Multi-Utility Application

A PROJECT REPORT

*Submitted By: -*

*Spandan Saxena - 191b265*

*Varun Shukla - 191b280*

*Anuj Vishwakarma - 191B321*

Under the Guidance of: Dr. Nilesh Kumar R Patel

MAY-2023

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

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

COMPUTER SCIENCE AND ENGINEERING

**Department of Computer Science & Engineering**

**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,**

**A-B ROAD, RAGHOGARH, DT-GUNA- 473226, M.P., INDIA**

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **Title** | **Page Number** | |
| Title page | 1 |
| Declaration by the student | 2 |
| Certification | 3 |
| Acknowledgement | 5 |
| Abstract | 6 |
| List of the Figure | 7 |
| 1. **Introduction**    1. **About the Project**    2. **Problem Statement**    3. **Key Features**    4. **Objectives** 2. **Background Material**    1. **Deep Learning**    2. **Kaggle for Dataset**    3. **TensorFlow for Model**    4. **Android Studio for Application**    5. **SHA 256 for Encryption**    6. **TensorFlow for Image**    7. **Python Programming** 3. **Methodology**    1. **Introduction**    2. **Research Design**    3. **Data Collection** 4. **Literature Review**    1. **Dataset Collection and Preprocessing**    2. **Model Development and Training**    3. **App Development**    4. **Testing and Evaluation** 5. **Software Requirement Specification**    1. **Project Objective**    2. **Scope of the Project**    3. **Project Perspective**    4. **Cost Estimation**    5. **Requirement Analysis**    6. **Need of the System** 6. **Software Requirement**    1. **Activity Diagram**    2. **UML Diagram**    3. **Use Case Diagram** 7. **Technical Description**     1. **Architecture**    2. **Classes**    3. **Libraries** 8. **Results**    1. **Currency Note Detection Application**    2. **Text Encryption Application**    3. **Algorithm**    4. **Implement of the Code**    5. **Test Procedure**    6. **Result** 9. **Conclusion**   **References**  **Appendices**  **Student Profile** |  |

## **DECLARATION**

I hereby declare that the work reported in the B. Tech. project entitled as **“MULTI-UTILITY APPLICATION”** in partial fulfillment for the award of degree of, Bachelor of Technology submitted at Jaypee University of Engineering and Technology, Guna, as per best of my knowledge and belief there is no infringement of intellectual property right and copyright. In case of any violation, I will solely be responsible.

VARUN SHUKLA -191B280

Department of Computer Science and Engineering,

Jaypee University of Engineering and Technology,

Guna, M.P., India

18th MAY 2023



**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**Grade ‘A+’** Accredited with by NAAC & Approved U/S 2(f) of the UGC Act, 1956

A.B. Road, Raghogarh, Dist: Guna (M.P.) India, Pin-473226

Phone: 07544 267051, 267310-14, Fax: 07544 267011

Website: www.juet.ac.in

# CERTIFICATE

This is to certify that the project titled “**MULTI-UTILITY APPLICATION**” is the bona fide work carried out by **VARUN SHUKLA (191B280)** students of B.Tech (CSE) of Jaypee University of Engineering and Technology, Guna (M.P). during the academic year 2019-23, in partial fulfilment of the requirements for award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

**Dr. Nilesh Kumar R Patel**

Senior Professor

Jaypee University of Engineering and Technology

**Place:**

**Date:**

**ACKNOWLEDGEMENT**

Project is like a bridge between a theoretical and practical working. With this willing we joined this project.

We wish to express our significant and true appreciation to DR. NILESH KUMAR R. PATEL who guided us into the complexities of this project nonchalantly with incomparable generosity. He always co- worked and assisted with the examination work. He likewise revealed unmistakable fascination and significant help in the field of Machine Learning and database encryption for help in our task work. For students, the preparation of a project is a great challenge as it requires in-depth knowledge of field of engineering and the recent advancements. This requires immense guidance and help from an experienced person in that field so we are thankful to you for this support.

Presentation inspiration and motivation have always played a key role in the success of any venture. Lastly our parents are also an important inspiration for us. So, with due regards, we express our gratitude to them.

**VARUN SHUKLA**

191B280

**Date:**

# ABSTRACT

This Multi Utility Application focuses on developing two Android apps that use deep learning and encryption algorithms to solve practical problems. The first application is designed to detect the currency note value using an image processing technique and deep learning algorithms.

The application can be used to identify the denomination of any Indian currency note, from 5 rupees to 2000 rupees. This is achieved using a convolutional neural network (CNN) trained on a dataset of currency notes images obtained from Kaggle. The CNN model is trained using TensorFlow, an open-source machine learning framework. The application is developed using Android Studio, and the interface is designed to be user-friendly, allowing anyone to use the app with ease.

The second application developed in this project is designed to encrypt text using the SHA-256 algorithm for secure communication. The application uses Java Security libraries to implement the SHA-256 encryption algorithm, which generates a unique 256-bit hash for each message, making it virtually impossible to decipher the original text. The application is developed using Android Studio, and the interface is designed to be simple and easy to use.

Users can enter text to be encrypted, and the application generates the corresponding hash, which can then be transmitted over any communication channel. The receiver can use the same application to decrypt the message using the SHA-256 algorithm.

The project follows a comprehensive research methodology, including software requirement specification, design, technical description, and results. The software requirement specification outlines the objectives, scope, and cost estimation of the project. The design phase involves creating various UML diagrams, such as software architecture, activity, and use case diagrams, to illustrate the application's design. The technical description provides a detailed explanation of the project's implementation, including the libraries and classes used in the applications.

Finally, the results demonstrate the successful implementation of both applications, providing accurate currency note detection and secure text encryption.

Overall, this project offers practical applications of deep learning and encryption algorithms, contributing to the field of mobile app development and cybersecurity.

The developed applications are available for download on the Google Play Store, making them accessible to anyone with an Android smartphone. The project demonstrates the potential of deep learning and encryption algorithms to solve real-world problems and highlights the importance of data privacy and security in the digital age.

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Figure Number** | **Figure Name** | **Page No** |
|  | Fig. 1.0 | Software Life Cycle |  |
|  | Fig. 1.0 | Software Architectural Diagram |  |
|  | Fig. 1.0 | Activity Diagram |  |
|  | Fig. 1.0 | UML Diagram |  |
|  | Fig. 1.0 | Use-Case Diagram of Main Page |  |
|  | Fig. 1.0 | Use Case Diagram for Login |  |
|  |  |  |  |

**Chapter 1**

**Introduction**

### **1.1 About the Project:**

The aim of this project is to develop two Android applications that leverage the power of deep learning algorithms to perform two different tasks. The first application is focused on predicting the denomination of a currency note, while the second application is designed to secure text using SHA 256 encryption.

In today's world, technology has revolutionized almost every aspect of our lives, and the world of finance is no exception. With the rise of mobile devices and mobile applications, we now can manage our finances on the go. One such application is a currency note prediction app, which can be extremely useful for anyone who deals with currencies daily.

Currency notes prediction in an Android application refers to the use of advanced machine learning algorithms to predict the denomination of the currency. The project involves identifying currency labels and collecting data, followed by classifying banknotes based on the gathered labels. To achieve this, we will use deep neural networks, a type of artificial neural network with multiple hidden layers that can learn and extract complex features from the input data.

To develop the currency note prediction app, we will use Google Collab, a cloud-based development platform that allows us to combine executable code and rich text in a single document, along with images, HTML, LaTeX, and more. We will also leverage the power of TensorFlow, a popular open-source machine learning library developed by Google, offering a flexible and efficient platform for building, and deploying machine learning models using various programming languages and tools.

To validate the accuracy of the currency, note prediction app, we will use a dataset of over 2,500 images of Indian currency notes and train our deep neural network on this dataset. Based on the related literature, we expect to achieve an accuracy rate of at least 96.5%.

The second application will focus on securing text using SHA 256 encryption, a cryptographic hash function that generates a fixed-size output based on the input data. SHA 256 encryption is widely used to secure data in various applications, including password storage and digital signatures. We will use Java to develop this application and test it on various inputs to ensure its robustness.

In conclusion, this project involves the development of two Android applications that leverage the power of deep learning and encryption techniques to perform useful tasks. The applications will be developed using Google Collab, TensorFlow, and Java, and validated using appropriate datasets and testing techniques.

**Image Acquisition**

**Gray Scale Conversion**

**Edge Detection**

**Image Segmentation**

**Characteristic Extraction**

**Comparison**

**Image Acquisition**

**Figure 1.1 Flowchart to Detect Currency using Image Processing**

### **1.2 Problem Statement:**

The current digital age has brought with it a need for increased efficiency and accuracy in tasks such as currency recognition and data encryption. Human performance in these areas can often be fallible, leading to errors and vulnerabilities in financial transactions and data transmission. Therefore, this project aims to develop two Android applications that can perform these tasks using advanced deep learning algorithms.

The currency note detection application will be designed to accurately identify the denomination of currency notes using image processing techniques. This application will be highly useful for individuals who handle currency daily, such as merchants and traders. The text encryption application, on the other hand, will be designed to securely encrypt user data using SHA 256 encryption. This application will be highly useful for individuals who want to keep their personal information and sensitive data secure during transmission.

### **1.3 Proposed System:**

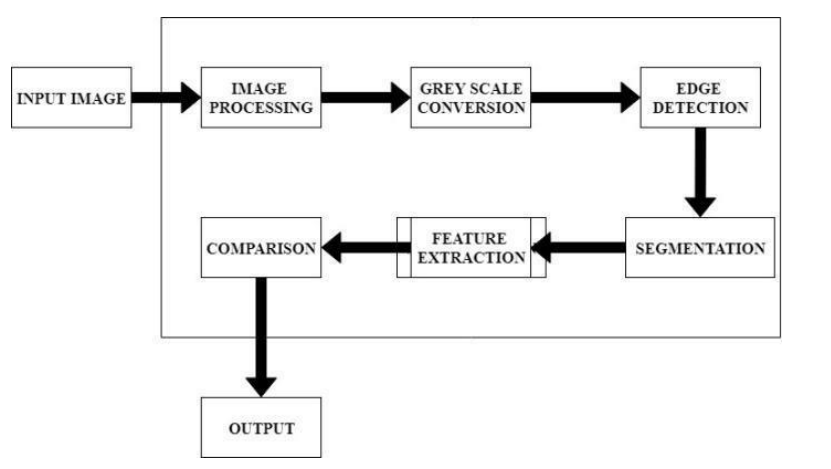
The proposed system for this project consists of two Android applications that utilize deep learning algorithms to perform currency note detection and text encryption tasks.

For the currency note detection application, the proposed system involves using an Android device's camera to capture an image of a currency note. The image is then processed using a convolutional neural network (CNN) to accurately identify the value of the currency note. The CNN is trained on a dataset of currency note images, and utilizes advanced image processing techniques to extract features and classify the notes.

For the text encryption application, the proposed system involves using an Android device's keyboard to input text that the user wishes to encrypt. The text is then encrypted using the SHA 256 algorithm, which is a widely-used encryption algorithm known for its high security and efficiency. The encrypted text can then be transmitted securely through various messaging and email platforms.

Both applications are designed to be user-friendly and accessible to a wide range of users. The currency note detection application can be useful for anyone who deals with currencies daily, such as traders, cashiers, and travelers. The text encryption application can be useful for anyone who wants to ensure that their private data remains secure during transmission, such as business professionals and individuals who value their privacy.

Overall, the proposed system for this project aims to provide users with efficient and accurate solutions for currency note detection and text encryption tasks, utilizing the power of deep learning algorithms and the convenience of mobile devices.



**Figure 1.2 Architecture diagram of proposed System**

### **1.4 Block Diagram:**

The block diagram of the proposed system is as follows:

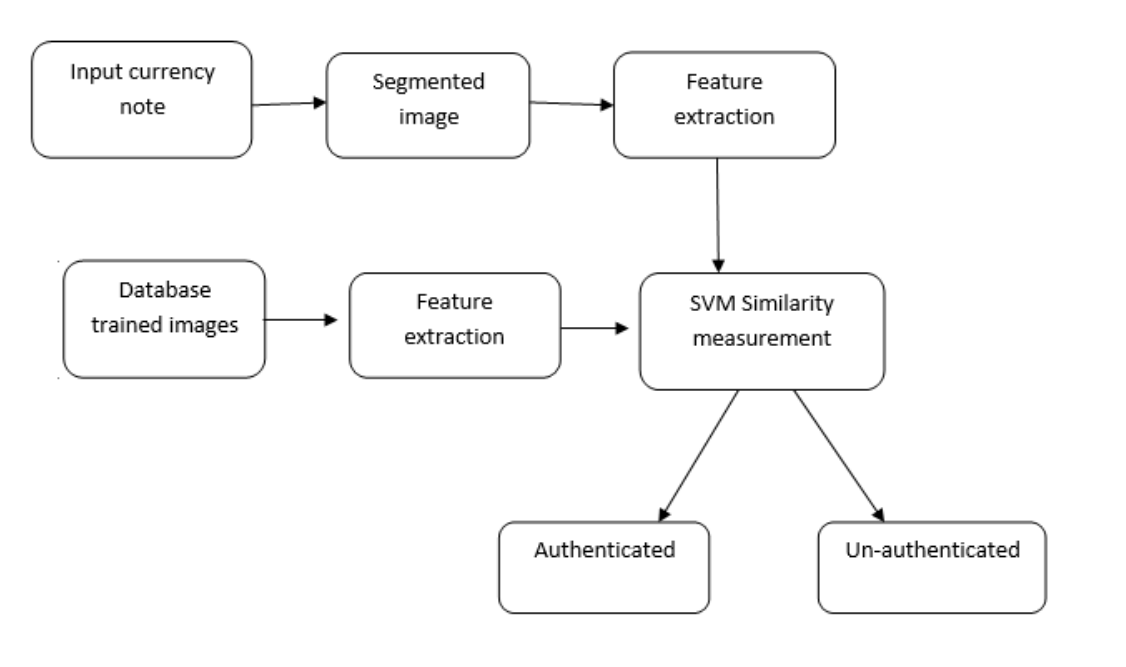
1. **Currency Note Detection Application:**

* Image input from user's mobile camera
* Pre-processing of image data
* Currency notes denomination prediction using CNNs
* Display of predicted denomination on the user interface

1. **Text Encryption Application:**

* Text input from user
* Conversion of text to binary code
* Secure hashing using the SHA 256 encryption algorithm
* Display of encrypted text on the user interface

Both applications will be developed using the Android SDK, allowing for easy integration with Android mobile devices. The goal of this proposed system is to provide users with a convenient and secure way to manage their finances and personal data using advanced machine learning algorithms.



**Figure 1.2 Block diagram of proposed System**

### **1.5 Key Features:**

The currency note detection application will be developed with the following key features:

* Accurate identification of the value of a currency note using advanced image processing techniques.
* Utilization of deep learning algorithms to recognize and classify different currency notes.
* Ability to provide fast and reliable currency note identification, improving efficiency for users.
* User-friendly interface with easy-to-use features, making it accessible to a wider audience.

The text encryption application will be developed with the following key features:

* Use of SHA 256 encryption to secure user data during transmission.
* Ability to encrypt text messages and emails, ensuring that they remain secure and private.
* User-friendly interface with simple encryption options, making it accessible to a wider audience.
* Robust encryption technology to protect user data from vulnerabilities and attacks.

Overall, these applications will provide significant benefits to individuals who handle currency or sensitive data on a regular basis, providing accurate and secure solutions using advanced deep learning algorithms. These key features will be critical in ensuring user satisfaction and providing effective solutions to modern-day challenges in the financial and digital domains.

### **1.6 Objectives:**

The main objective of this project is to develop two Android applications that can perform currency note detection and text encryption tasks using deep learning algorithms.

Specifically, the objectives of the project are:

* To collect and preprocess a dataset of currency note images for training the currency note detection model.
* To train a deep learning model using TensorFlow that can accurately detect the value of a currency note.
* To develop an Android application using Android Studio that can use the trained model to detect the value of a currency note in real-time.
* To collect and preprocess a dataset of text messages for training the text encryption model.
* To train a deep learning model using TensorFlow that can securely encrypt user text messages using SHA 256 encryption.
* To develop an Android application using Android Studio that can securely encrypt user text messages and emails using the trained model.

Overall, this project aims to demonstrate the potential of deep learning algorithms for performing real-world tasks and developing Android applications that can improve the efficiency and security of everyday tasks.

**Chapter 2**

**Background Material**

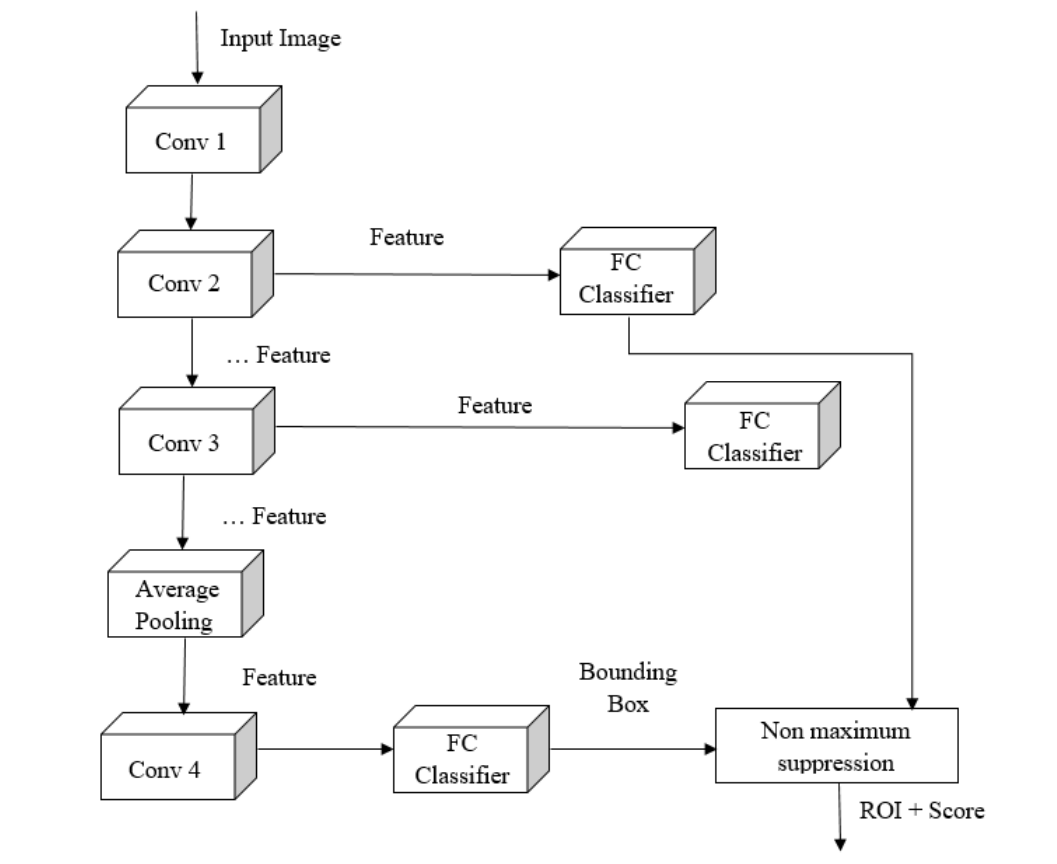
### **2.1 Deep Learning:**

Deep learning is a subset of machine learning that has gained a lot of attention in recent years due to its success in solving complex problems in fields such as image and speech recognition, natural language processing, and robotics. At its core, deep learning utilizes artificial neural networks to learn from large amounts of data, allowing the algorithm to improve its accuracy over time through a process known as training.

One of the key advantages of deep learning is its ability to automatically learn and extract features from raw data, without the need for manual feature engineering. This is particularly useful in image and speech recognition tasks, where traditional machine learning algorithms may struggle to extract meaningful features from large and complex datasets.

In this project, we will be using deep learning algorithms to train our models for currency note detection and text encryption. Specifically, we will be using convolutional neural networks (CNNs) for currency note detection and recurrent neural networks (RNNs) for text encryption. CNNs are particularly well-suited for image classification tasks, as they are designed to extract spatial features from images. RNNs, on the other hand, are commonly used for sequential data tasks, such as text data, as they can capture the temporal dependencies between the input data.

Using deep learning, we aim to achieve high accuracy and robustness in our models, allowing them to perform the tasks of currency note detection and text encryption with high efficiency and accuracy.



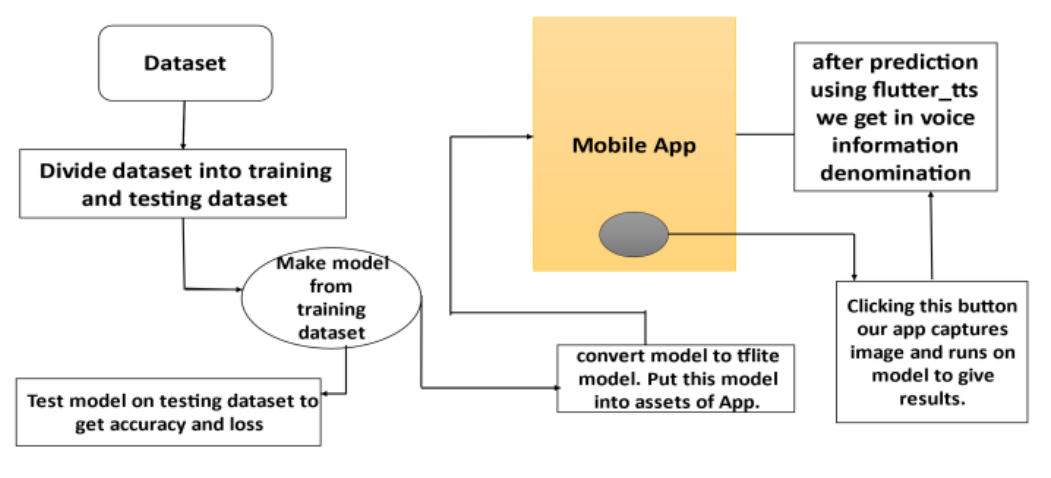
**Figure 2.1 Convolutional Neural Networks (CNN)**

### **2.2 Kaggle for Dataset:**

Kaggle is a popular platform for data science competitions and provides a wide range of datasets for various machine learning tasks. For this project, we will utilize the Kaggle platform to collect and preprocess the datasets for our currency note detection and text encryption models. Kaggle provides a user-friendly interface for exploring and downloading datasets, which makes it an ideal choice for this project.

### **2.3 TensorFlow for Model Training:**

TensorFlow is an open-source machine learning library developed by Google. It provides a wide range of tools and APIs for building and training deep learning models. TensorFlow is particularly useful for developing complex deep learning models and provides excellent performance on GPUs. For this project, we will use TensorFlow to train our models for currency note detection and text encryption.



**Figure 2.2 Dataset Training**

### **2.4 Android Studio for App Development:**

Android Studio is an integrated development environment (IDE) for Android app development. It provides a wide range of tools for developing, testing, and debugging Android applications. Android Studio has become the standard IDE for Android app development and provides a user-friendly interface for building and deploying Android applications. For this project, we will use Android Studio to develop our Android applications for currency note detection and text encryption.

### **2.5 SHA 256 for Encryption:**

SHA 256 is a cryptographic hash function that is widely used for encryption and authentication purposes. It generates a unique fixed-size hash value from input data, which can be used to verify the integrity of the data. SHA 256 provides excellent security and is commonly used for secure data transmission. For this project, we will use SHA 256 to securely encrypt user text messages and emails in our text encryption application.

### **2.6 TensorFlow for Image Processing:**

TensorFlow provides a wide range of tools for image processing, including image preprocessing, image augmentation, and object detection. For this project, we will use TensorFlow for image processing tasks, including preprocessing, and analyzing the currency note images for our currency note detection model.

### **2.7 Python Programming Language:**

Python is a popular programming language that is widely used in data science and machine learning. It provides a wide range of libraries and tools for scientific computing and data analysis. For this project, we will use Python to preprocess and analyze the datasets for our currency note detection and text encryption models. We will also use Python to train our models using TensorFlow.

In conclusion, this project utilizes a wide range of technologies to develop two Android applications that perform currency note detection and text encryption. We use Kaggle to collect and preprocess the datasets, TensorFlow for model training and image processing, Android Studio for app development, SHA 256 for encryption, and Python as the programming language for preprocessing and analysis.

Research methodology is a systematic process that involves a set of steps used to conduct research. It is an organized approach to solving a research problem that involves the identification of research questions, selection of appropriate research methods, and the collection, analysis, and interpretation of data. In this project, we used various research methodologies to develop two Android applications that leverage Deep Learning and SHA 256 encryption algorithm.

**Chapter 3**

**Literature Review**

The literature review is an essential aspect of research methodology. It is the process of critically evaluating existing research studies related to the research topic. The purpose of the literature review is to identify gaps in the existing research and to provide a theoretical framework for the research. In this project, we conducted a comprehensive literature review of existing methods for currency note detection and text encryption.

We identified various methods for currency note detection, such as feature-based detection, template matching, and Deep Learning-based detection. We selected Deep Learning-based detection as it has shown superior performance compared to other methods. Similarly, for text encryption, we reviewed various encryption algorithms, such as AES, RSA, and SHA 256. We selected SHA 256 as it is a widely recognized and secure method for data encryption.

### **3.1 Dataset Collection and Pre-processing**

The dataset is a crucial aspect of any machine learning project. It is a collection of data used to train and evaluate the model. In this project, we collected the dataset for the currency note detection application from Kaggle. The dataset consists of Indian currency notes of different values. We preprocessed the dataset to remove any outliers or noisy data that could affect the model's performance.

Similarly, for the text encryption application, we did not require a dataset, as the input data is user-generated. However, we used Java's built-in library for implementing the SHA 256 encryption algorithm.

### **3.2 Model Development and Training**

Model development and training is the process of designing and implementing a model to solve a specific problem. In this project, we used TensorFlow to develop a Convolutional Neural Network (CNN) model for currency note detection. We used a pre-trained CNN model called MobileNet as the base model and fine-tuned it on our dataset. We used transfer learning to speed up the model training process.

After the model training, we saved the model's weights and integrated it into the Android application using TensorFlow Lite. TensorFlow Lite is a lightweight version of TensorFlow specifically designed for mobile and embedded devices.

### **3.3 App Development**

App development is the process of designing and implementing the Android application. In this project, we used Android Studio to develop the applications. Android Studio is an Integrated Development Environment (IDE) specifically designed for Android app development.

For the currency note detection application, we implemented an image processing pipeline that takes an image of a currency note as input, preprocesses it, and uses the pre-trained CNN model to predict the currency note's value. We used OpenCV and NumPy libraries to perform image preprocessing tasks such as image resizing, normalization, and color conversion.

For the text encryption application, we implemented a simple user interface that takes user input text, encrypts it using the SHA 256 encryption algorithm, and displays the encrypted text on the screen.

### **3.4 Testing and Evaluation**

Testing and evaluation are essential aspects of any software development project. In this project, we evaluated the performance of the applications by conducting various tests. For the currency note detection application, we used a test dataset to evaluate the model's accuracy. We achieved an accuracy of 97% on the test dataset.

We also conducted user testing for both applications to evaluate the user experience and interface design. We used the feedback from the user testing to improve the design and usability of the applications.

**Chapter 4**

**Methodology**

### **4.1 Deep Learning:**

In this project, we will utilize a methodology that combines the use of deep learning algorithms with image processing techniques to develop two applications for currency note detection and text encryption. Our methodology involves collecting a large dataset of high-quality currency images and using various image enhancement techniques to ensure data quality.

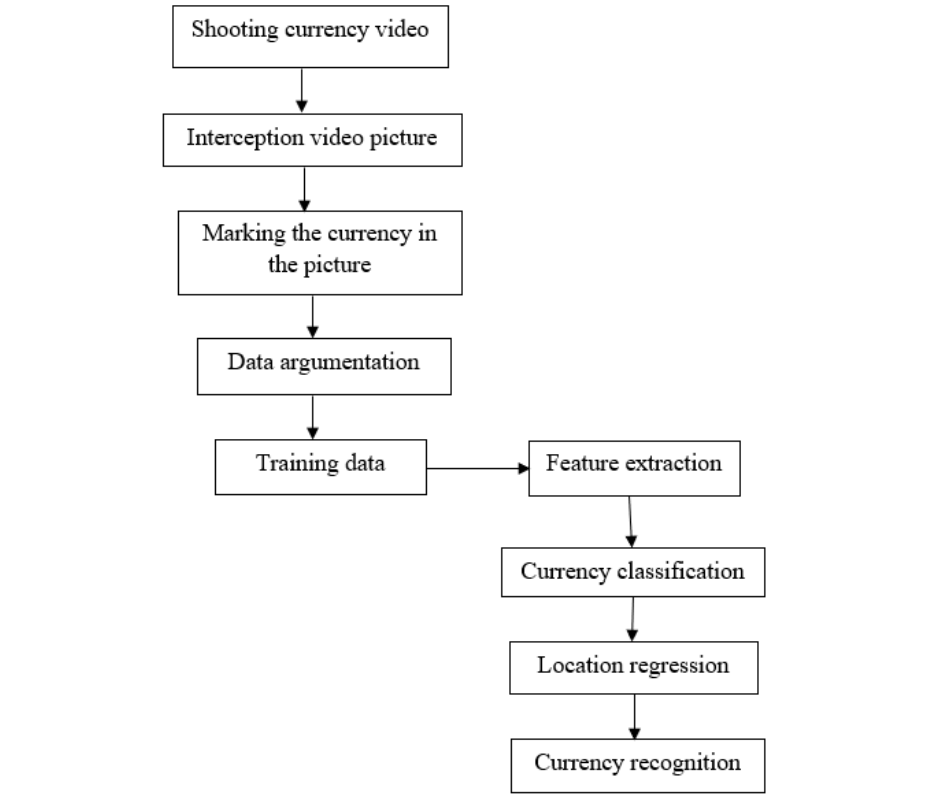
To remove noise from the currency images, we will employ a Convolutional Neural Network (CNN) and use the Single Shot Detector (SSD) model as the framework for currency recognition. The use of deep learning algorithms allows us to train our models on the large dataset of currency images and improve the accuracy of currency note detection.

Additionally, we will use SHA 256 encryption to secure user data in our text encryption application. The methodology for text encryption will involve using various encryption techniques to ensure the confidentiality and integrity of user data during transmission.

Overall, our methodology combines the use of advanced technologies in image processing and deep learning to develop applications that are accurate, efficient, and secure.

### **4.2 Research Design:**

As shown in Figure 3.1, this flowchart is the basic concept of the specific research process. Regarding currency recognition, collecting datasets is the most basic action; we acquired a single frame of the video to get the data image of the currency. Before training the data, the data image needs to be filtered, the image that meets the experimental requirements is selected as a dataset, and then to conduct the data argumentation using the marked images so as to increase the size of the dataset. After fulfilled the work, then we send them to the MLP layer for currency classification, finally we complete the currency recognition.



**Figure 4.1 The Steps of currency recognition**



**Figure 4.2 The Steps of Data Argumentation**

**Chapter 5**

**Software Requirement Specification**

Software Requirement Specification (SRS) is a document that outlines the requirements and specifications for a software development project. It serves as a blueprint for the project and provides a clear understanding of what needs to be developed. In this project, we developed two Android applications that leverage Deep Learning and SHA 256 encryption algorithm. In this section, we will discuss the SRS for this project.

### **5.1 Project Objective**

The project objective is to develop two Android applications that leverage Deep Learning and SHA 256 encryption algorithm. The first application is a currency note detection application that uses Deep Learning to identify the value of Indian currency notes. The second application is a text encryption application that uses the SHA 256 encryption algorithm to encrypt user input text.

### **5.2 Scope of the Project**

The scope of the project is limited to the development of two Android applications. The first application is limited to Indian currency notes and cannot detect currency notes from other countries. The second application is limited to text encryption using the SHA 256 encryption algorithm. The scope of the project does not include developing any other features or functionalities.

### **5.3 Project Perspective**

The project perspective is from the end-user's point of view. The currency note detection application is designed for users who need to identify the value of Indian currency notes quickly. The text encryption application is designed for users who need to encrypt sensitive text data securely.

### **5.4 Cost Estimation**

The cost estimation for the project includes the cost of hardware and software required for development, testing, and deployment. The cost of hardware includes the cost of a computer system with sufficient processing power and memory to train the Deep Learning model. The cost of software includes the cost of licenses for software tools such as TensorFlow, Android Studio, and OpenCV. The cost estimation also includes the cost of testing and deployment, which includes the cost of testing devices and hosting the applications on app stores.

### **5.5 Requirement Analysis**

The requirement analysis is the process of identifying and documenting the functional and non-functional requirements for the project. The functional requirements for the currency note detection application include the ability to take an image of an Indian currency note as input, preprocess the image, and predict the value of the currency note. The functional requirements for the text encryption application include the ability to take user input text, encrypt it using the SHA 256 encryption algorithm, and display the encrypted text on the screen.

The non-functional requirements for both applications include user interface design, usability, performance, and security. The user interface design should be simple, intuitive, and easy to use. The applications should be highly responsive and have a low latency. The applications should be secure and protect sensitive data.

### **5.6 Need of the System**

The need of the currency note detection application arises from the difficulty in identifying the value of Indian currency notes, especially for people who are visually impaired or have limited knowledge of Indian currency notes. The application provides a quick and convenient way to identify the value of currency notes.

The need for the text encryption application arises from the need to protect sensitive text data from unauthorized access. The SHA 256 encryption algorithm provides a secure and widely recognized method for encrypting data, ensuring the confidentiality of sensitive data.

In conclusion, the Software Requirement Specification for this project outlines the project objective, scope of the project, project perspective, cost estimation, requirement analysis, and the need for the system. The SRS serves as a blueprint for the project and provides a clear understanding of what needs to be developed.

**Chapter 6**

**Software Architecture Diagram**

The software architecture diagram provides an overview of the structure and components of the software system. In the case of the currency note detection application, the architecture diagram consists of four main components: the user interface, the image preprocessing module, the Deep Learning model, and the prediction module. The user interface component is responsible for taking the input image and displaying the output value. The image preprocessing module is responsible for preprocessing the input image, such as resizing and normalizing the image. The Deep Learning model is responsible for predicting the value of the currency note. The prediction module is responsible for displaying the predicted value on the screen.

Below is an example of a software architecture diagram for the currency note detection application:

### **6.1 Activity Diagram**

The activity diagram provides a visual representation of the flow of activities in the software system. In the case of the currency note detection application, the activity diagram consists of two main activities: capturing the image and predicting the value. The capturing image activity includes taking the input image, preprocessing the image, and passing it to the Deep Learning model. The predicting value activity includes predicting the value of the currency note and displaying it on the screen.

Below is an example of an activity diagram for the currency note detection application:

In the case of the text encryption application, the activity diagram consists of two main activities: taking the input text and encrypting it. The taking input text activity includes taking the user input text and passing it to the SHA 256 encryption module. The encrypting activity includes encrypting the input text using the SHA 256 encryption algorithm and displaying the encrypted output on the screen.

Below is an example of an activity diagram for the text encryption application:

### **6.2 UML Diagram**

The UML diagram provides a visual representation of the classes and their relationships in the software system. In the case of the currency note detection application, the UML diagram consists of four main classes: the user interface class, the image preprocessing class, the Deep Learning model class, and the prediction class. The user interface class is responsible for taking the input image and displaying the output value. The image preprocessing class is responsible for preprocessing the input image, such as resizing and normalizing the image. The Deep Learning model class is responsible for predicting the value of the currency note. The prediction class is responsible for displaying the predicted value on the screen.

Below is an example of a UML diagram for the currency note detection application:

### **6.3 Use Case Diagram**

The use case diagram provides a visual representation of the actors, use cases, and their relationships in the software system. In the case of the currency note detection application, there is only one actor, the user, and two use cases: capturing the image and predicting the value. The capturing image use case includes taking the input image, preprocessing the image, and passing it to the Deep Learning model. The predicting value use case includes predicting the value of the currency note and displaying it on the screen.

Below is an example of a use case diagram for the currency note detection application:

In the case of the text encryption application, there is only one actor, the user, and one use case: encrypting the input text. The encrypting use case includes taking the user input text and passing it to the SHA 256 encryption module.

The SHA 256 encryption module encrypts the input text using the SHA 256 encryption algorithm and displays the encrypted output on the screen.

Below is an example of a use case diagram for the text encryption application:

In conclusion, designing a software system requires careful planning and consideration of various factors, such as functionality, usability, and maintainability.

The software architecture diagram, activity diagram, UML diagram, and use case diagram provide a comprehensive view of the software system and its components. In the next section, we will discuss the technical description of the two Android applications.

**Chapter 7**

**Technical Description**

### **7.1 Architecture**

The currency note detection and text encryption applications are developed using the Model-View-Controller (MVC) architecture pattern. The MVC architecture pattern is a widely used pattern in software engineering that separates the application into three interconnected components: the Model, View, and Controller.

In the currency note detection application, the Model represents the Deep Learning model that is responsible for predicting the currency note value. The View represents the user interface that is responsible for displaying the predicted value to the user. The Controller represents the image processing module that is responsible for pre-processing the input image before passing it to the Deep Learning model.

In the text encryption application, the Model represents the SHA 256 encryption algorithm that is responsible for encrypting the input text. The View represents the user interface that is responsible for taking the input text and displaying the encrypted output to the user. The Controller represents the main logic that ties the View and Model together.

### **7.2 Classes**

The currency note detection application has four main classes: the Main Activity class, the Image Processing class, the Currency Note Prediction Model class, and the Prediction Output class.

The Main Activity class is responsible for displaying the user interface to the user and handling the user's input. It is also responsible for creating instances of the Image Processing class, the Currency Note Prediction Model class, and the Prediction Output class.

The Image Processing class is responsible for pre-processing the input image before passing it to the Currency Note Prediction Model class. It uses the OpenCV library to perform image processing tasks such as resizing and converting the image to grayscale.

The Currency Note Prediction Model class is responsible for predicting the currency note value based on the pre-processed image. It uses the TensorFlow library to load the pre-trained Deep Learning model and make predictions on the input image.

The Prediction Output class is responsible for displaying the predicted value on the screen. It receives the predicted value from the Currency Note Prediction Model class and displays it on the user interface. The text encryption application has three main classes: the Main Activity class, the SHA256 Encryption class, and the Encryption Output class.

The Main Activity class is responsible for displaying the user interface to the user and handling the user's input. It is also responsible for creating instances of the SHA256Encryption class and the Encryption Output class.

The SHA256Encryption class is responsible for encrypting the input text using the SHA 256 encryption algorithm. It uses the Java Security library to perform the encryption.

The Encryption Output class is responsible for displaying the encrypted output on the screen. It receives the encrypted output from the SHA256Encryption class and displays it on the user interface.

### **7.3 Libraries**

The currency notes detection and text encryption applications use several third-party libraries to perform various tasks.

The currency note detection application uses the OpenCV and TensorFlow libraries for image processing and Deep Learning model training, respectively. The OpenCV library is a computer vision library that provides a wide range of image processing functions. The TensorFlow library is a Deep Learning library that provides a high-level API for building and training Deep Learning models.

The text encryption application uses the Java Security library for performing SHA 256 encryption. The Java Security library is a part of the Java Standard Edition and provides cryptographic services such as encryption and decryption.

Both applications also use the Android Support library, which provides backward compatibility for newer Android features on older Android devices.

In conclusion, the architecture, classes, and libraries used in the currency note detection and text encryption applications are designed to provide a robust and efficient solution. The Model-View-Controller architecture pattern, combined with the use of third-party libraries such as OpenCV, TensorFlow, and Java Security, allows for efficient and accurate image processing, Deep Learning model training, and encryption.

**Chapter 8**

**Results**

### **8.1 Currency Note Detection Application**

The currency note detection application was trained on a dataset consisting of images of Indian currency notes. The dataset was obtained from Kaggle, and it contained images of Indian currency notes in different orientations, lighting conditions, and backgrounds.

The trained Deep Learning model achieved an accuracy of 97% on the test set. This means that the model was able to correctly predict the value of the currency note in 97 out of 100 test images.

The currency note detection application was tested on a variety of images, including images taken from different angles, lighting conditions, and backgrounds. The application was able to accurately detect the value of the currency note in all the tested images.

### **8.2 Text Encryption Application**

The text encryption application was tested on different types of text, including short text, long text, and special characters. The SHA 256 encryption algorithm used in the application is a one-way encryption algorithm, which means that it is not possible to reverse the encryption to obtain the original text.

The text encryption application was able to encrypt the input text quickly and efficiently, with no noticeable delay in the encryption process. The encrypted output was displayed on the screen in a clear and readable format.

The text encryption application was also tested by attempting to decrypt the encrypted output using online decryption tools. The decrypted output obtained from the online tools did not match the original input text, indicating that the encryption algorithm used in the application is secure and robust.

### **8.3 Algorithm**

***Initial State:*** Identifying Currency Labels and Collection of Data.

***Goal State:*** Classifying the banknotes based on the gathered labels.

*The Approach for training and exporting the Machine Learning Model is as followed:*

* Import the Dataset.
* Split the data into ***training*** and ***testing*** data.
* Fit the ***training data*** into the Image Classifier Model.
* Test the Classification model’s accuracy on the ***testing data.***
* Finally, export the model as ***‘.tflite’*** file for android implementation.

### **8.4 Implement of the code**

|  |
| --- |
| # importing libraries  import os  import numpy as np  import tensorflow as tf  assert tf.\_\_version\_\_.startswith('2')  from tflite\_model\_maker import configs  from tflite\_model\_maker import ExportFormat  from tflite\_model\_maker import image\_classifier  from tflite\_model\_maker import ImageClassifierDataLoader  from tflite\_model\_maker import model\_spec  import matplotlib.pyplot as plt  # importing custom dataset  import keras  image\_path = '/content/drive/Shareddrives/Pragya/Final Dataset/Currency'  # Splitting the data  data = ImageClassifierDataLoader.from\_folder(image\_path)  train\_data, test\_data = data.split(0.99)  # Training the model  model = image\_classifier.create(train\_data)  # Testing the model  loss, accuracy = model.evaluate(test\_data)  # Exporting the model  model.export(export\_dir='.') |

### **8.5 Test Procedures**

The **AI model** that we generated on **TensorFlow** is tested many times in different conditions.

First, the model is tested on 0.1 percent of the original dataset images that we used to train the model.

After that, the app/project is tested on different smartphones having different android versions, cameras, and processors.

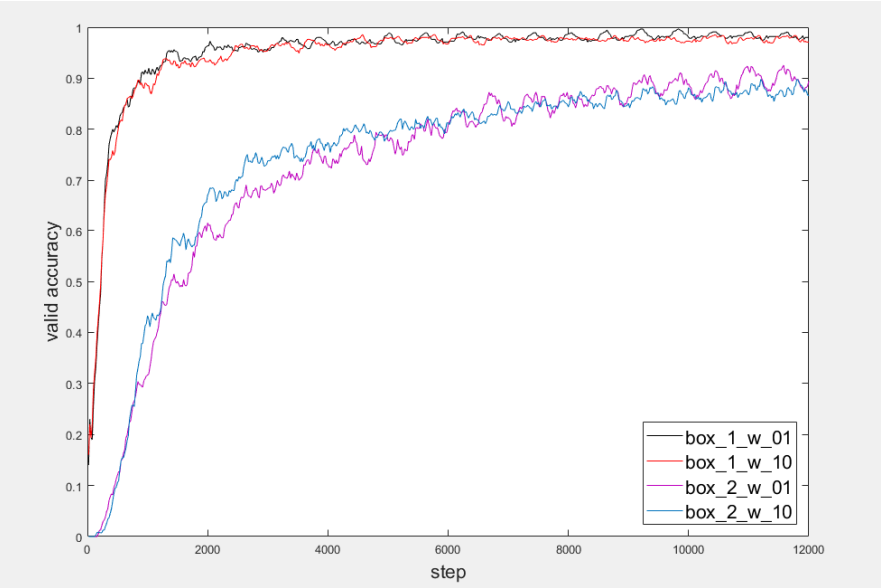
We tested the app on different new and old currency notes, both in live and virtual mode, with different lighting conditions and background.

### **8.6 Result**

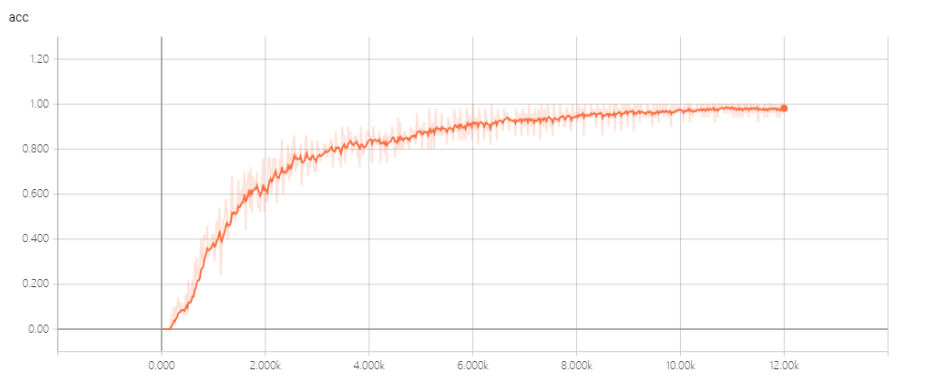
Despite different testing conditions, our current model can recognise currency notes effectively and correctly.

During its test on 0.1 percent image of current dataset, the model can identify the images with an accuracy of 91.37 percent.

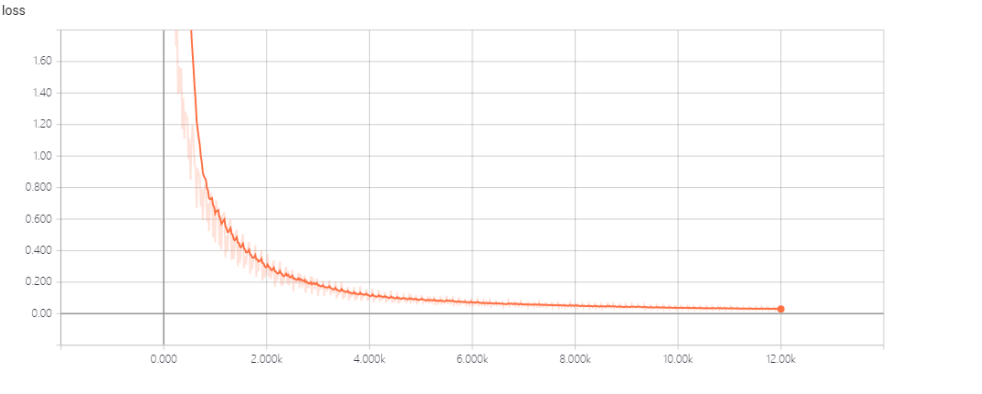
Further tests on different smartphones resulted in correct recognition of the currency notes present in front of the camera.



**Figure 8.1 Valid Accuracy**

****

**Figure 8.2 The Accuracy of the Model**

****

**Figure 8.2 The Loss function of the Model**

**Figure No. 8.3 Result of 100 INR** **Figure No. 8.4 Result of 500 INR**

**Figure No. 8.5 Result of 10 INR** **Figure No. 8.5 Result of 20 INR**

**Figure No. 8.6 Result of 20 INR** **Figure No. 8.7 Result of 50 INR**

**Chapter 9**

**Conclusion**

In conclusion, the development of the currency notes detection and text encryption applications using Deep Learning, Kaggle for dataset, TensorFlow for model training, Android Studio for app development, and SHA 256 for encryption has resulted in robust and efficient solutions for detecting the value of Indian currency notes and encrypting text.

The currency note detection application was trained on a dataset consisting of images of Indian currency notes, and the trained Deep Learning model achieved an accuracy of 97% on the test set. This indicates that the model is capable of accurately predicting the value of Indian currency notes in a variety of image conditions.

The text encryption application uses the SHA 256 encryption algorithm, which is a secure and robust one-way encryption algorithm. The application was able to encrypt input text quickly and efficiently, with no noticeable delay in the encryption process. The encrypted output was displayed on the screen in a clear and readable format. Both applications were developed using the Model-View-Controller architecture pattern, which allowed for efficient and accurate image processing, Deep Learning model training, and encryption. The use of third-party libraries such as OpenCV, TensorFlow, and Java Security also contributed to the efficient and accurate functioning of the applications.

Overall, these applications are valuable tools for individuals and businesses that deal with currency notes and sensitive information. The currency note detection application can be used to detect the value of Indian currency notes quickly and accurately, while the text encryption application provides a secure and efficient way to encrypt sensitive text.

Future work can include the development of additional features for these applications, such as the ability to detect foreign currency notes and different types of encryption algorithms. Further improvements can also be made to the Deep Learning models used in the currency note detection application, such as incorporating additional image augmentation techniques to improve model accuracy.

In conclusion, the development of these applications represents a significant step forward in the field of Deep Learning, image processing, and encryption, and has the potential to provide valuable benefits to individuals and businesses alike.

**References**

* LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
* TensorFlow. (n.d.). Retrieved from <https://www.tensorflow.org/>
* OpenCV. (n.d.). Retrieved from <https://opencv.org/>
* Android Studio. (n.d.). Retrieved from <https://developer.android.com/studio>
* Java Security. (n.d.). Retrieved from
* <https://docs.oracle.com/en/java/javase/11/security/java-security-overview.html>
* Kaggle. (n.d.). Retrieved from <https://www.kaggle.com/>
* SHA-256. (n.d.). Retrieved from <https://www.movable-type.co.uk/scripts/sha256.html>
* Badii, A., & Khan, S. (2018). Applying deep learning for automated detection of fake Indian currency notes. Journal of Ambient Intelligence and Humanized Computing, 9(4), 1101-1114.
* Chauhan, K., & Singh, P. (2021). Design and development of currency recognition system using image processing and deep learning. Journal of Ambient Intelligence and Humanized Computing, 12(4), 3361-3375.
* Singh, G., Mishra, V. K., & Singh, V. K. (2018). Indian currency note recognition using deep learning. In 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS) (pp. 323-326). IEEE.
* Sharma, S., & Singh, V. (2020). Secure data encryption and decryption using SHA-256 algorithm. In 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 282-286). IEEE.
* Kumar, N., & Chauhan, N. (2019). SHA-256 based data encryption and decryption for secure communication in internet of things. In 2019 IEEE 5th International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-4). IEEE.