A PROJECT REPORT ON

"CURRENCY RECOGNITION SYSTEM FOR VISUALLY IMPAIRED USING SIRB ALGORITHM"

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

MASTER OF COMPUTER APPLICATIONS

Submitted By

LINGAMPALLI KRUPA CHARAN PAUL

Reg No. 2251926013

Under the esteemed guidance of

Mr. R.SRIDHAR,M. Tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OFENGINEERING

Dr. B.R. AMBEDKAR UNIVERSITY, SRIKAKULAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGCOLLEGE OF ENGINEERING

Dr. B.R. AMBEDKAR UNIVERSITY, SRIKAKULAM

CERTIFICATE

This is to certify that the project entitled "**CURRENCY RECOGNITION SYSTEM FOR VISUALLY IMPAIRED USING SIRB ALGORITHM**" that is being submitted by **LINGAMPALLI KRUPA CHARAN PAUL(2251026013)** in partial fulfilment of requirements for the award of the degree in **MASTER OF COMPUTER APPLICATIONS** during 2022 - 2024, in **Dr. B. R. AMBEDKAR UNIVERSITY, COLLEGE OF ENGINEERING** is a record of bonafide work carried out by her under my guidance and super vision. The results embodied in this work have not being submitted to any other university or institute for the award of any degree or diploma.

PROJECT SUPERVISOR

Mr. R.SRIDHAR, M.Tech

Assistant Professor

HEAD OF THE DEPARTMENT

Mr.R.SRIDHAR,M.Tech

Assistant professor

External examiner

AKNOWLEDGEMENT

A Project is a golden opportunity for learning and self-development. I consider myself lucky and privileged to express my gratefulness and deep sense of gratitude and to our guide **Mr.R.SRIDHAR,M.Tech** , **Assistant professor, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,** for stimulating suggestions and encouragement that helped me at every stage of our project work, which made this project successful.

I also express my gratefulness gratitude to our **Mr.R.SRIDHAR , M.tech** Head of the Department of Computer Science and Engineering, **Dr.B.R. AMBEDKAR UNIVERSITY, SRIKAKULAM for**his support and encouragement throughout the project.

I also express my gratefulness gratitude to our **Prof. Ch.RAJSEKHARRAO, Ph,D., Principal, College of Engineering, Dr.B.R.AMBEDKAR UNIVERSITY, SRIKAKULAM for** his support and encouragement throughout the project.

Further-more, I would also like to acknowledge my thankfulness with much appreciation the crucial role of our **TEACHING STAFF, NONTEACHING STAFF, PARENTS AND FRIENDS** for their love,
support, encouragement and cooperation.

Place: Srikakulam
Date:

LINGAMPALLI KRUPA CHARAN PAUL
Regd.No: 2251926013

DECLARATION

I hereby declare that the project work entitled **"CURRENCY RECOGNITION SYSTEM FOR VISUALLY IMPAIRED USING SIRB ALGORITHM"** submitted by me for the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA) under the guidance of Mr.R.SRIDHAR, M.Tech, Assistant Professor.** Dr. B.R. AMBEDKAR UNIVERSITY , SRIKAKULAM. Is original and it has not been submitted earlier.

Place: Srikakulam
Date:

LINGAMPALLI KRUPA CHARAN PAUL
Regd.No:2251926025

ABSTRACT

It is relatively simple for a normal human to lead a life, but when it comes to visually impaired persons, leading a life is not that easy; they get obstacles in each and every aspect of their life. Among them, interpreting and understanding every banknote is the major difficulty, especially when it comes to paper currency.

Since currency plays such a vital role in our day to day lives and is crucial for every possible business and personal transaction, real-time detection and recognition of banknotes become a necessity for blind or visually impaired people. For that purpose, we are going to propose a real-time object detection system to help visually impaired people in their daily transactions.

In this proposal, we are going to develop a system that scans the currency note and helps the blind person to identify it, as the system voices it out with the audio of different languages like English, Telugu, for that respective currency note. We are going to use the SIRB algorithm in Machine learning in the back end to store the audio and photos of bank notes, and image processing, which is used while scanning the currency note.

Keywords- Machine learning, SIRB algorithm, Image processing.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF OUTPUTS

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

Visually Impaired are those people who have vision impairment or vision loss. 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 system helps visually impaired patients to recognize and detect money. We are going to develop a system that scans the currency note and helps the blind person to identify it, as the system voices it out with the audio of different languages like English, Telugu for that respective currency note. Despite the widespread usage of ATMs, Credit-Debit Cards, and other digital modes of payment like Google Pay, Paytm, and Phone Pay, money is still widely used for most daily transactions due to its convenience. Currency recognition or banknote recognition is a process of identifying the denominational value of a currency. It is a simple and straightforward task for the normal human beings, but if we consider the visually challenged people currency recognition is a challenging task. Visually handicapped people have a difficult time distinguishing between different cash denominations. Even though unique symbols are embossed on different currencies in India, the task is still too difficult and time-consuming for the blind. This brings a deep need for automatic currency recognition systems. So, our paper studies about the systems in order to help the visually challenged or impaired people; so that they can differentiate between various types of Indian currencies through implementation of image processing techniques. The study aims to investigate different techniques for recognising Indian rupee banknotes. From our work, the visually impaired people will be capable of recognizing different types of Indian currencies while their monetary transactions, so that they lead their life independently both socially and financially.We are going to propose a system that can recognize a currency note by scanning it and voicing out in three languages, such as English, Telugu to assist a blind person in recognizing it. We are going to use the SIRB algorithm in Machine Learning.The methods that we are going to use will work well on noisy images captured from a mobile phone. The proposed system also has advantages like high performance and simplicity. More accurate when compared to the existing system.

1.1 Project Overview

Recognition of banknotes becomes essential for blind or visually impaired persons because currency is so important to our daily lives and is required for every business and personal transaction. In order to assist those who are visually impaired in their daily activities, we will present a currency note recognition system.

1.2 Project Deliverables

The main aim of the project is to assist blind people to recognize a currency note by scanning it and voicing out in three languages, such as English, Telugu to assist a blind person in recognizing it.

1.3 Project Justification

Our main project justifies every visually impaired by assisting them with a voice that helps them to perform monetary transactions in simple manner.

1.4 Project Scope

Identification of a currency's denominational value is referred to as currency recognition or banknote recognition. Although it is a simple and uncomplicated exercise for many individuals, recognising cash might be difficult for persons who are vision impaired. People who are visually impaired have trouble telling the difference between various monetary denominations. The job is still too challenging and time-consuming for the blind even though different currencies in India have distinctive symbols engraved on them. The requirement for automatic money identification systems increases as a result. Therefore, the focus of our work is on systems that assist people who are blind or visually impaired in distinguishing between different forms of Indian currency using image processing techniques. The study's objective is to investigate various systems for classifying Indian rupee banknotes. Due to our efforts, visually impaired people would be able to distinguish between different Indian currencies while transacting money, allowing them to have independent social and financial lives.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

Prashengit Dhar et.al.[1]. In their work titled “Paper Currency Detection System Based on Combined SURF and LBP Features,” they said that monitoring systems also helps in a secure maintenance task. Paper currency detection is a part of the monitoring system. It is a helpful hand for those who are visually impaired. In banking systems paper currency detection can play a great role. Moreover, in different offices and markets, it is usually happened that someone lost his currency or fall down anywhere, in this case this system can act as a chief role in detecting paper currency.Thus, they proposed a new paper currency detection system for Bangladeshi paper currency using combined SURF and LBP features. The prediction is made by the SVM classifier, which suits best to the proposed model. Paper currency detection is a vital application for surveillance systems.In this paper, they proposed a new method to detect paper currency. They considered only paper currencies of Bangladesh. It is quite a challenging task to detect currency in different positions from different environments. This proposed system can detect currency in rotated positions also. This proposed system can also calculate the total amount of currency that exists in an image. We achieved an overall accuracy of 92.6%. But the main limitation of their work is their system cannot recognize the noisy data.So, they said that,later they will try to improve the performance for rotated position of currency.They will also focus on folding notes and overlapping notes. They will try to consider all possible sorts of notes. Moreover, they will investigate other local or global features to improve the performance better.

Barani.S et.al.[2], in their work titled as “Currency Identifier for Indian Denominations to Aid Visually Impaired”, said that identification of various denominations of currency is not an easy task for visually impaired people. In India though there are special symbols embossed on different denominations, still the task is tedious for blind people. The lack of identification devices motivated the need of a handheld device for segregation of different denominations. This work attempts to design a device using IR sensors, for Currency identification of Indian Denominations (CiID) to aid visually impaired people. The variation in the voltage level of various currencies is observed and detection has been implemented. This device CiID helps blind communities to differentiate various denominations of Indian rupee. Since IR sensors are used for scanning and the number of sensors used is also limited to eight, the device is cost effective. The accuracy of the device upon testing is 86 %. The sensors used are sensitive to light hence the design of the device should be in such a way that light intensity does not affect the performance of the device. The pattern of signals acquired from sensors could be trained using a neural network. Patterns trained in such a way can lead to improvement of device accuracy. The main

limitation of this is that the size of the device is large and uncomfortable. So, they decided that the size of the device should also be reduced in future for easy portability.

Zóra Solymár et.al.[3], in their work titled as "Banknote Recognition for Visually Impaired", said that The Bionic Eyeglass is a portable device that helps blind and visually impaired people in everyday navigation, orientation and recognition tasks requiring visual input. In countries where banknotes cannot be distinguished by their size people with visual impairments have real difficulties in determining their face values. Several currencies have been designed with tactile marks embossed on the banknotes, however, due to the usage and softness of banknotes they quickly lose most of the tactile information and become practically unusable for the intended target group. They have shown that portrait objects are more likely to be detected without significant loss at a certain threshold level than numbers. This phenomenon can be explained by the location of the two objects. The bounding defines sub-classes representing different binarizations of the same object, two for each class (super-class). For each subclass the positive examples for the training were the elements of the sub-class, the negative examples were the samples in all other super-classes, whereas examples from the same superclass but from different subclasses have been left out. They have used k-means clustering on the Zernike moment feature vector to split the classes. They achieved the accuracy of 95.89% Based on these votes, the subjects could correctly identify the banknote in 98.95% of the cases.

Nada N. Saeed et.al.[4], in their work titled as "Android-Based Object Recognition for the Visually Impaired", as per the World Health Organization (WHO) statistics of June 2012, there lives a sum of 285 million visually impaired individuals across the globe. Moreover, according to the same statistics, 90% out of this sum resides in developing countries where modern assistive gadgets are either unavailable or unaffordable. The aim of this paper is to develop an object recognition system that is easily attainable, manages to achieve high accuracy and is fast, portable and low-cost. Therefore, it will be designed as a smartphone application for Android that uses the device's built-in camera. The algorithm should be of low complexity so that it uses the least amount of power possible and manages to achieve real-time performance for improved user-friendliness. In comparison to its commercial iOS counterpart, this application was developed for the open source Android Operating System which makes it more affordable and easily accessible. It managed to achieve real-time performance as well as a high accuracy of 95.36%.

Swapnil Bhole et.al.[5], in their work titled as "Deep Learning based Object Detection and Recognition Framework for the Visually-Impaired",said that vision impairment or blindness is one of the top ten disabilities in humans, and unfortunately, India is home to the world's largest

visually impaired population. In this study, they presented a novel framework to assist the visually impaired in object detection and recognition, so that they can independently navigate, and be aware of their surroundings. The paper employs transfer learning on Single-Shot Detection (SSD) mechanism for object detection and classification, followed by recognition of human faces and currency notes, if detected, using Inception v3 model. The SSD detector is trained on a modified PASCAL VOC 2007 dataset, in which a new class is added, to enable the detection of currency as well. Furthermore, separate Inception v3 models are trained to recognize human faces and currency notes, thus making the framework scalable and adaptable according to the user preferences. Ultimately, the output from the framework can then be presented to the visually impaired person in audio format. Mean Accuracy and Precision (mAP) scores of standalone SSD detectors of the added currency class was 67.8 percent, and testing accuracy of person and currency recognition of Inception v3 model were 92.5 and 90.2 percent respectively.

Suriya Singh et.al.[6], in their work titled “Currency Recognition on Mobile Phones”, said that their paper introduces Visual object recognition on a mobile phone which has many applications. In this paper, they focus on the problem of recognition of currency bills on a low-end mobile phone. This is an immediate requirement for the visually impaired individuals. There are around 285 Million people estimated to be visually impaired worldwide, out of which 39 Million are blind and 246 Million have low vision. The differences in texture or length of currency bills are not really sufficient for identification by the visually impaired. Moreover, bills are not as easy to distinguish by touch as coins. Certain unique engravings are printed on the bills of different currencies but they tend to wear away. The problem of currency recognition using computer vision techniques has been studied in the past. Neural networks have as an instance recognition under clutter. We are able to use a thin index structure to make the application efficient and compact. Also note that our problem is considerably different from typical image instance retrieval (e.g. buildings). Their method is generic and scalable to multiple domains including those beyond the currency bills. Our solution uses a visual Bag of Words (BoW) based method for recognition. To enable robust recognition in a cluttered environment, we first segment the bill from the background using an algorithm based on iterative graph cuts. They formulate the recognition problem as an instance retrieval task. This is an example of fine-grained instance retrieval that can run on mobile devices. We evaluate the performance on a set of images captured in diverse natural environments, and report an accuracy of 96.7% on 2584 images.

Md. Ferdousur Rahman Sarker et.al.[7], in their work titled as “Real-time Bangladeshi Currency Detection System for Visually Impaired Person”, said that their one of the main problems suffered by a visually impaired person to identify paper currencies because of the similarity of paper size and texture between different banknotes. As currencies are the commonly

used stuff in everyday life, understanding the value of the banknotes is a very important task for them. Therefore, they have proposed a real-time system that will help them to recognize the currencies and resolve this crisis to make visually impaired people feel confidence in the financial dealings, not depending on others. They have applied the widely used ORB based feature descriptor for recognizing the traditional Bangladeshi paper currencies. The average recognition rate for each of the different types of banknotes are documented in the experimental results. The recognition rate is higher than any other methods applied for experimentation and the average matching rate is also quite satisfactory considering a real-time system. The presented system could be very helpful for the visually impaired personals, who can use the mobile application to recognize the banknotes very accurately.

CHAPTER 3

PROBLEM ANALYSIS

3. PROBLEM ANALYSIS

3.1 Existing System:

- In the existing system, they proposed a paper currency detection system for Bangladeshi paper currency, where the currency was recognised using SURF and LBP features .
- The prediction is made by the SVM classifier with accuracy of 92%.

3.1.1 Challenges:

- If the user supplies incorrect data, then abnormality is detected as the system could not classify features precisely.
- It tends to be difficult to access and implement currency detection.

3.2. Proposed system:

- We are going to propose a system that can recognize a currency note by scanning it and voicing it out in English to assist a blind person in recognizing it.
- The methods used work well on noisy images captured from a mobile phone.
- We are going to use the SIRB algorithm in Machine Learning.

3.2.1. Advantages:

- Works efficiently on noisy images captured from a mobile phone.
- Results in high performance and simplicity.
- More accurate when compared to the existing system.

CHAPTER 4

SYSTEM ANALYSIS

4. SYSTEM ANALYSIS

Systems analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. System analysis is the process of studying a procedure in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way.

The development of a computer-based information system includes a systems analysis phase which produces or enhances the data-model which itself is a precursor to creating or enhancing a database. There are a number of different approaches to system analysis. When a computer-based information system is developed, systems analysis would constitute the following steps:

- The development of a feasibility study, involving determining whether a project is economically, socially, technologically and organizationally feasible.
- Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of work on the existing system.
- Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on.

4.1 System requirement specifications:

A requirement is a feature that the system must have or a constraint that it must to be accepted by the client. Requirement engineering aims at defining the requirements of the system under construction.

Requirement engineering include two main activities, requirement elicitation, which result in the specification of the system that client understands, and analysis which in analysis model that the developer can unambiguously interpret. A requirement is a statement about what the proposed system will do. Requirements can be divided into two major categories: Functional requirements and Non-Functional requirements.

4.1.1 Functional requirements

Functional requirements describe the interactions between the system and its environment independent of its implementation. The environment includes the user and any other external system with which the system interacts. Functional requirements capture the intended behavior of the system, this behavior may be expressed as services, tasks or functions the system is required to perform.

In product development, it is useful to distinguish between the baseline functionality necessary for any system to compete in that product domain, and features that differentiate the system from competitors' products and from variants in your company's own product line/family. Features may be additional functionality or differ from the basic functionality along some quality attribute (such as performance and memory utilization).

4.1.2 Non-Functional Requirements

Non-functional requirements describe the aspects of the system that are not directly related to the functional behavior of the system. Non-functional requirements include a broad variety of requirements that apply to many different aspects of the system, from usability to performance.

- **Portability**: The degree of conversion of our application to target system is easy (e.g., Windows, Unix etc.).
- **Efficiency**: Our application uses the CPU cycles, memory and disk space efficiently.
- **Accuracy:** Our application gives accurate results to what clients expect.
- **Usability:** It is very easy to learn and operate the system.
- **Cost and development time:** The cost and development time is very less.

4.2 Object Oriented Analysis

In the case of object-oriented analysis, the process is varying. But these two are identical at use case analysis. Actually, the steps involved in the analysis phase are

- Identify the actors
- Develop a simple business process model using UML activity diagram
- Prepare interaction diagrams
- Classification- develop a static UML class diagram Identify classes, relationships, attributes and methods.

System models

- Scenarios
- Use Case Model

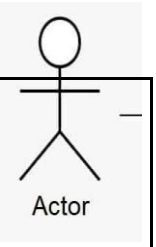
Use case is a description of the behavior of the system. That description is written from the point of view of a user who just told the system to do something particular.

4.2.1 Use-Case Scenario

Consider the use case where the user gives input data. The input data which is in raw format is then given for pre-processing. In pre-processing, missing values get eliminated and the input scales in between (0,1) to maintain uniformity. Then pre-processed data is given to analyze.

Use-Case Diagrams

An important part of the Unified Modelling Language (UML) is the facilities for drawing use case diagrams. Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors and use cases. Actors represent roles that can be played by users of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems. Use cases describe the behavior of the system.

Actor	An actor as mentioned is a user of the system and is depicted using a stick figure. The role of the user is written beneath the icon. Actors are not limited to humans. If a system communicates with another application and expects input or delivers output then that application can also be considered as an actor.	
Use case	A Use Case is the functionality provided by the system typically described as verb + object (e.g.: Register Car, Delete User). Use Cases are depicted with an ellipse. The name of the Use Case is written within the ellipse.	

Directed Association	Associations are used to link Actors with use cases and indicate that an actor participates in the Use Case in some form. Directed Association is the same as association but the difference is that it is represented by a line having an arrowhead.	
System boundary boxes	You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not.	

Table 4.1 Graphical Notations for Use Case Diagram

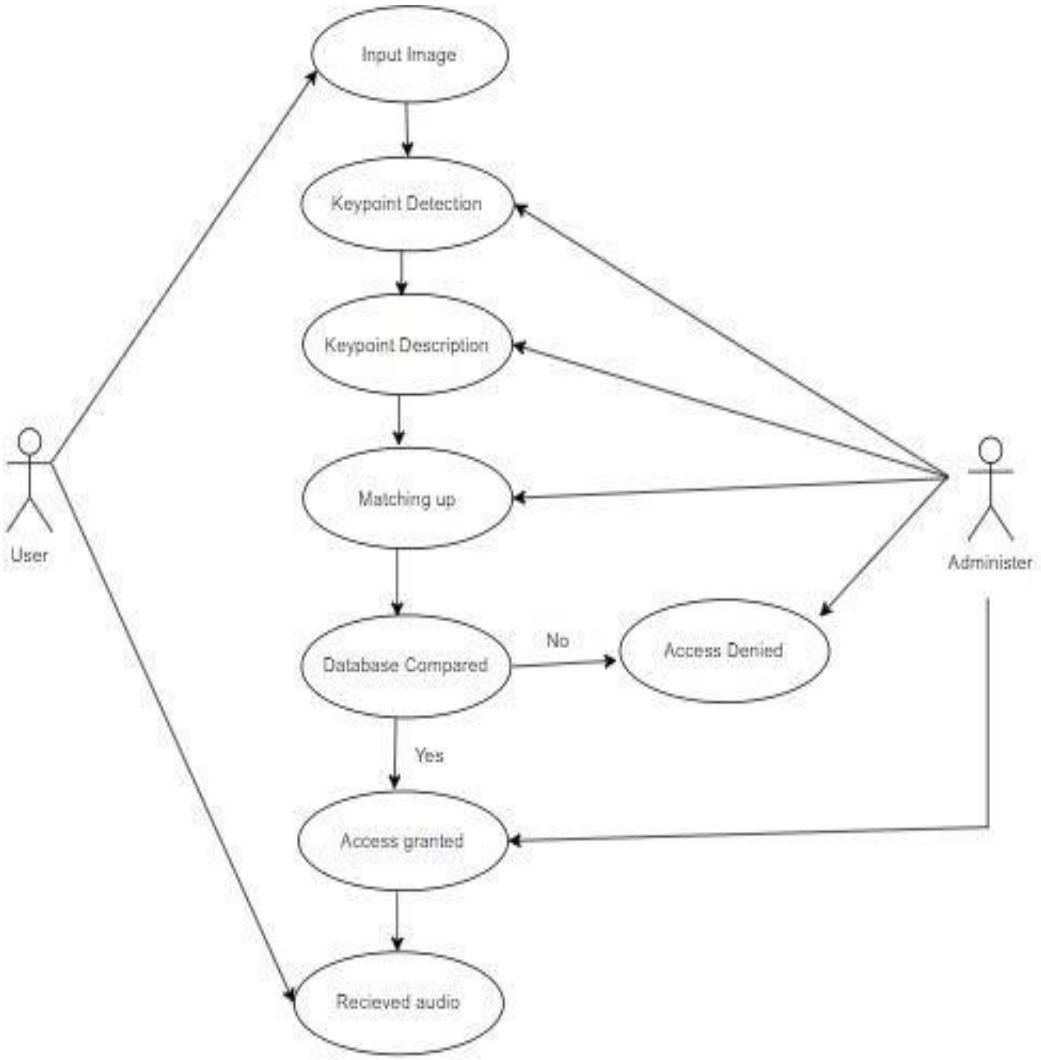


Fig 4.1 Use Case Diagram of Currency Recognition System

4.3 System Requirements

System requirements specification is a detailed statement of the effects that a system is required to achieve. A good specification gives a complete statement of what the system is to do, without making any commitment as to how the system is to do it.

A system requirements specification is normally produced in response to a user requirement specifications or other expression of requirements, and is then used as the basis for system design. The system requirement specification typically differs from expression of requirements in both scope and precision; the latter may cover both the envisaged system and the environment in which it will operate, but may leave many broad concepts unrefined.

4.3.1 Software requirements

- VS CODE
- IDLE (Python 3.9 64 bit)
- Scanner

4.3.2 Hardware requirements

- Processor : i5 (or Above)
- RAM : 32 GB (or 16 GB of 1600 MHz DDR3 RAM)
- Storage : 300 GB

CHAPTER 5

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 Introduction

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

Systems design mainly concentrates on defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture.

The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to be able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory, operations research and computer science in general.

5.1.1 Class Diagram

The Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting 17 different aspects of a system but also for constructing executable code of the software application. It describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams which can be mapped directly with object-oriented languages.

Purpose

The purpose of the class diagram is to model the static view of an application. The class diagrams are only diagrams which can be directly mapped with object-oriented languages and

thus widely used at the time of construction [9]. The UML diagrams like activity diagram, sequence.

The diagram can only give the sequence flow of the application but the class diagram is a bit different. So, it is the most popular UML diagram in the coder community. So, the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe the responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Active Class

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.

Visibility

Use visibility markers to signify who can access the information which is in a class. There are three possibilities. They are:

- Private visibility hides information from anything outside the class partition.
- Public visibility allows all other classes to view the marked information.
- Protected visibility allows child classes to access information which is inherited from a parent class.

Associations

Associations represent static relationships between the classes. Place the association names above, on or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles at the end of an association. Roles represent how the two classes see each other.

Association

Multiplicity (Cardinality)

Place multiplicity notations at the ends of an association. These symbols indicate the number of instances of one class linked to the instances of another class.

Constraint

Place constraints inside curly braces {}

Composition and Aggregation

Composition and Aggregation links a semantic association between two classes in the UML diagram. They are used in class diagrams. They both differ in their symbols.

Composition

Aggregation

Generalization

It is a specification relationship in which objects of the specialized element (the child) are suitable for objects of the generalization element (the parent). It is used in class diagrams.

Generalization

Class Diagram

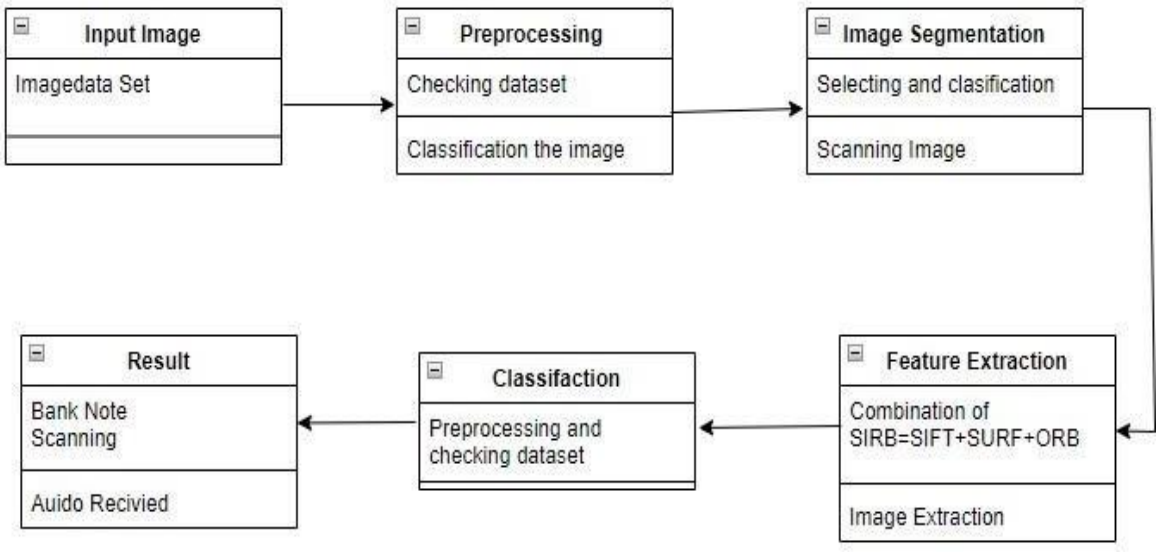


Fig 5.1 Class Diagram of Currency Recognition System

5.1.2 Activity Diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It puts emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

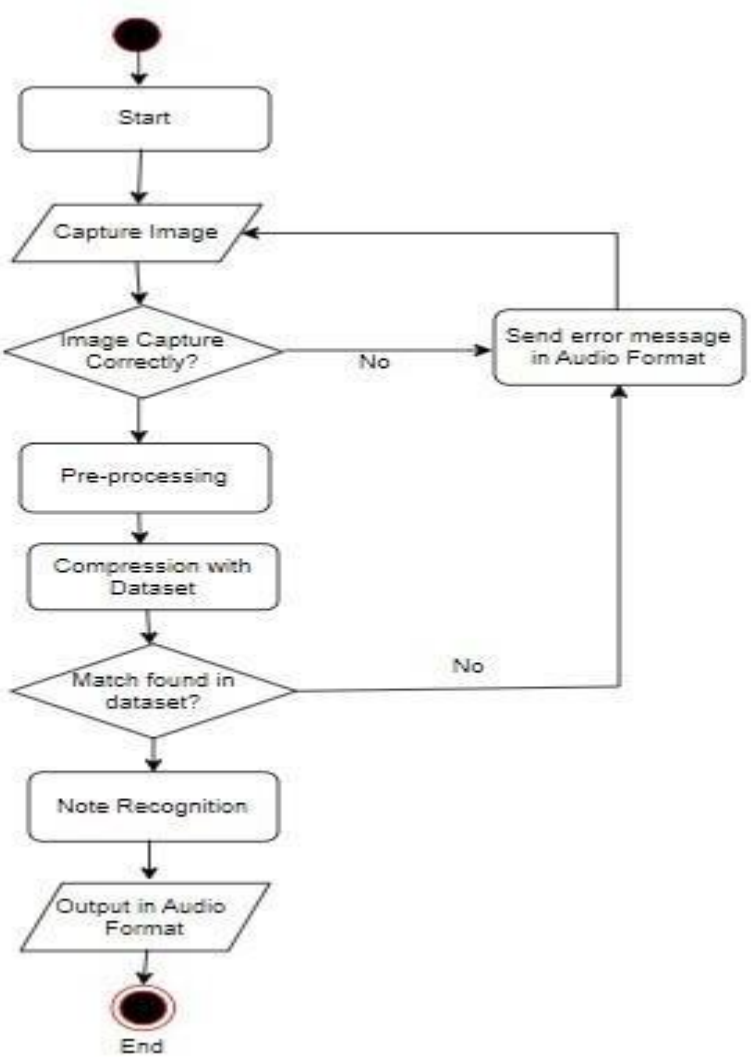
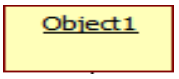


Fig 5.2 Activity Diagram of Currency Recognition System

Basic Notation of the Activity Diagram

Initial Node 	A black circle is the standard notation for an initial state before an activity takes place. It can either stand alone or you can use a note to further elucidate the starting point.
Activity 	The activity symbols are the basic building blocks of an activity diagram and usually have a short description of the activity they represent.
Control Flow	Arrows represent the direction flow of the flow chart. The arrow points in the direction of progressing activities.
Branch 	A marker shaped like a diamond is the standard symbol for a decision. There are always at least two paths coming out of a decision and the condition text lets you know which options are mutually exclusive.
Fork 	A fork splits one activity flow into two concurrent activities
Join 	A join combines two concurrent activities back into a flow where only one activity is happening at a time.
	The final flow marker shows the ending point for a process in a flow. The difference between a final flow node and the end state node is that the latter represents the end of all flows in an activity.
 **Complete Activity Flow**	The black circle that looks like a selected radio button is the UML symbol for the end state of an activity. As shown in two examples above, notes can also be used to explain an end state.
Notes 	The shape used for notes.

Object	Objects are instances of classes and are arranged horizontally. The pictorial representation for an Object is class (a rectangle) with the name prefixed by the object name (optional).	
Actor	Actors can also communicate with objects so they too can be listed as a column.An Actor is modeled using the stick figure.	

Lifeline	The Life Line identifies the existence of the object over time. The notation for a lifetime is a vertical dotted line extending from an object.	
Activation	Activation modeled as rectangular boxes on the lifeline indicate when the object is performing an action.	

Table 5.1 Graphical Representation of ActivityDiagram

5.2 System Architecture

- Gather a dataset of images of different currencies. You can either take pictures of currencies yourself or find a pre-existing dataset online.
- Preprocess the images by resizing, cropping, and adjusting their brightness and contrast.
- Use SIFT and ORB feature detectors and descriptors to extract features from the images.
- Train a machine learning model, such as a Support Vector Machine (SVM) on the feature vectors to recognize different currencies.
- Evaluate the performance of the model on a test set of images and fine-tune the model if necessary.
- Develop a user-friendly interface that can take input images from a camera or a file and output the recognized currency using text-to-speech or other auditory feedback methods.
- Test the interface with visually impaired individuals to get feedback and improve the user experience.

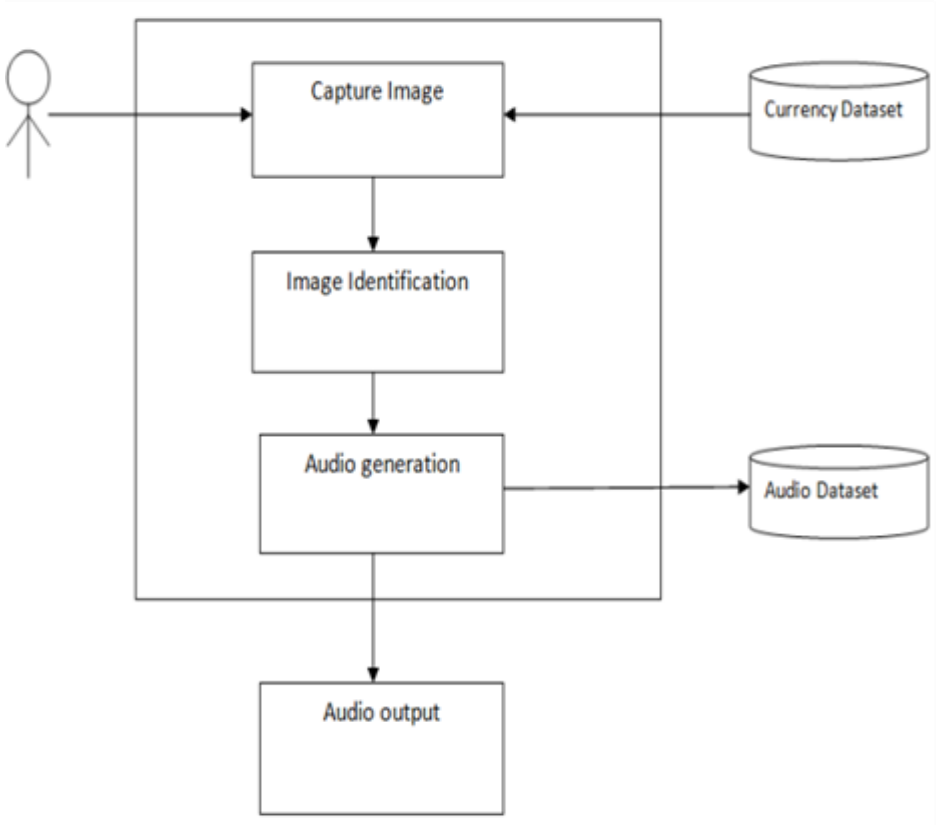


Fig 5.3: System Architecture of Currency recognition System

5.3 Algorithm Description

The SIRB Algorithm:

- In this proposal we are going to use the SIRB algorithm which is a combination of three algorithms namely SIFT+SURF+ORB which are used for feature detectors in image scanning and processing.

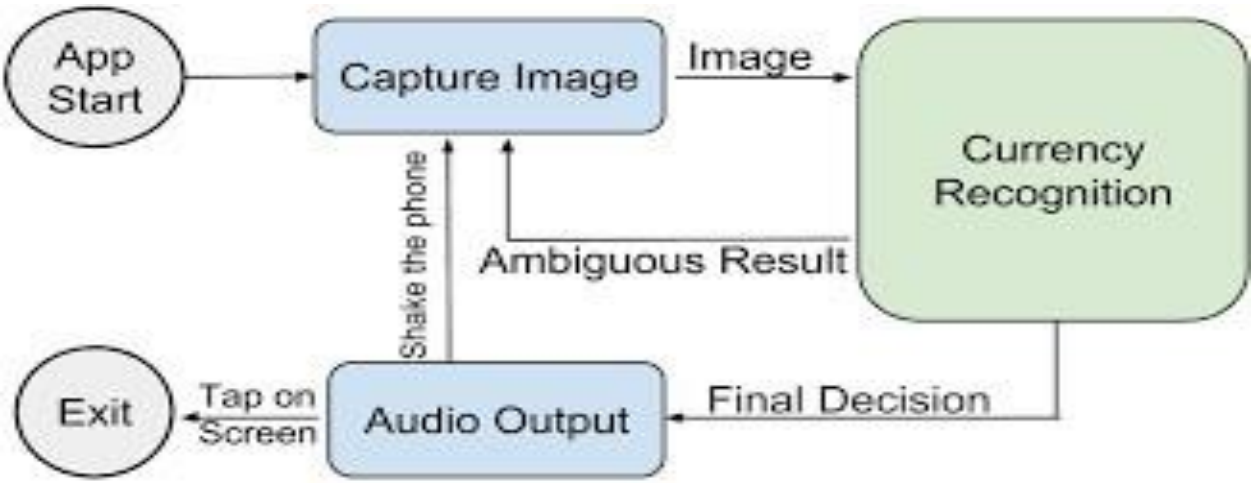


Fig 5.4: Algorithm Description Diagram for Currency Recognition System

Oriented FAST and Rotated BRIEF (ORB):

- The algorithm is a feature detector and descriptor that is widely used in image stitching and registration systems such as simultaneous localization and mapping (SLAM) systems. ORB algorithm has improved the speed of processing and robustness.
- Oriented FAST and rotated BRIEF (ORB) is a fast robust local feature detector, first presented by Ethan Rublee et al. in 2011, that can be used in computer vision tasks like object recognition or 3D reconstruction.
- ORB performs as well as SIFT on the task of feature detection (and is better than SURF) while being almost two orders of magnitude faster. ORB builds on the well-known FAST keypoint detector and the BRIEF descriptor. Both of these techniques are attractive because of their good performance and low cost.

The Steps for ORB Algorithm:

- Step 1: Determine scale and estimated location of prominent key points.
- Step 2: Refine their scale and location.
- Step 3: For each key point determine the orientations.
- Step 4: For each key point determine the descriptors.

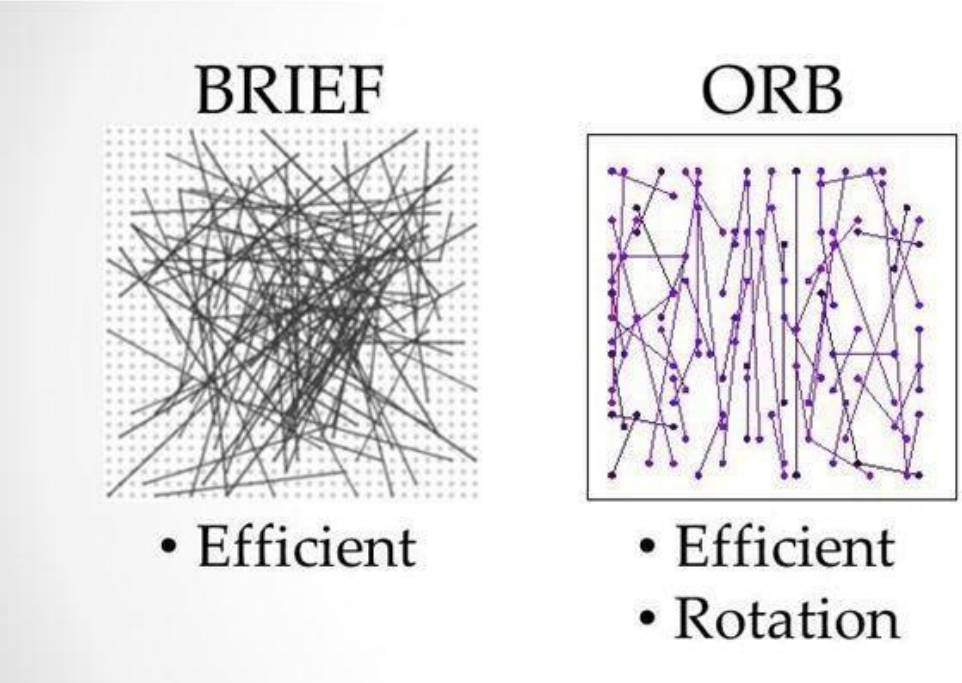


Fig 5.5 : ORB Algorithm for Object Recognition

The Scale-Invariant Feature Transform (SIFT):

- SIFT (Scale-Invariant Feature Transform) is a powerful technique for image matching that can identify and match features in images that are invariant to scaling, rotation, and affine distortion. It is widely used in computer vision applications, including image matching, object recognition, and 3D reconstruction.
- It is a computer vision algorithm to detect, describe, and match local features in images. Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, individual identification of wildlife and match moving.
- The main features of SIFT are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint.

The Steps for SIFT Algorithm:

- Step 1: Feature point (also called keypoint) detection.
- Step 2: Feature point localization.
- Step 3: Orientation assignment.
- Step 4: Feature descriptor generation.

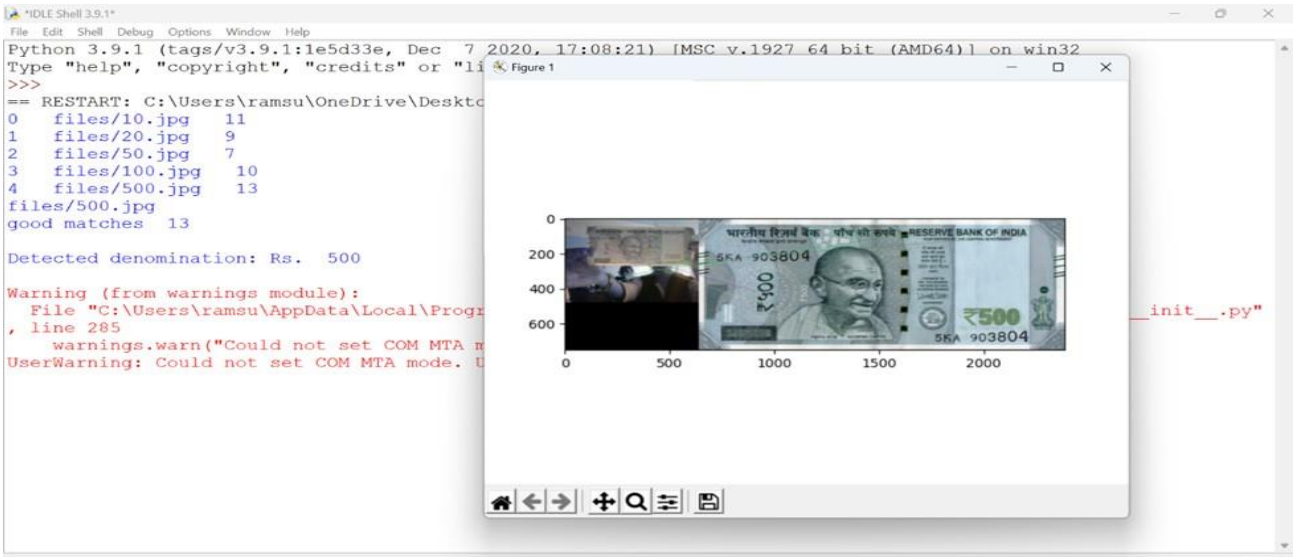


Fig 5.6 : Output for Currency Recognition

The Speeded Up Robust Features (SURF):

- The SURF algorithm is based on the same principles and steps as SIFT; but details in each step are different. The algorithm has three main parts: interest point detection, local neighborhood description, and matching.

- In this method is a fast and robust algorithm for local, similarity invariant representation and comparison of images. The main interest of the SURF approach lies in its fast computation of operators using box filters, thus enabling real-time applications such as tracking and object recognition.
- The two main advantages of SURF over SIFT is that SURF uses Laplacian of Gaussian so as to have a distinction between background and foreground features, and secondly, SURF uses only 64 dimensional vector compared to 128 dimensional vector for SIFT.

The Steps for SURF Algorithm:

SURF is composed the main two steps

Step 1 : Feature Extraction

Step 2 : Feature Description

Fig 5.7 : Feature Extraction for Currency Recognition for Rs.20

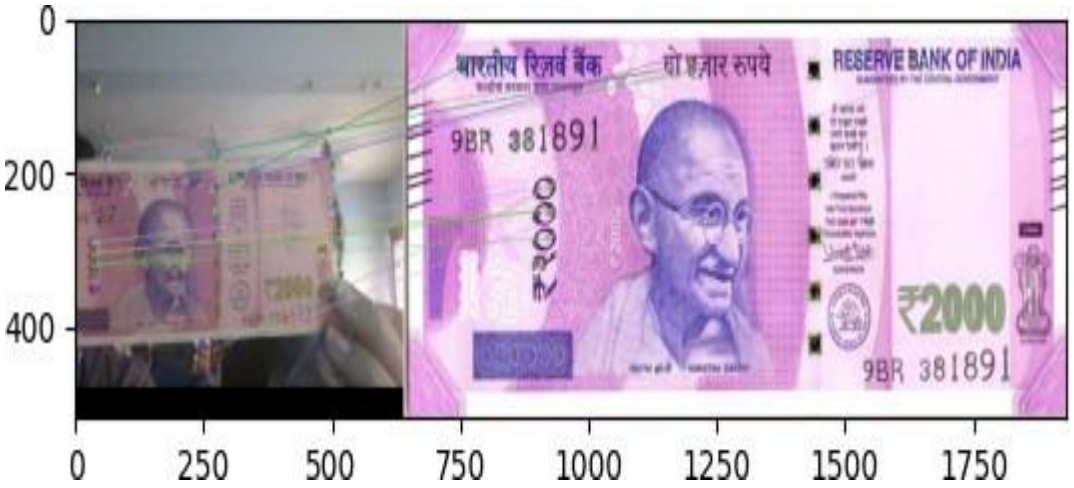


Fig 5.8: Feature Extraction for Currency Recognition for Rs.2000

CHAPTER 6

IMPLEMENTATION

6. IMPLEMENTATION

6.1 Technology Description:

6.1.1 Python:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Features of Python

- Simple
- Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudocode nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.
- Easy to Learn
- As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.
- Free and Open Source.

- Python is an example of FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good. It has been created and is constantly improved by a community who just want to see a better Python.
- Python can be used to develop both desktop and web apps and complex scientific and numerical applications.
- Python also has automatic memory management eliminating the need to manually allocate and free memory in the code.

NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the nd array object.

This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of a nd array will create a new array and delete the original.

The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.

NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

Pandas DataFrames

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can

accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Pandas support different kinds of data like:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and unordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data need not be labeled at all to be placed into a pandas data structure

OpenCV:

Opencv is an open source library which is very useful for computer vision applications such as video analysis, CCTV footage analysis and image analysis. OpenCV is written by C++ and has more than 2,500 optimized algorithms. When we create applications for computer vision that we don't want to build from scratch we can use this library to start focusing on real world problems. There are many companies using this library today such as Google, Amazon, Microsoft and Toyota. Many researchers and developers contribute. We can easily install it in any OS like Windows, Ubuntu and MacOS.OpenCV is written in the programming language C++, as is its primary interface, but it still retains a less comprehensive though extensive older C interface. All newer developments and algorithms appear in the C++ interface. There are language bindings in Python, Java, and MATLAB/Octave. The application programming interface (API) for these interfaces can be found in the online documentation. Wrapper libraries in several languages have been developed to encourage adoption by a wider audience.

In version 3.4, JavaScript bindings for a selected subset of OpenCV functions were released as OpenCV.js, to be used for web platforms.OpenCV runs on the desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD and OpenBSD as well as mobile operating systems: Android, iOS, Maemo, BlackBerry 10 and QNX.

gTTS:

gTTS (*Google Text-to-Speech*), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Writes spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout. It features flexible pre-processing and tokenizing.

6.2 Sample Source Code

Try.py :

```
# test file

# TODO:

#       Figure out four point transform

#       Figure out testing data warping

#       Use webcam as input

#       Figure out how to use contours

#       Currently detects inner rect -> detect outermost rectangle

#       Try using video stream from android phone

from utils import *

from matplotlib import pyplot as plt

import subprocess

from gtts import gTTS

import pyglet

import cv2

#from playsound import playsound

# histogram(img)

# fourier(img)

# img = harris_edge(img)

# display('image',img)

# show the original image and the edge detected image

# cv2.imshow("Image", image)

# cv2.imshow("Edged", edged)
```

```
# cv2.waitKey(0)

# cv2.destroyAllWindows()

# find the contours in the edged image, keeping only the

# largest ones, and initialize the screen contour

# must define here

max_val = 8

max_pt = -1

max_kp = 0

orb = cv2.ORB_create()

# orb is an alternative to SIFT

#test_img = read_img('files/test_100_2.jpg')

#test_img = read_img('files/test_100_2.jpg')

#test_img = read_img('files/test_100_3.jpg')

#test_img = read_img('files/test_20_4.jpg')

cap = cv2.VideoCapture(0)

ret,frame = cap.read()

cap.release()

cv2.imwrite('photo.jpg',frame)

test_img = read_img('photo.jpg')

# resizing must be dynamic

original = resize_img(test_img, 0.4)

display('original', original)

# keypoints and descriptors

# (kp1, des1) = orb.detectAndCompute(test_img, None)

(kp1, des1) = orb.detectAndCompute(test_img, None)
```

```
training_set = ['files/10.jpg','files/20.jpg', 'files/50.jpg', 'files/100.jpg',
'files/500.jpg','files/500.jpg']

for i in range(0, len(training_set)):

        # train image

        train_img = cv2.imread(training_set[i])

        (kp2, des2) = orb.detectAndCompute(train_img, None)

        # brute force matcher

        bf = cv2.BFMatcher()

        all_matches = bf.knnMatch(des1, des2, k=2)

        good = []

        # give an arbitrary number -> 0.789

        # if good -> append to list of good matches

        for (m, n) in all_matches:

                if m.distance < 0.789 * n.distance:

                        good.append([m])


        if len(good) > max_val:

                max_val = len(good)

                max_pt = i

                max_kp = kp2

        print(i, ' ', training_set[i], ' ', len(good))

if max_val != 8:

        print(training_set[max_pt])

        print('good matches ', max_val)

        train_img = cv2.imread(training_set[max_pt])
```

```
        img3 = cv2.drawMatchesKnn(test_img, kp1, train_img, max_kp, good, 4)
        note = str(training_set[max_pt])[6:-4]
        print('\nDetected denomination: Rs. ', note)
        audio_file = 'audio/' + note + '.mp3'
        audio = pyglet.media.load(audio_file,streaming=False)
        audio.play()
        pyglet.clock.schedule_once(lambda dt:pyglet.app.exit() , audio.duration)
        pyglet.app.run()
        # audio_file = "value.mp3
        # tts = gTTS(text=speech_out, lang="en")
        # tts.save(audio_file)
        #return_code = subprocess.call(["afplay", audio_file])
        (plt.imshow(img3), plt.show())
else:
        print('No Matches')
```

App.py :

```
from flask import Flask, request, jsonify
import flask
import werkzeug
import re
# from detect import *
app = Flask(__name__)
@app.route('/', methods=['GET'])
def check():
    return 'OK'
```

```
@app.route('/post', methods=['POST'])
def post_something():
    param = request.form.get('name')
    print(param)
    # You can add the test cases you made in the previous function, but in our case here you are just testing the POST functionality
    if param:
        return jsonify({
            "Message": "Welcome {param} to our awesome platform!!",
            # Add this option to distinct the POST request
            "METHOD": "POST"
        })
    else:
        return jsonify({
            "ERROR": "no name found, please send a name."
        })
@app.route('/image', methods=['POST'])
def handle_request():
    files_ids = list(flask.request.files)
    image_num = 1
    file_name = ""
    for file_id in files_ids:
        imagefile = flask.request.files[file_id]
        filename = werkzeug.utils.secure_filename(imagefile.filename)
        print("Image Filename : " + imagefile.filename)
```

```
    imagefile.save(filename)
    file_name = filename
    image_num = image_num + 1
from detect import helper
note = helper(file_name)
note += ".jpg"
print("Detected note: ", note)
currency = ""
if(re.findall(".*[2][0][0][0].*", note)):
    currency = "2000"
elif(re.findall(".*[2][0][0][^0].*", note)):
    currency = "200"
elif(re.findall(".*[2][0][^0].*", note)):
    currency = "20"
elif(re.findall(".*[1][0][0][^0].*", note)):
    currency = "100"
elif(re.findall(".*[1][0][^0].*", note)):
    currency = "10"
elif(re.findall(".*[5][0][0].*", note)):
    currency = "500"
elif(re.findall(".*[5][0][^0].*", note)):
    currency = "50"
else:
    currency = "-1"
print("Detected Currency: ", currency)
```

```
    if currency != "-1":
        return jsonify({
            "note": currency
        })
    else:
        return jsonify({
            "note": -1
        })
if __name__== "__main__":
    app.run(host="0.0.0.0", port=4555, debug=True)
# from flask import Flask, request
# import flask
# import werkzeug
# import re
# import cv2
# from detect import helper
# import time
# import requests
# app = Flask(__name__)
# @app.route('/', methods=['GET', 'POST'])
# def check():
#     return 'OK'
# @app.route('/image', methods=['POST'])
# def handle_request():
#     camera = cv2.VideoCapture(0)
```

```
#     _, img = camera.read()  # Captures the image from the camera
#     # Write the captured image to disk
#     cv2.imwrite('currency_note.jpg', img)
#     # Release the camera
#     camera.release()
#     # Capture the image from the camera
#     cap = cv2.VideoCapture(0)
#     ret, frame = cap.read()
#     cap.release()
#     # Save the image
#     cv2.imwrite("note.jpg", frame)
#     # Process the image using the detect module
#     note = helper("note.jpg")
#     note += ".jpg"
#     print("Detected note: ", note)
#     currency = ""
#     if(re.findall(".*[2][0][0][0].*", note)):
#         currency = "2000"
#     elif(re.findall(".*[2][0][0][^0].*", note)):
#         currency = "200"
#     elif(re.findall(".*[2][0][^0].*", note)):
#         currency = "20"
#     elif(re.findall(".*[1][0][0][^0].*", note)):
#         currency = "100"
#     elif(re.findall(".*[1][0][^0].*", note)):
```

```
#         currency = "10"
#     elif(re.findall(".*[5][0][0].*", note)):
#         currency = "500"
#     elif(re.findall(".*[5][0][^0].*", note)):
#         currency = "50"
#     else:
#         currency = "-1"
#     print("Detected Currency: ", currency)
#     if currency != "-1":
#         return jsonify({
#             "note": currency
#         })
#     else:
#         return jsonify({
#             "note": -1
#         })
# if __name__== "__main__":
#     app.run(host="0.0.0.0", port=4555, debug=True)
```

CHAPTER 7

TESTING

7.TESTING

7.1 INTRODUCTION

Testing is the major quality control measure employed for software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document which is usually textual and non-textual. After the coding phase, computer programs are available that can be executed for testing purposes. This implies that testing has to uncover errors introduced during coding phases. Thus, the goal of testing is to cover requirement, design, or coding errors in the program. The purpose is to exercise the different parts of the module code to detect coding errors. After this, the modules are gradually integrated into subsystems, which are then integrated themselves to eventually form the entire system. During the module integration, testing is performed. The goal is to detect designing errors, while focusing the interconnecting between the modules. After the system is put together, system testing is performed. Here the system is tested against the system requirements to see if all requirements were met and the system performs as specified by the requirements. Finally, testing is performed to demonstrate to the client for the operation of the system.

For the testing to be successful, proper selection of the test case is essential. There are two different approaches for selecting test cases. The software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called “black box testing”.

The focus here is on testing the external behavior of the system. In structural testing, the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of statements in the code. The two forms of testing are complementary: one tests the external behavior, the other tests the internal structure. Often structural testing is used for lower levels of testing, while functional testing is used for higher levels.

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Frequently the testing process starts with the test plan. This plan identifies all testing related activities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested; different units to be tested, and the manner in which the module will be integrated together. Then for different test units, a test case specification document is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing.

During the testing of the unit the specified test cases are executed and the actual results are compared with the expected outputs. The final output of the testing phase is the testing report and the error report are a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the error encountered and the action taken to remove the error.

Testing approaches

Testing is a process, which reveals the error in a program. It is a major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test case and output is evaluated to determine whether the program is performing as expected. In order to make sure that the system does not have errors, the different levels of testing strategies are applied at differing phases of software development.

7.1.1 Unit Testing

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

7.1.2 Each module can be tested using the following two strategies

7.1.2.1 Black Box Testing

Internal system design is not considered in this type of testing. Tests are based on the requirements and the functionality. This testing is used to find the errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Performance errors
- Initialization and termination errors.

In this testing, only the output is checked for correctness but the logical flow of the data is not checked.

7.1.2.2 White Box Testing

This testing is based on the knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code, statements etc. It is used to generate the test cases in the following cases:

• Guarantee that all the independent paths have been executed.

• Execute all the logical decisions on their true and false sides.

• Execute all the loops at their boundaries and within their operational

• Execute the internal data structures to ensure their validity.

7.1.3 Integration Testing

Integration testing ensures that the software and the subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly or not when integrated together.

7.1.4 System Testing

It involves in-house testing of the entire system before the delivery to the user. Its aim is to satisfy the user and the system that meets all the requirements of the client's specifications.

7.1.5 Acceptance Testing

It is a pre-delivery testing in which the entire system is tested at the client's site on the real-world data to find errors.

7.1.6 Validation Testing

The system is tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

7.1.7 Compiling Test

It was a good idea to do our stress testing early, because it gives us time to fix some of the unexpected exceptions and stability problems that only occur when the components are exposed to very high transaction volumes.

7.1.8 Execution Test

Finally, the program was successfully loaded and executed.

7.1.9 Output Test

The successful output screens are placed in the output screens section.

7.2 Test cases

S.No.	Description	Input	Expected value	Actual value	Result
1.	Adding New Data	Image	Displays the Image of currency notes	Displays the Image of currency notes	Pass
2.	Preprocessing the currency images	Image	Preprocessed The new Data	Removes Noisy Data	Pass
3.	Preprocessing New Data with Trained Data	Image	Input Data is compared with Training Data	New Data is Analyzed	Pass
4.	Feature Extraction of currency Images	Image	Predicting Data	Predicted Data	Pass

Table 7.1 Test Case

CHAPTER 8

RESULTS AND DISCUSSIONS

8. RESULTS AND DISCUSSIONS

These are the sample outputs with respect to different currency notes :

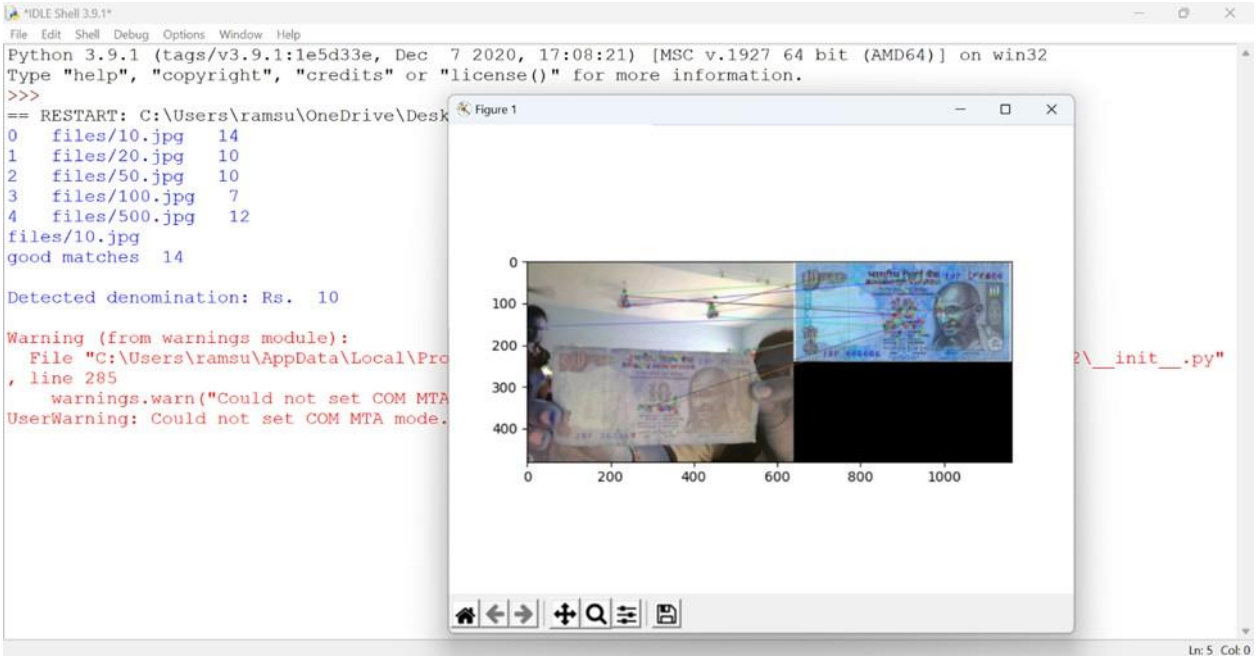


Fig 8.1: Currency Recognition for Rs.10

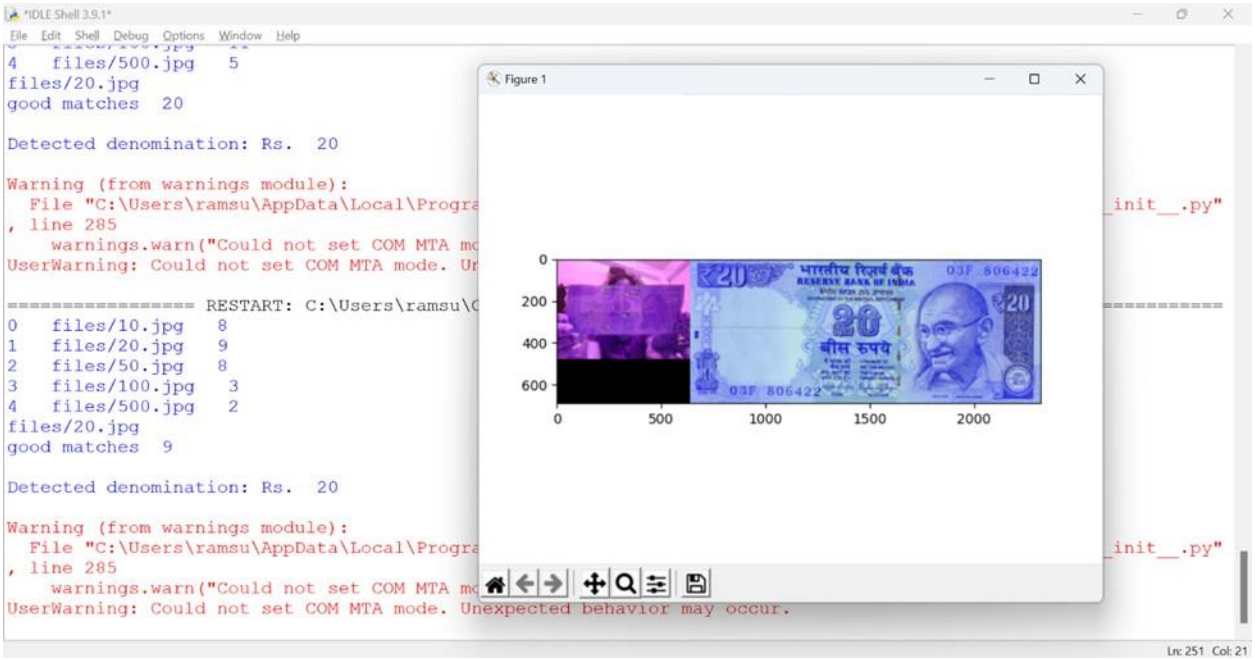


Fig 8.2: Currency Recognition for Rs.20

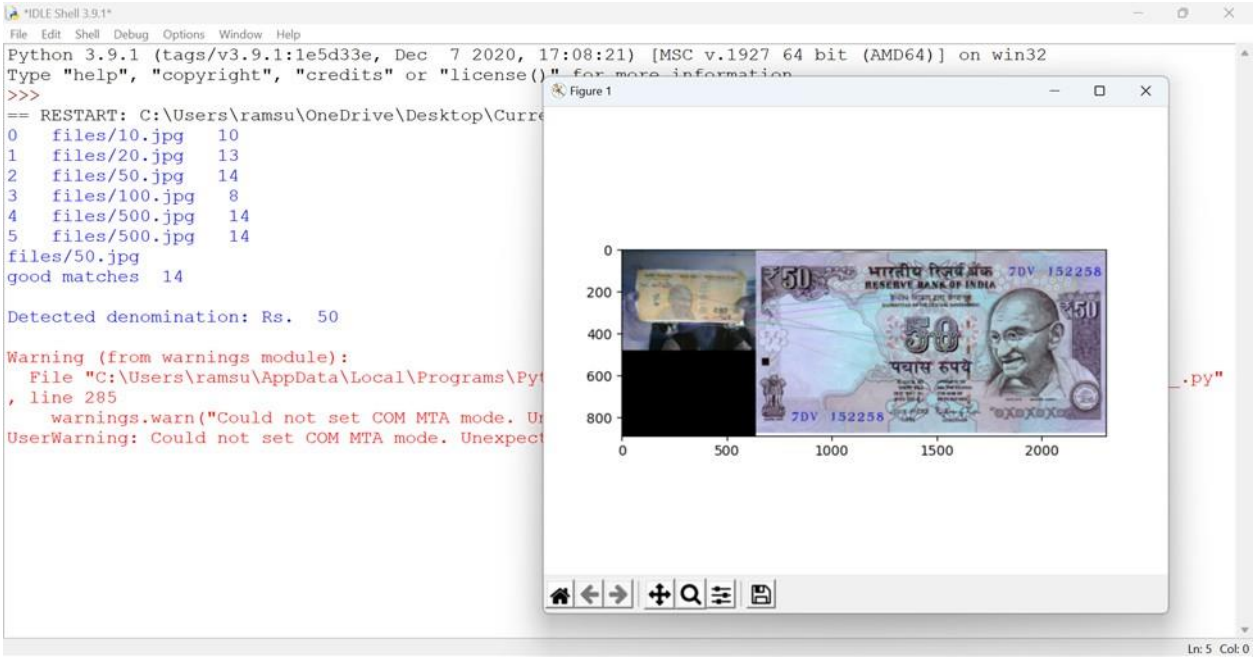


Fig 8.3: Currency Recognition for Rs.50

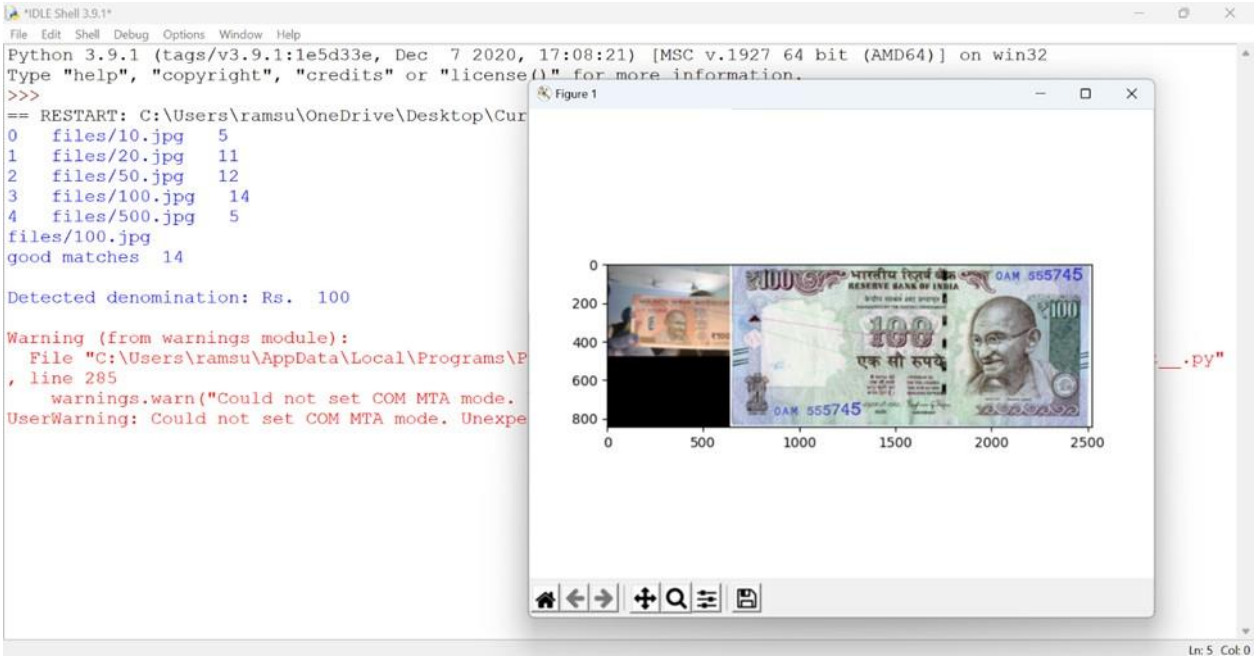


Fig 8.4: Currency Recognition for Rs.100

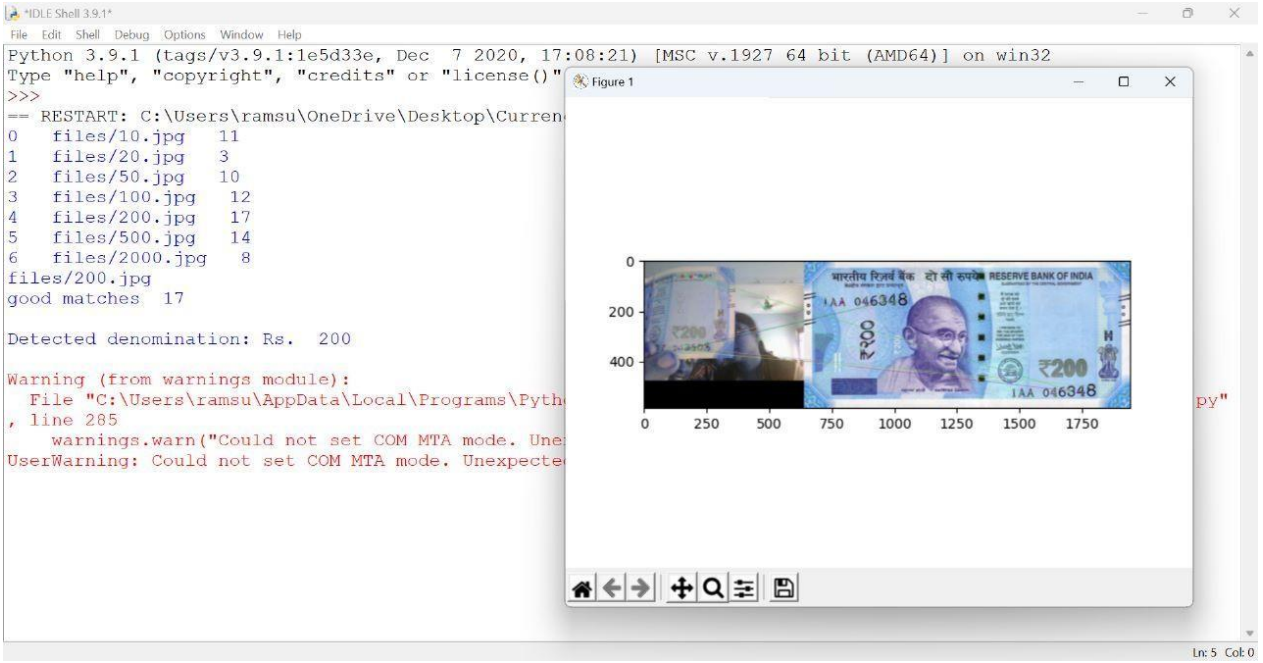


Fig 8.5: Currency Recognition for Rs.200

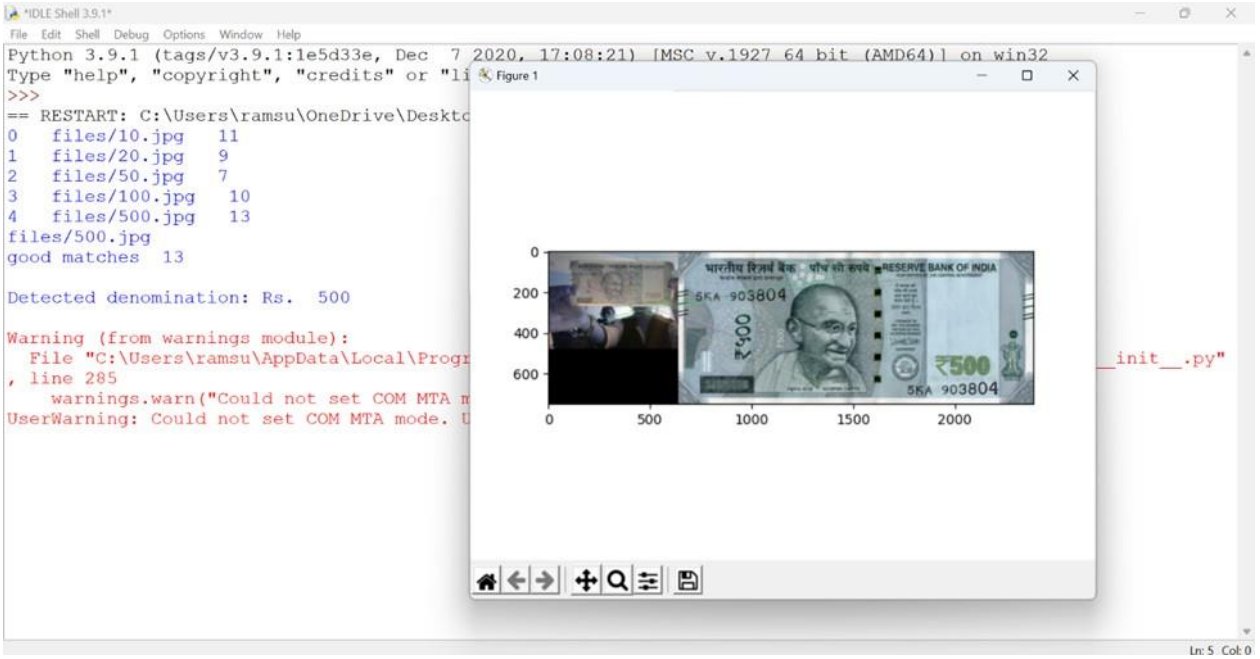


Fig 8.6: Currency Recognition for Rs.500

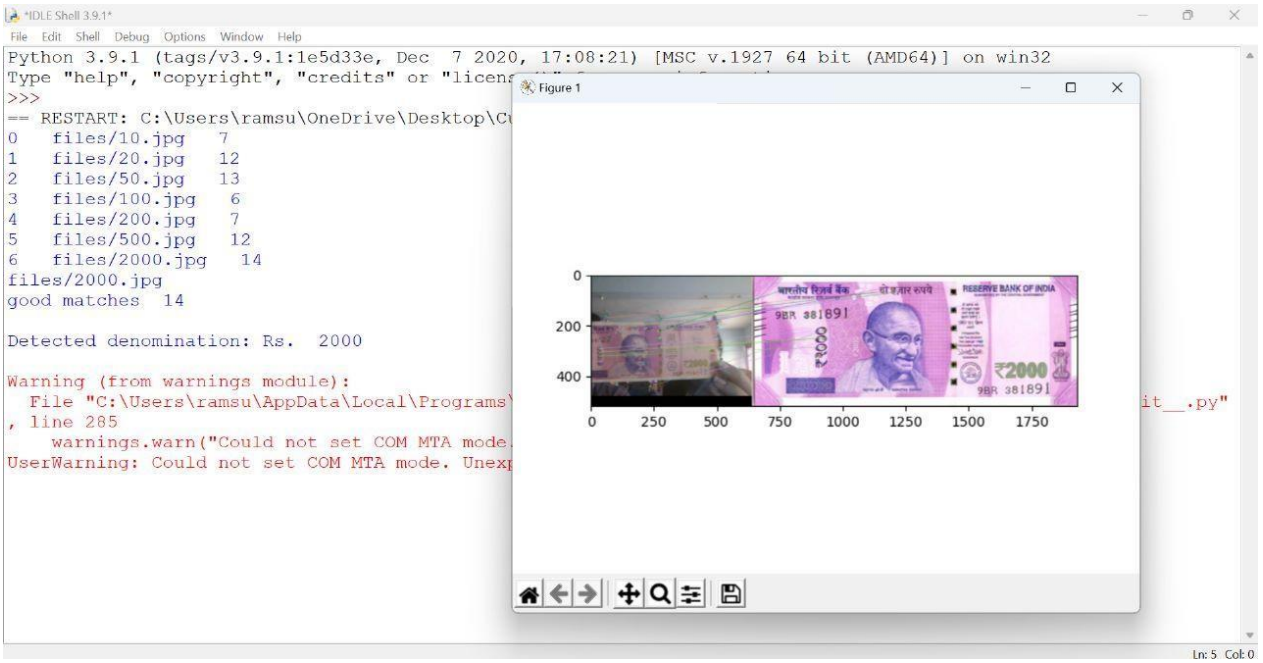


Fig 8.7: Currency Recognition for Rs.2000

CHAPTER 9

CONCLUSION

9. CONCLUSION

We presented a system of currency recognition for the people who are visually challenged. Our system is made to scan the Indian currency and voice out an audio in English. Our system helps them to scan any currency note and based on audio provided they can recognize the value of each banknote.

The system we generated is based on the combination of three algorithms. The ORB algorithm is used to analyze the size and approximate the position of major critical points. Localization of feature points is done by using the SIFT algorithm. The features of the image captured are extracted by using SURF algorithm.

The methods we used will work well on noisy images which are shot with a cell phone. The recommended approach also offers advantages like high performance and simplicity. compared to the present method, more accurate.

In the future we are planning to develop an Android application for this system. We are going to enlarge the audio dataset by adding extra languages like Hindi and Telugu which can increase the users.

Our project with the Course Outcomes CO1 to CO5 has attained, Program Outcomes PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12 and Program Specific Outcomes PSO1, PSO2, PSO3

PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
3	3	2	2	3	2	2	3	3	3	2	2	3	2	2

CHAPTER 10

REFERENCES

10. REFERENCES

1. Dhar,Prashengit, Md Burhan Uddin Chowdhury, and Tonoy Biswas."Paper currency detection system based on combined SURF and LBP features." *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*. IEEE, 2018.

2. Barani, S. (2015, March). Currency identifier for Indian denominations to aid visually impaired. In *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]* (pp. 1-4). IEEE.

3. Solymár, Z., Stubendek, A., Radványi, M., & Karacs, K. (2011, August). Banknote recognition for visually impaired. In *2011 20th European Conference on Circuit Theory and Design (ECCTD)* (pp. 841-844). IEEE.

4. Saeed, N. N., Salem, M. A. M., & Khamis, A. (2013, December). Android-based object recognition for the visually impaired. In *2013 IEEE 20th international conference on electronics, circuits, and systems (ICECS)* (pp. 645-648). IEEE.

5. Bhole, S., & Dhok, A. (2020, March). Deep learning based object detection and recognition framework for the visually-impaired. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 725-728). IEEE.

6. Singh, S., Choudhury, S., Vishal, K., & Jawahar, C. V. (2014, August). Currency recognition on mobile phones. In *2014 22nd International Conference on Pattern Recognition* (pp. 2661-2666). IEEE.

7. Sarker, M. F. R., Raju, M. I. M., Al Marouf, A., Hafiz, R., Hossain, S. A., & Protik, M. H. K. (2019, September). Real-time Bangladeshi Currency Detection System for Visually Impaired Persons. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-4). IEEE.

8. Semary, N. A., Fadl, S. M., Essa, M. S., & Gad, A. F. (2015, December). Currency recognition system for visually impaired: Egyptian banknote as a study case. In *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)* (pp. 1-6). IEEE.

9. Jyothi, C. R., SundaraKrishnan, Y. K., & SrinivasaRao, V. (2016, March). Paper currency recognition for color images based on Artificial Neural Network. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (pp. 3402-3406). IEEE.

10. Kamal, S., Chawla, S. S., Goel, N., & Raman, B. (2015, December). Feature extraction and identification of Indian currency notes. In *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)* (pp. 1-4). IEEE.

11. Abburu, V., Gupta, S., Rimitha, S. R., Mulimani, M., & Koolagudi, S. G. (2017, August). Currency recognition system using image processing. In *2017 Tenth International Conference on Contemporary Computing (IC3)* (pp. 1-6). IEEE.

12. Hassanpour, H., Yaseri, A., & Ardeshiri, G. (2007, February). Feature extraction for paper currency recognition. In *2007 9th International Symposium on Signal Processing and Its Applications* (pp. 1-4). IEEE.

13. Sargano, A., Sarfraz, M., & Haq, N. (2013, May). Robust features and paper currency recognition system. In *Proceedings of the 6th International Conference on Information Technology*.

14. Doush, I. A., & Sahar, A. B. (2017). Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms. *Journal of King Saud University-Computer and Information Sciences*, *29*(4), 484-492.

15. Mirza, R., & Nanda, V. (2012). Paper currency verification system based on characteristic extraction using image processing. *International Journal of Engineering and Advanced Technology (IJEAT)*, *1*(3), 68-71.

16. Yadav, B. P., Patil, C. S., Karhe, R. R., & Patil, P. H. (2014). An automatic recognition of fake Indian paper currency note using MATLAB. *Int. J. Eng. Sci. Innov. Technol*, *3*, 560-566.

17. Aggarwal, H., & Kumar, P. (2012). Indian currency note denomination recognition in color images. *International Journal on Advanced Computer Engineering and Communication Technology*, *1*(1), 12-18.

18. Karacs, K., Lazar, A., Wagner, R., Bálya, D., Roska, T., & Szuhaj, M. (2006, November). Bionic eyeglass: an audio guide for visually impaired. In *2006 IEEE*

Biomedical Circuits and Systems Conference (pp. 190-193). IEEE.

19. Sevkli, M., Turkyilmaz, A., & Aksoy, M. S. (2002, May). Banknote recognition using inductive learning. In *Proceedings of Int. Conf. On Fuzzy Syst. And Soft Computational Intelligence in Management and Industrial Eng-15. FSSCIMIE* (Vol. 2, pp. 122-128).