EMPOWERING VISUALLY IMPAIRED WITH CURRENCY RECOGNITION

A PROJECT REPORT

Submitted by

MUKESH P R [REGISTER NO: 211420243034]

RAHUNATH M [REGISTER NO: 211420243042]

RAVISHANKAR S [REGISTER NO: 211420243043]

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ARITFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

March 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project work **"Empowering Visually Impaired with Currency Recognition"** is the bonafide work of **"**MUKESH P R (211420243034), RAHUNATH M (211420243042), RAVISHANKAR S (211420243043)**"** who carried out the project work under my supervision.

SIGNATURE

Dr. S. MALATHI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT,
DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING COLLEGE,
POONAMALLEE,
CHENNAI-600 123

SIGNATURE

Dr. K. JAYASHREE, M.E., Ph.D.,
PROFESSOR,
DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING COLLEGE,
POONAMALLEE,
CHENNAI-600 123

Certified that the above mentioned students were examined in End Semester project (AD8811) held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENTS

We MUKESH P R (211420243034), RAHUNATH M (211420243042), RAVISHANKAR S (211420243043), hereby declare that this project report titled "Empowering Visually impaired with Currency Recognition", under the guidance of **Dr. K. JAYASHREE, M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D.** and **Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the AI&DS Department, **Dr. S. MALATHI, M.E., Ph.D.,** for thesupport extended throughout the project.

We would like to thank our supervisor **Dr. K. JAYASHREE M.E., Ph.D.,** coordinators **Dr. K. JAYASHREE & Dr. P. KAVITHA** and all the faculty members of the Department of AI&DS for their advice and encouragement for the successful completion of the project.

1. **MUKESH P R**

2. RAHUNATH M

3. RAVISHANKAR S

ABSTRACT

This project work focuses on the implementation of Open Computer Vision technology for the recognition of Indian currency. With the increasing prevalence of computer vision applications, the research explores the potential of Open Computer Vision in automating the process of currency identification. The system employs image processing techniques to analyze and recognize various denominations of Indian currency notes, contributing to the development of efficient and reliable currency recognition systems. The investigation includes the pre-processing of currency images, feature extraction, and the application of machine learning algorithms within the Open Computer Vision framework to enhance accuracy and speed in currency identification. The findings of this research hold significance for the automation of tasks such as currency sorting, counterfeit detection, and other financial transactions, thereby contributing to advancements in the field of computer vision and financial technology.

TABLE OF CONTENT

LIST OF FIGURES

SECTION-I

1. INTRODUCTION:

In the realm of accessibility and inclusivity, technological advancements continue to bridge gaps and empower individuals with diverse abilities. Among those often faced with challenges in everyday transactions are individuals with visual impairments, who encounter obstacles in independently managing currency. However, emerging technologies offer promising solutions to this longstanding issue. By harnessing the power of currency recognition systems, visually impaired individuals can gain autonomy in handling money, fostering greater financial independence and confidence in their daily lives. This introduction explores the transformative potential of currency recognition technology in empowering the visually impaired, highlighting its significance in enhancing accessibility and promoting equality in financial interactions.

1.1. RELATED WORK:

Egocentric vision (a.k.a. first-person vision–FPV) applications have thrived over the past few years, thanks to the availability of affordable wearable cameras and large annotated datasets. The position of the wearable camera (usually mounted on the head) allows recording exactly what the camera wearers have in front of them, in particular hands and manipulated objects. This intrinsic advantage enables the study of the hands from multiple perspectives: localizing hands and their parts within the images; understanding what actions and activities the hands are involved in; and developing human-computer interfaces that rely on hand gestures. In this survey, we review the literature that focuses on the hands using egocentric vision, categorizing the existing approaches into: localization (where are the hands or parts of them?); interpretation (what are the hands doing?); and application (e.g., systems that

used egocentric hand cues for solving a specific problem). Moreover, a list of the most prominent datasets with hand-based annotations is provided.

1.1.1. Disadvantages:

- Limited scope of use.
- They focused on only hands recognition.
- Limited scalability.
- They did not build a classification model.

1.1.2. DESCRIPTION:

The Indian currency recognition system using OpenCV (Open Source Computer Vision) technology aims to facilitate automated identification and classification of Indian banknotes through image processing techniques. Leveraging the power of OpenCV, the system employs advanced algorithms to extract key features from currency images, such as color, texture, and patterns. Through the application of image segmentation and object recognition methodologies, the system can accurately distinguish between different denominations of Indian currency notes. This technology holds significant potential for applications in various sectors, including banking, retail, and automated teller machines (ATMs), streamlining the currency handling process and enhancing overall efficiency. By harnessing the capabilities of OpenCV, this system contributes to the development of robust and reliable solutions for real-time currency recognition, addressing the growing need for automation in financial transactions and related domains.

1.1.3. Data Science

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

1.1.4. ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals. Leading AI textbooks define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers.

Machine Learning

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are

supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labelling to learn data has to be labelled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Proposed system

The proposed system for empowering visually impaired individuals with currency recognition technology entails a mobile application interface equipped with accessibility features, integrating smartphone cameras to capture images of banknotes. Advanced image processing algorithms and machine learning models, such as convolutional neural networks (CNNs), analyze these images to identify currency denominations accurately in real-time. Users receive immediate auditory and tactile feedback regarding the recognized denomination, with offline functionality ensuring usability in areas with limited internet connectivity. Continuous improvement mechanisms, including user feedback loops, are integrated to refine the system's performance over time, ultimately fostering financial independence and inclusion for visually impaired individuals.

Deep learning is a subfield of artificial intelligence (AI) that focuses on the development and implementation of artificial neural networks to simulate the way the human brain works. These neural networks, inspired by the structure and functioning of the human brain, consist of interconnected layers of nodes, or artificial neurons. Deep learning algorithms can automatically learn and adapt from vast amounts of data, allowing them to perform tasks such as image and speech recognition, natural language processing, and even complex decision-making.

2. LITERATURE SURVEY

General

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period.

Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them. Loan default trends have been long studied from a socio-economic stand point.

Most economics surveys believe in empirical modeling of these complex systems in order to be able to predict the loan default rate for a particular individual. The use of machine learning for such tasks is a trend which it is observing now. Some of the survey's to understand the past and present perspective of loan approval or not.

Review of Literature Survey

Title : CURRECNCY RECOGNIZEION SYSTEM USING IMAGES PREPROCESSING

Author : SANDEEP KUMAR CHAUBEY

Year : 2022

It is difficult for people to recognize currencies from different countries. Our aim is to help people solve this problem. However, currency recognition systems that are based on image analysis entirely are not sufficient. Our system is based on image processing and makes the process automatic and robust. We use and Indian Rupees as examples to illustrate the technique. Color and shape information are used in our algorithm.

Title : INDIAN CURRENCY DETECTION SYSTEM FOR BLIND PEOPLE

Author: Mhaske V.D, Sanika Bagade, Pratik Dhaygude, Aditi Lande, Harshwardhan Nikam

Year : 2024

This project addresses the challenges faced by India's visually impaired population (18 million people) in managing currency independently. We aim to develop an accessible Android application using advanced image recognition to confidently identify Indian currency notes. The system ensures 99 percent accuracy in currency recognition and 90 percent success in counterfeit detection. Beyond India, the vision is to provide global currency recognition for financial independence. The proposed solution incorporates an emergency alert feature for prompt assistance and location sharing. The current lack of universally accessible solutions makes quick and efficient transactions challenging for blind individuals. Our innovation strives to bridge this gap by offering an efficient, user-friendly solution through an Android application, enabling visually impaired individuals to manage their money effortlessly

Title : Indian Currency Denomination Recognition and Fake Currency Identification

Author : Padmaja , P Bhargav Naga Shyam , H Ganga Sagar , B Diwakar Nayak and MBhushan Rao

Year : 2021

Visually impaired and senior citizens find it difficult to identify different banknotes, driving the need for an automated system to recognize currency notes. This study proposes recognizing Indian currency notes of various denominations using Deep Learning through the CNN model. While not recognizing currency notes is one issue, identifying fake notes is another major issue. Currency counterfeiting is the illegal imitation of currency to deceive its recipient. The current existing methodologies for identifying a phony note rely on hardware. A method completely devoid of hardware that relies on specific security features to help distinguish a legitimate currency note from an illegitimate one is much needed. These features are extracted using the boundary box region of interest (ROI) and Canny Edge detection in OpenCV implemented in Python, and the multi scale template matching algorithm is applied to match the security features and differentiate fake notes fromlegitimatenotes.

Title : An Android Application for Indian Currency Recognition and Detection for Visually Impaired People

Author : Dhananjay V. Pawar, Mansi N. Sharma, Sachin N. Pisar

Year : 2023

The currency recognition system using TensorFlow is an innovative application of deep learning technology that can recognize and classify different currencies from around the world. This system uses a convolutional neural

network (CNN) to extract meaningful features from banknote images, and then a classifier to predict the currency type. The training of the system is performed on a large dataset of banknote images, and the system achieves high accuracy in recognizing different currencies. The system has a wide range of potential applications for visually impaired people, including ATM machines, vending machines, and self-service kiosks, where accurate and efficient currency recognition is essential. The implementation of the system using TensorFlow provides a scalable and efficient solution that can be easily deployed on various platforms.The currency recognition system using TensorFlow is an exciting and valuable application of deep learning technology. It has the potential to revolutionize the way we handle cash transactions and improve the overall efficiency of financial systems. This abstract provides an overview of the key features and potential applications of the system, highlighting its potential to benefit a wide range of industries and users.

Title : INDIAN FAKE CURRENCY DETECTION USING COMPUTER VISION

Author : Devid Kumar, Surendra Chauhan

Year : 2020

This paper developed a computer vision based approach for Indian paper currency detection. In this approach, extract currency feature and develop an own datasets used for the currency detection. By using feature extraction method of front and back side Rs. 200 denomination security feature of Indian currency note. The mainly use ORB (Oriented FAST and Rotated BRIEF) and Brute-Force matcher approach to extract the feature of paper currency, so that can more accurately detection the denomination of the banknote both obverse and reverse. Our main contribution is through using ORB and BF matcher in

OpenCV based, the average accuracy of detection is up to 95.0% and tested this method on different denominations of Indian banknote.

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period.

Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

3. IMPLEMENTATIONS AND DETAILS

Implementations of currency recognition systems for visually impaired individuals typically involve the integration of image processing techniques and machine learning algorithms into user-friendly mobile applications. These applications leverage smartphone cameras to capture images of banknotes, which are then analyzed using advanced algorithms, such as convolutional neural networks (CNNs), to accurately identify currency denominations in real-time. The system provides immediate feedback to users through auditory and tactile cues, with offline functionality ensuring accessibility even in areas with limited internet connectivity. Continuous refinement mechanisms, including user feedback loops, are incorporated to enhance the system's accuracy and usability over time, ultimately empowering visually impaired individuals with increased independence in financial transactions.

3.1. PREPARING THE DATASET:

This dataset contains lot of train and test image records of features extracted of Rupies images, which were then classified into classes for computer vision.

We proposed a system to develop the project using Deep learning algorithm. Recently, Deep learning and Artificial intelligence has plays a big role in various industries for their improvement and development. So we tried to implement Deep learning algorithm to diagnosis the thyroid. We collected the previous record of patient who had the thyroid and who doesn't had the disease and those who had symptoms. By collection those peoples information our machine is tried to identifies the pattern of the datasets by various performing calculations. After identifies the pattern in image using various deep learning algorithm the model can able to predict the instance based on previous information's. If you image related to his has given to input then the algorithm can tell whether it is affected or not.

3.1.2. Advantages:

- We use a deep learning approach to build a classification model.
- We recognize Indian currency
- High scalability.
- Exponential scope.
- We build a classification model

3.2. SYSTEM STUDY

Aim:

The aim of the project is to develop an efficient Indian currency recognition system using OpenCV (Open Source Computer Vision) technology. This system aims to automate the process of identifying and classifying various

denominations of Indian currency notes through image processing and computer vision techniques. By leveraging advanced algorithms and machine learning models, the project seeks to enhance accuracy and speed in currency recognition, facilitating applications in sectors such as automated teller machines (ATMs), cash handling systems, and financial transactions. This initiative not only streamlines currency management processes but also contributes to the overall modernization of financial systems by integrating cutting-edge technology for enhanced efficiency and reliability in currency recognition.

Objectives:

The objective of the project "Indian Currency Recognition using OpenCV Technology" is to develop a robust and efficient system that can automatically identify and classify Indian currency notes through the implementation of Open Computer Vision (OpenCV) technology. This project aims to enhance accessibility for visually impaired individuals and streamline currency verification processes. By leveraging advanced image processing techniques, the system will be designed to accurately recognize and differentiate various denominations of Indian currency notes, contributing to financial inclusivity and facilitating seamless transactions in the digital era.

3.3. OUTLINE OF THE PROJECT

The proposed project aims to address provide a brief overview of the problem or domain], recognizing its significance in [mention the broader context]. The primary objectives of this undertaking are to [clearly define the project goals and specific outcomes], with a focus on benefiting [identify the target stakeholders and beneficiaries]. The methodology encompasses a meticulous approach to data collection, involving [describe the data sources and types], and the utilization of [mention methods for data acquisition, cleaning,

and pre-processing. The technology and tools chosen for implementation include [specify programming languages, frameworks, and tools], selected based on their suitability for [provide rationale]. The heart of the project lies in the model architecture, which comprises [outline the proposed model or algorithm] and details its neural network structure or statistical methods involved.

4. SYSTEM ANALYSIS

The proposed system for empowering visually impaired individuals with currency recognition technology entails a mobile application interface equipped with accessibility features, integrating smartphone cameras to capture images of banknotes. Advanced image processing algorithms and machine learning models, such as convolutional neural networks (CNNs), analyze these images to identify currency denominations accurately in real-time. Users receive immediate auditory and tactile feedback regarding the recognized denomination, with offline functionality ensuring usability in areas with limited internet connectivity. Continuous improvement mechanisms, including user feedback loops, are integrated to refine the system's performance over time, ultimately fostering financial independence and inclusion for visually impaired individuals.

4.1. OVERVIEW OF THE SYSTEM

The system built on deep learning principles represents a sophisticated approach to solving complex problems in various domains. At its core, deep learning involves the utilization of artificial neural networks, inspired by the structure and functioning of the human brain, to automatically learn and extract meaningful patterns from vast amounts of data. The primary objective of this deep learning system is to address [specify the problem or challenge it aims to tackle], leveraging the power of neural networks to achieve high-level abstraction and representation.

4.1.1. Project Requirements

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
    A. Hardware requirements
    B. software requirements

Functional requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

4.2. Environmental Requirements:

1. Software Requirements:

Operating System : Windows

Tool : Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor : Intel core i3

Hard disk : minimum 300 GB

RAM : minimum 4 GB

SOFTWARE DESCRIPTION

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system "Conda". The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

PYCHARM:

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. Code faster and with more easily in a smart and configurable editor with code completion, snippets, code folding and split windows support.

PyCharm Features

- **Intelligent Coding Assistance** – PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactorings and rich navigation capabilities.
- **Intelligent Code Editor** – PyCharm's smart code editor provides first-class support for Python, JavaScript, CoffeeScript, TypeScript, CSS, popular template languages and more. Take advantage of language-aware code completion, error detection, and on-the-fly code fixes!
- **Smart Code Navigation** – Use smart search to jump to any class, file or symbol, or even any IDE action or tool window. It only takes one click to switch to the declaration, super method, test, usages, implementation, and more.
- **Fast and Safe Refactorings** – Refactor your code the intelligent way, with safe Rename and Delete, Extract Method, Introduce Variable, Inline Variable or Method, and other refactorings. Language and framework-specific refactorings help you perform project-wide changes.
- **Built-in Developer Tools** – PyCharm's huge collection of tools out of the box includes an integrated debugger and test runner; Python profiler; a built-in terminal; integration with major VCS and built-in database

tools; remote development capabilities with remote interpreters; an integrated ssh terminal; and integration with Docker and Vagrant.

- **Debugging, Testing and Profiling** – Use the powerful debugger with a graphical UI for Python and JavaScript. Create and run your tests with coding assistance and a GUI-based test runner. Take full control of your code with Python Profiler integration.
- **VCS, Deployment and Remote Development** – Save time with a unified UI for working with Git, SVN, Mercurial or other version control systems. Run and debug your application on remote machines. Easily configure automatic deployment to a remote host or VM and manage your infrastructure with Vagrant and Docker.
- **Database tools** – Access Oracle, SQL Server, PostgreSQL, MySQL and other databases right from the IDE. Rely on PyCharm's help when editing SQL code, running queries, browsing data, and altering schemas.
- **Web Development** – In addition to Python, PyCharm provides first-class support for various Python web development frameworks, specific template languages, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js, and more.
- **Python Web frameworks** – PyCharm offers great framework-specific support for modern web development frameworks such as Django, Flask, Google App Engine, Pyramid, and web2py, including Django templates debugger, manage.py and appcfg.py tools, special auto completion and navigation, just to name a few.
- **JavaScript & HTML –** PyCharm provides first-class support for JavaScript, Coffee Script, Type Script, HTML and CSS, as well as their modern successors. The JavaScript debugger is included in PyCharm and is integrated with the Django server run configuration.
- **Live Edit** – Live Editing Preview lets you open a page in the editor and the browser and see the changes being made in code instantly in the

browser. PyCharm auto-saves your changes, and the browser smartly updates the page on the fly, showing your edits.

- **Scientific Tools** – PyCharm integrates with Ipython Notebook, has an interactive Python console, and supports Anaconda as well as multiple scientific packages including Matplotlib and NumPy.
- **Interactive Python console** – You can run a REPL Python console in PyCharm which offers many advantages over the standard one: on-the-fly syntax check with inspections, braces and quotes matching, and of course code completion.
- **Scientific Stack Support** – PyCharm has built-in support for scientific libraries. It supports Pandas, Numpy, Matplotlib, and other scientific libraries, offering you best-in-class code intelligence, graphs, array viewers and much more.
- **Conda Integration** – Keep your dependencies isolated by having separate Conda environments per project, PyCharm makes it easy for you to create and select the right environment.
- **Customizable and Cross-platform IDE** – Use PyCharm on Windows, Mac OS and Linux with a single license key. Enjoy a fine-tuned workspace with customizable color schemes and key-bindings, with VIM emulation available.
- **Customizable UI** – Enjoy a fine-tuned workspace with customizable color schemes and key-bindings.
- **Plugins** – More than 10 years of IntelliJ platform development gives PyCharm 50+ IDE plugins of different nature, including support for additional VCS, integrations with different tools and frameworks, and editor enhancements such as Vim emulation.
- **Cross-platform IDE** – PyCharm works on Windows, Mac OS or Linux. You can install and run PyCharm on as many machines as you have, and use the same environment and functionality across all your machines.

METHODOLOGY

Pre-processing and Training the model (CNN): The dataset is pre-processed such as Image reshaping, resizing and conversion to an array form. Similar processing is also done on the test image. A dataset consisting of about 4 different knee osteoarthritis is obtained, out of which any image can be used as a test image for the software.

CNN Model steps:

Conv2d:

The 2D convolution is a fairly simple operation at heart: you start with a kernel, which is simply a small matrix of weights. This kernel "slides" over the 2D input data, performing an elementwise multiplication with the part of the input it is currently on, and then summing up the results into a single output pixel.

MaxPooling2D layer

Down samples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool_size) for each channel of the input. The window is shifted by strides along each dimension.

ARCHITECTURE OF CNN

CONVOLUTIONAL NEURAL NETWORK:

A Convolutional neural network (CNN) is one type of Artificial Neural Network. A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data.

TYPES OF CNN

1. 1 VGG-16

2 LENET

FEASIBILITY STUDY

Splitting the dataset:

The data use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. It has the test dataset (or subset) in order to test our models and it will do this using Tensorflow library in Python using the Keras method.

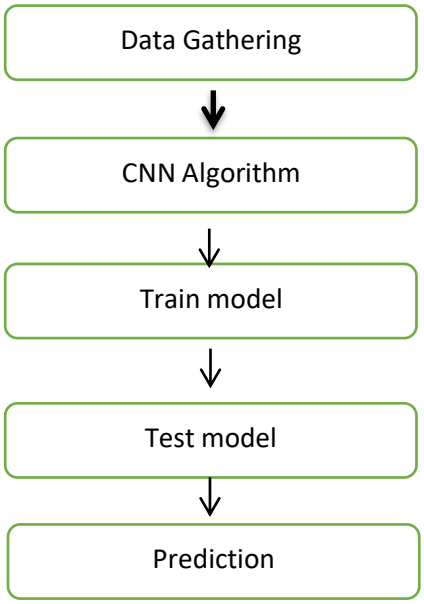


Fig1. Steps of dataflow diagram

DESIGN ARCHITECTURE

Design is meaningful engineering representation of something that is to be built. Software design is a process design is the perfect way to accurately translate requirements in to a finished software product. Design creates a representation or model, provides detail about software data structure,

architecture, interfaces and components that are necessary to implement a system.

System Architecture:

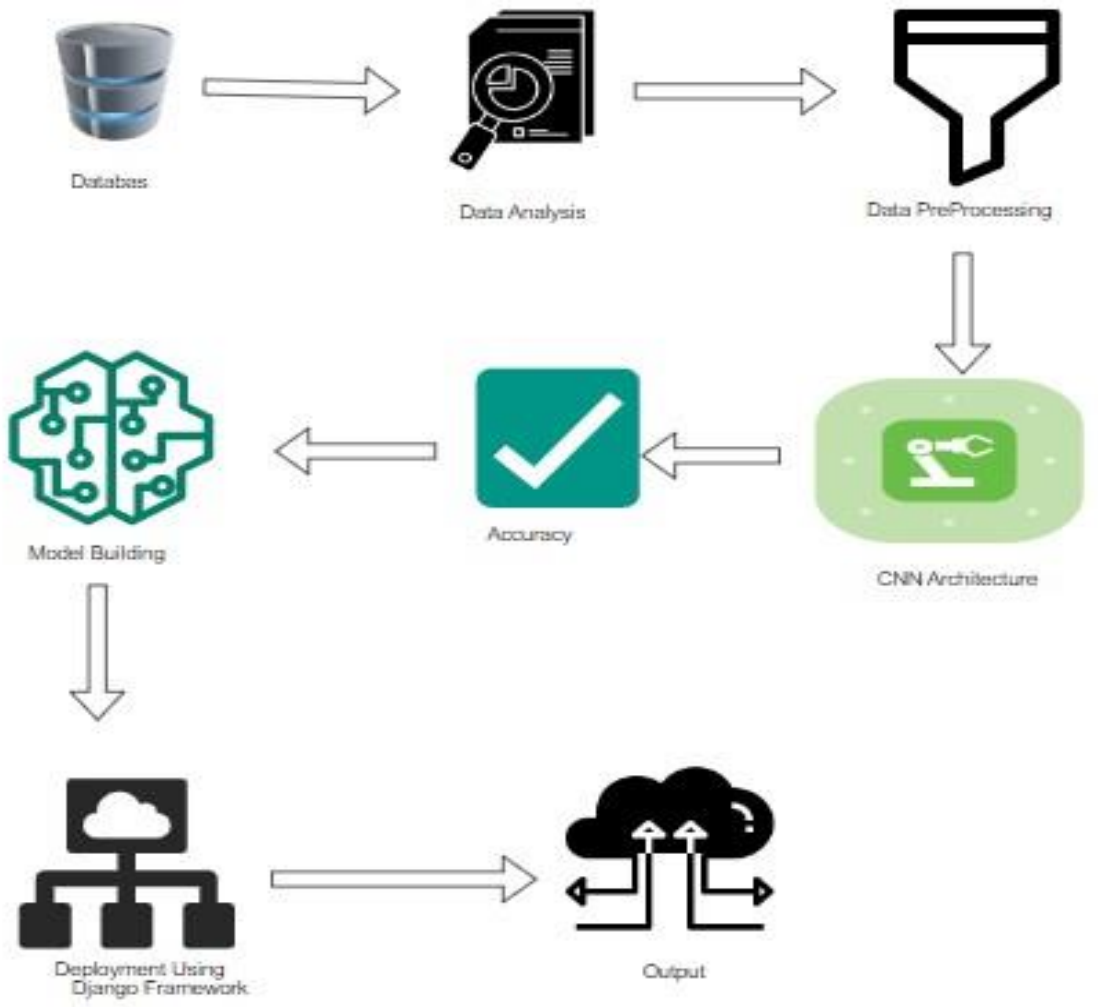


Fig2. System Architecture

The above diagram proposed a system to develop the project using Deep learning algorithm. Recently, Deep learning and Artificial intelligence has plays a big role in various industries for their improvement and development. So we tried to implement Deep learning algorithm to diagnosis the thyroid. We collected the previous record of patient who had the thyroid and who doesn't had the disease and those who had symptoms. By collection those peoples

information our machine is tried to identifies the pattern of the datasets by various performing calculations. After identifies the pattern in image using various deep learning algorithm the model can able to predict the instance based on previous information's. If you image related to his has given to input then the algorithm can tell whether it is affected or not.

Work flow diagram:

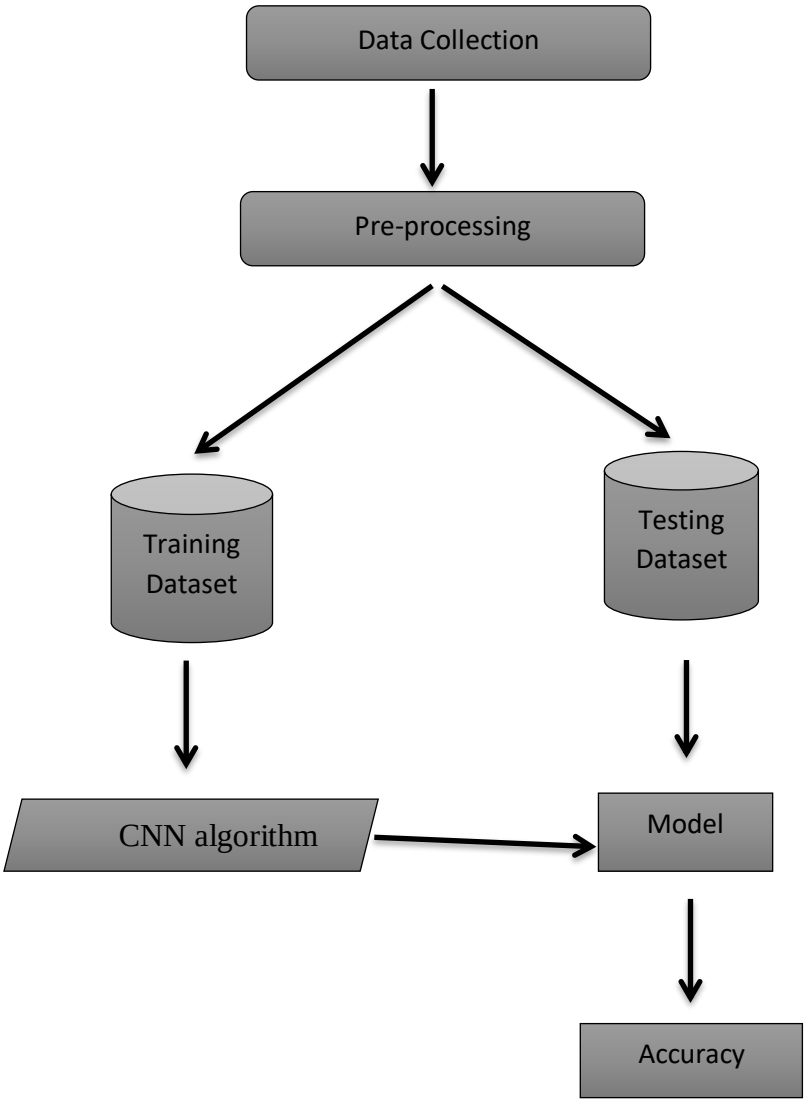


Fig 3. Work Flow

The system's architecture is characterized by the presence of multiple hidden layers in its neural networks, hence the term "deep" learning. This depth allows the system to learn hierarchical and intricate features directly from raw data, eliminating the need for manual feature engineering. The choice of deep neural networks, such as Convolutional Neural Networks (CNNs) for image-related tasks or Recurrent Neural Networks (RNNs) for sequential data, underscores the versatility of the system across different applications.

In terms of implementation, the system involves meticulous data collection, preprocessing, and the use of cutting-edge technologies such as graphics processing units (GPUs) or specialized processing units like Tensor Processing Units (TPUs) to handle the computational demands of training deep models. The model's performance is continuously refined through rigorous testing, validation, and fine-tuning processes.

USECASE DIAGRAM:

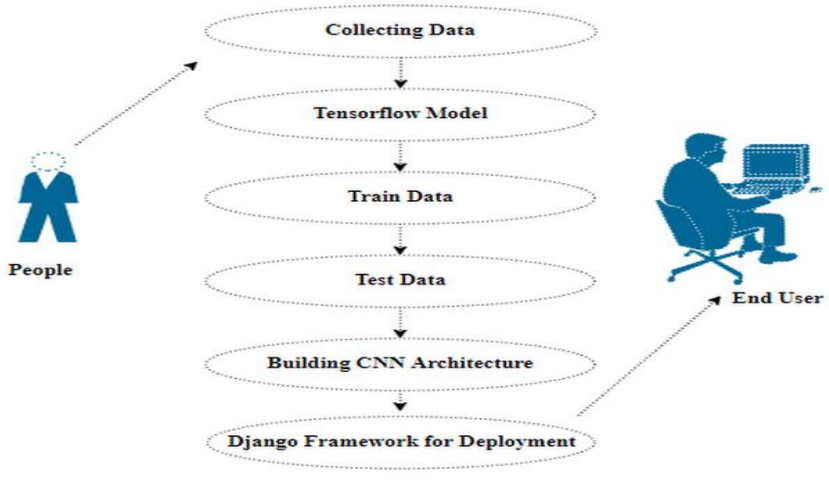


Fig 4. Usecase diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities

are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

Working Process:

- Download and install anaconda and get the most useful package for machine learning in Python.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- Machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming.

The best way to get started using Python for machine learning is to complete a project.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird's eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well-known steps:

- Define Problem.

- Prepare Data.
- Evaluate Algorithms.
- Improve Results.
- Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.

CLASS DIAGRAM

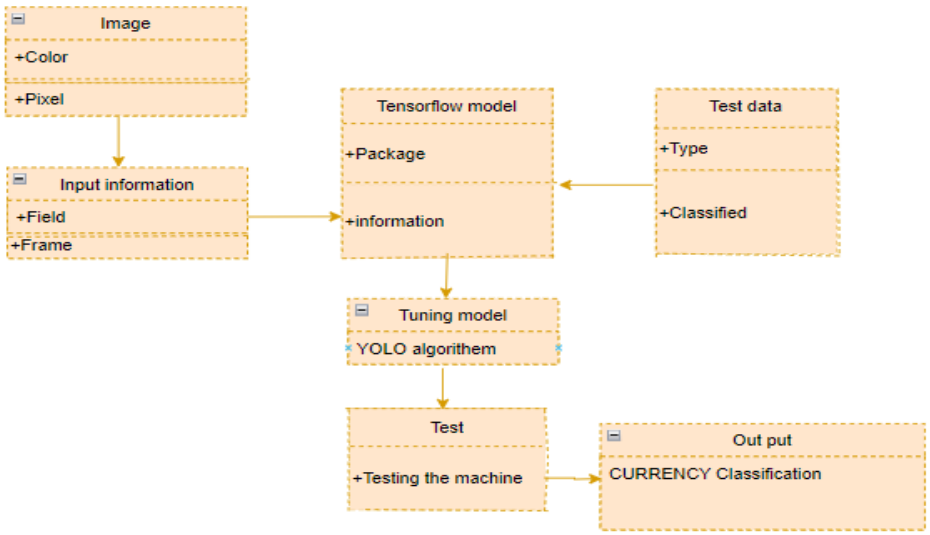


Fig 5. Class diagram

A class diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model. Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top down approach to Systems Design. Symbols and Notations Used in DFDs Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.

Data store: files or repositories that hold information for later use, such as a database table or a membership form.

Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

DFD levels and layers A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish. DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

ACTIVITY DIAGRAM

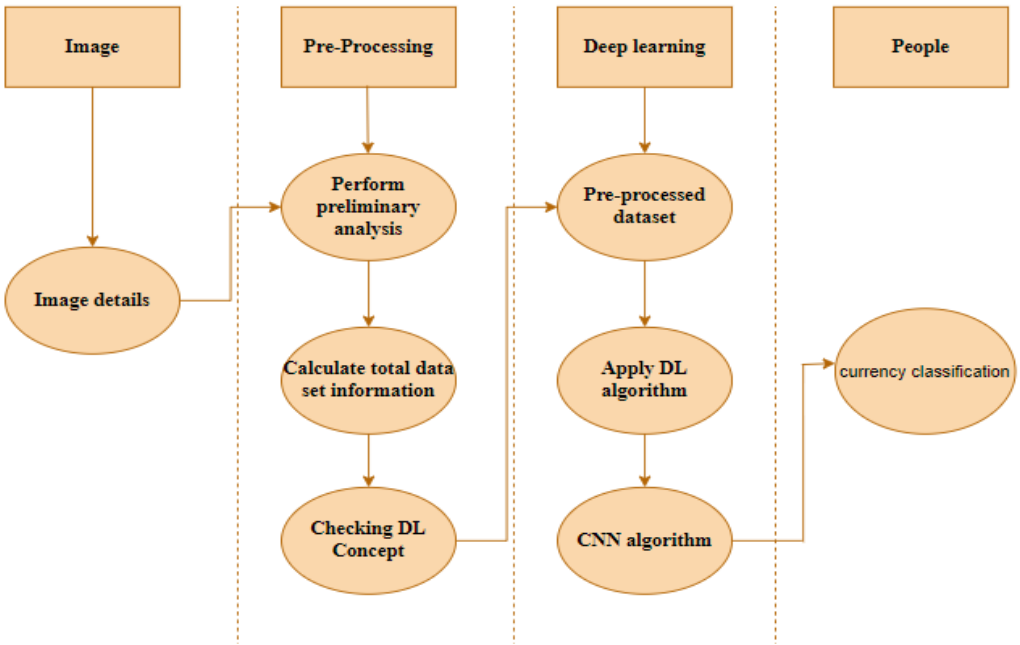


Fig 6. Activity diagram

A activity diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model. Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top down approach to Systems Design. Symbols and Notations Used in DFDs Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.

Data store: files or repositories that hold information for later use, such as a database table or a membership form.

Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

DFD levels and layers A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish. DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

SEQUENCE DIAGRAM

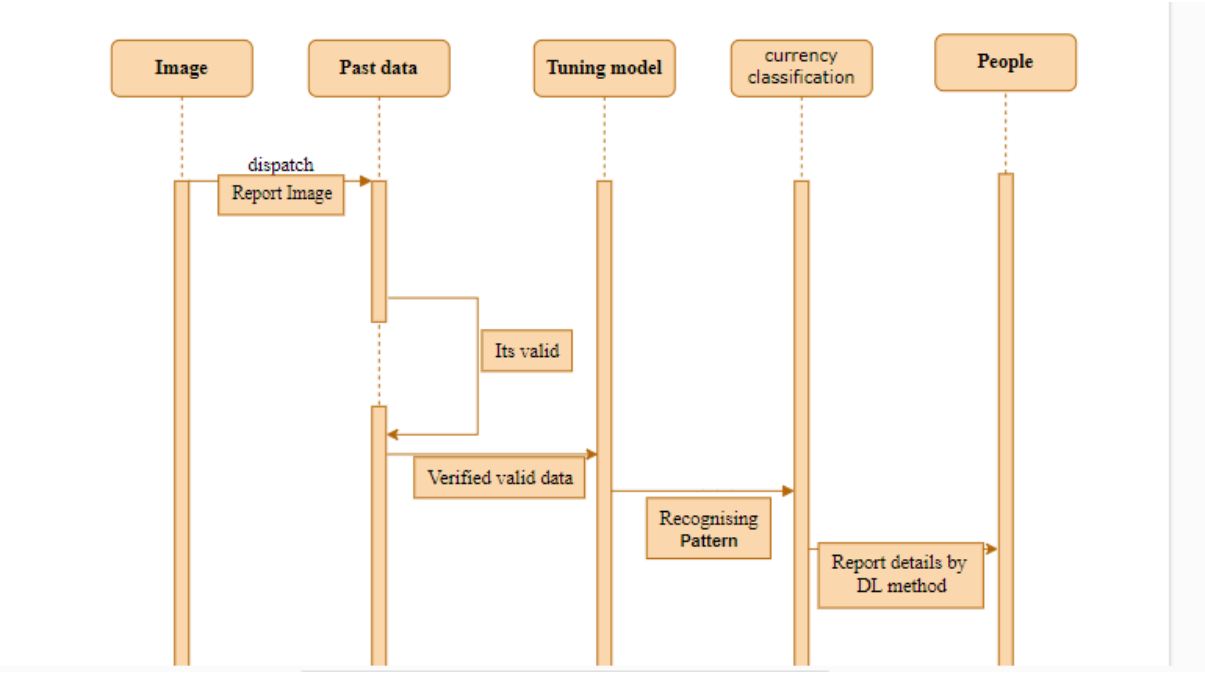


Fig 7. Sequence diagram

A sequence diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model. Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top down approach to Systems Design. Symbols and Notations Used in DFDs Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.

Data store: files or repositories that hold information for later use, such as a database table or a membership form.

Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

DFD levels and layers A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish. DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

5. MODULES

1. Manual Net
2. ALEX NET
3. LENETS
4. Deploy

5.1. MODULE DESCRIPTION

IMPORT THE GIVEN IMAGE FROM DATASET:

We have to import our data set using keras preprocessing image data generator function also we create size, rescale, range, zoom range, horizontal flip. Then we import our image dataset from folder through the data generator function. Here we set train, test, and validation also we set target size, batch size

and class-mode from this function we have to train using our own created network by adding layers of CNN.

TO TRAIN THE MODULE BY GIVEN IMAGE DATASET:

To train our dataset using classifier and fit generator function also we make training steps per epoch's then total number of epochs, validation data and validation steps using this data we can train our dataset.

DEPLOY

In this module the trained deep learning model is converted into hierarchical data format file (.h5 file) which is then deployed in our django framework for providing better user interface and predicting the output.

Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text

accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

CSS

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users.

CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language.

CSS Syntax

```
Selector {
        Property 1 : value;
        Property 2 : value;
        Property 3 : value;
    }
```

For example:

```
h1
  {
    Color: red;
     Text-align: center;
  }
#unique
 {
```

```
    color: green;
}
```

- Selector: selects the element you want to target
- Always remains the same whether we apply internal or external styling
- There are few basic selectors like tags, id's, and classes
- All forms this key-value pair
- Keys: properties(attributes) like color, font-size, background, width, height,etc
- Value: values associated with these properties

CSS Comment

- Comments don't render on the browser
- Helps to understand our code better and makes it readable.
- Helps to debug our code
- Two ways to comment:
    - Single line

CSS How-To

- There are 3 ways to write CSS in our HTML file.
    - Inline CSS
    - Internal CSS
    - External CSS

- Priority order

Inline > Internal > External

Inline CSS

- Before CSS this was the only way to apply styles
- Not an efficient way to write as it has a lot of redundancy
- Self-contained
- Uniquely applied on each element
- The idea of separation of concerns was lost
- Example:

```
<h3 style=" color:red"> Have a great day </h3>

<p  style =" color: green"> I did this , I did that </p>
```

Internal CSS

- With the help of style tag, we can apply styles within the HTML file
- Redundancy is removed
- But the idea of separation of concerns still lost
- Uniquely applied on a single document
- Example:

```
 < style>

        h1{

             color:red;

           }
```

```
</style>

<h3> Have a great day </h3>
```

External CSS

- With the help of <link> tag in the head tag, we can apply styles
- Reference is added
- File saved with .css extension
- Redundancy is removed
- The idea of separation of concerns is maintained
- Uniquely applied to each document
- Example:

```
<head>

<link rel=”stylesheet” type=”text/css” href=”name of the Css file”>

</head>

        h1{

                color:red;        //.css file

            }
```

CSS Selectors

- The selector is used to target elements and apply CSS
- Three simple selectors
    - Element Selector

    - Id Selector
    - Class Selector

- Priority of Selectors

CSS Colors

- There are different colouring schemes in CSS
- **RGB**-This starts with RGB and takes 3 parameter
- **HEX**-Hex code starts with # and comprises of 6 numbers which are further divided into 3 sets
- **RGBA**-This starts with RGB and takes 4 parameter

CSS Background

- There are different ways by which CSS can have an effect on HTML elements
- Few of them are as follows:
    - Color – used to set the color of the background
    - Repeat – used to determine if the image has to repeat or not and if it is repeating then how it should do that
    - Image – used to set an image as the background
    - Position – used to determine the position of the image
    - Attachment – It basically helps in controlling the mechanism of scrolling

CSS BoxModel

- Every element in CSS can be represented using the BOX model
- It allows us to add a border and define space between the content

- It helps the developer to develop and manipulate the elements
- It consists of 4 edges
    - Content edge – It comprises of the actual content
    - Padding edge – It lies in between content and border edge
    - Border edge – Padding is followed by the border edge

Margin edge – It is an outside border and controls the margin of the element.

6. RESULTS AND DISCUSSIONS

CODING

MANUAL NET ARCHITECTURE

```
# Import the necessary libraries.

import numpy as np

import matplotlib.pyplot as plt

import os

# Import the necessary libraries.

import tensorflow as tf

import glob

from PIL import Image

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, Dense, MaxPooling2D, Flatten,
Convolution2D, Dropout, BatchNormalization
```

```
# Ignoring the warings
import warnings
warnings.filterwarnings("ignore")
Rs_10 = 'DATASET/TRAIN/Rs_10'
Rs_20 = 'DATASET/TRAIN/Rs_20'
Rs_50 = 'DATASET/TRAIN/Rs_50'
Rs_100 = 'DATASET/TRAIN/Rs_100'
Rs_200 = 'DATASET/TRAIN/Rs_200'
Rs_500 = 'DATASET/TRAIN/Rs_500'
Rs_2000 = 'DATASET/TRAIN/Rs_2000'
def plot_images(item_dir, n=5):
    all_item_dir = os.listdir(item_dir)
    item_files = [os.path.join(item_dir, file) for file in all_item_dir][:n]
      plt.figure(figsize=(35, 10))
    for idx, img_path in enumerate(item_files):
        plt.subplot(2, n, idx+1)
        img = plt.imread(img_path)
        plt.imshow(img, cmap='gray')
        plt.axis('off')

    plt.tight_layout()
def Images_details_Print_data(data, path):
```

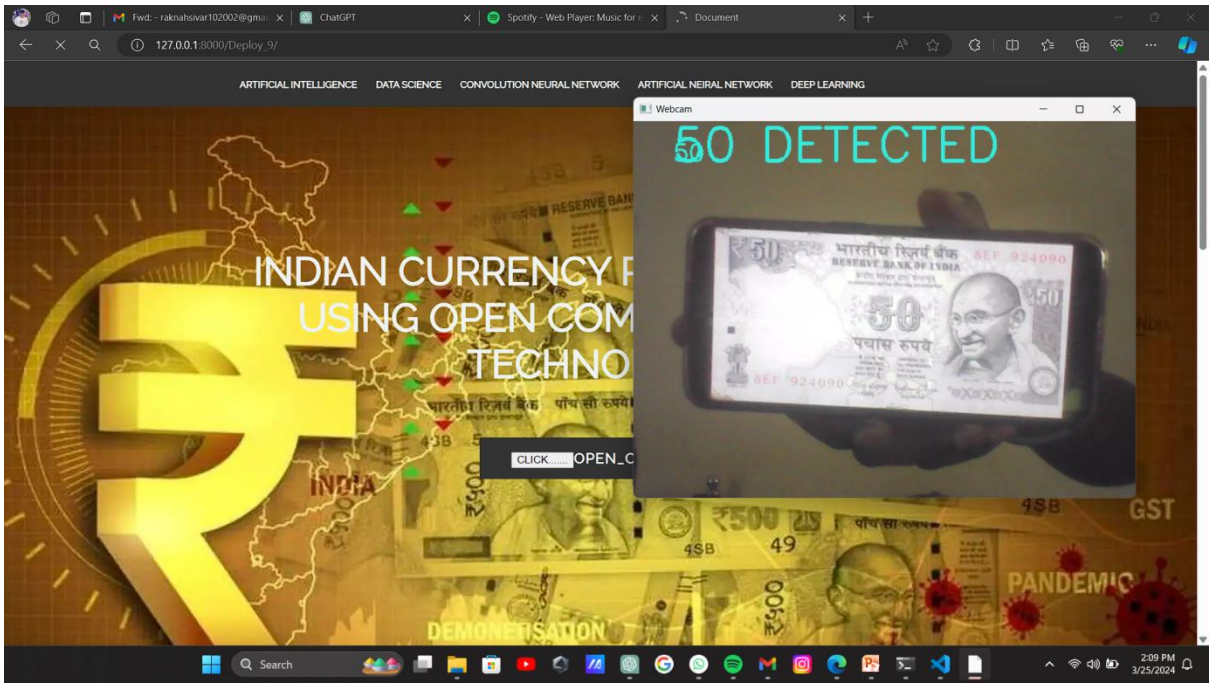
```
    print(" ====== Images in: ", path)
    for k, v in data.items():
        print("%s:\t%s" % (k, v))
def Images_details(path):
    files = [f for f in glob.glob(path + "**/*.*", recursive=True)]
    data = {}
    data['images_count'] = len(files)
    data['min_width'] = 10**100  # No image will be bigger than that
    data['max_width'] = 0
    data['min_height'] = 10**100  # No image will be bigger than that
    data['max_height'] = 0
    for f in files:
        im = Image.open(f)
        width, height = im.size
        data['min_width'] = min(width, data['min_width'])
        data['max_width'] = max(width, data['max_height'])
        data['min_height'] = min(height, data['min_height'])
        data['max_height'] = max(height, data['max_height'])

    Images_details_Print_data(data, path)
print("")
plt.title('Model Loss')
```

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.show()

Output



7. Conclusion:

In conclusion, the application of Open Computer Vision (OpenCV) technology in Indian currency recognition marks a significant advancement in the field of automated financial systems. The utilization of computer vision algorithms and image processing techniques has enabled the development of robust and efficient systems capable of accurately identifying and categorizing Indian currency notes. This innovation not only enhances the speed and accuracy of currency recognition processes but also contributes to the overall efficiency of financial transactions, fostering a seamless and technologically advanced monetary ecosystem in India. The successful integration of OpenCV in currency recognition underscores the transformative potential of cutting-edge

technologies in shaping the future of financial systems and underscores the importance of continued research and development in this domain.

8. Future Work:

The future work for the project on "Indian Currency Recognition by Using Open Computer Vision Technology" could focus on enhancing the system's accuracy and versatility. This may involve incorporating advanced machine learning algorithms to improve currency detection in varying lighting conditions and angles. Additionally, the system could be extended to recognize and authenticate various denominations and new currency releases. Collaborations with financial institutions and regulatory bodies could be explored to integrate real-time updates on currency design changes. Furthermore, exploring the integration of mobile applications or wearable devices for real-world applications, such as assisting visually impaired individuals in identifying currency, would add societal value to the technology. Continued research and development in this domain can contribute to the evolution of secure and efficient financial systems.

9. References:

[1] Jayashree, K & Sheila Anand 2013, ‘Web Service Diagnoser Model for Managing faults in web services’, Computer standards and Interfaces, Elsevier Publications, vol. 36,no. 1, pp. 154-164.

[2] J. Lehrner, S. Kogler, C. Lamm, D. Moser, S. Klug, G. Pusswald, P. Dal-Bianco, W. Pirker, and E. Auff, ‘‘Awareness of memory deficits in subjective cognitive decline, mild cognitive impairment, Alzheimer’s disease and Parkinson’s disease,’’ Int. Psychogeriatrics, vol. 27, no. 3, p. 357, 2015.

[3] J. Lehrner, S. Kogler, C. Lamm, D. Moser, S. Klug, G. Pusswald, P. Dal-Bianco, W. Pirker, and E. Auff, ‘‘Awareness of memory deficits in subjective

cognitive decline, mild cognitive impairment, Alzheimer's disease and Parkinson's disease,'' Int. Psychogeriatrics, vol. 27, no. 3, p. 357, 2015.

[4] H. Ji, Z. Liu, W. Yan, and R. Klette, ''Early diagnosis of Alzheimer's disease using deep learning,'' in Proc. 2nd Int. Conf. Control Comput. Vis., 2019, pp. 87–91.

[5] M. A. Ebrahimighahnavieh, S. Luo, and R. Chiong, ''Deep learning to detect Alzheimer's disease from neuroimaging: A systematic literature review,'' Comput. Methods Programs Biomed., vol. 187, Apr. 2020, Art. no. 105242.

[6] R. C. Petersen, ''Mild cognitive impairment,'' Continuum Lifelong Learn. Neurol., vol. 22, no. 2, p. 404, 2016.

[7] S. Ishiai, Y. Koyama, K. Seki, S. Orimo, N. Sodeyama, E. Ozawa, E. Y. Lee, M. Takahashi, S. Watabiki, R. Okiyama, T. Ohtake, and M. Hiroki, ''Unilateral spatial neglect in AD: Significance of line bisection performance,'' Neurology, vol. 55, no. 3, pp. 364–370, Aug. 2000.

[8] M. Tanveer, B. Richhariya, R. U. Khan, A. H. Rashid, P. Khanna, M. Prasad, and C. T. Lin, ''Machine learning techniques for the diagnosis of Alzheimer's disease: A review,'' ACM Trans. Multimedia Comput., Commun., Appl., vol. 16, no. 1s, pp. 1–35, 2020.

[9] Babu Rajendiran Jayashree Kanniappan "A Generic Model for Identifying QoS Parameters Interrelations in Cloud Services Selection Ontology during Runtime" Symmtery vol.13, no 4 pp 1-18, 2021.

[10] M. Nguyen, T. He, L. An, D. C. Alexander, J. Feng, and B. T. T. Yeo, ''Predicting Alzheimer's disease progression using deep recurrent neural networks,'' NeuroImage, vol. 222, Nov. 2020, Art. no. 117203.

[11] T. Jo, K. Nho, and A. J. Saykin, ‘‘Deep learning in Alzheimer’s disease: Diagnostic classification and prognostic prediction using neuroimaging data,’’ Frontiers Aging Neurosci., vol. 11, p. 220, Aug. 2019.

[12] I. Beheshti, H. Demirel, and H. Matsuda, ‘‘Classification of Alzheimer’s disease and prediction of mild cognitive impairment-to-Alzheimer’s conversion from structural magnetic resource imaging using feature ranking and a genetic algorithm,’’ Comput. Biol. Med., vol. 83, pp. 109–119, Apr. 2017.

[13] V. S. Nori, C. A. Hane, Y. Sun, W. H. Crown, and P. A. Bleicher, ‘‘Deep neural network models for identifying incident dementia using claims and EHR datasets,’’ PLoS ONE, vol. 15, no. 9, 2020, Art. no. e0236400.

[14] J. Wang, M. J. Knol, A. Tiulpin, F. Dubost, M. de Bruijne, M. W. Vernooij, H. H. H. Adams, M. A. Ikram, W. J. Niessen, and G. V. Roshchupkin, ‘‘Gray matter age prediction as a biomarker for risk of dementia,’’ Proc. Nat. Acad. Sci. USA, vol. 116, no. 42, pp. 21213–21218, Oct. 2019.

[15] H. Li and Y. Fan, ‘‘Early prediction of Alzheimer’s disease dementia based on baseline hippocampal MRI and 1-year follow-up cognitive measures using deep recurrent neural networks,’’ in Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI), Apr. 2019, pp. 368–371.

[16] H. Li, M. Habes, D. A. Wolk, and Y. Fan, ‘‘A deep learning model for early prediction of Alzheimer’s disease dementia based on hippocampal magnetic resonance imaging data,’’ Alzheimer’s Dementia, vol. 15, no. 8, pp. 1059–1070, Aug. 2019.

[17] M. Ganguli, Y. Jia, T. F. Hughes, B. E. Snitz, C.-C.-H. Chang, S. B. Berman, K. J. Sullivan, and M. I. Kamboh, ‘‘Mild cognitive impairment that does not progress to dementia: A population-based study,’’ J. Amer. Geriatrics Soc., vol. 67, no. 2, pp. 232–238, Feb. 2019.