impaired person with an object detection and navigation system. The proposed module is divided into two, one is static object detection using a raspberry-pi setup. Secondly the system deals with the dynamic obstacles too. This algorithm detects the shape of the object and searches from the database provided in the hardware and converts the textual information to audio information. It presents an ultrasonic waist belt which provides electronic navigation for visually impaired and blind persons. Ultrasonic waist belt detects obstacles in front, bottom, left and right directions using a network of ultrasonic sensors. To inform the person about the detected obstacle and its distance, the information from the ultrasonic waist belt is passed to the mobile application which then alerts the user through voice instructions.
- Mobile Assistive Application for Blind People in the work by Ádám Csapó, György Wersényi, Hunor Nagy and Tony Stockman, (2015) is a mobile assistive application has been developed to aid visually impaired individuals in obtaining direction information using color targets. This innovative solution addresses the challenges associated with detecting and tracking targets accurately using the mobile camera. To overcome the limitations of current methods, the proposed approach leverages the CamShift algorithm to enhance the location and identification of color targets. By incorporating this algorithm, the

application aims to provide more reliable and precise orientation guidance for visually impaired users, reducing the likelihood of detection failures and inaccurate tracking.

- In the paper by A. Ganz, J. Schafer, E. Puleo, C. Wilson and M. Robertson,(2012)[12] the CamShift algorithm-based method offers significant improvements in detecting and tracking color targets compared to existing techniques. By utilizing this algorithm, the mobile assistive application can effectively locate and identify the color targets in real-time. This allows visually impaired individuals to receive more accurate orientation information, helping them navigate their surroundings with greater confidence and independence. In D. Jain's(2014)[13] work, the enhanced performance of the application minimizes detection errors and ensures that users can rely on the provided directions to reach their desired destinations efficiently.
- Smart Walking Stick with Image Recognition in Albawi, S., Mohammed,(2020)[14] introduces a cost-effective solution for assisting visually impaired individuals by incorporating a camera into a smart walking stick. The camera captures images and employs image recognition technology to identify obstacles in the surrounding environment. Users can conveniently take photos using a joystick switch, and in Ziad O. Abu-Faraj, Paul Ibrahim, Elie Jabbour and Anthony Ghaoui Year-(2018)[15] the system then verbally announces the objects recognized in the captured direction. Unlike ultrasonic-based alternatives, this approach minimizes infrastructure expenses by leveraging image processing techniques.
- Traditional Methods and OCR Software in the paper by J. Hansen (2014)[16] shows that in the realm of assisting individuals with visual impairments, traditional tools such as guide dogs and walking canes have long been relied upon for navigation. However, the advent of modern technology has ushered in a new era of accessibility. With the emergence of advanced Optical Character Recognition (OCR) software and text-to-speech engines, individuals with visual impairments now have access to a plethora of innovative assistive devices. These technologies, which convert written text into audible speech, have become the fundamental building blocks for a range of cutting-edge devices designed to enhance independence and mobility for the visually impaired.

- In the work of W. Bieniecki, S. Grabowski and W. Rozenberg(1995)[17], OCR software plays a crucial role in enabling individuals with visual impairments to access written information. By using advanced algorithms to analyze text and convert it into a digital format, OCR software makes it possible for visually impaired individuals to access printed materials such as books, documents, and even signage. Paired with text-to-speech engines, which convert the digitized text into spoken words, these technologies have opened up new avenues for accessibility. In the journal by E. Vanitha, P. K. Kasarla and E. Kumaraswamy,July (2015)[18] and W. Bieniecki, S. Grabowski and W. Rozenberg, May 2007 With the help of these tools, visually impaired individuals can now have important information read aloud to them, allowing them to navigate their environment more effectively.
- Artificial Vision for Blind Person in Gagandeep Singh, Omkar Kandale, Kevin Takhtani and Nandini Dadhwal, (2020)[19] shows that the ultrasonic waist belt is an innovative solution designed to enhance the mobility and independence of visually impaired and blind individuals. Equipped with a network of ultrasonic sensors, the waist belt effectively detects obstacles in multiple directions including front, bottom, left, and right. By leveraging this sensor network, the device provides real-time feedback to the user about the presence and proximity of obstacles in their surroundings. To relay this information, the waist belt seamlessly connects to a mobile application that converts the sensor data into voice instructions. Through this intuitive interface, the user receives immediate alerts and guidance, enabling them to navigate their environment with greater confidence and safety. This ultrasonic waist belt represents a significant advancement in assistive technology for the visually impaired and blind community. By employing a comprehensive sensor network, it offers a holistic view of the surrounding environment, empowering users to anticipate and avoid potential obstacles. The integration of the mobile application further enhances the device's functionality by transforming the sensor data into easily understandable voice instructions. This combination of hardware and software creates a seamless and intuitive user experience, providing vital navigational assistance to those who rely on it. With the ultrasonic waist belt, visually impaired individuals can gain greater

freedom and independence in their daily lives, navigating through their surroundings with increased ease and confidence.

- Smartphone-based Room Type Recognition in N. Moore(2000)[20] features a study that has proposed a groundbreaking solution to empower individuals with visual impairments by harnessing the power of acoustic fingerprinting on smartphones. This innovative method leverages image processing algorithms to enable accurate identification of room types and the detection of occupants. By utilizing this technology, visually impaired individuals gain valuable situational awareness as they can receive real-time information about the number of people present, their age groups, and even their gender. This breakthrough holds immense potential to enhance the independence and navigation capabilities of visually impaired individuals in diverse environments. The study's approach integrates acoustic fingerprinting with image processing algorithms, forming a powerful tool for visually impaired individuals. By capturing and analyzing acoustic signals using smartphones, the system can differentiate between various room types and accurately identify the presence of occupants. The image processing algorithms complement this by effectively counting the number of people in the room and providing additional insights such as their age groups and gender. This information empowers visually impaired individuals by equipping them with a heightened sense of their surroundings, enabling them to navigate and interact with different environments more confidently. The study's findings pave the way for potential advancements in assistive technologies that promote inclusivity and independence for individuals with visual impairments. The Smart Assistive System for Blind People using Raspberry PI in Maria Anu V.; L.Mary Gladence; Ramnadh Velaga; Bharadwaj Valluri(2020)[21] gives one approach to developing software for character recognition and line-by-line reading is by leveraging the power of OpenCV libraries, which are renowned for their capabilities in computer vision tasks. These libraries offer functionalities for capturing images of text and performing optical character recognition (OCR). By utilizing OpenCV, developers can build assistive technology tools specifically tailored for individuals with visual impairment and limited vision. These tools typically consist of OCR software, which extracts text from images, and text-to-speech engines, which convert the extracted text into spoken words, enabling users to access and comprehend written content effectively. OCR software plays a crucial role in the accessibility of digital content for individuals with visual impairments. By employing OpenCV's image capturing capabilities and OCR algorithms, the software can accurately extract characters from images containing text. This extracted text is then processed by text-to-speech engines, which convert it into audible output, facilitating an inclusive reading experience. By combining the power of OpenCV, OCR, and text-to-speech technologies, developers can create software solutions that empower individuals with visual impairments to access written information.

- Image Recognition based SMART AID FOR VISUALLY CHALLENGED PEOPLE by D R Krishna Raju,, M S Vinay Prasad,, Parth Rohit,S J Ranganatha Gowda(2019)[22]shows a new and innovative solution for aiding visually challenged individuals is being proposed - the development of a smart walking stick equipped with image recognition capabilities. Unlike traditional navigation aids like sticks, this smart stick would have the ability to capture images and analyze them to identify obstacles in the user's path. The user would initiate the image capture by using a joystick switch to point the camera in the desired direction. The captured image would then be processed by the system, which would verbally communicate the presence of objects detected in that particular direction. By utilizing image-based operations, the infrastructure required for this smart stick is significantly reduced, making it a more cost-effective alternative to ultrasonic-based solutions.
- Object Detection and Navigation System in G. Kleege(2014)[23] aims to address the challenges faced by visually impaired individuals by developing an advanced object detection and navigation system. The system incorporates a Raspberry Pi setup, which effectively detects static objects in the environment. Additionally, a comprehensive database is utilized for shape recognition, allowing the system to provide accurate information about the objects detected. The textual information is then converted into audio feedback, enabling visually impaired users to navigate their surroundings with greater ease and confidence. By leveraging the capabilities of the Raspberry Pi and a robust shape recognition database, this project presents an innovative solution for visually impaired

individuals. The system's object detection functionality ensures that static objects are identified, enhancing safety and enabling users to navigate more effectively. Furthermore, the conversion of textual information into audio feedback empowers visually impaired individuals to access valuable information about their environment, contributing to their independence and improving their overall quality of life. This project holds great potential to positively impact the daily lives of visually impaired individuals by providing them with a reliable and intuitive navigation tool.

Conclusion:

The literature review highlights various approaches and technologies employed in visual assist projects for visually challenged individuals. These projects utilize camera-operated mechanisms, wearable devices, object detection algorithms, and image recognition techniques to enhance navigation, object recognition, and situational awareness.

CHAPTER 3

PROBLEM IDENTIFICATION

3.1 Navigating Around Places

Blindness is a daunting condition. Blind people learn how to interact with others and how to do things, regardless of their visual impairment. It is estimated that around 2% to 8% of blind individuals use their cane to navigate. Others rely on their guide dog, their partial sight or their sighted guide. Apart from navigation, blind individuals can do pretty much everything a sighted person can; they can cook, put on makeup and, simply, be independent. With the help of accessible technology or products, and their own willpower, blind people can be independent.

3.2 Difficulty in Finding Reading Material

According to a report by the World Health Organization in 2013, there are an estimated 40 to 45 million people who are blind and 135 million have low vision. Blind people face troubles due to inaccessible infrastructure and social challenges. Blind people have a tough time finding good reading materials in accessible formats. People read Braille by moving their fingertips from left to right across the lines of dots. Braille is a system of raised dots that can be read with the fingers by people who are blind or who have low vision. Teachers, parents, and others who are not visually impaired ordinarily read braille with their eyes. Braille is not a language. Also finding braille for each and every script/text is very difficult. This limits visually disabled people to read and learn from textual data.

3.3 Visual Assistive Devices Not Readily Available

a) The market of assistive devices is very scattered and not in easy access.

b) Most assistive devices being of from foreign brands are not pocket friendly.

c) Most assistive devices have slow processing and take time to produce the result; causing great inconvenience to the users.

d) Most assistive device are limited to provide wide-range of features to the user; the user has to use multiple aids and tools.

CHAPTER 4
METHODOLOGY (OR) EXPERIMENTAL SETUP

4.1 Project Idea

We propose a visual assistance device for visually challenged people. The objective of the proposed visual assistance eyewear device for visually challenged people is to provide a cost-effective solution that converts the visual world into an audio world, allowing individuals with visual disabilities to navigate and engage in daily activities independently. The device aims to offer essential features that can be utilized without the need for an internet connection, eliminating the reliance on multiple devices and enabling users to perform tasks free-hand. Additionally, the device prioritizes fast processing to minimize time delays and prevent potential negative consequences for the users.

The ESP-32 camera module will capture images of the target(s) (like surrounding objects, people, text document, currency, road, traffic signal, traffic sign board, etc). This captured image in real time will be processed by multiple algorithms. Then information extracted will be converted into audio based signals which will act as feedback for the user.

4.1.1 Object detection

The blind person's life becomes easier, and they can go anywhere they want without anyone's help. They can walk alone through the streets, they do not need anyone to assist them, they can handle themselves correctly. To prevent users from dangerous locations, our aim is to collect from the environment (cameras, sensors, scanners, etc) and transmit them to the users in the audio format. When sensors identify objects it sends the data to the audio module and gets converted into audio clips. An obstacle as close as 4 cm can be detected by these modules. The objective of these projects is to provide a cost effective way to allow path planning for blind people. With the help of an object detection system and navigation system the blind person can easily detect the object through cameras or through some sensors.

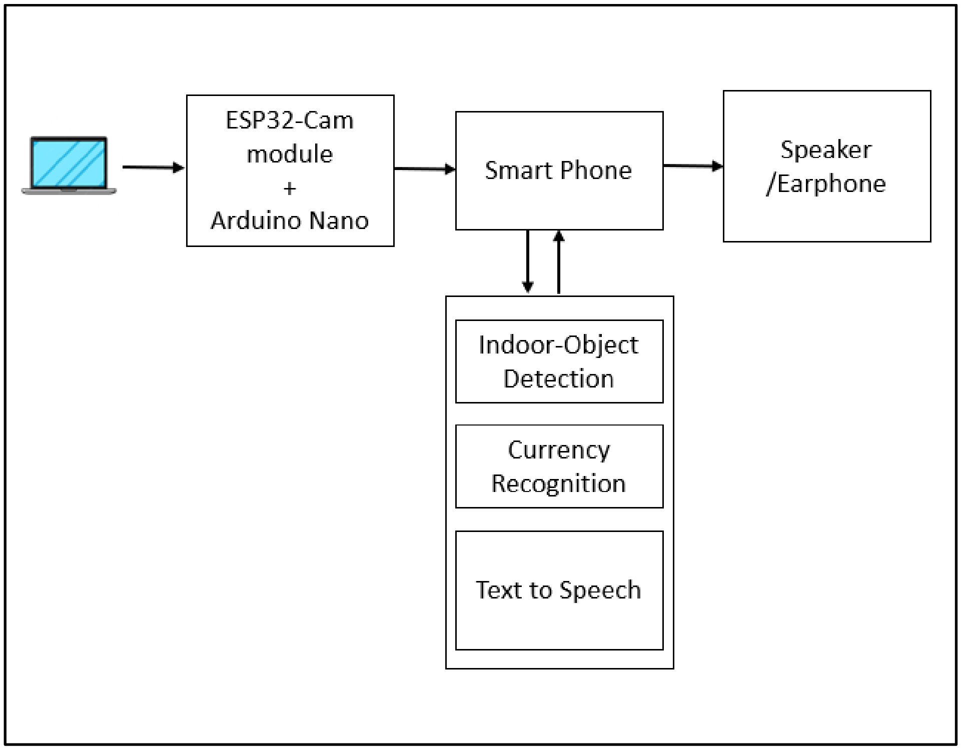


Fig 4.1 Block Diagram of Proposed Solution

4.1.2 Currency recognition

Problems faced by the visually impaired in performing daily activities are in great numbers. They also face a lot of difficulties in monetary transactions. They are unable to recognize the paper currencies due to similarity of paper texture and size between different categories. This project helps visually impaired patients to recognize and detect money.

4.1.3 Text to speech conversion

Text-to-speech (TTS) is a type of assistive technology that reads digital text aloud. It's sometimes called "read aloud" technology. With a click of a button or the touch of a finger, TTS can take words on a computer or other digital device and convert them into audio. TTS is very helpful for kids and adults who struggle with reading. But it can also help with writing and editing, and even with focusing. TTS works with nearly every personal digital device, including computers, smartphones, and tablets. All kinds of text files can be read aloud, including Word and Pages documents. Even online web pages can be read aloud. The voice in TTS is computer-generated, and reading speed can usually be sped up or slowed down. Many TTS tools highlight words as they are read aloud. This allows the user to see text and hear it at the same time. Some TTS tools can also read text aloud from images. For example, a user could take a photo of a street sign on their phone and have the words on the sign turned into audio.

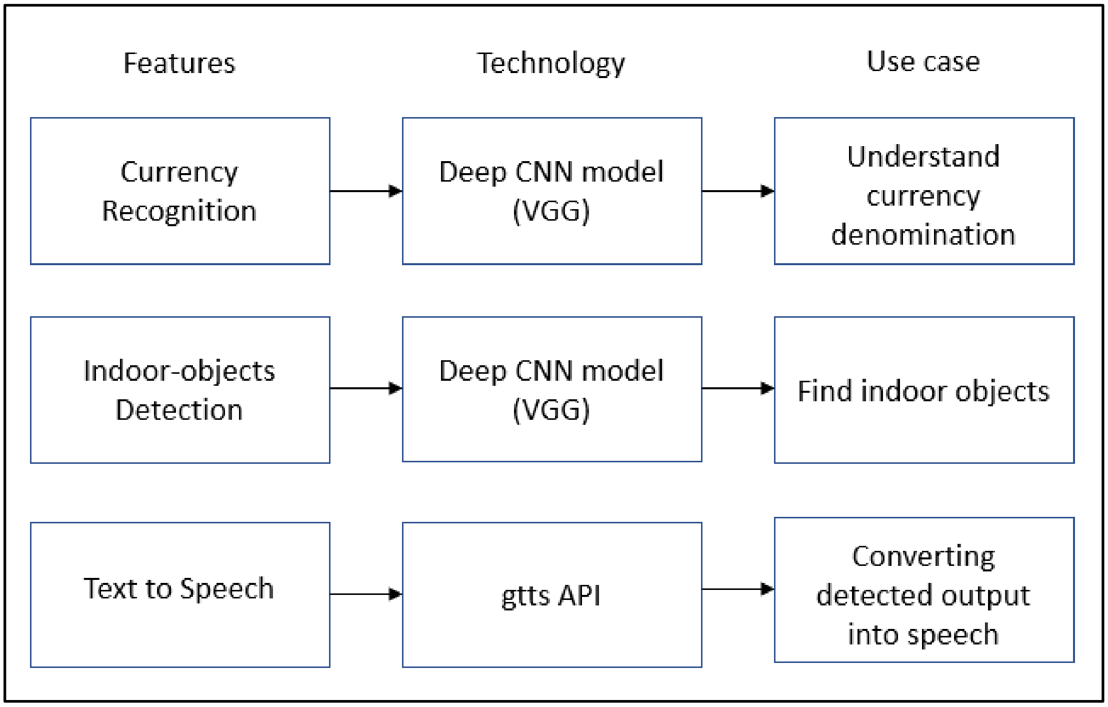


Fig 4.2 Detailed Working of Proposed Solution

4.1.4 Overall Working

This project is a microcontroller based virtual assistance device for the visually challenged people. The ESP32 camera module will capture images of the target(s) like surrounding objects,people,text documents, currency, road, traffic signal, etc. This captured image will be sent to a smartphone device and will be processed in real time by multiple algorithms. Then information extracted will be converted into audio based signals which will act as a feedback for the user. The code can be referred from Appendix I.

4.2 Hardware Requirements

4.2.1 ESP32-CAM Module

ESP32-CAM in [24] is a low-cost ESP32-based development board with an onboard camera, small in size. It is an ideal solution for IoT application, prototypes, constructions and DIY projects. The board integrates WiFi, traditional Bluetooth and low power BLE , with 2 high- performance 32-bit LX6 CPUs.

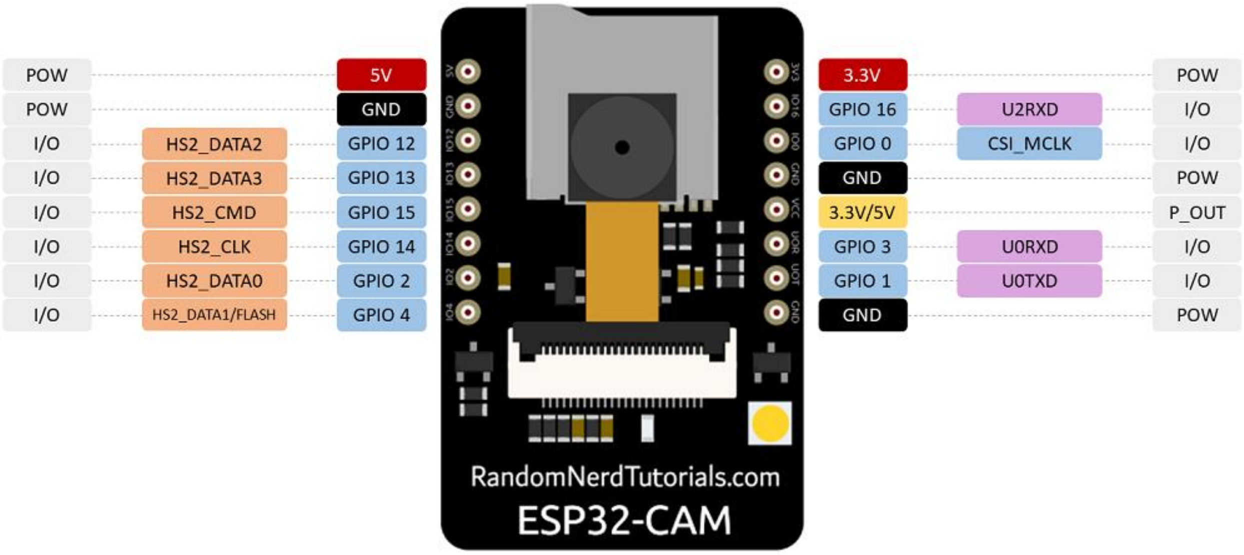


Fig 4.3 Pin Diagram of ESP-32 CAM Module

4.2.2 Arduino nano

The ESP32-CAM AI-Thinker module is an ESP32 development board with an OV2640 camera, microSD card support, on-board flash lamp and several GPIOs to connect peripherals. However, it doesn't have a built-in programmer. So it needs an external programmer to connect it to the computer and upload code. The model uses an Arduino Nano[n] for this. It will command the camera sensor to capture images and send it to the desired location for processing.

4.2.3 Smartphone

The sent images are received by any smartphone device and further algorithms for object detection, text recognition and currency identification are implemented on it. For these purposes, implementation of machine learning and artificial intelligence is needed.

4.2.4 Speakers/Earphones

After the processing of the image, the output is generated in the form of voiced instructions which can be heard via smartphone speaker or the person can use earphones like in [25] if preferred. This whole process is done in real time and the voiced instructions will help a blind person in navigation as well as text reading.

Fig 4.4 Wired Earphones

4.3 Software Requirements

4.3.1 Deep learning (Resnet-50)

Deep Learning is used for detection and identification of currency. Resnet-50 is a convolutional neural network that is 50 layers deep. ResNet stands for Residual Network. It solves complex problems with improved accuracy and performance. Basically, deep learning is a way to enhance human gain like used for automated driving, stop signs ,and traffic lights.[26]

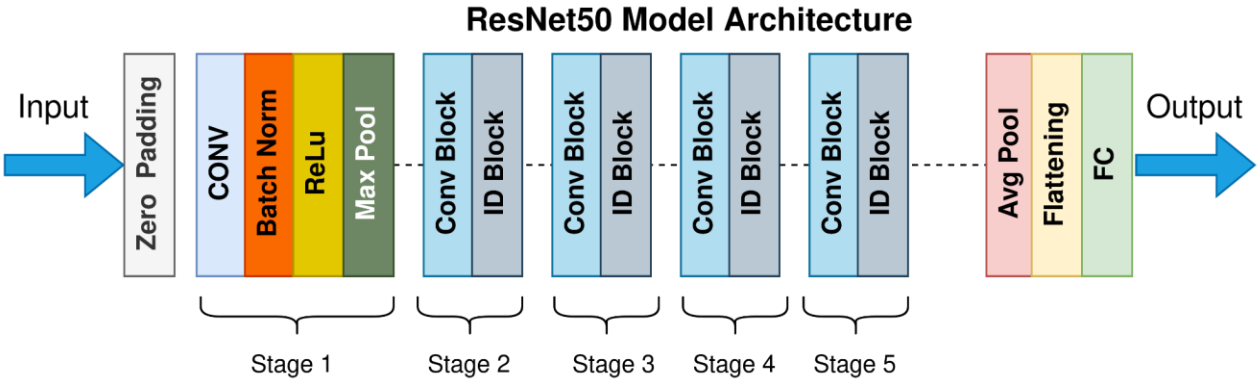


Fig 4.5 Architecture of ResNet

4.3.2 gTTS

gTTS (google Text-to-Speech) is used for speech translation. A Python library and CLI tool to interface with Google Translate's text-to-speech API. The text variable is a string used to store the user's input. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more.[27]

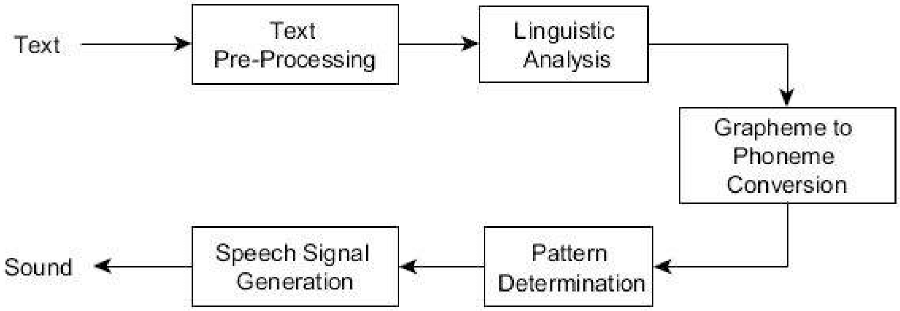


Fig 4.6 Flow Diagram of gTTS

4.4 Survey Analysis (after first presentation)

Fig 4.7 Survey of a Trust for Blind People

SENIOR BLIND HOSTAL
2/72 TELIWARA, GURUDWARA ROAD,
YOUR DONATION IS THE LIFE OF BLIND
A/C NO - 3585511235
ANY SMALL HAPPYNESS & COULD BE SHARE WITH US
IE LIFE OF BLIND
C CODE - CBIN0283323

Fig 4.8 Pictures From Survey

- **Do you have any other impairment other than visual like hearing or speaking?**

  The blind people involved in the survey were not having any other impairment.

- **What are the current apps or technologies that you are using for your day to day tasks?**

  There are apps for currency detection, color recognition and document reading .The note is placed in front of the camera and it detects the denomination.There is also a read aloud feature in smartphones named **Talkback[28]** that helps the blind in using the smartphones.The current device that they are using is a cane having ultrasonic sensor that detects the obstacles but the limitation is that it does not gives the details whether it is a dog or heap of stones. It is based on touch and it is a bit slow.

- **What is the cost of the assistive device and do you go for a brand while purchasing it?**

  The cost of the can is around 4000 and at times they do nut run after branded items

- **Will you be willing to wear a device that will help you with visual handicappedness? Are you comfortable wearing a spectacle of x gms?**

The blind men agreed to wear the spectacles up to 100 grams.

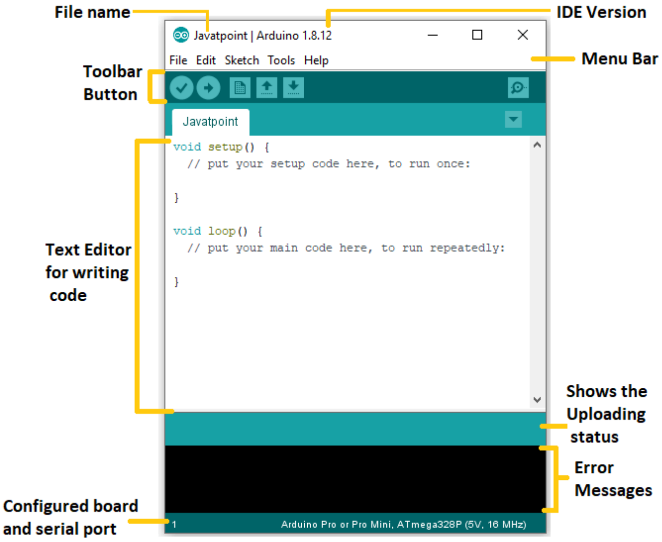
- **Do you have a smartphone and how do you use it?Do you have internet connectivity all the time?**

  Mostly everyone has a smartphone and a good internet connection. According to them if a device helps 60% percent of the visually challenged people it is considered as useful and successful.

4.5 Feasibility Of The Project (after first presentation)

4.5.1 Coding using Arduino IDE

Arduino programs [29] are written in the Arduino Integrated Development Environment (IDE). Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution. The code can be referred from Appendix I.


File name
IDE Version
Javatpoint | Arduino 1.8.12
File Edit Sketch Tools Help
Menu Bar
Toolbar Button
Javatpoint
Text Editor for writing code
Shows the Uploading status
Error Messages
Configured board and serial port

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
1          Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz)
```

Fig 4.9 Using Arduino IDE

4.5.2 Connecting the Arduino nano module

There are two jumper cables connected[30], one is for ESP32CAM connected between GND and IO0, and another is for Arduino UNO GND and RESET pins. ESP 32 CAM module is powered from the Arduino UNO 5v and GND pins connected to the ESPCAM. The data transfer pins UOT and UOR from the ESP32CAM are connected to TX and RX pins of Arduino respectively.

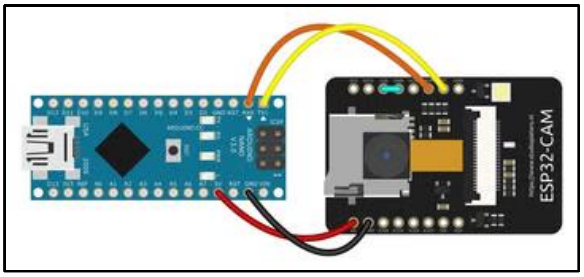
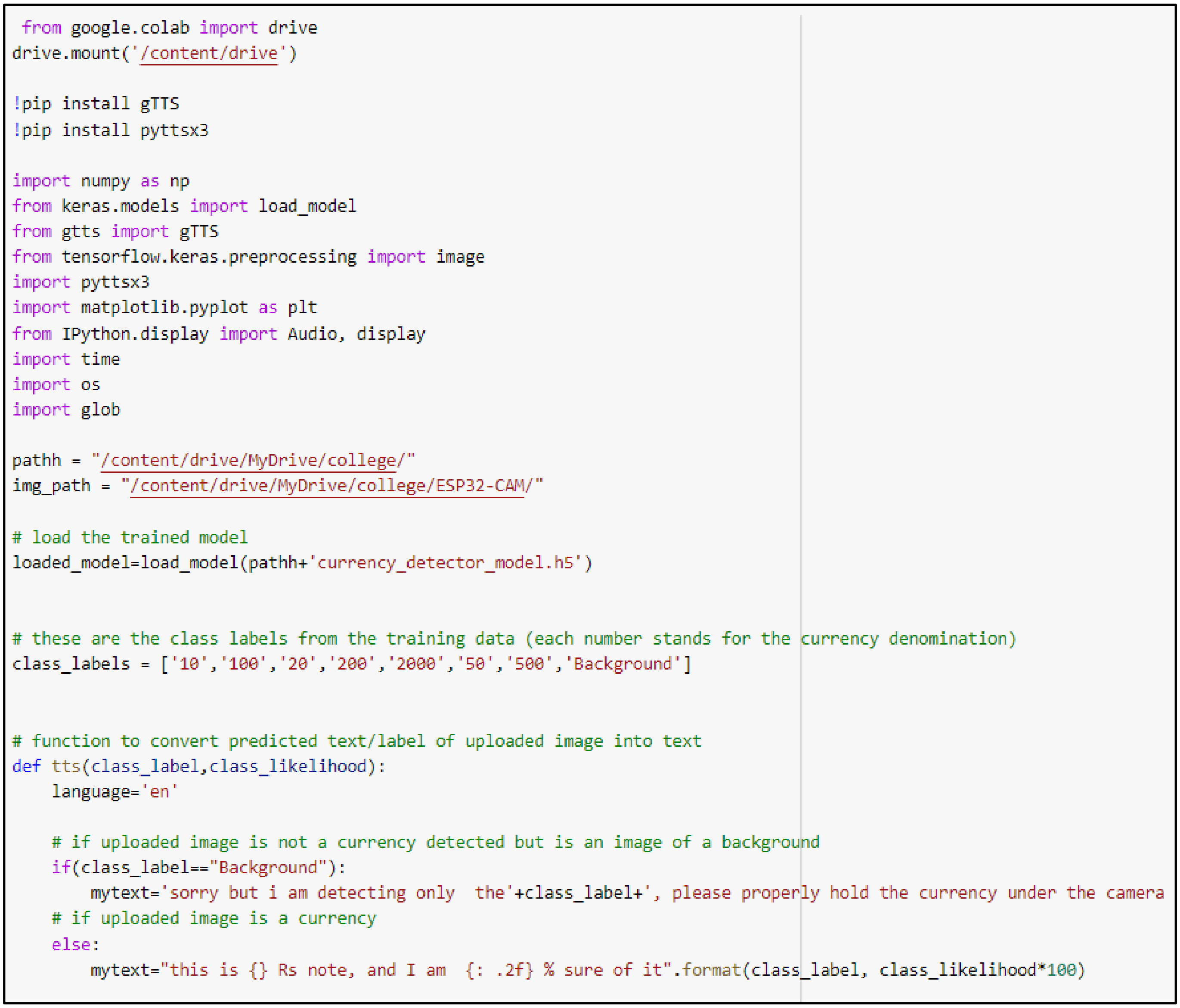


Fig 4.10 Connecting Arduino Nano With ESP32-CAM Module

4.5.3 ML based software for currency identification

The complete code can be referred from Appendix II.



```
 from google.colab import drive
drive.mount('/content/drive')

!pip install gTTS
!pip install pyttsx3

import numpy as np
from keras.models import load_model
from gtts import gTTS
from tensorflow.keras.preprocessing import image
import pyttsx3
import matplotlib.pyplot as plt
from IPython.display import Audio, display
import time
import os
import glob

pathh = "/content/drive/MyDrive/college/"
img_path = "/content/drive/MyDrive/college/ESP32-CAM/"

# load the trained model
loaded_model=load_model(pathh+'currency_detector_model.h5')


# these are the class labels from the training data (each number stands for the currency denomination)
class_labels = ['10','100','20','200','2000','50','500','Background']


# function to convert predicted text/label of uploaded image into text
def tts(class_label,class_likelihood):
    language='en'

    # if uploaded image is not a currency detected but is an image of a background
    if(class_label=="Background"):
        mytext='sorry but i am detecting only  the'+class_label+', please properly hold the currency under the camera
    # if uploaded image is a currency
    else:
        mytext="this is {} Rs note, and I am  {: .2f} % sure of it".format(class_label, class_likelihood*100)
```

Fig 4.11 Machine Learning Software

4.6 Project Cost Estimation (after first presentation)

The total estimated cost of the project is estimated to be around **Rs. 1,500** only.

Component	**Cost**
ESP32-Cam module (having OV2640 camera)	Rs. 500
Arduino Nano	Rs. 300
Batteries	Rs. 200
Miscellaneous (foam, tape, superglue, cardboard)	Rs. 200
Spectacles	Rs. 150
Jumper wires	Rs. 150

Table 4.1 Project Components Cost

4.7 Integration of Software Algorithm to Hardware (after second presentation)

Based on feedback from the jury, we integrated the software algorithm[31] of currency detection with the designed hardware structure to understand the feasibility of the project. The code can be referred from Appendix I.

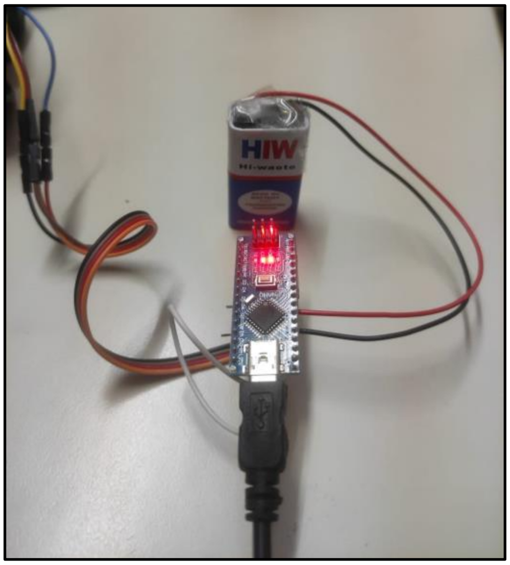


Fig 4.12 Image of Circuit Connection

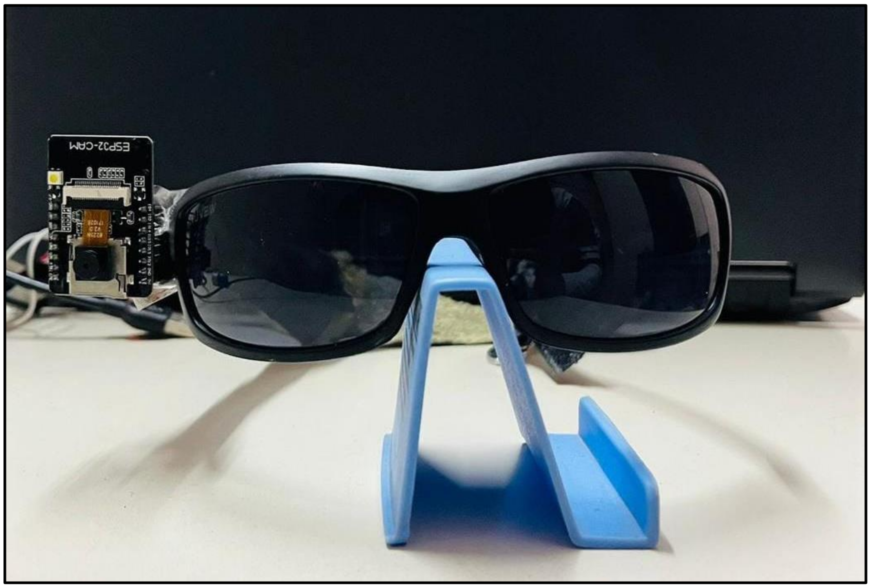


Fig 4.13 Picture of Prototype

4.8 Research Paper Presented in International Conference at AKGEC-2023 (after second presentation)

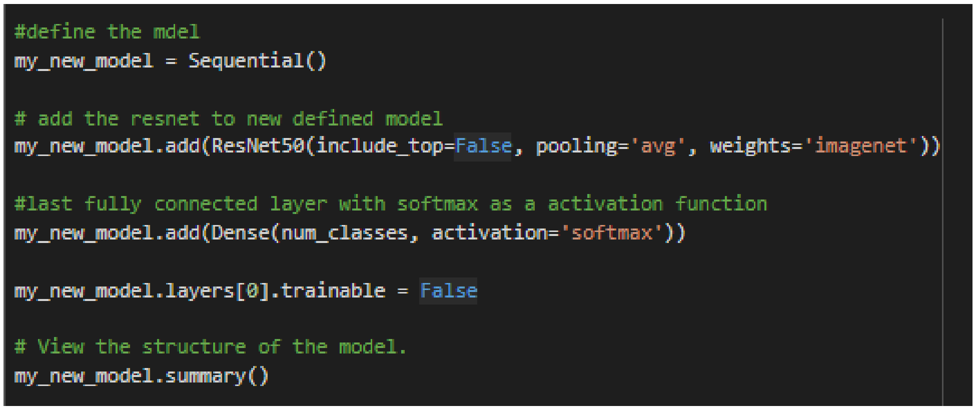
- Both the jury members were happy with the novelty and objective of this project.
- They also appraised us for a detailed literature review and understanding regarding the limitations of existing products/solutions in the market.
- The only feedback which was suggested by the jury members regarding further improvement was at software level to use segmentation algorithm in the place of current detection algorithm to predict denomination of the currency.
- We have implemented changes as suggested by them in the recent working prototype of our project.

4.9 Training the ML Model For Object Detection (after third presentation)

The model was trained using images captured in real time to test the prototype on more diverse objects like pen, book, laptop in addition to currency. This was done to show the prototype's capability to be scaled in future to more objects. The code can be referred from Appendix II.

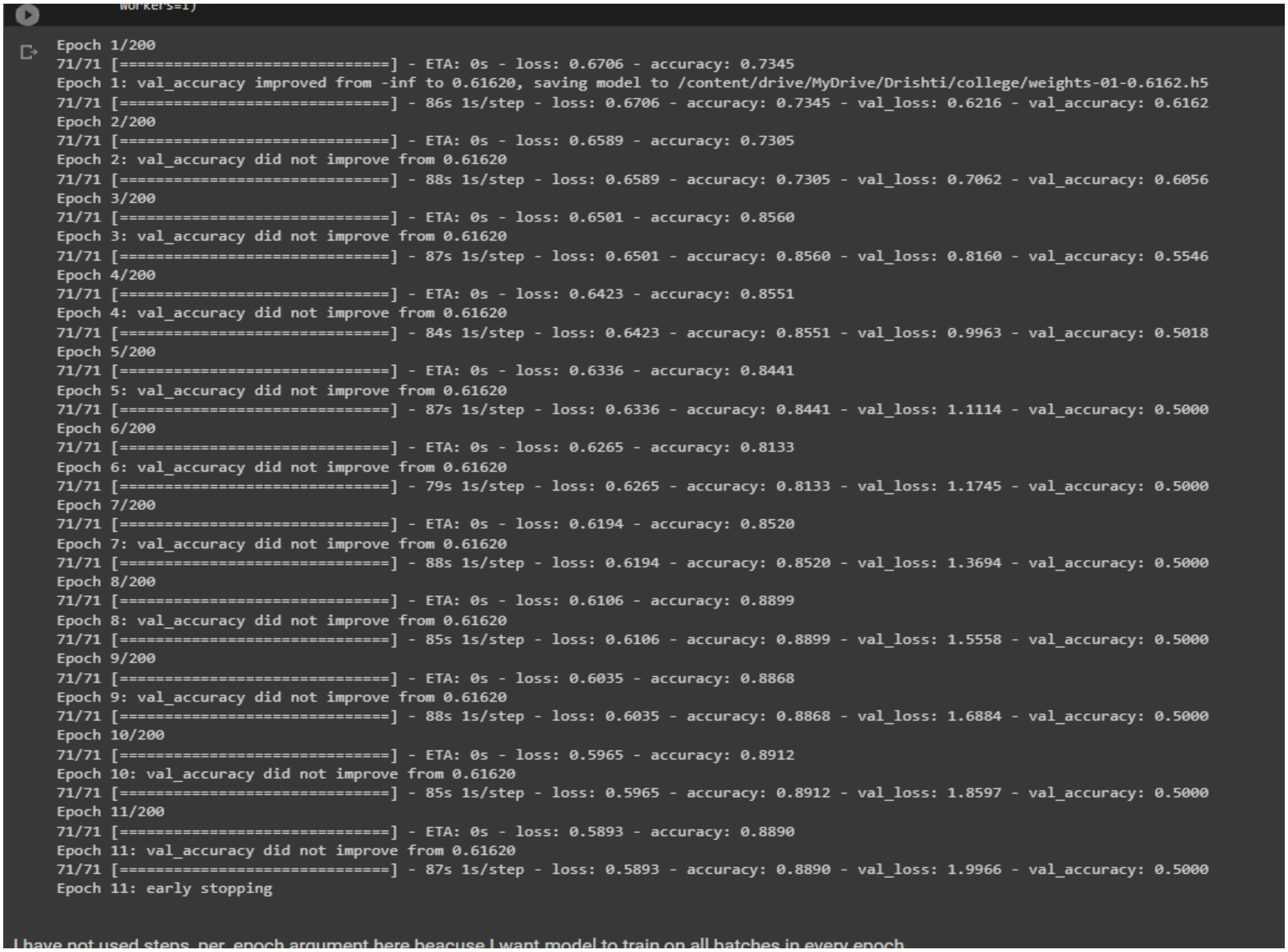
When training was done using first model architecture, the model was getting overfit i.e. the training accuracy was getting way more higher than the test accuracy. This shows that the model was just over learning features of the training dataset and was not able to generalize well to the unseen images in the test dataset.

Therefore, architecture was changed and more deeper fully connected layers were added in the model to better understand the image representations as well as not to get overfit on training dataset and generalize well to the unseen test images as well. This showed a positive result as now both training and test accuracies were increasing in the same manner.



```
#define the mdel
my_new_model = Sequential()

# add the resnet to new defined model
my_new_model.add(ResNet50(include_top=False, pooling='avg', weights='imagenet'))

#last fully connected layer with softmax as a activation function
my_new_model.add(Dense(num_classes, activation='softmax'))

my_new_model.layers[0].trainable = False

# View the structure of the model.
my_new_model.summary()
```

Fig 4.15 First Algorithm for Object Detection

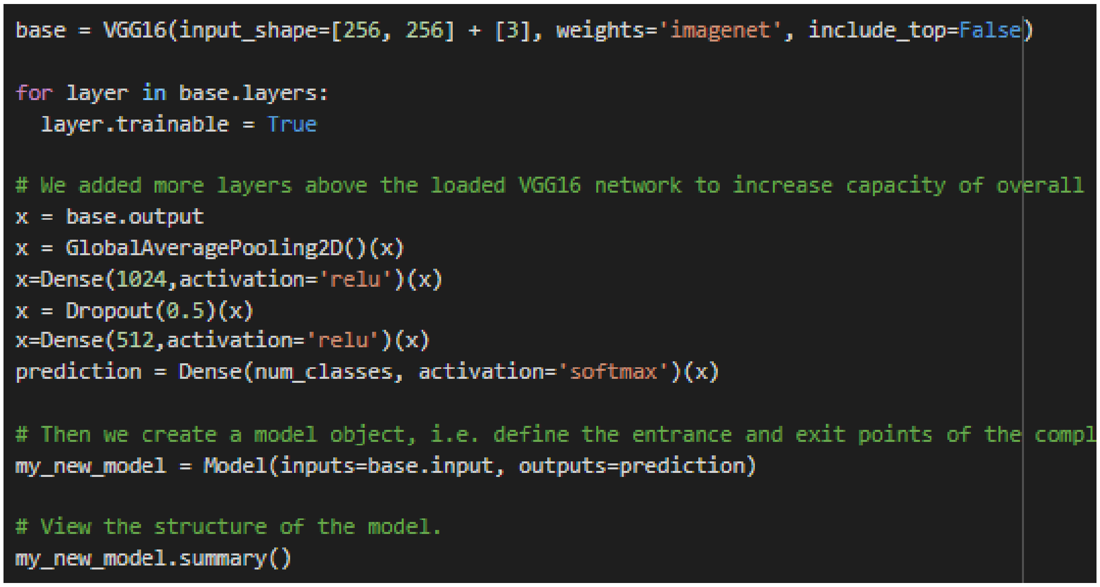


```
        workers=1)
```

```
Epoch 1/200
71/71 [==============================] - ETA: 0s - loss: 0.6706 - accuracy: 0.7345
Epoch 1: val_accuracy improved from -inf to 0.61620, saving model to /content/drive/MyDrive/Drishti/college/weights-01-0.6162.h5
71/71 [==============================] - 86s 1s/step - loss: 0.6706 - accuracy: 0.7345 - val_loss: 0.6216 - val_accuracy: 0.6162
Epoch 2/200
71/71 [==============================] - ETA: 0s - loss: 0.6589 - accuracy: 0.7305
Epoch 2: val_accuracy did not improve from 0.61620
71/71 [==============================] - 88s 1s/step - loss: 0.6589 - accuracy: 0.7305 - val_loss: 0.7062 - val_accuracy: 0.6056
Epoch 3/200
71/71 [==============================] - ETA: 0s - loss: 0.6501 - accuracy: 0.8560
Epoch 3: val_accuracy did not improve from 0.61620
71/71 [==============================] - 87s 1s/step - loss: 0.6501 - accuracy: 0.8560 - val_loss: 0.8160 - val_accuracy: 0.5546
Epoch 4/200
71/71 [==============================] - ETA: 0s - loss: 0.6423 - accuracy: 0.8551
Epoch 4: val_accuracy did not improve from 0.61620
71/71 [==============================] - 84s 1s/step - loss: 0.6423 - accuracy: 0.8551 - val_loss: 0.9963 - val_accuracy: 0.5018
Epoch 5/200
71/71 [==============================] - ETA: 0s - loss: 0.6336 - accuracy: 0.8441
Epoch 5: val_accuracy did not improve from 0.61620
71/71 [==============================] - 87s 1s/step - loss: 0.6336 - accuracy: 0.8441 - val_loss: 1.1114 - val_accuracy: 0.5000
Epoch 6/200
71/71 [==============================] - ETA: 0s - loss: 0.6265 - accuracy: 0.8133
Epoch 6: val_accuracy did not improve from 0.61620
71/71 [==============================] - 79s 1s/step - loss: 0.6265 - accuracy: 0.8133 - val_loss: 1.1745 - val_accuracy: 0.5000
Epoch 7/200
71/71 [==============================] - ETA: 0s - loss: 0.6194 - accuracy: 0.8520
Epoch 7: val_accuracy did not improve from 0.61620
71/71 [==============================] - 88s 1s/step - loss: 0.6194 - accuracy: 0.8520 - val_loss: 1.3694 - val_accuracy: 0.5000
Epoch 8/200
71/71 [==============================] - ETA: 0s - loss: 0.6106 - accuracy: 0.8899
Epoch 8: val_accuracy did not improve from 0.61620
71/71 [==============================] - 85s 1s/step - loss: 0.6106 - accuracy: 0.8899 - val_loss: 1.5558 - val_accuracy: 0.5000
Epoch 9/200
71/71 [==============================] - ETA: 0s - loss: 0.6035 - accuracy: 0.8868
Epoch 9: val_accuracy did not improve from 0.61620
71/71 [==============================] - 88s 1s/step - loss: 0.6035 - accuracy: 0.8868 - val_loss: 1.6884 - val_accuracy: 0.5000
Epoch 10/200
71/71 [==============================] - ETA: 0s - loss: 0.5965 - accuracy: 0.8912
Epoch 10: val_accuracy did not improve from 0.61620
71/71 [==============================] - 85s 1s/step - loss: 0.5965 - accuracy: 0.8912 - val_loss: 1.8597 - val_accuracy: 0.5000
Epoch 11/200
71/71 [==============================] - ETA: 0s - loss: 0.5893 - accuracy: 0.8890
Epoch 11: val_accuracy did not improve from 0.61620
71/71 [==============================] - 87s 1s/step - loss: 0.5893 - accuracy: 0.8890 - val_loss: 1.9966 - val_accuracy: 0.5000
Epoch 11: early stopping
```

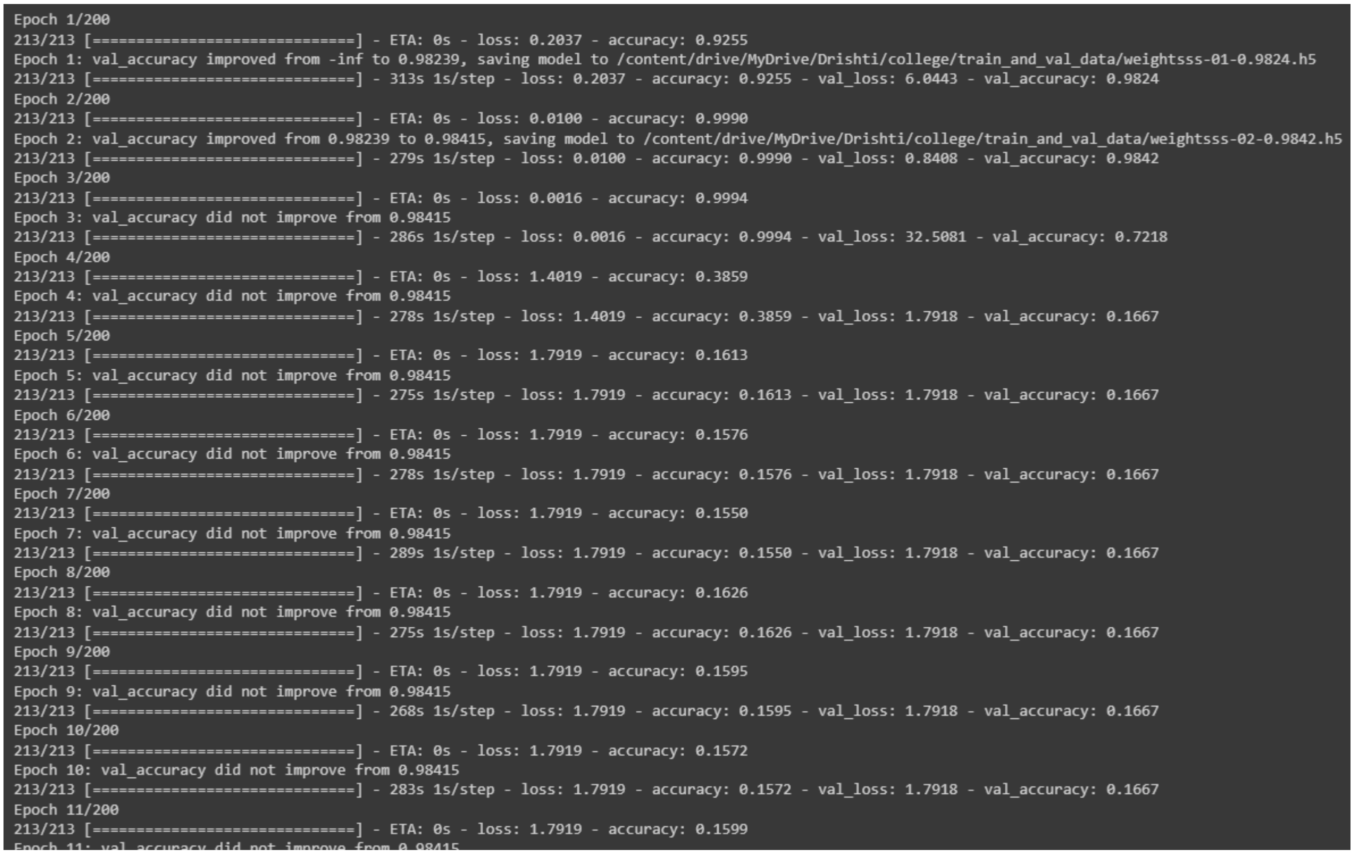
I have not used steps_per_epoch argument here beacuse I want model to train on all batches in every epoch

Fig 4.16 First Model Training (Case of Overfitting)



```
base = VGG16(input_shape=[256, 256] + [3], weights='imagenet', include_top=False)

for layer in base.layers:
  layer.trainable = True

# We added more layers above the loaded VGG16 network to increase capacity of overall
x = base.output
x = GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x)
x = Dropout(0.5)(x)
x=Dense(512,activation='relu')(x)
prediction = Dense(num_classes, activation='softmax')(x)

# Then we create a model object, i.e. define the entrance and exit points of the compl
my_new_model = Model(inputs=base.input, outputs=prediction)

# View the structure of the model.
my_new_model.summary()
```

Fig 4.17 Second Algorithm for Object Detection



```
Epoch 1/200
213/213 [==============================] - ETA: 0s - loss: 0.2037 - accuracy: 0.9255
Epoch 1: val_accuracy improved from -inf to 0.98239, saving model to /content/drive/MyDrive/Drishti/college/train_and_val_data/weightsss-01-0.9824.h5
213/213 [==============================] - 313s 1s/step - loss: 0.2037 - accuracy: 0.9255 - val_loss: 6.0443 - val_accuracy: 0.9824
Epoch 2/200
213/213 [==============================] - ETA: 0s - loss: 0.0100 - accuracy: 0.9990
Epoch 2: val_accuracy improved from 0.98239 to 0.98415, saving model to /content/drive/MyDrive/Drishti/college/train_and_val_data/weightsss-02-0.9842.h5
213/213 [==============================] - 279s 1s/step - loss: 0.0100 - accuracy: 0.9990 - val_loss: 0.8408 - val_accuracy: 0.9842
Epoch 3/200
213/213 [==============================] - ETA: 0s - loss: 0.0016 - accuracy: 0.9994
Epoch 3: val_accuracy did not improve from 0.98415
213/213 [==============================] - 286s 1s/step - loss: 0.0016 - accuracy: 0.9994 - val_loss: 32.5081 - val_accuracy: 0.7218
Epoch 4/200
213/213 [==============================] - ETA: 0s - loss: 1.4019 - accuracy: 0.3859
Epoch 4: val_accuracy did not improve from 0.98415
213/213 [==============================] - 278s 1s/step - loss: 1.4019 - accuracy: 0.3859 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 5/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1613
Epoch 5: val_accuracy did not improve from 0.98415
213/213 [==============================] - 275s 1s/step - loss: 1.7919 - accuracy: 0.1613 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 6/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1576
Epoch 6: val_accuracy did not improve from 0.98415
213/213 [==============================] - 278s 1s/step - loss: 1.7919 - accuracy: 0.1576 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 7/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1550
Epoch 7: val_accuracy did not improve from 0.98415
213/213 [==============================] - 289s 1s/step - loss: 1.7919 - accuracy: 0.1550 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 8/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1626
Epoch 8: val_accuracy did not improve from 0.98415
213/213 [==============================] - 275s 1s/step - loss: 1.7919 - accuracy: 0.1626 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 9/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1595
Epoch 9: val_accuracy did not improve from 0.98415
213/213 [==============================] - 268s 1s/step - loss: 1.7919 - accuracy: 0.1595 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 10/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1572
Epoch 10: val_accuracy did not improve from 0.98415
213/213 [==============================] - 283s 1s/step - loss: 1.7919 - accuracy: 0.1572 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 11/200
213/213 [==============================] - ETA: 0s - loss: 1.7919 - accuracy: 0.1599
Epoch 11: val_accuracy did not improve from 0.98415
```

Fig 4.18 Second Model Training

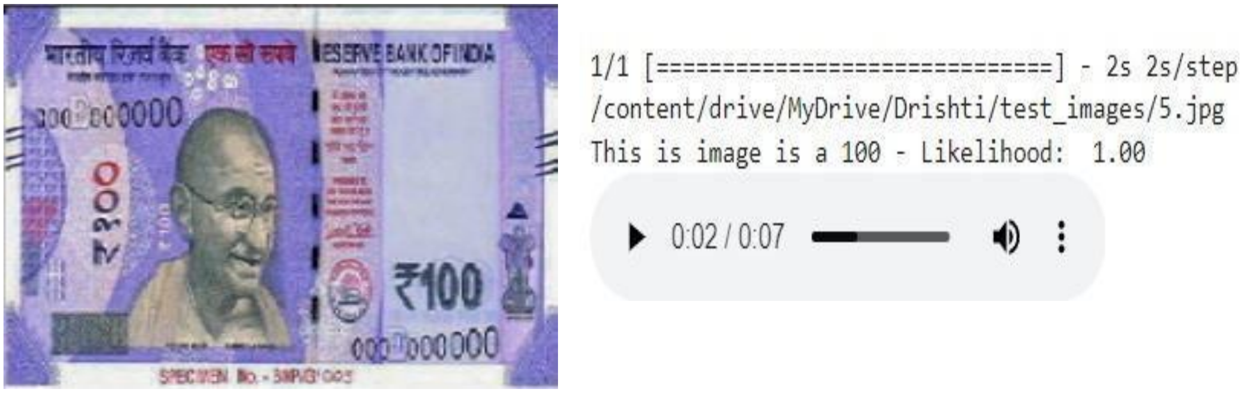
CHAPTER 5

RESULTS

5.1 Currency Detection

5.1.1 Result of Rs. 100 note

- A 100 rupees note is given as an input.
- Captures the image and analyzes it.
- Image is processed and the algorithm is applied.
- Gives output as 10 with a likelihood or probability of 0.89
- Output is given in the form of audio as well as text.



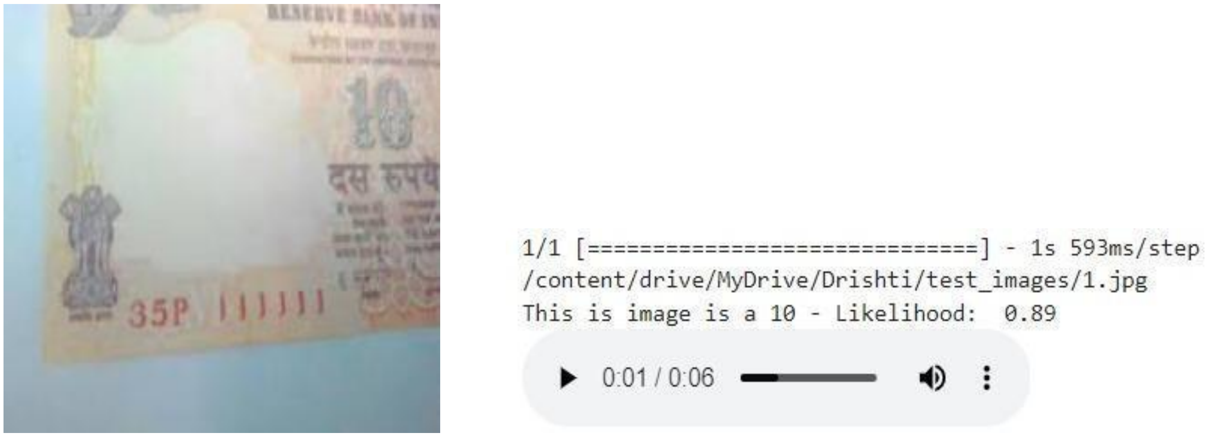
```
1/1 [==============================] - 2s 2s/step
/content/drive/MyDrive/Drishti/test_images/5.jpg
This is image is a 100 - Likelihood:  1.00
```

Fig 5.1 Input of INR 100 With Model’s Text And Audio-Based Prediction

5.1.2 Result of Rs. 10 note

- An image of a 10 rupees note is provided as input for analysis.
- The image is captured and processed using an algorithm.
- After analysis, the output is determined to be 10 rupees with a high likelihood or probability of 0.89.

- The output is presented both in audio format and as text.



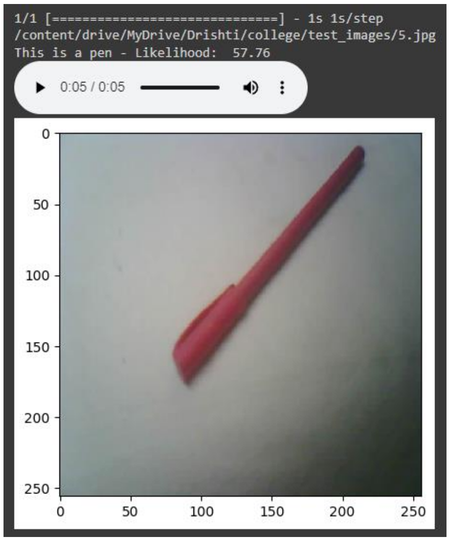
```
1/1 [==============================] - 1s 593ms/step
/content/drive/MyDrive/Drishti/test_images/1.jpg
This is image is a 10 - Likelihood:  0.89
```

Fig 5.2 Input of INR 10 With Model’s Text And Audio-Based Prediction

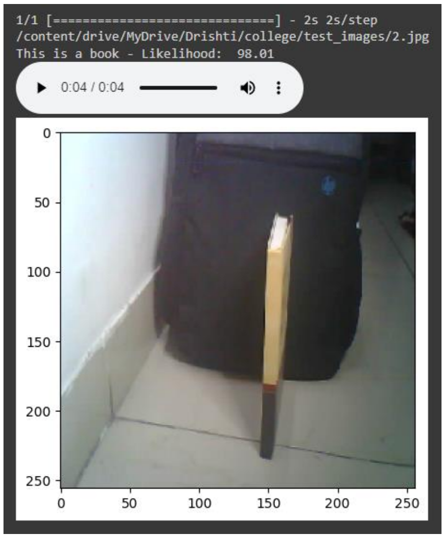
5.2 Indoor Object Detection

5.2.1 Results after first model training

- An image of a red pen is provided as input for analysis.
- The image is captured and processed using an algorithm.
- The Train accuracy is of 73.4%   and the test accuracy = 61.6%
- The output is presented both in audio format and as text.



```
1/1 [==============================] - 1s 1s/step
/content/drive/MyDrive/Drishti/college/test_images/5.jpg
This is a pen - Likelihood:  57.76
```

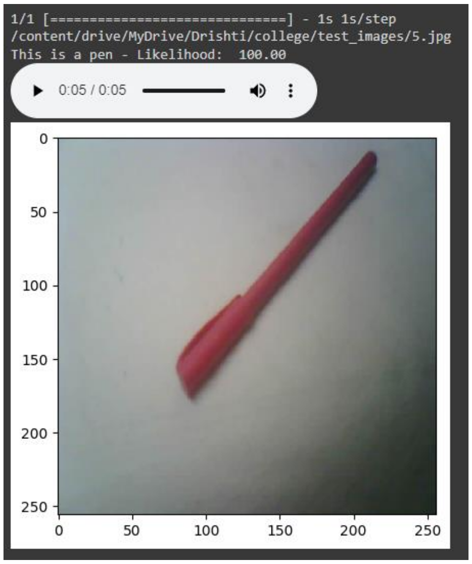


```
1/1 [==============================] - 2s 2s/step
/content/drive/MyDrive/Drishti/college/test_images/2.jpg
This is a book - Likelihood:  98.01
```

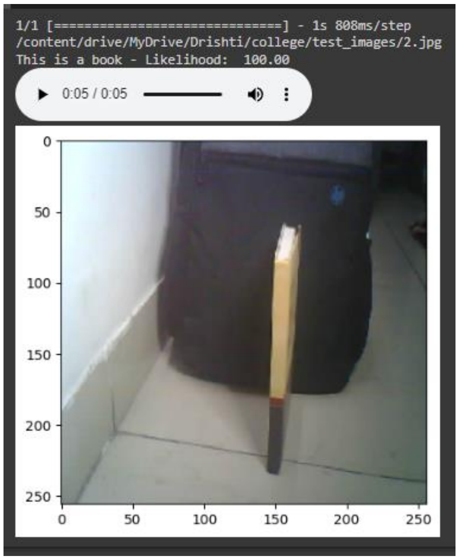
Fig 5.3 Input of a pen and a book With Model's Output

5.2.2 Results after second model training

- The image of a red pen is analyzed using an algorithm.
- The algorithm has been trained and tested, with a training accuracy of 92.5% and a test accuracy of 98.2%.
- The results of the analysis are presented both in audio format and as text.



```
1/1 [==============================] - 1s 1s/step
/content/drive/MyDrive/Drishti/college/test_images/5.jpg
This is a pen - Likelihood:  100.00
```



```
1/1 [==============================] - 1s 808ms/step
/content/drive/MyDrive/Drishti/college/test_images/2.jpg
This is a book - Likelihood:  100.00
```

Fig 5.4 Input of a Pen And a Book With Model's Output (improved)

CHAPTER 6
CONCLUSION AND FUTURE SCOPE OF WORK

Visually impaired people are those who are either totally blind or have a very low vision that is legally considered as blindness. The number of visually impaired people has increased in the recent decades and the difficulties they face in their everyday life are becoming more and more serious with the new technologies, buildings, population and so on. Blind people have a tough time finding good reading materials in accessible formats. People read Braille by moving their fingertips from left to right across the lines of dots. Braille is a system of raised dots that can be read with the fingers by people who are blind or who have low vision.

This project is intended to help this type of people to widen their scope of independence by giving them a description of the live scenes delivered in an audio format. The project is implemented using Arduino as a main software program, and the single board wifi camera computer the ESP-32 CAM as the main hardware unit. The ESP-32 camera module captures images of the target(s) and then sends it to google drive from where it is picked up in real time and processed by multiple algorithms. Then information extracted is converted into audio based signals which act as feedback for the user. The project is implemented with emphasis on cost effectiveness, which is kept as low as possible, this appears in the cost analysis given in Table 4.1, the final prototype costs Rs.1500 only.

To enhance future work, the addition of more optimization functions would greatly benefit the system.

- By expanding the range of available optimization methods, users would have a wider selection to choose from based on their specific requirements.
- This would allow for more tailored and efficient optimization processes, ultimately leading to improved overall performance and user satisfaction.
- Additionally, incorporating user testing results and feedback into the development process would provide valuable insights for further enhancements.

- By actively involving users in the testing phase, developers can gain a deeper understanding of their needs and preferences, leading to more targeted improvements and a more user-centric system.
- One valuable feature that could be implemented to enhance the system's capabilities is text recognition using TesseractOCR.
- This optical character recognition engine has proven to be effective in extracting text from images and documents that lack a text layer. By integrating TesseractOCR into the system, users would be able to easily extract and convert printed text into searchable formats such as text files or PDFs.
- This would greatly enhance the system's versatility and usability, enabling users to work with text-based content more efficiently.
- Based on the conclusions drawn from experiment results, it is evident that the quality of the camera plays a vital role in achieving accurate results.
- Therefore, a recommended improvement for the system would be to incorporate a higher-quality camera.
- By upgrading the camera component, the system would be better equipped to capture clear and detailed images, resulting in improved accuracy during text recognition and other image-based tasks.
- A better camera would ensure that the system can effectively process visual information, enhancing its overall performance and delivering more precise outcomes to the users.

REFERENCES

[1] WHO. 2022. Blindness and vision impairment.

[2] Samruddhi Deshpande and Revati Shriram. 2016. Real time text detection and recognition on hand held objects to assist blind people. International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT).

[3] B. Deepthi Jain, S. M. Thakur and K. V. Suresh. 2017. Visual Assistance for Blind Using Image Processing. International Conference on Communication and Signal Processing (ICCSP).

[4] J. Tao, H. Wang, X. Zhang, X. Li and H. Yang. 2017. An object detection system based on YOLO in traffic scenes. 6th International Conference on Computer Science and Network Technology (ICCSNT).

[5] Roberto Netoa and Nuno Fonsecaa. 2014. Utilization of computer framework advances for navigational purposes. International Journal and Magazine of Engineering, Technology, Management and Research.

[6] Marko Lazic, Sad Anat and Perišić Predrag Šiđanin. 2015. Theoretical Framework of Computer Cognition for Use in Urban Design Process. South East European Journal of Architecture and Design.

[7] M. Ebling. 2009. Virtual Senses. New Products in Pervasive Computing IEEE.

[8] R. Lima, L. Barreto, A. Amaral and S. Paiva. 2023. Visually Impaired People Positioning Assistance System Using Artificial Intelligence. IEEE Sensors Journal.

[9] Edwards and Alistair D.N. 2006. Modern Stage of SNE Development: Implementation of Inclusive Education. Icts In Education For People With Special Needs.

[10] Rakesh Chandra Joshi, Saumya Yadav, Malay Kishore Dutta, and Carlos M. Travieso-Gonzalez. 2020. Efficient Multi-Object Detection and Smart Navigation Using Artificial Intelligence for Visually Impaired People. Centre for Advanced Studies, Dr. A.P.J. Abdul Kalam Technical University.

[11] Ádám Csapó, György Wersényi, Hunor Nagy and Tony Stockman. 2015. A survey of assistive technologies and applications for blind users on mobile platforms: a review and foundation for research. Journal on Multimodal User Interfaces.

[12] A. Ganz, J. Schafer, E. Puleo, C. Wilson and M. Robertson. 2012. Quantitative and Qualitative Evaluation of PERCEPT Indoor Navigation System for Visually Impaired Users. Engineering in Medicine and Biology Society (EMBC).

[13] D. Jain. 2014. Path-Guided Indoor Navigation for the Visually Impaired. 16th international ACM SIGACCESS conference on Computers & accessibility.

[14] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network. International Conference on Engineering and Technology (ICET), 2017.

[15] Ziad O. Abu-Faraj, Paul Ibrahim, Elie Jabbour and Anthony Ghaoui. 2018. It Looks Beautiful but Scary: How Low Vision People Navigate Stairs and Other Surface Level Change. 20th International ACM SIGACCESS Conference.

[16] J. Hansen. 2012. A Matlab Project in Optical Character Recognition (OCR).

[17] W. Bieniecki, S. Grabowski and W. Rozenberg. 2007. Image Preprocessing for Improving OCR Accuracy. Perspective Technologies and Methods in MEMS Design.

[18] E. Vanitha, P. K. Kasarla and E. Kumaraswamy. 2015. Implementation of Text-To-Speech for Real Time Embedded System Using Raspberry Processor. International Journal and Magazine of Engineering,Technology, Management and research.

[19] Gagandeep Singh, Omkar Kandale, Kevin Takhtani and Nandini Dadhwal. 2020. A Smart Personal AI Assistant for Visually Impaired People. International Research Journal of Engineering and Technology (IRJET).

[19] Therese Yamuna Mahesh, Parvathy S S, Shibin Thomas, Shilpa Rachel Thomas and Thomas Sebastian. 2021. CICERONE- A Real Time Object Detection for Visually Impaired People. IOP Conference Series: Materials Science and Engineering.

[20] N. Moore. 2000. The Information Needs of Visually Impaired People. A review of research for the RNIB. ACUMEN.

[21] Maria Anu V., L.Mary Gladence, Ramnadh Velaga and Bharadwaj Valluri. 2020. Smart Assistive System for Blind People using Raspberry PI. International Conference on System, Computation, Automation and Networking (ICSCAN).

[22] D R Krishna Raju, M. S. Vinay Prasad, Parth Rohit and S. J. Ranganatha Gowda. 2019. Image Recognition based SMART AID FOR VISUALLY CHALLENGED PEOPLE. International Conference on Communication and Electronics Systems (ICCES).

[23] G. Kleege. 2014. Blind Imagination: Pictures into Words. Gale Literature Resource Center.

[24] https://i0.wp.com/randomnerdtutorials.com/wp-content/uploads/2020/03/ESP32-CAM-pinout-new.png?resize=840%2C374&ssl=1 (accessed on 2 November 2022)

[25] https://n1.sdlcdn.com/imgs/b/c/i/Sony-MH750-Earbuds-Wired-Earphones-SDL695196466-1-c8076.jpg (accessed on 25 November 2022)

[26] https://miro.medium.com/max/1200/0*tH9evuOFqk8F41FG.png (accessed on 13 October 2022)

[27] https://www.researchgate.net/profile/Suhas-Mache/publication/304601298/figure/fig3/AS:667867041247263@1536243319820/Block-diagram-of-Text-to-Speech-System-Techniques-of-speech-synthesis-5-a.png (accessed on 2 November 2022)

[28] https://f-droid.org/en/packages/com.google.android.marvin.talkback/(Talkback app)

[29] https://static.javatpoint.com/tutorial/arduino/images/arduino-ide.png (accessed 2 November 2022)

[30] https://all3dp.com/2/esp32-cam-arduino-tutorial/#:~:text=Restart%20Arduino%20IDE%20and%20connect,AI%20Thinker%20ESP32%2DCAM%E2%80%9D.Connecting the Arduino nano module to ESP32

[31] https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e RCNN model

APPENDIX I

ARDUINO CODE

- **espcam.ino file**

```
// Enter your WiFi ssid and password
const char* ssid     = "DESKTOP-FQRO1VJ 1261";   //your network SSID
const char* password = "0aC7633(";   //your network password

String myScript = "/macros/s/AKfycbzK6Z0rh-OXX8JUxJFkrfFspxGxcYGf1bO_Mgu5lSUewSECLWAMHcLqYzMRWeGVNyyl/exec";
//Create your Google Apps Script and replace the "myScript" path.
String myLineNotifyToken = "myToken=**********";    //Line Notify Token. You can set the value of xxxxxxxxxx empty if you don't want to send picture to Linenotify.
String myFoldername = "&myFoldername=ESP32-CAM";
String myFilename = "&myFilename=ESP32-CAM.jpg";
String myImage = "&myFile=";
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "Base64.h"
#include "esp_camera.h"

#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
```

```
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM       5
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

void setup()
{
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);
  delay(10);
  WiFi.mode(WIFI_STA);
  Serial.println("");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  long int StartTime=millis();
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    if ((StartTime+10000) < millis()) break;   //here is a 10000
  }
  Serial.println("");
  Serial.println("STAIP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("");
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Reset");
    ledcAttachPin(4, 3);
    ledcSetup(3, 5000, 8);
    ledcWrite(3,10);
    delay(200);
    ledcWrite(3,0);
    delay(200);
    ledcDetachPin(3);
    delay(1000);
    ESP.restart();
  }
  else {
    ledcAttachPin(4, 3);
    ledcSetup(3, 5000, 8);
    for (int i=0;i<5;i++) {
      ledcWrite(3,10);
      delay(200);
      ledcWrite(3,0);
      delay(200);
    }
    ledcDetachPin(3);
  }
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
```

```
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;  //0-63 lower number means higher quality
    config.fb_count = 2;
  }
else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    config.fb_count = 1;
  }
  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
  }
  //drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_VGA);  //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
}
void loop()
{
  SendCapturedImage();
  delay(10000);                                    //here is a 10000
}
String SendCapturedImage() {
  const char* myDomain = "script.google.com";
  String getAll="", getBody = "";
  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
    return "Camera capture failed";
  }
  Serial.println("Connect to " + String(myDomain));
  WiFiClientSecure client_tcp;
```

```
client_tcp.setInsecure();   //run version 1.0.5 or above
if (client_tcp.connect(myDomain, 443)) {
  Serial.println("Connection successful");
  char *input = (char *)fb->buf;
  char output[base64_enc_len(3)];
  String imageFile = "data:image/jpeg;base64,";
  for (int i=0;i<fb->len;i++) {
    base64_encode(output, (input++), 3);
    if (i%3==0) imageFile += urlencode(String(output));
  }
  String Data = myLineNotifyToken+myFoldername+myFilename+myImage;
  client_tcp.println("POST " + myScript + " HTTP/1.1");
  client_tcp.println("Host: " + String(myDomain));
  client_tcp.println("Content-Length: " + String(Data.length()+imageFile.length()));
  client_tcp.println("Content-Type: application/x-www-form-urlencoded");
  client_tcp.println("Connection: keep-alive");
  client_tcp.println();
  client_tcp.print(Data);
  int Index;
  for (Index = 0; Index < imageFile.length(); Index = Index+1000) {
    client_tcp.print(imageFile.substring(Index, Index+1000));
  }
  esp_camera_fb_return(fb);
  int waitTime = 10000;   // timeout 10 seconds        //here is a 10000
  long startTime = millis();
  boolean state = false;
  while ((startTime + waitTime) > millis())
  {
    Serial.print(".");
```

```
   delay(100);
   while (client_tcp.available())
   {
     char c = client_tcp.read();
     if (state==true) getBody += String(c);
     if (c == '\n')
     {
      if (getAll.length()==0) state=true;
      getAll = "";
     }
     else if (c != '\r')
      getAll += String(c);
     startTime = millis();
    }
    if (getBody.length()>0) break;
  }
  client_tcp.stop();
  Serial.println(getBody);
 }
 else {
  getBody="Connected to " + String(myDomain) + " failed.";
  Serial.println("Connected to " + String(myDomain) + " failed.");
 }
 return getBody;
}
String urlencode(String str)
{
  String encodedString="";
  char c;
```

```
}
```

```
    char code0;
    char code1;
    char code2;
    for (int i =0; i < str.length(); i++){
      c=str.charAt(i);
      if (c == ' '){
        encodedString+= '+';
      } else if (isalnum(c)){
        encodedString+=c;
      } else{
        code1=(c & 0xf)+'0';
        if ((c & 0xf) >9){
            code1=(c & 0xf) - 10 + 'A';
        }
        c=(c>>4)&0xf;
        code0=c+'0';
        if (c > 9){
            code0=c - 10 + 'A';
        }
        code2='\0';
        encodedString+='%';
        encodedString+=code0;
        encodedString+=code1;
      }
      yield();
    }
    return encodedString;
}
```

- **Base64.cpp file**

```
#if (defined(__AVR__))
#include <avr\pgmspace.h>
#else
#include <pgmspace.h>
#endif
const char PROGMEM b64_alphabet[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
"abcdefghijklmnopqrstuvwxyz"
"0123456789+/";
/* 'Private' declarations */
inline void a3_to_a4(unsigned char * a4, unsigned char * a3);
inline void a4_to_a3(unsigned char * a3, unsigned char * a4);
inline unsigned char b64_lookup(char c);
int base64_encode(char *output, char *input, int inputLen) {
	int i = 0, j = 0;
	int encLen = 0;
	unsigned char a3[3];
	unsigned char a4[4];

	while(inputLen--) {
		a3[i++] = *(input++);
		if(i == 3) {
			a3_to_a4(a4, a3);

			for(i = 0; i < 4; i++) {
				output[encLen++] = pgm_read_byte(&b64_alphabet[a4[i]]);
			}
			i = 0;
		}
	}
```

```
    if(i) {
        for(j = i; j < 3; j++) {
            a3[j] = '\0';
        }
        a3_to_a4(a4, a3);
        for(j = 0; j < i + 1; j++) {
            output[encLen++] = pgm_read_byte(&b64_alphabet[a4[j]]);
        }
        while((i++ < 3)) {
            output[encLen++] = '=';
        }
    }
    output[encLen] = '\0';
    return encLen;
}
int base64_decode(char * output, char * input, int inputLen) {
    int i = 0, j = 0;
    int decLen = 0;
    unsigned char a3[3];
    unsigned char a4[4];
    while (inputLen--) {
        if(*input == '=') {
            break;
        }
        a4[i++] = *(input++);
        if (i == 4) {
            for (i = 0; i <4; i++) {
                a4[i] = b64_lookup(a4[i]);
            }
            a4_to_a3(a3,a4);
            for (i = 0; i < 3; i++) {
                output[decLen++] = a3[i];
            }
            i = 0;
        }
    }
    if (i) {
        for (j = i; j < 4; j++) {
            a4[j] = '\0';
        }
        for (j = 0; j <4; j++) {
            a4[j] = b64_lookup(a4[j]);
        }
        a4_to_a3(a3,a4);
        for (j = 0; j < i - 1; j++) {
            output[decLen++] = a3[j];
        }
    }
    output[decLen] = '\0';
    return decLen;
}
int base64_enc_len(int plainLen) {
    int n = plainLen;
    return (n + 2 - ((n + 2) % 3)) / 3 * 4;
}
int base64_dec_len(char * input, int inputLen) {
    int i = 0;
    int numEq = 0;
```

```
        for(i = inputLen - 1; input[i] == '='; i--) {
                numEq++;
        }
        return ((6 * inputLen) / 8) - numEq;
}
inline void a3_to_a4(unsigned char * a4, unsigned char * a3) {
        a4[0] = (a3[0] & 0xfc) >> 2;
        a4[1] = ((a3[0] & 0x03) << 4) + ((a3[1] & 0xf0) >> 4);
        a4[2] = ((a3[1] & 0x0f) << 2) + ((a3[2] & 0xc0) >> 6);
        a4[3] = (a3[2] & 0x3f);
}

inline void a4_to_a3(unsigned char * a3, unsigned char * a4) {
        a3[0] = (a4[0] << 2) + ((a4[1] & 0x30) >> 4);
        a3[1] = ((a4[1] & 0xf) << 4) + ((a4[2] & 0x3c) >> 2);
        a3[2] = ((a4[2] & 0x3) << 6) + a4[3];
}
inline unsigned char b64_lookup(char c) {
        if(c >='A' && c <='Z') return c - 'A';
        if(c >='a' && c <='z') return c - 71;
        if(c >='0' && c <='9') return c + 4;
        if(c == '+') return 62;
        if(c == '/') return 63;
        return -1;
}
```

- **Base64.h file**

```
#ifndef _BASE64_H
#define _BASE64_H
extern const char b64_alphabet[];
int base64_encode(char *output, char *input, int inputLen);
int base64_decode(char *output, char *input, int inputLen);
int base64_enc_len(int inputLen);
int base64_dec_len(char *input, int inputLen);

#endif // _BASE64_H
```

APPENDIX II

MACHINE LEARNING CODE

- **testing.py file**

```
from google.colab import drive
drive.mount('/content/drive')

!pip install gTTS
!pip install pyttsx3

import numpy as np
from keras.models import load_model
from gtts import gTTS
from tensorflow.keras.preprocessing import image
import pyttsx3
import matplotlib.pyplot as plt
from IPython.display import Audio, display
import time
import os
import glob
from google.colab import output

pathh = "/content/drive/MyDrive/college/"
img_path = "/content/drive/MyDrive/college/ESP32-CAM/"

# load the trained model
loaded_model=load_model(pathh+'currency_detector_model.h5')


# these are the class labels from the training data (each number stands for the currency
denomination)
class_labels = ['10','100','20','200','2000','50','500','Background']


# function to convert predicted text/label of uploaded image into text
def tts(class_label,class_likelihood):
    language='en'

    # if uploaded image is not a currency detected but is an image of a background
    if(class_label=="Background"):
        mytext='sorry but i am detecting only  the'+class_label+', please properly hold the currency
under the camera or upload the proper image of the currency'
    # if uploaded image is a currency
    else:
        mytext="this is {} Rs note, and I am  {: .2f} % sure of it".format(class_label,
class_likelihood*100)

    # gTTS() converts text into the audio and supports multiple languages
    myobj = gTTS(text=mytext, lang=language, slow=False)

    #store audio result
    file=pathh+'result.mp3'
    myobj.save(file)


# predict label of uploaded image (if uploaded image is a currency or not. if its currency then
what is its denomination)
def prediction(file_name):
    # load/read the uploaded image
    img = image.load_img(file_name, target_size=(256,256))

    # convert image into array
    image_to_test = image.img_to_array(img)

    # as Keras expects a list of images, not a single image, thus add a fourth dimension to the
image
    list_of_images = np.expand_dims(image_to_test, axis=0)

    # make a prediction using the loaded trained model
    results = loaded_model.predict(list_of_images)

    # as we are only testing one image, we only need to check the first result
    single_result = results[0]
    # from likelihood score for all possible classes, find out which class had the highest score.
    # the class with highest likelihood is predicted as the result.
    most_likely_class_index = int(np.argmax(single_result))
    class_likelihood = single_result[most_likely_class_index]

    # get the name of the most likely class
    class_label = class_labels[most_likely_class_index]

    # print the result
    print(file_name)
    print("This is image is a {} - Likelihood: {: .2f}".format(class_label, class_likelihood))

    # convert the actual prediction result text into audio file.
    tts(class_label,class_likelihood)


def automate():
  print("process started")
  print("waiting for actual images to come")
  time.sleep(50)

  files = glob.glob(img_path+"*")
  for f in files:
      os.remove(f)

  count = 0

  while True:
    lst = os.listdir(img_path)
    if len(lst)>0:
      time1 = time.time()

      # path to the uploaded image file
      file_to_predict = img_path+"ESP32-CAM.jpg"        #img_path+"ESP32-CAM.jpg"
      print("picked "+ str(count) + " sequence image")

      # display the uplaoded image
      img = image.load_img(file_to_predict, target_size=(256,256))
      plt.imshow(img)

      #predict the label of upload image file and convert the label into audio
      prediction(file_to_predict)
      path=pathh+'result.mp3'

      #play the output audio
      display(Audio(path, autoplay=True))
      print("predicting output of "+ str(count) + " sequence image")

      time2 = time.time()
      sub_time = time2-time1
      if sub_time<8: #if whole process of prediction completes within time less than 8s then make
it wait....because next image will come after 10s
      time_left = 8 - sub_time
      time.sleep(time_left)

      #if more than 8s, then let process continues

      files = glob.glob(img_path+"*")
      for f in files:
          os.remove(f)

      output.clear()
      count += 1
```

- **training.py file**

```
#On the GPU of this notebook for training


from google.colab import drive
drive.mount('/content/drive')


"""# Importing libraries"""
```

```
# Commented out IPython magic to ensure Python compatibility.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# %matplotlib inline
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.utils import plot_model
from keras.optimizers import Adam
from keras.layers import Input, Dense, Flatten, Dropout, GlobalAveragePooling2D
from keras.applications import VGG16
from keras.models import Model


import os
import random

#change_1
path = '/content/drive/MyDrive/Drishti/college/train_and_val_data'

#change_2
# total number of classes
num_classes = 6

"""# Data pre-processing pipeline is "preprocessing.py". Run it on local system after you downloaded the raw data folder.
```

Just run this below code to see if any data is corrupt or not. If corrupt, delete it before starting the training

```
"""

import os
import cv2
bad_list=[]
dir= path+'/processed_data'

subdir_list=os.listdir(dir) # create a list of the sub directories in the directory ie train or test
for d in subdir_list:  # iterate through the sub directories train and test
    dpath=os.path.join (dir, d) # create path to sub directory
    if d in ['test', 'train']:
        class_list=os.listdir(dpath) # list of classes ie dog or cat
       # print (class_list)
        for klass in class_list: # iterate through the two classes
            class_path=os.path.join(dpath, klass) # path to class directory
            #print(class_path)
            file_list=os.listdir(class_path) # create list of files in class directory
            for f in file_list: # iterate through the files
                fpath=os.path.join (class_path,f)
                index=f.rfind('.') # find index of period infilename
                ext=f[index+1:] # get the files extension
                if ext  not in ['jpg', 'png', 'bmp', 'gif']:
                    print(f'file {fpath}  has an invalid extension {ext}')
                    bad_list.append(fpath)
                else:
                    try:
```

```
                img=cv2.imread(fpath)
                size=img.shape
            except:
                print(f'file {fpath} is not a valid image file ')
                bad_list.append(fpath)

print(bad_list)

"""# Load dataset and image agumentaion

Total of 12 classes 10rs, 20rs, 100rs, 200rs, 500rs, pen, copy, chair, table, banana, apple and background.
"""

data_agumentation=ImageDataGenerator(rescale=1./255,
                        shear_range=0.2,
                        zoom_range=0.2,
                        horizontal_flip=True,
                        vertical_flip=True)

# load training data
train_data=data_agumentation.flow_from_directory(directory=path+'/processed_data/train',
                                                 target_size=(256,256),
                                                 class_mode='categorical',
                                                 batch_size=32,
                                                 shuffle = True)

#load validation data
val_data=ImageDataGenerator().flow_from_directory(directory=path+'/processed_data/test',
                                        target_size=(256,256),
                                        class_mode='categorical')

"""# Model

### define model add layers and compile

train 1
"""

# #define the mdel
# my_new_model = Sequential()

# # add the resnet to new defined model
# my_new_model.add(ResNet50(include_top=False, pooling='avg', weights='imagenet'))

# #last fully connected layer with softmax as a activation function
# my_new_model.add(Dense(num_classes, activation='softmax'))

# my_new_model.layers[0].trainable = False

# # View the structure of the model.
# my_new_model.summary()

"""train 2"""

base = VGG16(input_shape=[256, 256] + [3], weights='imagenet', include_top=False)

for layer in base.layers:
  layer.trainable = True

# We added more layers above the loaded VGG16 network to increase capacity of overall
network to understand and remember features of training images.

x = base.output
x = GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x)
x = Dropout(0.5)(x)
x=Dense(512,activation='relu')(x)
prediction = Dense(num_classes, activation='softmax')(x)

# Then we create a model object, i.e. define the entrance and exit points of the complete model
which is about to be trained. This complete model includes VGG16 + additional layers added on
top of VGG16

my_new_model = Model(inputs=base.input, outputs=prediction)

# View the structure of the model.
my_new_model.summary()

"""### compile the model with optimizer and loss function"""

# compile the model with adam optimizer, categorical_croosentropy loss function
my_new_model.compile(optimizer=Adam(learning_rate=0.0001),loss='categorical_crossentropy'
,metrics=['accuracy'])

"""### early stopping and callback"""
```

```
checkpoint = ModelCheckpoint(path+"/weightsss-{epoch:02d}-{val_accuracy:.4f}.h5",
monitor='val_accuracy',verbose=1, save_best_only=True, save_weights_only=False,
mode='max')
early = EarlyStopping(monitor='val_loss', min_delta=0, patience=20, verbose=1, mode='auto')

"""### fit the model with train and val dataset"""

# fit the model with train data and validation data
r = my_new_model.fit(
        train_data,
        epochs = 200,
        validation_data=val_data,
        callbacks=[checkpoint, early],
        use_multiprocessing=True,
        workers=1)

# Then we plot loss curves
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()

# and accuracies curves
plt.plot(r.history['accuracy'], label='train acc')
plt.plot(r.history['val_accuracy'], label='val acc')
plt.legend()
plt.show()
```