NORTH SOUTH UNIVERSITY

Department of Electrical and Computer Engineering

Project Topic: Counterfeit Detection of Bangladesh Banknotes

Course Name: Junior Design

Course Code: CSE299

Section: 19

Faculty: Md. Ishan Arefin Hossain (IAH)

Date of submission: 26th November, 2024

Team AM

Group members:

Name	ID
Asifur Rahman	2132835042
Mohammad Hossain	2131770042
Khandaker Anjuman Parvez	2212536042

Approval:

Asifur Rahman (ID-2132835042), Mohammad Hossain (ID-2131770042) and Khandaker Anjuman Parvez(ID-2212536042) from Electrical and Computer Engineering Department of North South University have worked on the Junior Design Project titled “Counterfeit Detection of Bangladesh Banknotes” under the supervision of Ishan Arefin Hossain partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

Supervisor’s Signature

……

Ishan Arefin Hossain

Lecturer

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Chairman’s Signature

……

Dr. Mohammad Abdul Matin

Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Declaration:

This is to confirm that we are the original creators of this project. This work has never been submitted in whole or in part for another degree or diploma award. Project-related data will be kept private and not shared without the project supervisor's permission. This study has appropriately acknowledged and cited pertinent prior works. According to the supervisor, the plagiarism guideline has been upheld.

Students' names & Signatures

1. Mohammad Hossain

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

2. Asifur Rahman

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

3. Khandaker Anjuman Parvez

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Acknowledgment:

For his invaluable support, precise guidance, and advice regarding the experiments, research, and theoretical studies conducted during the current project and in the preparation of the current report, the authors would like to extend their sincere gratitude to their project supervisor, Ishan Arefin Hossain, Lecturer, Department of Electrical and Computer Engineering, North South University, Bangladesh.

Additionally, the research was made possible by the Department of Electrical and Computer Engineering at North South University in Bangladesh, for which the authors are grateful. Additionally, the writers would like to thank their loved ones for their unwavering support and sacrifices.

Table of content

List of Figures

List of Figures

Abstract:

Counterfeit notes are a big problem for the economy. The increase in counterfeit notes leads to economic losses, such as inflation, instability, and personal financial losses in the country's economy, which threaten a country's stability and development. Due to the widespread issue of fake notes, the Bangladeshi economy faces numerous challenges, including reduced trust in financial systems and increased transaction risks. The project aims to create a web-based solution to detect counterfeit Bangladeshi banknotes using machine learning and deep learning. The application uses a convolutional neural network (CNN) model trained on a large dataset of real and fake bill photos from trusted sources. This results in a user-friendly web interface where users can upload pictures for banknote verification so that the system is efficient and accurate in detecting counterfeit notes, thereby further helping to reduce economic corruption and protect the vulnerable population of Bangladesh. The project is highly scalable and integrated, making it suitable for banking, financial transactions, and e-commerce platforms that provide secure exchange. It will improve fraud detection while reducing economic corruption in Bangladesh and protecting vulnerable communities.

Keywords— Machine learning, Deep learning, Counterfeit detection, Bangladeshi banknotes, Image classification, Web-based counterfeit detection, Economic stability, and Financial security.

Introduction:

Suppose you are in a situation where you have received a note and need clarification about whether the note is fake or real. In that case, our counterfeit detection of Bangladesh banknote project will help you detect whether the banknote is counterfeit. The threat of counterfeit banknotes is widespread worldwide, not only in Bangladesh. The economy can suffer from counterfeit money, especially when it comes to inflation. The money supply rises when fake notes are used, which causes an imbalance between the actual values of goods and money. This excess of money has the potential to destabilize the economy, raise prices, and reduce purchasing power. Due to the potential for financial gain, this threat is more severe in poorer nations than industrialized ones, and high-denomination notes are always targeted more frequently than low-denomination ones. We constantly combat counterfeiting, and those who engage in it follow us closely. It undermines the balance of the socioeconomic system. It is also dangerous, particularly for a nation like ours. Thus, educating the public about fake currency is essential to our mission. We periodically see such instances where police arrest people and seize counterfeit money from their possession in various nationwide campaigns, according to news reports and media coverage.

According to a survey conducted in Narayanganj City 2016, fake notes were made on randomly selected people of various ages and professions of 2000 people with some questionnaires. The survey found that 40% needed help distinguishing between fake and genuine notes. So, our AI project will help with this problem. This project uses some machine learning and deep learning features. In the case of machine learning, the method employs evaluation tools to verify how effectively the model distinguishes between real and fake notes and cleans and prepares the data to ensure it is suitable for training. It adds modifications to the dataset, such as flipping and rotating photos, to improve the model's accuracy in real-world situations. The deep learning model offers high accuracy for real-time application, automatically learns specific attributes like watermarks and printing defects without human intervention, and adapts to different counterfeit notes using advanced techniques like transfer learning and data augmentation. We ran a total of 6 models and got excellent accuracy in every model. However, we used the mobile net transfer learning model because it is lightweight compared to other models. With an outstanding accuracy rate of 99.08%.

A dataset of 2280 images is used in this project. Where the dataset is split into two categories. One category is real notes, and another category is fake notes. In every category, there are two splits: one is a 500tk note and another 1000tk note. After augmentation, the dataset is increased to 9120 images. Using these images, we trained the models. After the model training, we test the models with dataset images and real images. Some models made the wrong prediction in some cases; overall, we got accurate results in some models. At the end, we convert the model to a TFlite model.

We developed a website that any user can access, but Google authentication is necessary when anyone wants to detect a note. After that, anyone can upload an image of a note and detect whether it is fake. The website was developed using HTML, CSS, Bootstrap, and JavaScript. To connect the model to the website, we used Python Flask. However, our project can detect only the front side of a note. The back side of an image can lead to wrong predictions. So, our project is efficient only for the front side as the dataset we used is appropriate for the front side.

Literature Review:

[1]The authors have developed a three-layered Deep Convolutional Neural Network model that effectively identifies counterfeit Indian rupee notes. This model comprises the following main steps: image acquisition, image processing, image transformation, edge detection, image segmentation, feature extraction, pattern matching, and printing the result. These steps go in order, as written here. The study examined 306 photos that had undergone augmentation; 80% were used for training and the remaining 20% for validation. The learning rate was found to be 0.001. The work's two main accomplishments are the ability to accurately identify counterfeit money, which is 96.6 with a success rate of 80%, and the development of mobile and web-based applications to do so.

[2]The authors consider three different architectures in the implementation of "Fake Banknote Recognition Using Deep Learning," where they develop a method to identify the best thresholds of the model by transfer learning, namely sequential, residual, and inception. The authors also implement a custom model where transfer is achieved by learning faster estimation in the embedded system. Then, the authors compare the accuracy of the custom model with that of transfer learning. Finally, the authors achieve the accuracy of different models for fake banknote recognition using deep learning 100% and 99.52% in the case of a custom network. Moreover, Using the transfer learning model achieved 99.52% accuracy for VGG16, 99.52% for VGG19, 99.52% for ResNet50, and 99.52% for InceptionV3. Using the custom network and transfer learning models achieved similar accuracy results, with ResNet18 achieving the highest accuracy of 100%. Therefore, the authors believe that custom networks capable of fast estimating fake banknote recognition are most suitable for real-time applications.

[3]The authors work on a convolutional neural network (CNN) classifier for counterfeit detection that can classify banknote images as genuine or counterfeit using visible light, IR reflectance, and IR transmission images of banknotes as input data. They mainly work with counterfeit notes from four national currencies (EURO, USD, Korean, and Jordan) datasets. They used a model in which they trained a convolutional neural network classifier using preprocessing images of banknotes to detect counterfeit banknotes. They also evaluated the classifier's performance using accuracy, precision, and recall metrics. The authors achieved an average accuracy of 97.11% for note recognition in banknote datasets from four countries with their convolutional neural network model. Also, the authors worked

more in-depth on using smartphones to detect fake notes and succeeded in developing this method. As a result, the smartphone camera achieves a high accuracy rate in terms of convenience and acceptability

[4]The journal is about identifying other countries' currencies and fake Indian currencies. In this process, they used MATLAB software for the computational and image processing. The system must be uploaded via a file folder in MATLAB to identify the currency name of the country to which the input image belongs. Calculating parameters like PSNR, mean square error, and the similarity index is essential to this currency recognition system. In cases of fake note detection, an image of a currency note is selected and converted from color (RGB) to grayscale. Edge detection is applied to the entire grayscale image, and then specific features of the note are cropped and segmented. These features are analyzed, and their intensity is measured. Based on this analysis, the note is classified as real or fake. However, their currency recognition system works for some country's currencies.

[5]"Image Processing Based Feature Extraction of Bangladeshi Banknotes" investigates using sophisticated image processing methods to identify counterfeit money. Because of the advances in counterfeiting technologies, traditional methods of detecting counterfeit goods, like chemical and physical property-based techniques, as Bangladeshi bank notes following features, are becoming less and less effective. As a result, researchers are concentrating on feature-based techniques that examine security features, including watermarks, optically variable ink, and microprinting incorporated into currency (OVI). Ahmed [1] suggests a software-based solution that uses characteristics like ultraviolet lines, security threads, micro-letters, and others specific to Bangladeshi banknotes. Optical Character Recognition (OCR), Speeded Robust Features (SURF), Principal Component Analysis (PCA), and contour analysis are among the well-known image processing techniques they use. These are all implemented within the OpenCV library.

[6]Counterfeit currency detection has also become entirely modernized by incorporating different technological methods. Early methods were based on some image processing techniques, including Fourier Descriptors, morphological operations, and edge detection. These techniques gave moderately good results, lying in the range of 90-93% accuracies presented by Sawant et al. and Manikandan. Recent studies have looked into using DWT and LBP methods for feature extraction. These approaches have boosted productivity by cutting down on complex calculations. Researchers have also tried other classifiers in this field, including cutting-edge machine-learning tools like Neural

Networks, K-means clustering, and Support Vector Machines.
Convolutional neural networks have proven to be the most effective of these. These models don't need manual feature extraction because they can achieve high accuracy by looking at raw images of currency notes. The study we're talking about uses transfer learning and 2,000 images to sort notes into real or fake categories. It does this without taking too much time to process.

[7]The rise of better printing tech has led to more fake money in circulation, making it necessary to find good ways to spot counterfeits. At first, people looked at things you could see, like color and size, but this didn't work well for old or dirty bills. Now, we use image processing to do the job. This includes finding edges, turning images to grayscale, and pulling out key features. We use tools like MATLAB for this work. We check things like security threads, watermarks, and special marks to see if a bill is real. The new method takes pictures of money under UV light, breaks down the image, and figures out how bright it is. This helps sort real bills from fake ones. We think the bill is real if the brightness is above 70%. This works well to catch fake ₹500 and ₹2000 notes. We might be able to build this into machines someday.

[8]Earlier techniques mostly paid attention to the visible features, such as color, size, and serial numbers. These methods detected features at the basic level, which caused noise problems and inefficiencies in feature extraction for dirty or damaged notes. Grayscale conversion, edge detection, and image segmentation are examples of operations that increased accuracy but were still inefficient for complex datasets.
The arrival of machine learning models such as Support Vector Machines (SVM) and K-nearest neighbors (KNN) improved detection. KNN was quite good for small datasets, grouping data based on their similarity. Yet, problems like feature noise and dataset limits still existed.
The discovery of fake money using deep learning is the most important breakthrough in this field. CNNs (Convolutional Neural Networks) are very good at detecting that. They can make high-quality images and don't have any actions that need to be performed by someone to extract the features. These make them very accurate. One of the other ways of knowing the realness of a thing besides CNNs is by doing the wavelet transformation on it, which then reveals things such as watermarks, security threads, and hidden images. They can detect false alarms so well that they are only 0.1% of all the signals.

The ability to quickly catch fake money and to be able to deal with many things at the same time is the key advantage of the new systems. Money places like banks and other places use that method. Next, they hope these schemes will be upgraded nicely and smartly. The data they will use will be a larger amount, and they will design very intelligent deep-learning models to resolve the glitches the current models have come up with.

[9]"Fake Currency Detection Utilizing Machine Learning" shows that we are beginning to switch from hardware and digital imaging technologies to deep-learning-based methods, which are more efficient in recognizing disloyal currencies. The traditional approaches that mainly required human intervention and using physical/manual features were ineffective and time-consuming and, thus, were used before.

On the other hand, recent progress is based on convolutional neural networks (CNNs), whose strength lies in the fact that they naturally practice feature learning in the area of two-class image classification using raw images.

Previous studies commissioned by Laavanya & Vijayaraghavan (2019) and Agasti et al. (2017) employed neural networks and image-processing techniques for currency recognition. The system is highly susceptible to the limitations of the traditional methods since it uses data augmentation and preprocessing that strengthen the CNN models, thereby enabling quick and accurate real-time detection. Future research is suggested to explore, for instance, a system with additional security features or improved noise handling.

[10]“Detection of Fake Currency Using MT” describes how the approach of catching fraudulent money has broadened from conventional methods. Traditional manual inspection and elementary image processing were tedious and prone to errors. Machine learning algorithms like Neural Networks, SVMs, and decision trees played a vital role in this regard by automating the feature analysis and concentrating on texture, bleed lines, and number panels. ORB and SSIM, thus allowing advanced feature extraction and higher accuracy. Previous research recorded outstanding achievements in deploying CNNs, hyperspectral imaging, and OCR to withstand inflation by different currencies. This clearly indicates that machine learning is a suitable technology for counteracting the increasing trend of counterfeiting.

[11]“Fake Currency Detection Using Convolution Neural Network” indicates improvements in artificial intelligence for counterfeiting detection. Previous models mostly depended on human inspection and simple image processing, which was inaccurate and inefficient. However, recent methods use deep learning and Convolutional Neural Networks (CNNs) in particular to investigate features such as edges, textures, and patterns for classification.

Earlier research covered several models, including custom CNNs, predetermined architectures (e.g. ResNet, AlexNet), and competitive methods using algorithms such as KNN and SVM. Despite the success of accurate identification, introducing the solution to different types of currencies and imaging conditions is a real technical challenge. Nevertheless, the work accomplished is more than just the results of these studies; they introduce a CNN model that was trained with a dataset comprising a whole range of example data, enhancing the detection of faults and reliability.

[12]Counterfeit currency detection is considered critical by economists, and traditional methods, whether hardware-based or those using simple image processing techniques, have essentially focused on characteristics related to size, texture, and security features such as watermarks. Unfortunately, these methods have some setbacks in feature extraction and processing efficiency. Modern techniques utilize deep learning, especially CNNs and transfer learning with models such as fine-tuned AlexNet, and excel more in performing feature extraction and classification with optimization techniques such as Adam. Recent systems achieve accuracies of 81.5% for real notes and 75% for counterfeits, surpassing traditional methods by 26%. Future research aims to improve detection through advanced preprocessing, noise reduction, and surface pattern analysis.

Motivation:

Throughout history, currency issuers have faced one common threat: the threat of counterfeiting. Despite the introduction of electronic currency, banknotes remain in abundance. The amount of counterfeit currency in circulation at any moment threatens the confidence in the currency. Counterfeit currency threatens economies worldwide, particularly in developing nations like Bangladesh. A fake currency detection system varies depending on the specific features of a country's bank notes. For Bangladeshi bank notes, the following features are considered: Microprinting, watermark, optically variable ink, iridescent ink, security thread, ultraviolet lines, and many detection systems. However, counterfeiting has been heavily reduced by using those features. Therefore, existing detection systems are either too costly, limited in availability, or insufficiently accurate. A simple survey was conducted at Narayanganj City in 2016 about fake notes on randomly selected people of various ages and professions of 2000 people with some questionnaires. This survey emphasized two sections:

1. How many people are aware of fake notes?

2. How many people are familiar with the features of genuine notes?

Sl.	Question	Positive Response	Negatives Response
1	Do you have any idea about fake notes?	90%	10%
2	Are you able to make difference between Fake and Genuine notes?	60%	40%

Fig 1 - A sample survey result

(Source: Raising Ethics regarding Fake Notes: Impact on our Economy)

So, this project seeks to address these challenges by creating an affordable solution that anyone can use for personal or commercial work and that is easy to use by phone—a website-based, user-friendly solution. By leveraging modern image processing techniques, this project aims to provide a counterfeit detection tool that is accurate and accessible to a broader audience, ultimately mitigating the economic and social damages caused by counterfeit currency circulation.

Aim/Goal:

This project aims to develop a robust, cost-effective, web-based system that is reliable and effective in detecting counterfeit Bangladeshi bank notes using Image processing techniques. This project will focus on identifying unique and distinguishing features. Bangladeshi Bank notes have special features like Microprinting, watermarks, optically variable ink, iridescent ink, security thread, and ultraviolet lines. This web-based project detects fake currency by extracting existing features of banknotes, such as micro-printing and optical variables. Ink (OVI), watermark, iridescent ink, security thread, and ultraviolet lines using OCR (Optical Character Recognition). Analysis, Face Recognition, Speeded UP Robust Features (SURF), and Canny Edge & Hough transformation for watermark and security thread verification.

Therefore, the goal is to provide a reliable solution that can be widely adopted across various sectors and ensure high accuracy, speed, and affordability in the counterfeit detection process, providing a scalable and user-friendly solution suitable for use in Banks, retail stores, and commercial sectors. Personal use for individuals concerned about counterfeit currency.

Specification/Features of the project:

Our AI project will detect whether a note is counterfeit by uploading images on the website. This is the feature of this project.

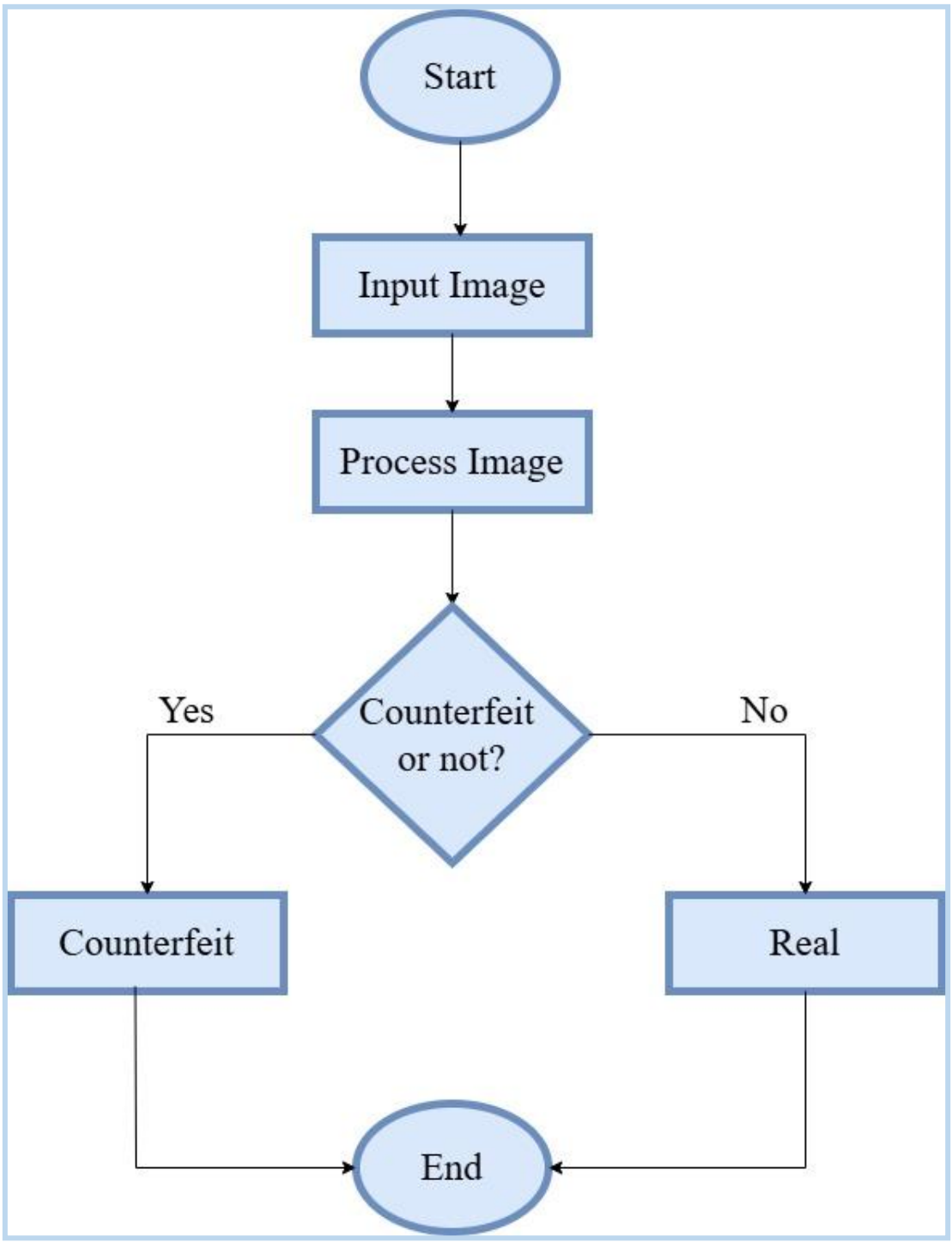


Fig 2 – Algorithm for counterfeit note detection

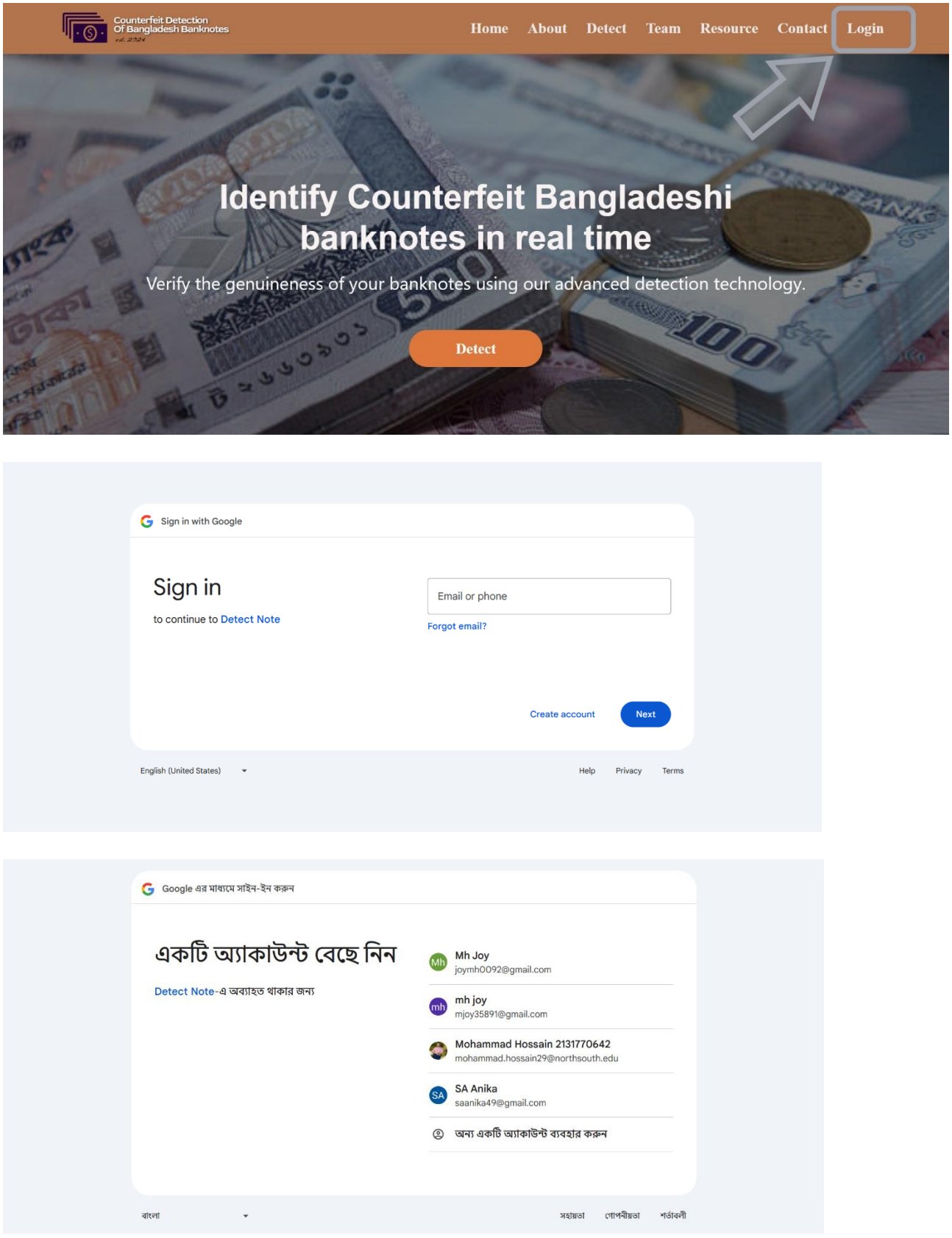


Fig 3 – Google Authentication

Google authentication is another crucial component of the project. Users can use their Google account to log in.

AI Detect Section:

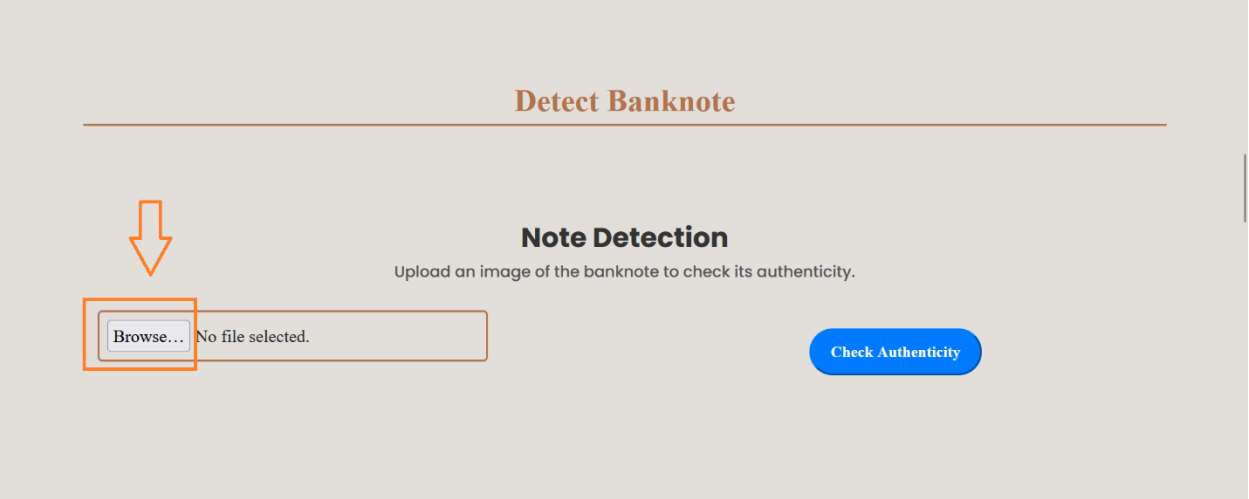


Fig 4 – AI Detection Part

Result-Note Detection (Predict Real note):

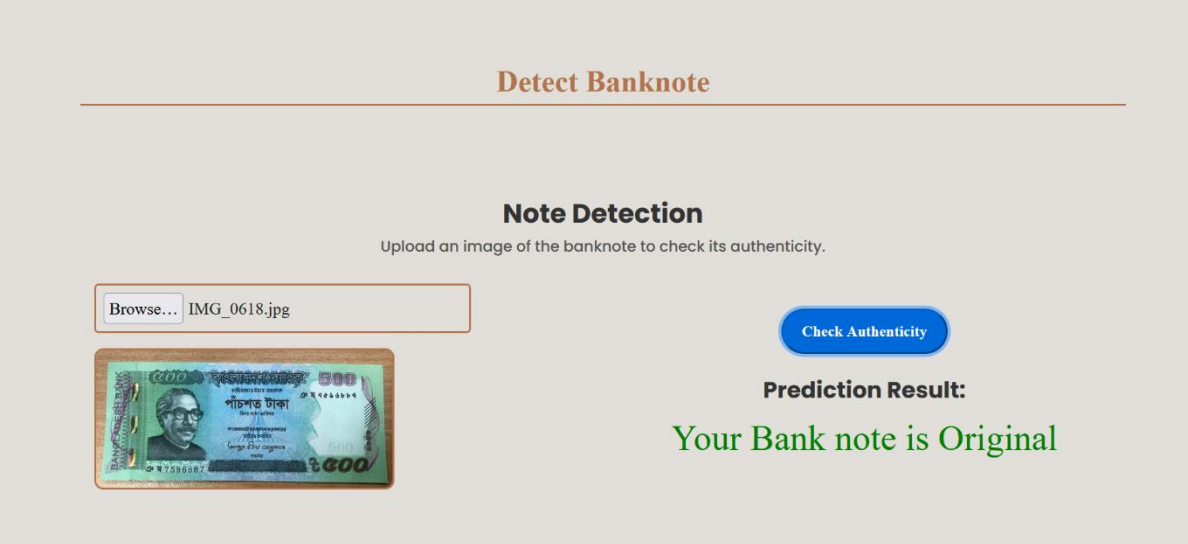


Fig 5- AI Detection (Predict Real note)

Result-Note Detection (Predict Fake note):

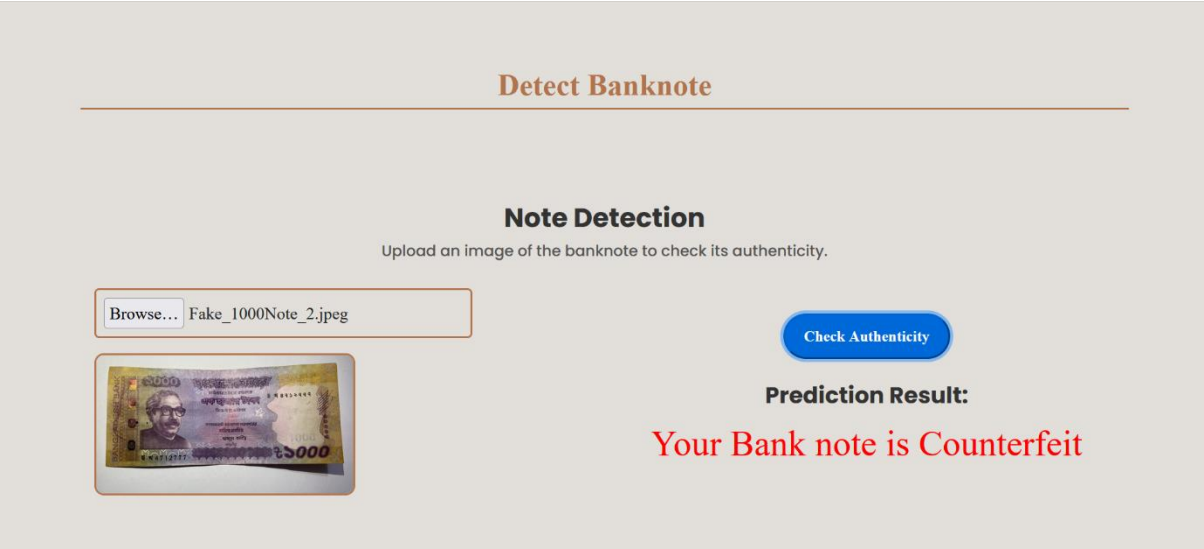


Fig 6 - AI Detection (Predict Fake note)

Application of the project in real life:

Basically, this method solves a major issue with counterfeit money by detecting false notes of 500 and 1000. It plays an essential role in the banking sector during cash deposits and withdrawals, ensuring that only genuine high-value notes enter the financial system. For businesses, especially those with large transactions, this system offers effective solutions at point-of-sale (POS) terminals that help provide additional security and protect their financial interests.

The system's ease of use has made it more acceptable to individual users, allowing them to verify banknotes confidently and avoid personal financial loss. Law enforcement authorities can benefit from detecting and managing the circulation of counterfeit 500 and 1000 notes to prevent fraudulent actions. It may also be connected with e-commerce platforms' cash-on-delivery services, making transactions more safe and dependable. By focusing on these specific notes, the system plays an important role in reducing financial corruption in the country and increasing public confidence.

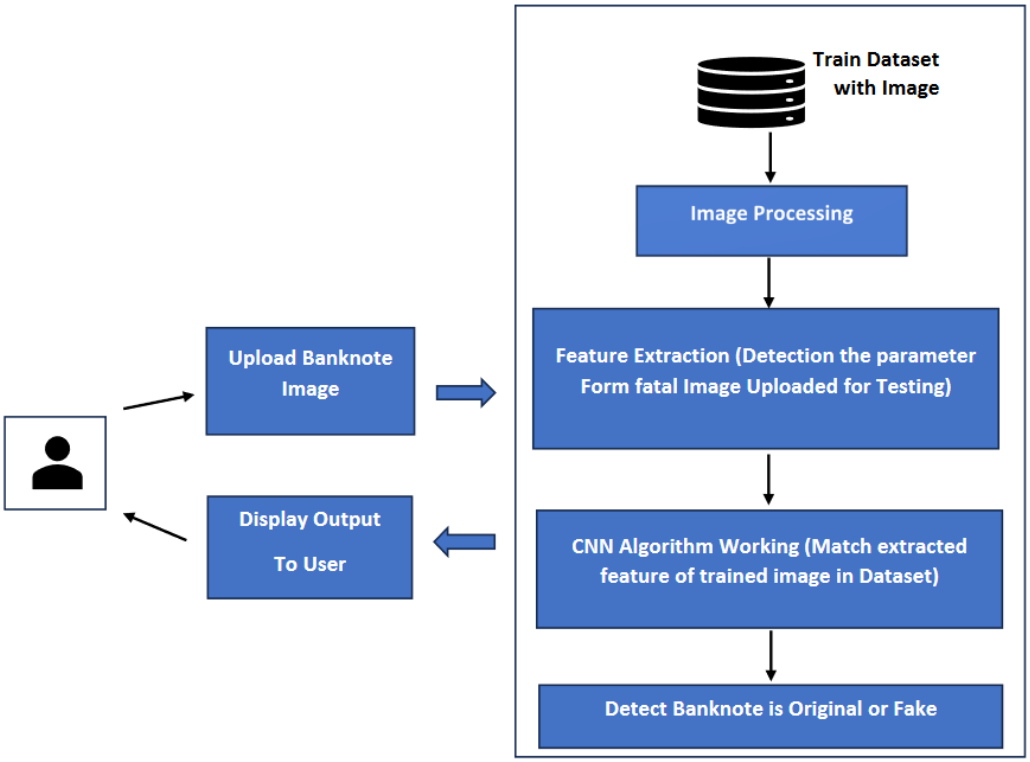


Fig: System Architecture Block Diagram

Note Detection (Predict Real note):

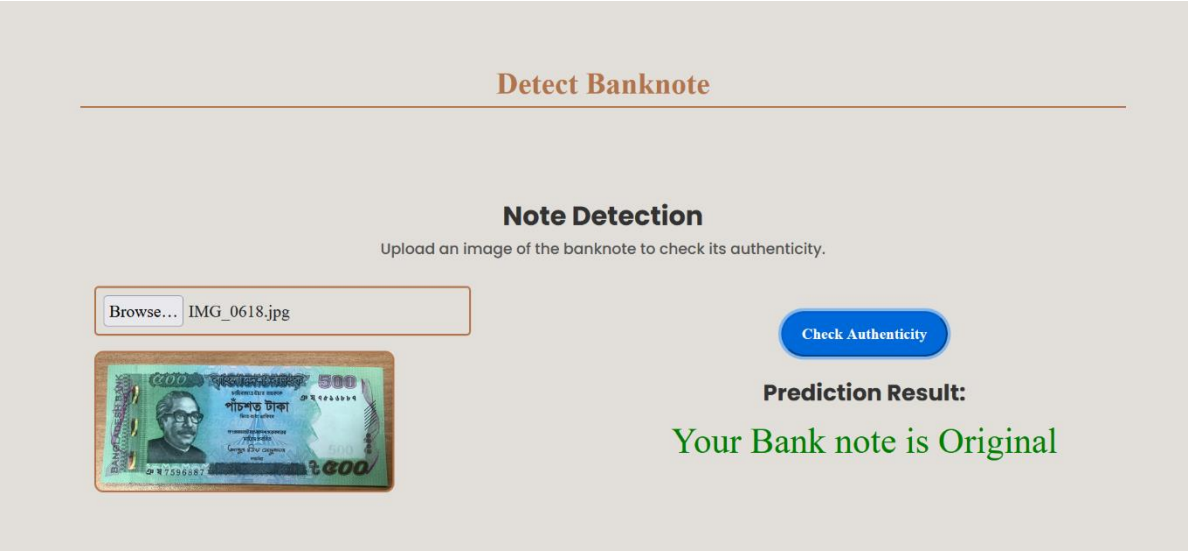


Fig 7- Note Detection (Predict Real note)

Note Detection (Predict Fake note):

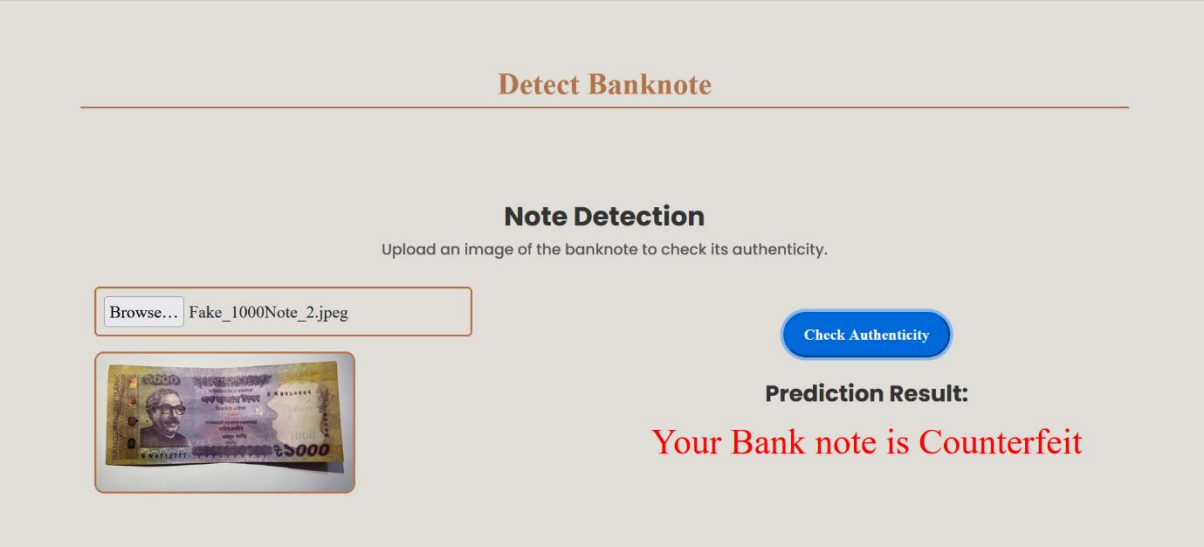


Fig 8- Note Detection (Predict Fake note)

Tools needed and short description:

As we build a web application implementing machine learning, we can divide our project into Frontend, Backend, and (Machine learning and Deep learning models).

Frontend

- **HTML :** For Structure the web application.
- **CSS:** For basic design and styling.
- **Javascript:** For dynamic and interactive elements
- **Bootstrap:** For responsive and mobile friendly design

Backend

- **Python Flask:**To connect the machine learning model with the web application and handle user authentication
- **Google Authentication:** For secure user login using Google accounts.

Machine Learning Model

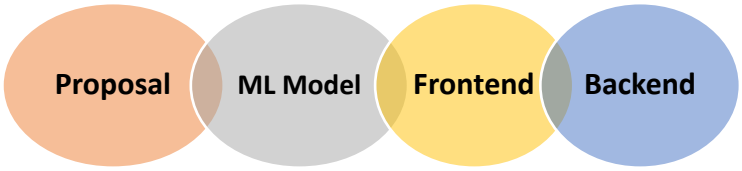
- **Python:** Used for model development
  Frameworks and Libraries :
- **NumPy:** For numarical computation.
- **Pandas:** For data preprocessing
- **Matplotlib:** For visualizing performance metrics
- **Scikit-learn:** For preprocessing and evaluation
- **PyTorch**: To train the Model
- **OpenCV:** For image Preprocessing
- **TensorFlow Lite (TFLite)** : To convert the trained model into a lightweight format.

Additional Tools

- **GitHub**: For version control.
- **VS Code**: Used as the code editor for developing the web application
- **Google Colab**: For training and testing the machine learning model.

Tentative Schematic diagram:

The project implementation can be divided into four major parts. The tentative flow diagram shows the flow of project works and implementations.



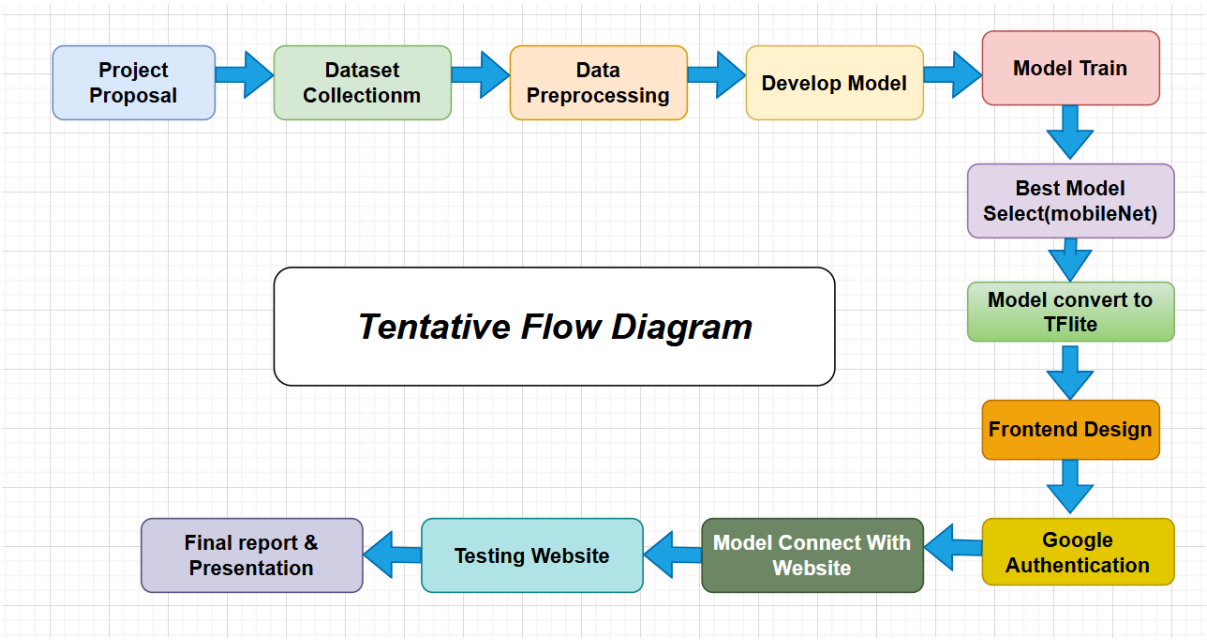


Fig 9 - Flow of project works and implementations

That flow diagram shows tentative steps from the project proposal to the final presentation. It gives a proper idea about the implementation of the project.

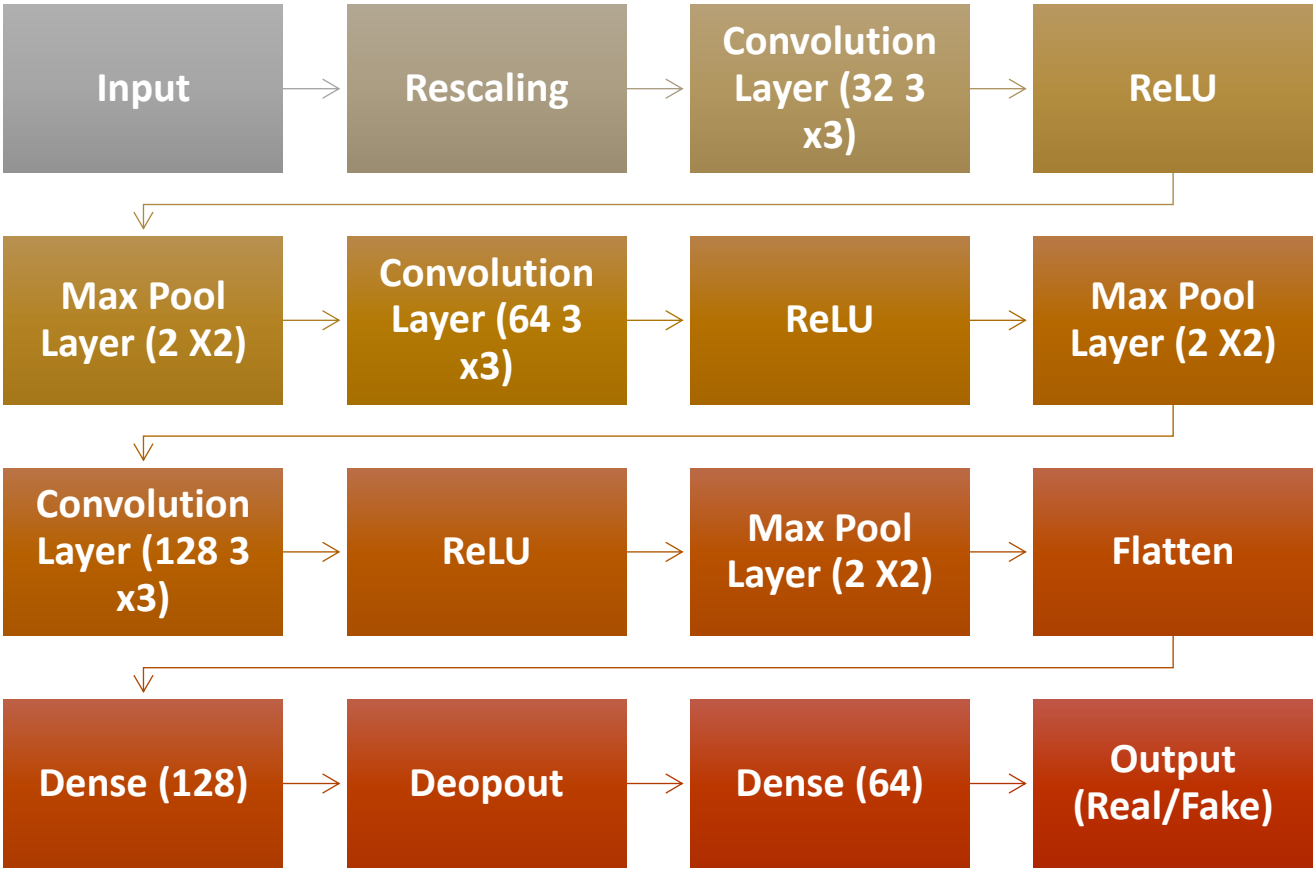


Fig 10 - Tentative flow diagram for CNN

Project Plan:

Our project aims to create a system using software to identify fake Bangladesh banknotes during detection. Users can utilize the system via a web platform to submit a banknote image to verify and detect counterfeit banknotes. We will complete our project in several steps. For this, we will complete our project by planning a good project.

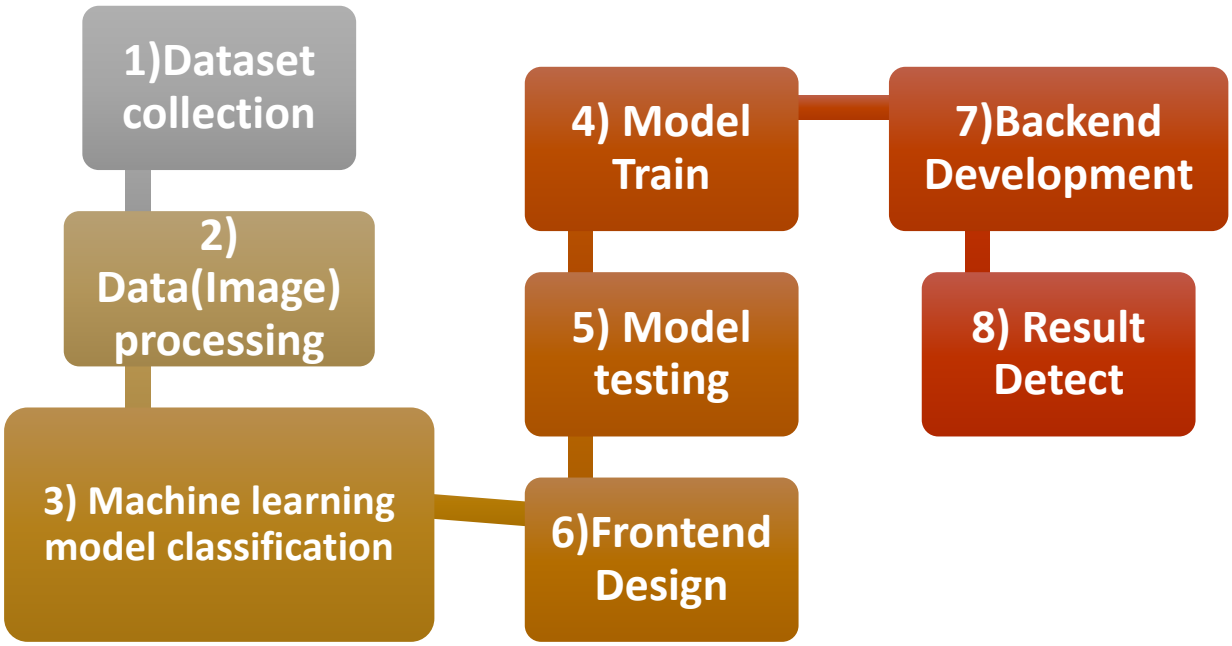


Fig 11 – Project Plan

First, we must collect a dataset for Counterfeit Detection on the Bangladesh Banknote project. Next, we need to do image processing on that data. Then, the meshing learning model classifies that model. After training, the model should be tested. Then, we connect the trained model with the backend of our web application, and we can achieve our results.

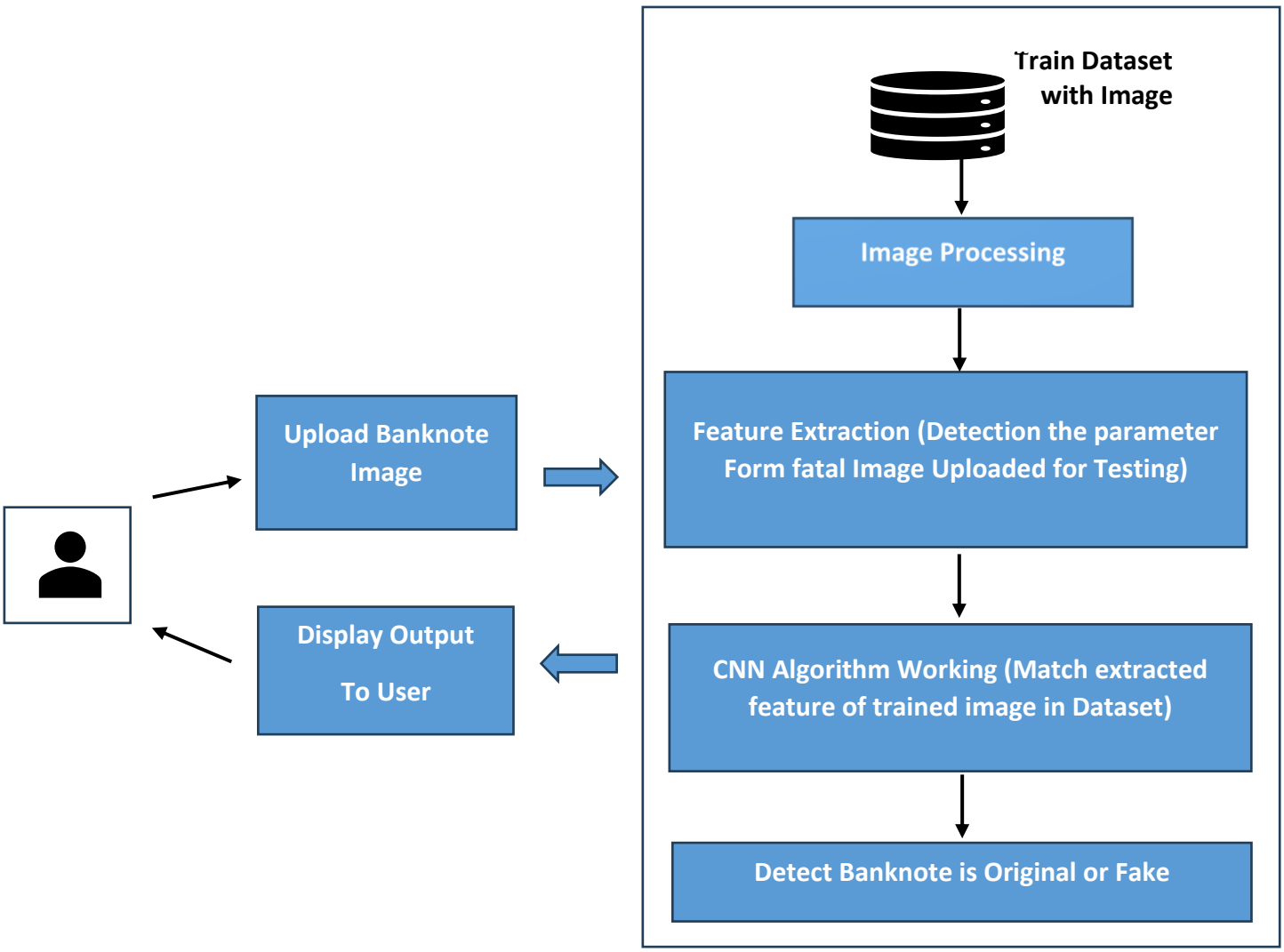


Fig 12 - System Architecture Block Diagram

Methodology:

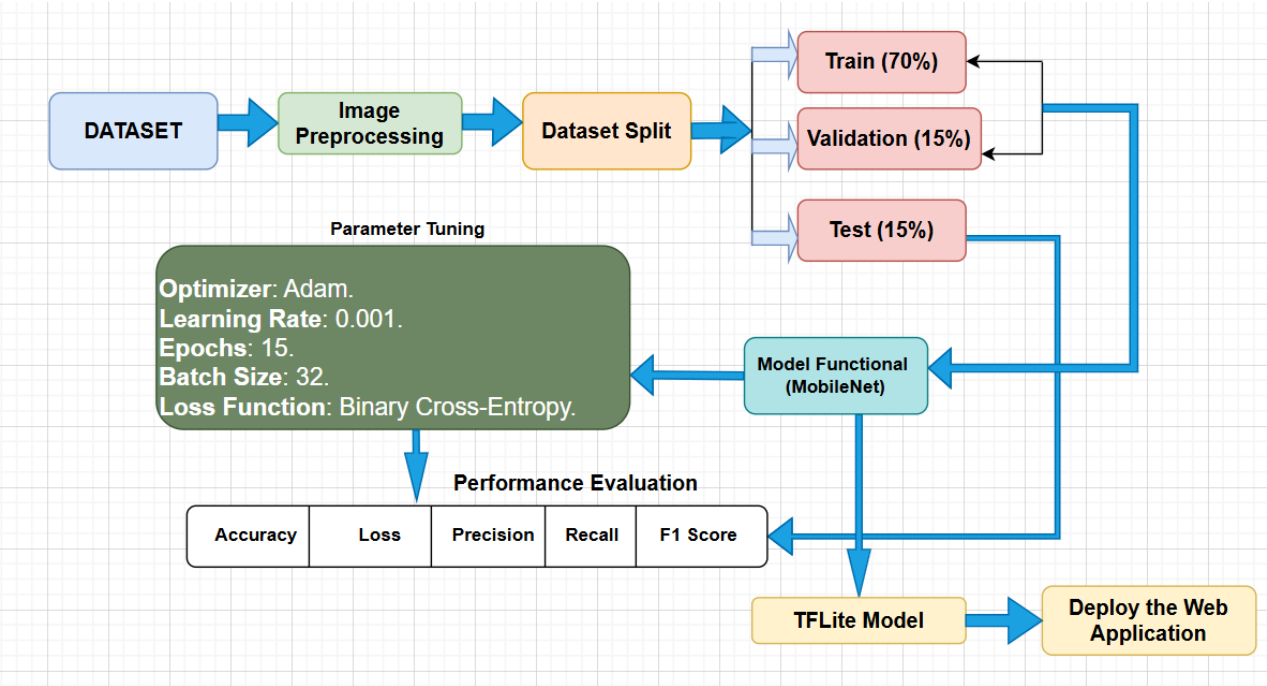


Fig 13 - Pipeline of the project.

Dataset Use:

A dataset from the Kaggle repository is

(Available at: https://www.kaggle.com/datasets/bibhaschowdhury/bangladeshi-banknote-authentication-dataset/data)

Used to validate the MobileNet model's performance for counterfeit detection. Images of 500 Taka and 1000 Taka Bangladeshi banknotes are included in the dataset. There are two class designations for each denomination: authentic and false. The meticulously selected dataset provides a solid basis for training and testing the model, which guarantees a fair representation of genuine and fake banknotes.

Fig 14 - Sample authentic images.

Fig 15 - Sample fake images.

Image Preprocessing:

This study used an efficient and systematic picture preprocessing methodology to prepare the dataset for training and testing. Both authentic and phony Bangladeshi banknotes went through the following steps:

Noise Reduction: Gaussian Blur was applied to remove noise and create smooth images.

Normalization: Scaled pixel values between 0 and 255 for uniform brightness and contrast.

Augmentation:

- **Rotation**: Rotated by -15°, 0°, and 15°.
- **Flipping**: Applied horizontal flips.
- **Center Cropping**: Retained 80% of the central region for key features.

Format Consistency: Converted all non-RGB images to RGB format.

```
#Save the images in google drive after preprocessing
real_directory = '/content/drive/MyDrive/Main Dataset/Processed_image/Real_processed_images/'

process_images(Real_notes, real_directory)
```

Fig - Real note image save directory

```
fake_directory = '/content/drive/MyDrive/Main Dataset/Processed_image/fake_processed_images/'
process_images(Fake_notes, fake_directory)
```

Fig 16 - Fake note image save directory

DataSplit:

Using the split-folder library, the dataset is split into train (70%), test (15%) and validation (15%).

```
#Directories for the source paths containing real and fake note after image preprocessing
Real_directory = '/content/drive/MyDrive/Main Dataset/Processed_image/Real_processed_images'
Fake_directory = '/content/drive/MyDrive/Main Dataset/Processed_image/fake_processed_images'

#Directories for train, validation, and test
Data_split_directory = '/content/drive/MyDrive/Main Dataset/data_split/'

Train_directory = os.path.join(Data_split_directory, 'Train Data/')
Validation_directory = os.path.join(Data_split_directory, 'Validation Data/')
Test_directory = os.path.join(Data_split_directory, 'Test Data/')
```

Fig 17 - Data split directory

```
Found 6384 images belonging to 2 classes.
Found 1368 images belonging to 2 classes.
Found 1368 images belonging to 2 classes.
```

Fig 18 - After the Data split train, validation, test

Feature Extraction:

In this project, the MobileNet-based Convolutional Neural Network (CNN) model was used to feature extraction to classify Bangladeshi banknotes as **real** or **fake**. The model architecture leverages MobileNet's lightweight and efficient design, which includes depthwise separable convolutions for reduced computational complexity.

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
conv1 (Conv2D)	(None, 112, 112, 32)	864
conv1_bn (BatchNormalization)	(None, 112, 112, 32)	128
conv1_relu (ReLU)	(None, 112, 112, 32)	0
conv_dw_1 (DepthwiseConv2D)	(None, 112, 112, 32)	288
conv_dw_1_bn (BatchNormalization)	(None, 112, 112, 32)	128
conv_dw_1_relu (ReLU)	(None, 112, 112, 32)	0
conv_pw_1 (Conv2D)	(None, 112, 112, 64)	2,048
conv_pw_1_bn (BatchNormalization)	(None, 112, 112, 64)	256
conv_pw_1_relu (ReLU)	(None, 112, 112, 64)	0
conv_pad_2 (ZeroPadding2D)	(None, 113, 113, 64)	0
conv_dw_2 (DepthwiseConv2D)	(None, 56, 56, 64)	576
conv_dw_2_bn (BatchNormalization)	(None, 56, 56, 64)	256
conv_dw_2_relu (ReLU)	(None, 56, 56, 64)	0
conv_pw_2 (Conv2D)	(None, 56, 56, 128)	8,192
conv_pw_2_bn (BatchNormalization)	(None, 56, 56, 128)	512
conv_pw_2_relu (ReLU)	(None, 56, 56, 128)	0
conv_dw_3 (DepthwiseConv2D)	(None, 56, 56, 128)	1,152
conv_dw_3_bn (BatchNormalization)	(None, 56, 56, 128)	512
conv_dw_3_relu (ReLU)	(None, 56, 56, 128)	0

conv_pw_6_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_7 (DepthwiseConv2D)	(None, 14, 14, 512)	4,608
conv_dw_7_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_dw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_7 (Conv2D)	(None, 14, 14, 512)	262,144
conv_pw_7_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_pw_7_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_8 (DepthwiseConv2D)	(None, 14, 14, 512)	4,608
conv_dw_8_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_dw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_8 (Conv2D)	(None, 14, 14, 512)	262,144
conv_pw_8_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_pw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_9 (DepthwiseConv2D)	(None, 14, 14, 512)	4,608
conv_dw_9_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_dw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_9 (Conv2D)	(None, 14, 14, 512)	262,144
conv_pw_9_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_pw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_10 (DepthwiseConv2D)	(None, 14, 14, 512)	4,608
conv_dw_10_bn (BatchNormalization)	(None, 14, 14, 512)	2,048
conv_dw_10_relu (ReLU)	(None, 14, 14, 512)	0
conv_pw_3 (Conv2D)	(None, 56, 56, 128)	16,384
conv_pw_3_bn (BatchNormalization)	(None, 56, 56, 128)	512
conv_pw_3_relu (ReLU)	(None, 56, 56, 128)	0
conv_pad_4 (ZeroPadding2D)	(None, 57, 57, 128)	0
conv_dw_4 (DepthwiseConv2D)	(None, 28, 28, 128)	1,152
conv_dw_4_bn (BatchNormalization)	(None, 28, 28, 128)	512
conv_dw_4_relu (ReLU)	(None, 28, 28, 128)	0
conv_pw_4 (Conv2D)	(None, 28, 28, 256)	32,768
conv_pw_4_bn (BatchNormalization)	(None, 28, 28, 256)	1,024
conv_pw_4_relu (ReLU)	(None, 28, 28, 256)	0
conv_dw_5 (DepthwiseConv2D)	(None, 28, 28, 256)	2,304
conv_dw_5_bn (BatchNormalization)	(None, 28, 28, 256)	1,024
conv_dw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65,536
conv_pw_5_bn (BatchNormalization)	(None, 28, 28, 256)	1,024
conv_pw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pad_6 (ZeroPadding2D)	(None, 29, 29, 256)	0
conv_dw_6 (DepthwiseConv2D)	(None, 14, 14, 256)	2,304
conv_dw_6_bn (BatchNormalization)	(None, 14, 14, 256)	1,024
conv_dw_6_relu (ReLU)	(None, 14, 14, 256)	0
conv_pw_6 (Conv2D)	(None, 14, 14, 512)	131,072
conv_pw_6_bn (BatchNormalization)	(None, 14, 14, 512)	2,048

conv_dw_12_relu (ReLU)	(None, 7, 7, 512)	0
conv_pw_12 (Conv2D)	(None, 7, 7, 1024)	524,288
conv_pw_12_bn (BatchNormalization)	(None, 7, 7, 1024)	4,096
conv_pw_12_relu (ReLU)	(None, 7, 7, 1024)	0
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9,216
conv_dw_13_bn (BatchNormalization)	(None, 7, 7, 1024)	4,096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1,048,576
conv_pw_13_bn (BatchNormalization)	(None, 7, 7, 1024)	4,096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dense (Dense)	(None, 1024)	1,049,600
dense_1 (Dense)	(None, 2)	2,050

Total params: 4,280,516 (16.33 MB)
Trainable params: 1,051,650 (4.01 MB)
Non-trainable params: 3,228,864 (12.32 MB)
Optimizer params: 2 (12.00 B)

Fig 19 - Summary of the Model.

Parameter Tuning:

The model's performance was improved by thorough parameter adjustment. The binary cross-entropy loss function was chosen because it is appropriate for binary classification problems, such as discriminating between genuine and counterfeit banknotes. The Adam optimizer was selected because it efficiently adjusts the model's weights and accelerates the training process. The model was thoroughly evaluated using performance indicators such as accuracy, precision, and recall. Early stopping was used to prevent overfitting, which ensured that the training process would cease if the validation loss did not improve after ten consecutive epochs. These tactics combined to produce a highly optimized and dependable model for counterfeit detection.

```
# Creating the model
model = Model(inputs=base_model.input, outputs=predictions)

model.compile(optimizer=Adam(learning_rate=0.0001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
#Early stopping callback to prevent overfitting
early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

#Train the model
history = model.fit(
    train_generator,
    epochs=10,
    validation_data=validation_generator,
    callbacks=[early_stop]
)
```

Fig 20- Parameter Tuning.

Performance Evaluation:

The MobileNet model was comprehensively evaluated using various metrics and visualizations. The model's performance on the test dataset was assessed by calculating accuracy, precision, recall, F1-score, and ROC-AUC score, ensuring a thorough analysis of its effectiveness in counterfeit detection.

```
#Evaluate the model on the test data
test_loss, test_acc = model.evaluate(test_generator)
print(f'Test accuracy: {test_acc*100:.2f}%')

model.save('/content/drive/MyDrive/Main Dataset/models/Graphmobilenet_transfer_learning_model_for_counterfeit_detection_adam.h5')
```

Fig 21 - Save Model directory

```
Classification Report:
              precision    recall  f1-score   support

        fake       1.00      0.99      0.99       684
        real       0.99      1.00      0.99       684

    accuracy                           0.99      1368
   macro avg       0.99      0.99      0.99      1368
weighted avg       0.99      0.99      0.99      1368

Precision: 0.99
Recall: 0.99
F1 Score: 0.99
```

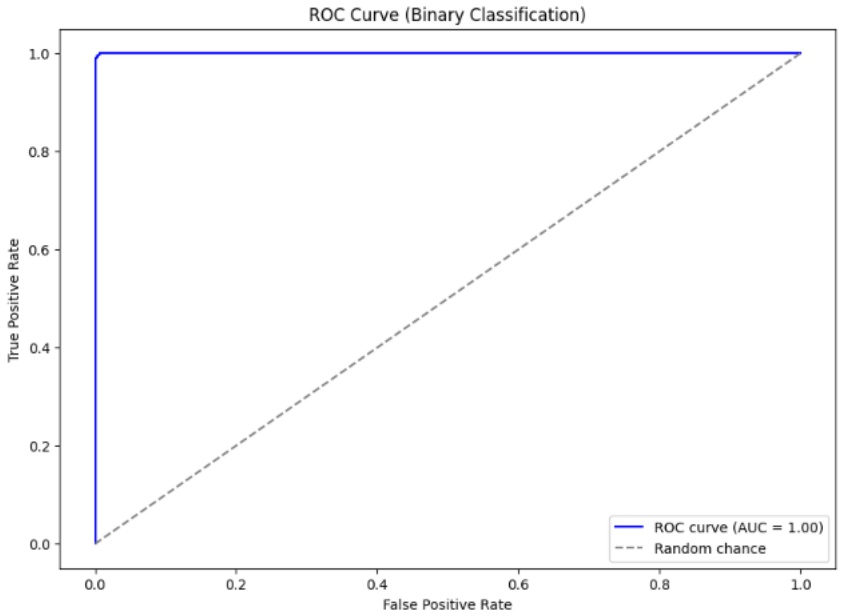


Fig 22 - ROC Curve

```
#Testing the model after model train using images from our dataset

from tensorflow.keras.models import load_model
model = load_model('/content/drive/MyDrive/Main Dataset/models/Graphmobilenet_transfer_learning_model_for_counterfeit_detection_adam.h5')

def predict_banknote(image_path):
    image = preprocess_image(image_path)
    prediction = model.predict(image)

    #[0] = 'fake', [1] = 'real'
    if prediction[0][1] > 0.5:
        print("The banknote is real.")
    else:
        print("The banknote is fake.")

predict_banknote('/content/drive/MyDrive/Main Dataset/Check Note/Picture taken from phone(real)/IMG_0614.jpg')
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step
The banknote is real.
```

Fig 23 - Detection Result

Convert to Lightweight Model and test performance:

First, load the .h5 model and then use the TFLite Converter to transform the model into a smaller, optimized .tflite format. Save the .tflite model. Then, test the performance of the .tflite model.

```
#Load .h5 model
keras_model_path = '/content/drive/MyDrive/Main Dataset/models/mobilenet_transfer_learning_model_for_counterfeit_detection.h5'
model = tf.keras.models.load_model(keras_model_path)

#Convert the model to TFLite to connect with website
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()

#Save the TFLite model
tflite_model_path = '/content/drive/MyDrive/Main Dataset/TFlite Model/mobilenet_transfer_learning_model_for_counterfeit_detection_quantized.tflite'
with open(tflite_model_path, 'wb') as f:
    f.write(tflite_model)
```

Fig 24 - Convert tflite model.

Design Implementation:

Home page without Login:

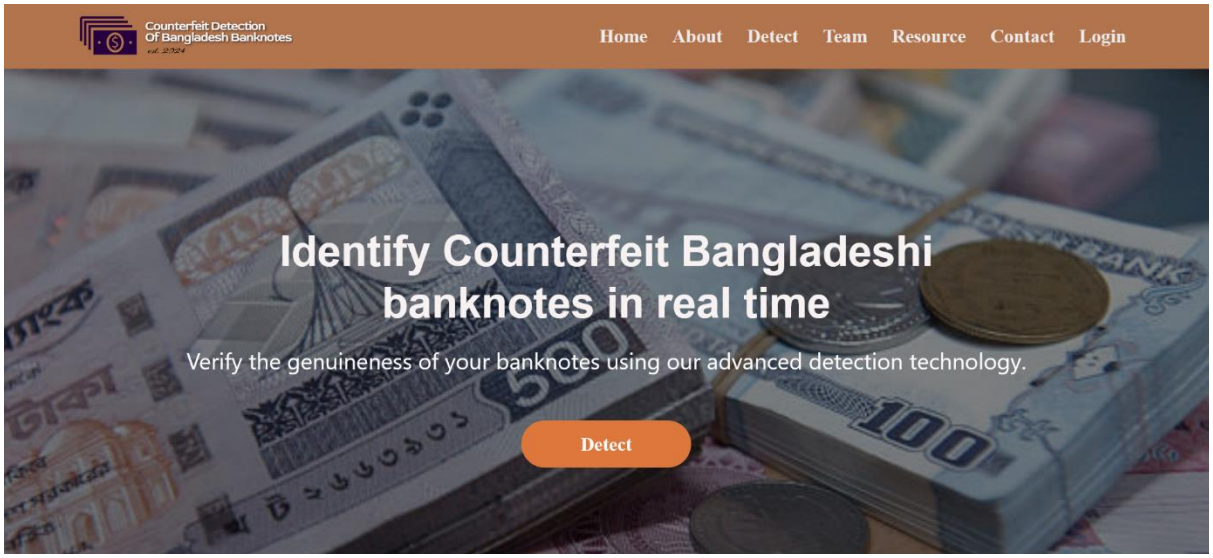


Fig 25 – Home page without Login

Google Authentication:

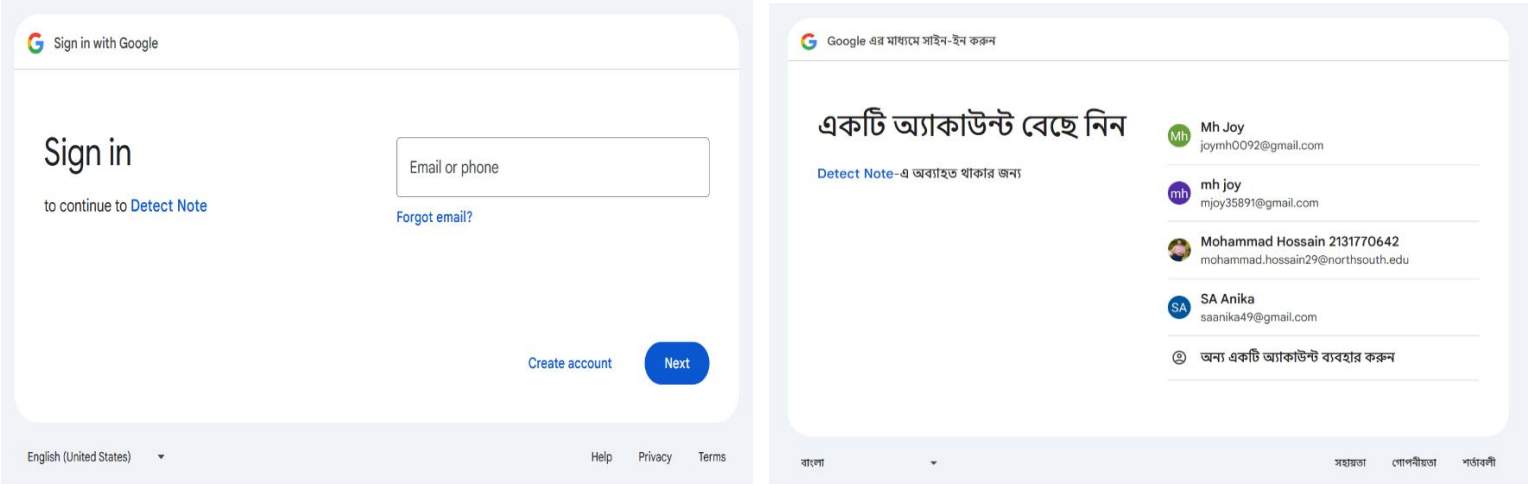


Fig 26 – Google Authentication

Home page with Login after Google Authentication:

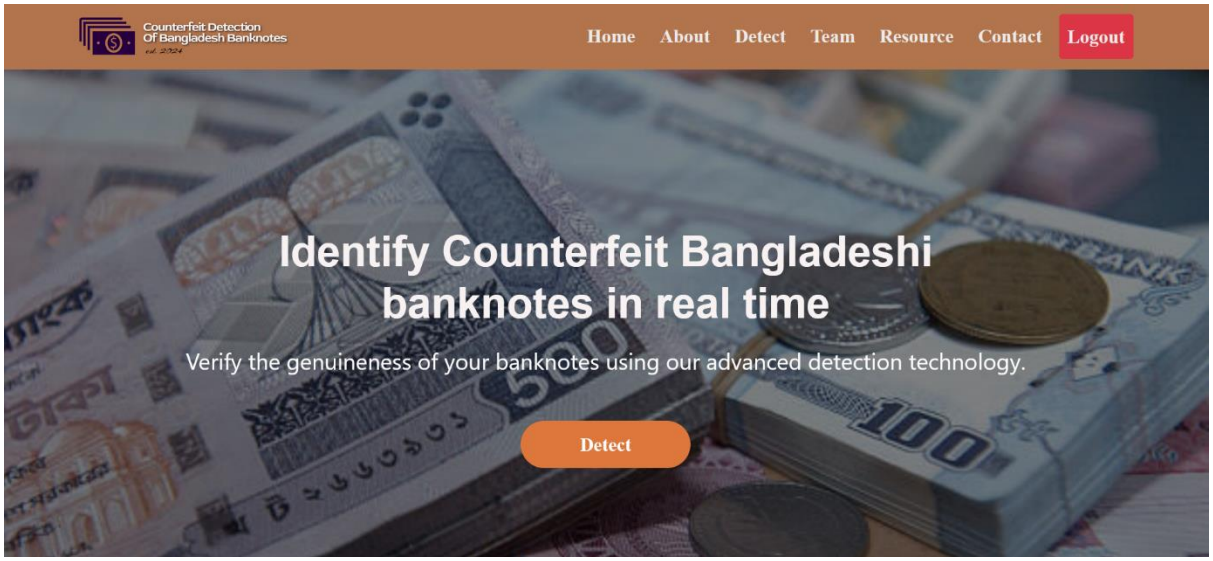


Fig 27 - Home page with Login after Google Authentication

About:

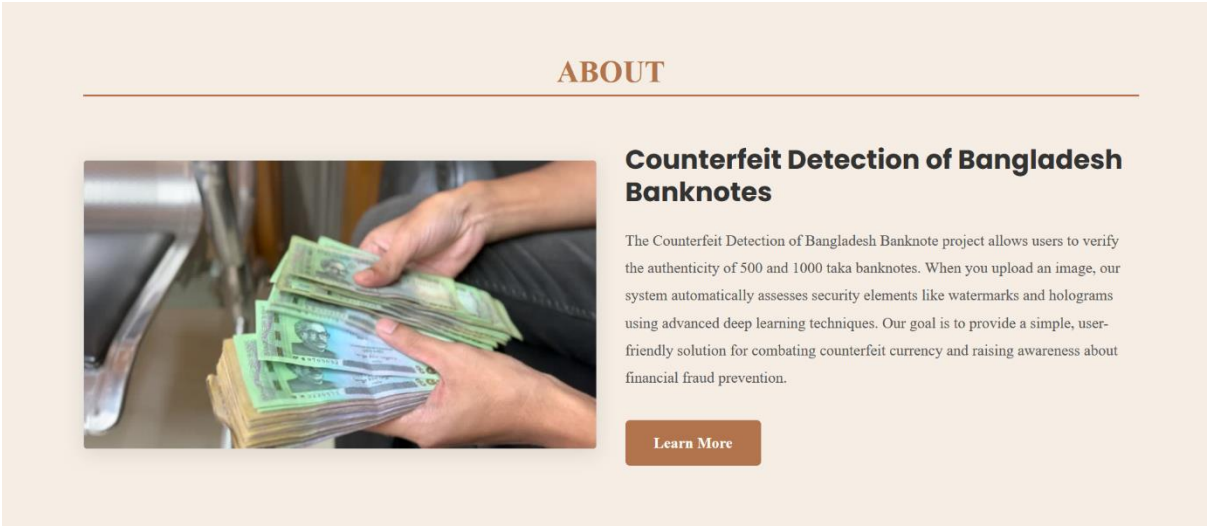


Fig 28 - About

Note Detection (Select a file):

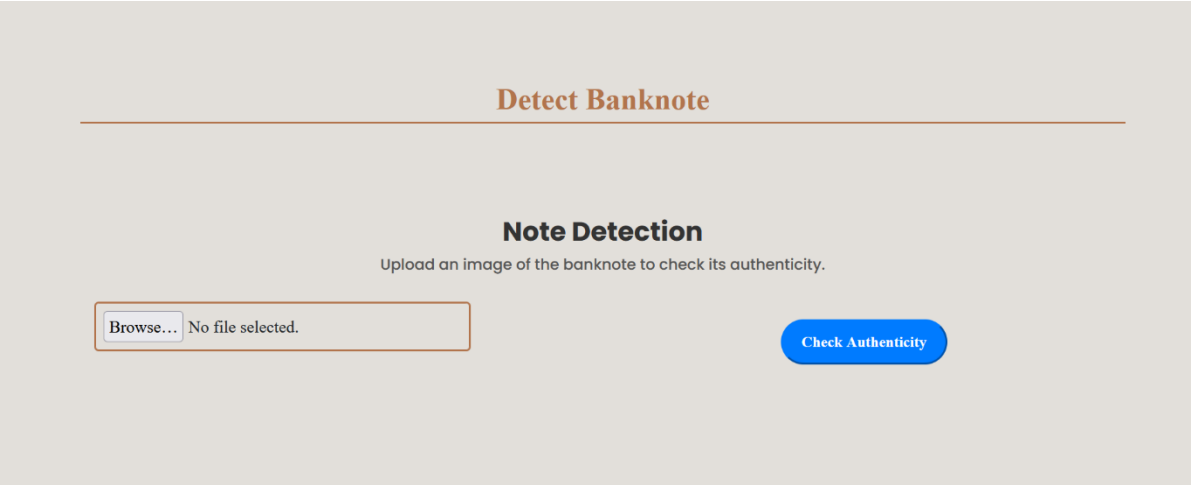


Fig 29 -Note Detection (Select a file)

Note Detection (Upload an image):

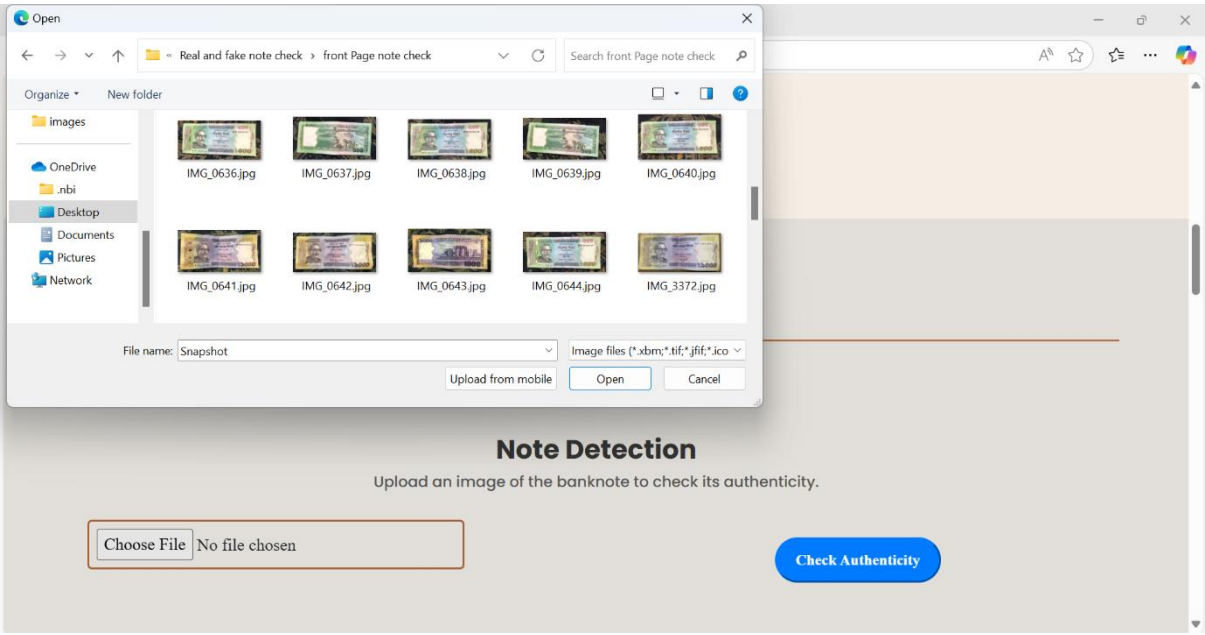


Fig 30 -Note Detection (Upload an image)

Note Detection (Predict Real note):

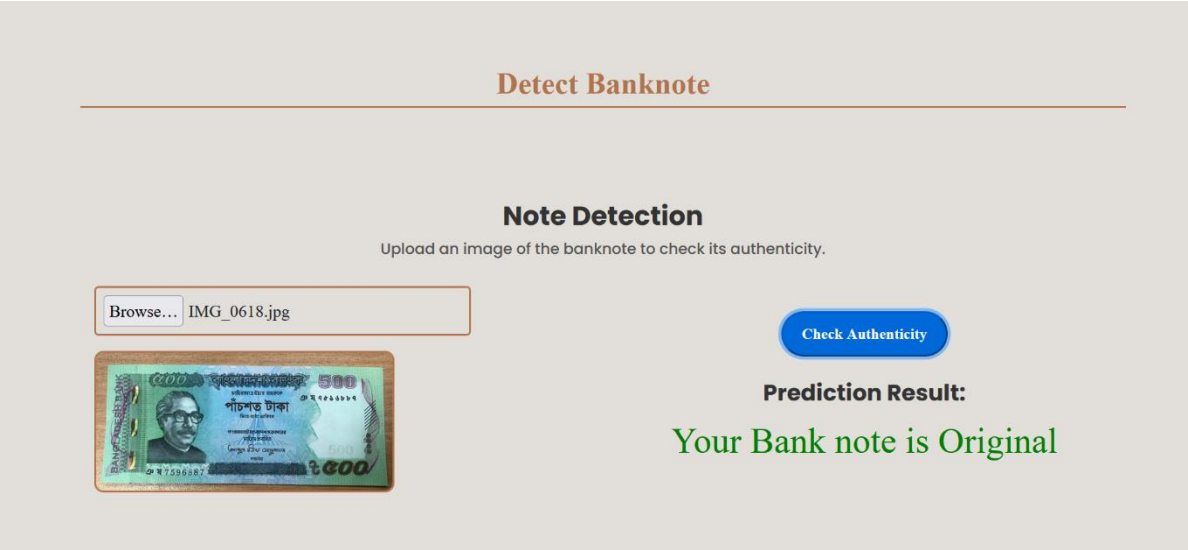


Fig 31 - Note Detection (Predict Real note)

Note Detection (Predict Fake note):

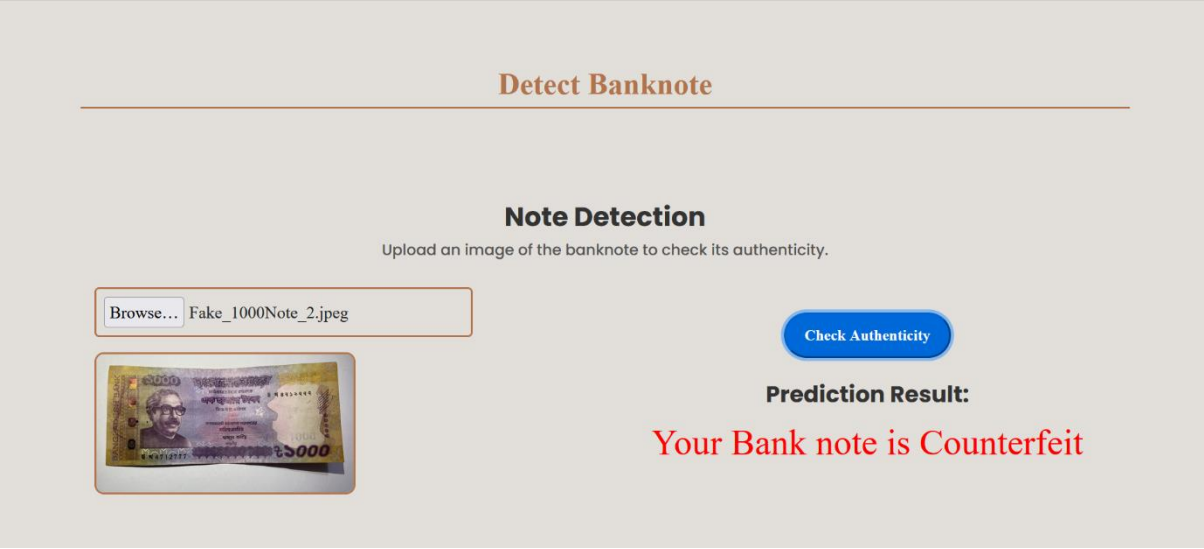


Fig 32 - Note Detection (Predict fake note)

Our Team:

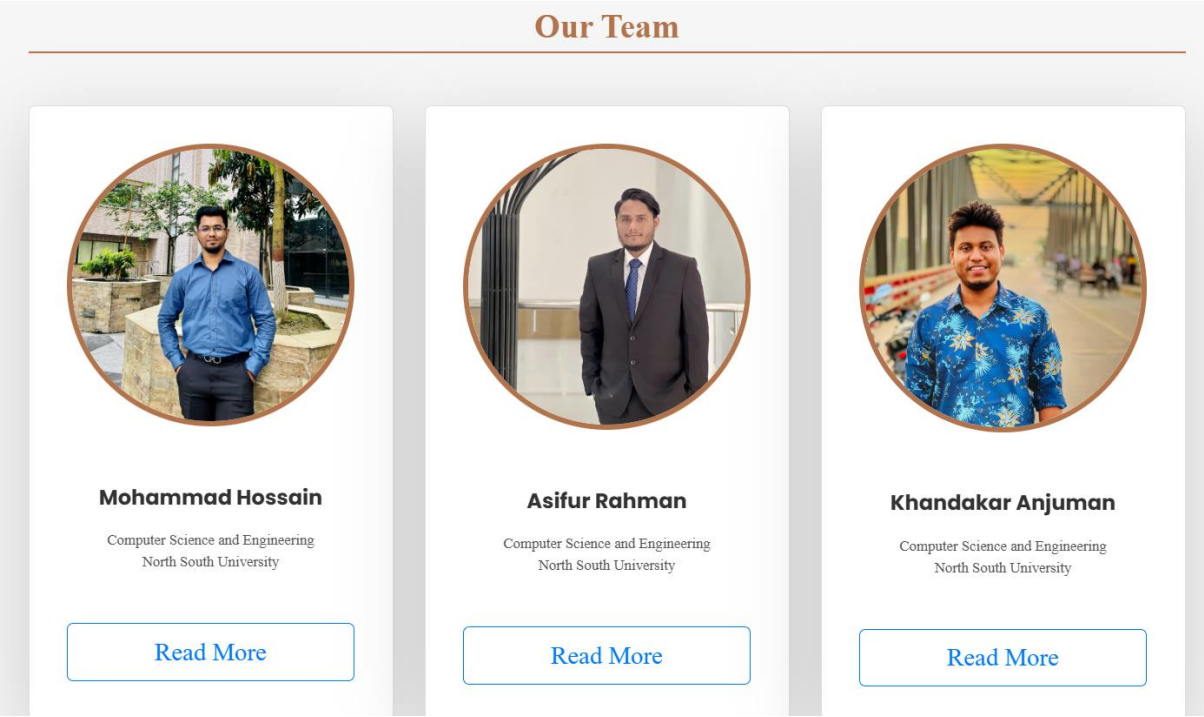


Fig 33 - Our team

Educational Resource:

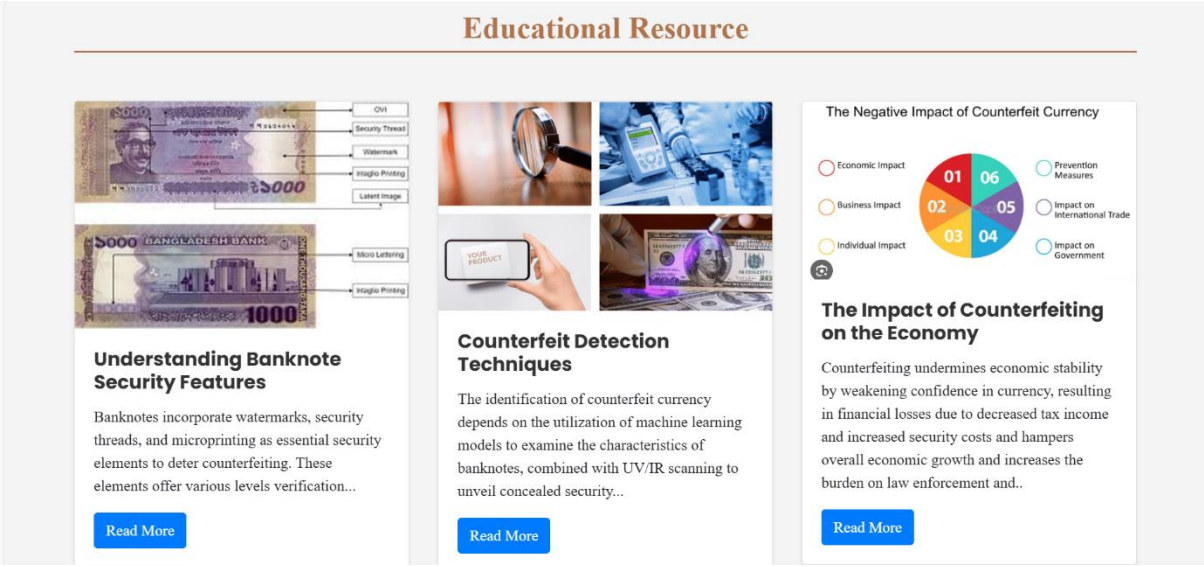


Fig 34 – Educational Resource

Contact Us:

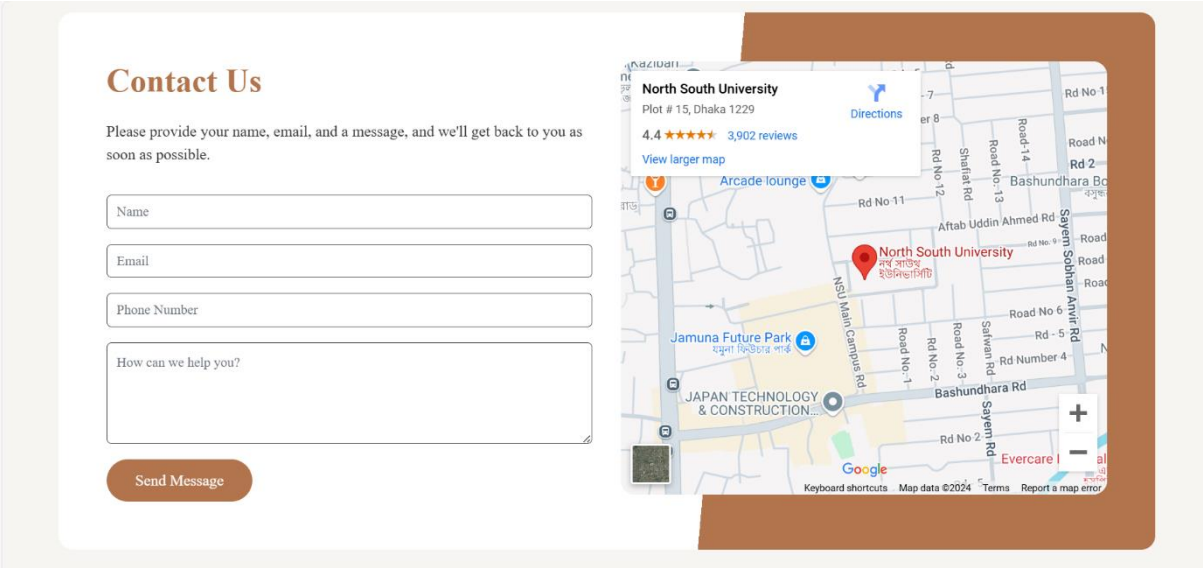


Fig 35 – Contact Us

Footer Section:

Fig 36 – Footer Part

Results Analysis and Evaluation:

Results of Different Models:

Model	Accuracy	Precision	Recall	F1 score	AUC	Size(MB)
MobileNet	0.9908	0.99	0.99	0.99	1.00	24.6
Resnet50	0.9954	0.98	1.0	0.99	0.99	93.5
DenseNet121	0.9912	0.99	0.99	0.99	1.00	29.6
Vgg16	0.9234	0.94	0.94	0.94	0.95	60.3
MobileNetV2	0.9992	1.00	1.00	1.00	1.00	47.1
InceptionV3	0.9965	0.99	0.99	0.99	1.00	193

Table 1 – Model Result

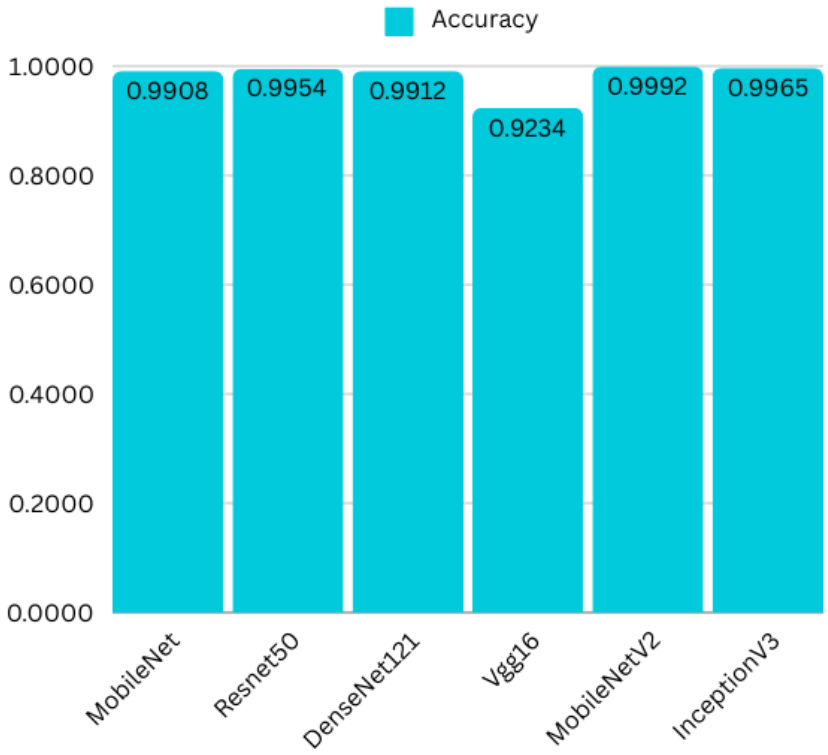


Fig 37 – Accuracy Bar Chart

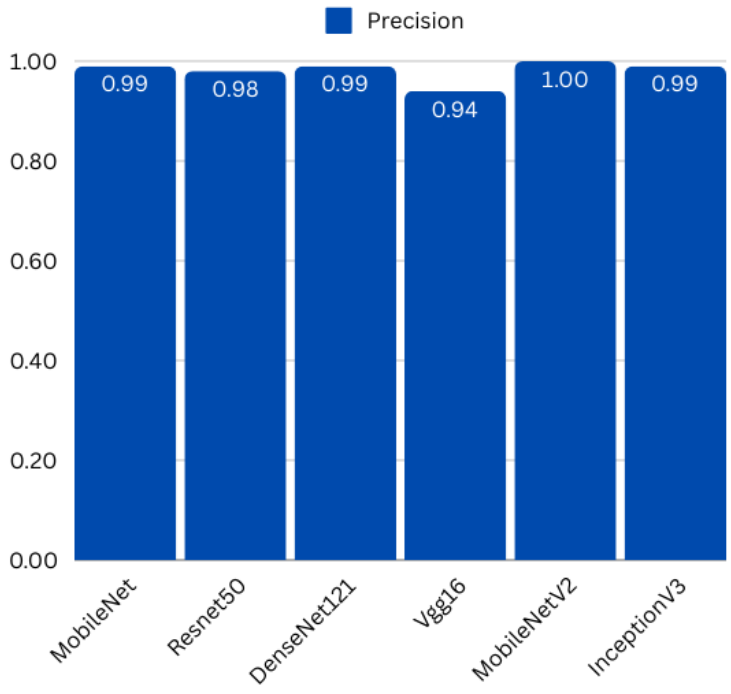


Fig 38 – Precision Bar Chart

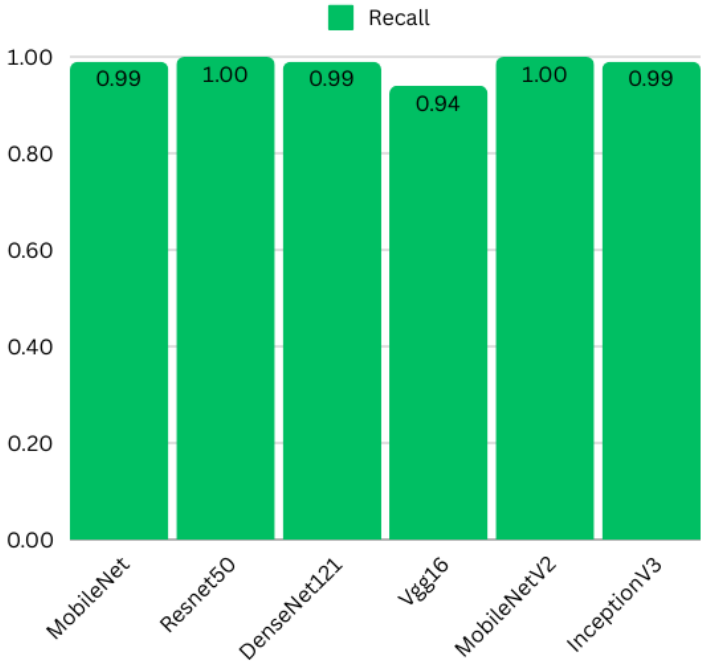


Fig 39 – Recall Bar Chart

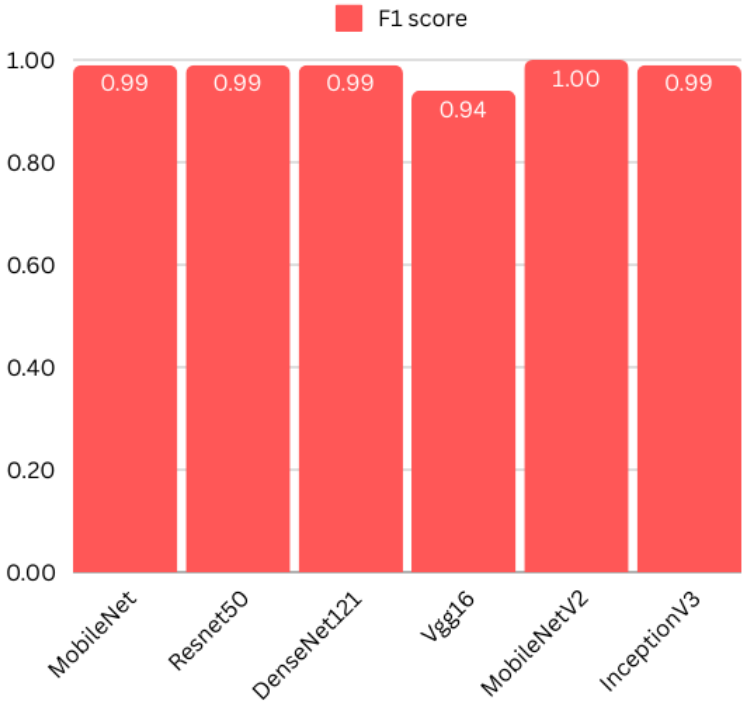


Fig 40 – F1 score Bar Chart

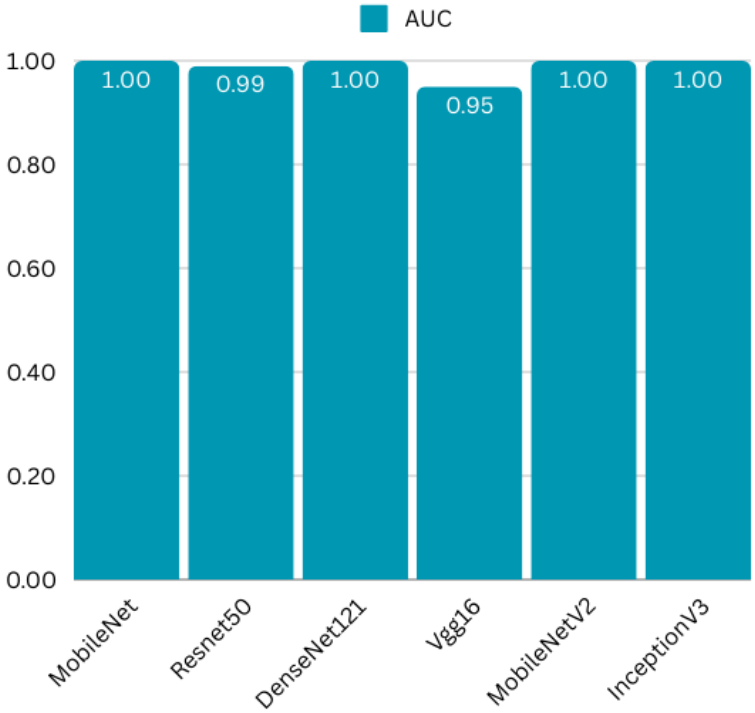


Fig 41 – AUC Bar Chart

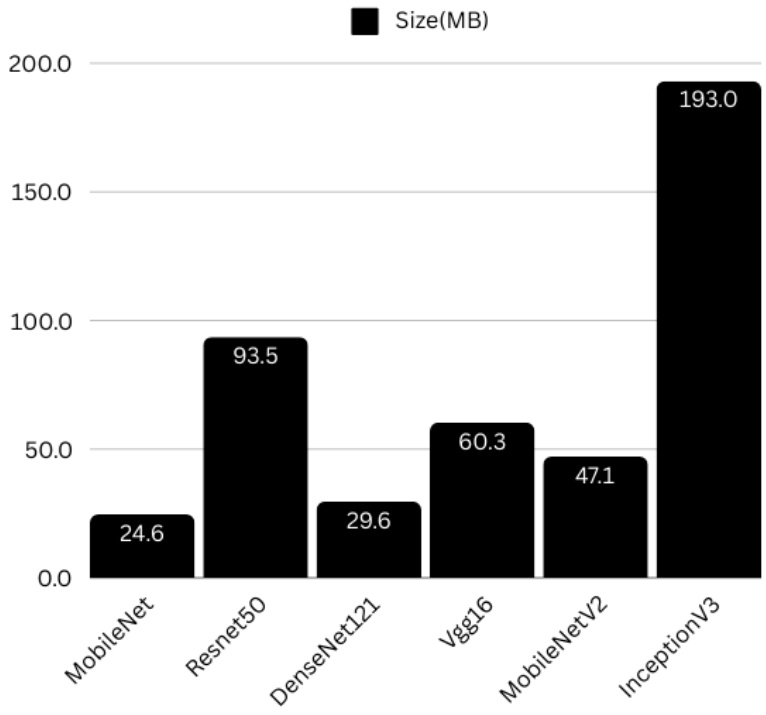


Fig 42 – Size Bar Chart

Every model gave almost the same accuracy, precision, recall, F1 score and AUC except VGG16. We chose MobileNet because its size is very small compared to other models, and a miniature model in case of size gives faster results.

Result for different optimizers:

Optimizer	Epochs	Accuracy	Validation Loss
SGD	10	0.9885	0.0302
Adam	10	0.9908	0.0192

Table 2 – Optimizer Result

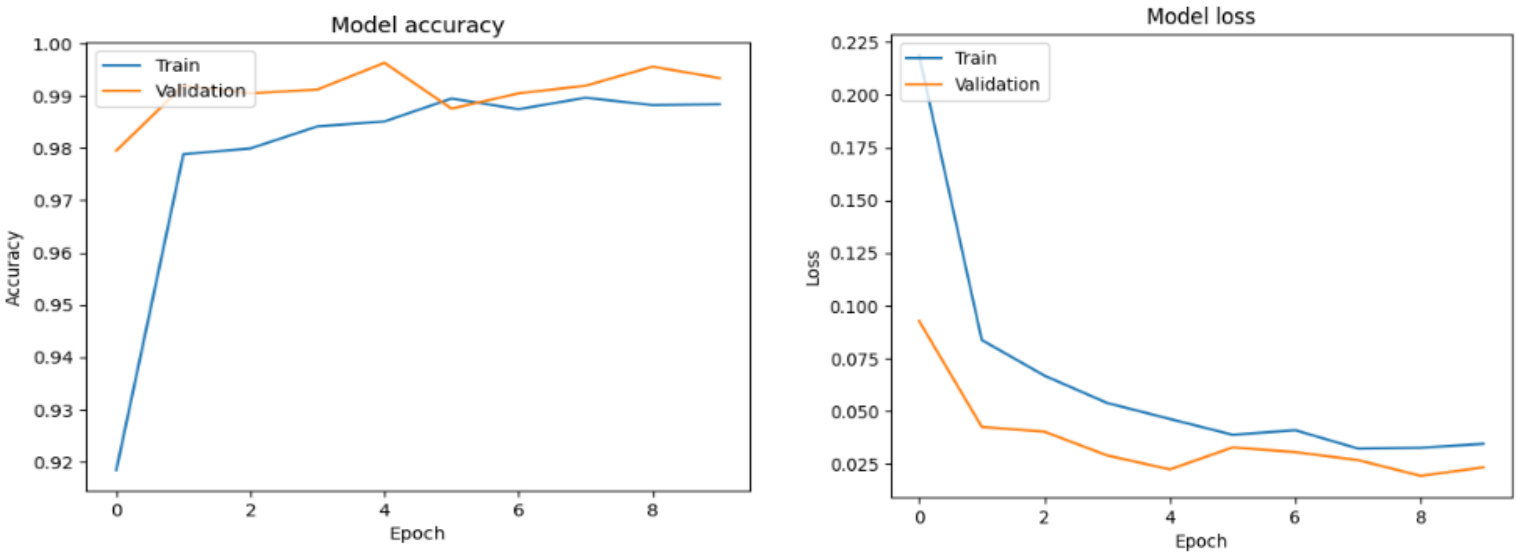


Fig 43 – Accuracy and loss of Adam

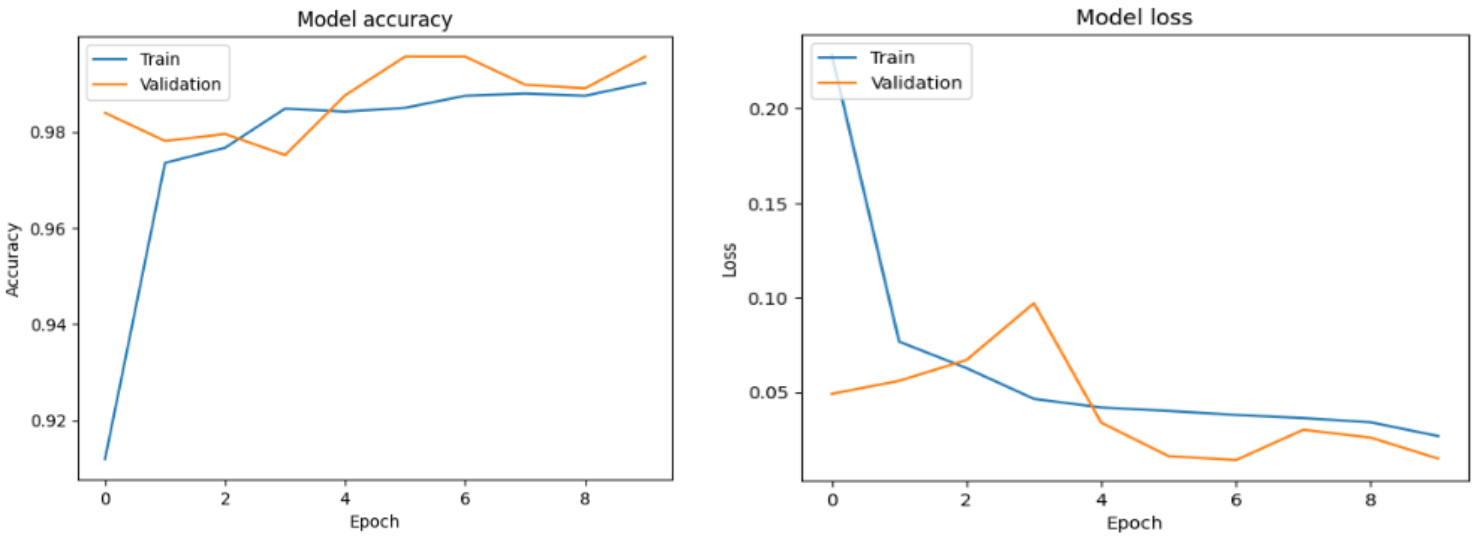


Fig 44 – Accuracy and loss of SGD

The Adam optimizer outperforms SGD when evaluated over the same number of epochs.

Future work:

As the making of counterfeit notes increases daily and the quality of these counterfeit notes gets better, our future responsibility is to improve this project to make it more efficient and accurate. Although this project can detect counterfeit notes accurately, we can improve this project by increasing the size of the dataset. In this project, we decreased the dataset size because the drive's storage size is limited. So, we could only use part of the dataset. Our future goal is to merge more datasets. Currency security features and design changes are continuous. This project can detect only the recent 500 and 1000 notes. If there is a change in currency in the future, we will update our dataset with new currency images to detect the new currencies, whether they are fake or not. One of the restrictions of this project is that it can detect fakes or not using the image of the front side of a note. Our project can not detect accurately if you upload a photo of the back of a note. Collecting a dataset of images of the back side of a note is another future goal so that a user can also identify using the image of the back side of a note. A scanning option will also be added to our website for user convenience.

One day, this project could be a game changer in decreasing the effect of counterfeit notes on Bangladesh's economy.

Project Cost:

The project cost is zero as we didn't buy any hardware equipment for this project.

Conclusion:

The emergence of fake currency is increasing daily, negatively affecting our economy and inflation rate. It has made life for low-income people difficult. Our counterfeit note detection using AI is essential to addressing this crisis. We trained different types of models with a dataset of 2280 images. Using the MobileNet model, we built the overall project with good accuracy, which is 99%. Our project website is also user-friendly. Just click on the image of the note and then upload it to our web application to detect whether the note is fake after Google Authentication. The general public also has a vital role to play. People should report counterfeit money suspicions by detecting this web app to assist law enforcement. By this project, we can help strengthen the economy, lower inflation, and keep the financial system running smoothly.

Contribution:

1. **Asifur Rahman**
    - Dataset Collection, Data analysis & Reorganized Dataset
    - Dataset Pre-Processing
    - Improve Dataset Pre-Processing
    - Three Model Trained, Test, Evaluate(MobileNet, VGG16, InceptionV3)
    - Testing with unseen images in the model
    - Connect model with Website
    - Testing website with Unseen Images

2. **Mohammad Hossain**
    - Dataset Collection, Data analysis & Reorganized Dataset
    - Data Split from Pre-Processing Dataset
    - Structure of our project's website
    - Improve Data Split from Pre-Processing Dataset
    - Three Model Trained, Test Evaluate(ResNet50, DensNet121, MobileNetV2)
    - Testing with unseen images in the model
    - Full Website design & Responsive this Website
    - Model Conversion to Lightweight TFLite Format
    - Testing website with Unseen Images

3. **Khandaker Anjuman Parvez**
    - Google Authentication Login Design
    - Google Authentication connects with the website

References:

[1] S Naresh Kumar; Gaurav Singal; Shwetha Sirikonda; R. Nethravathi;, "A Novel Approach for Detection of Counterfeit Indian Currency Notes Using Deep Convolutional," in *IOP Conference Series: Materials Science and Engineering*, Warangal, India, 2020.

[2] César G. Pachón; Dora M. Ballesteros; Diego Renza;, "Fake Banknote Recognition Using Deep Learning," *Applied Sciences*, vol. 11, no. 3, pp. 1-20, 2021.

[3] Tuyen Danh Pham; Chanhum Park; Dat Tien Nguyen; Ganbayar Batchuluun; Kang Ryoung Park;, "Deep Learning-Based Fake-Banknote Detection for the Visually Impaired People Using Visible-Light Images Captured by Smartphone Cameras," *IEEE Access*, vol. 8, pp. 63144 - 63161, 2020.

[4] P. Ashok babu; P. Sridhar; Rajeev Ratna Vallabhuni;, "Fake Currency Recognition System Using Edge Detection," in *2022 Interdisciplinary Research in Technology and Management (IRTM)*, Kolkata, India, 2022.

[5] Zahid Ahmed; Sabina Yasmin; Md Nahidul Islam; Raihan Uddin Ahmed;, "Image processing based Feature extraction of Bangladeshi banknotes," in *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*, Dhaka, Bangladesh, 2014.

[6] B. Naga Lakshmi; G. Sarath Kumar;, "FAKE CURRENCY DETECTION USING MACHINE," *Journal of Engineering Sciences*, vol. 13, no. 12, pp. 223-229, 2022.

[7] Tushar Agasti; Gajanan Burand; Pratik Wade; P Chitra;, "Fake currency detection using image processing," in *IOP Conference Series: Materials Science and Engineering*, 2017.

[8] Sarangam Kodati; M Dhasaratham; Srikanth Veldandi; K.Meenendranath Reddy;, "Detection of Fake Currency Using Machine Learning Models," *International Journal of Research In Science & Engineering*, vol. 04, no. 01, pp. 31-38, 2023.

[9] Karthik K; Gowtham R; Nandan K R; Mrs. Varsha;, "FAKE CURRENCY DETECTION USING MACHINE LEARNING," *International Journal For Technological Research In Engineering*, vol. 8, no. 1, pp. 17-19, 2020.

[10] Karangula Navya; Baksam Chiranjeevi; Danush M; Hariharan M S; Lalith Kumar S;, "DETECTION OF FAKE CURRENCY USING MACHINE LEARNING TECHNIQUES,"

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS - IJCRT, vol. 12, no. 5, pp. 536-541, 2024.

[11] Ayush Antre; Om Kalbhor; Pratik Jagdale; Ganesh Dhanne; Prof. Nilesh R. Sonawane;, "FAKE CURRENCY DETECTION USING CONVOLUTION," *International Research Journal of Modernization in Engineering Technology and Science,* vol. 5, no. 4, pp. 2916-2920, 2023.

[12] M. Laavanya; V. Vijayaraghavan;, "Real Time Fake Currency Note Detection using," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 9, no. 1S5, pp. 95-98, 2019.

Appendix:

Index.html Code:

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Counterfeit Detection of Bangladesh Banknotes</title>

   <link rel="stylesheet" href="style.css">

   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css">

   <!-- bootstrap links -->

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

      integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

   <!-- bootstrap links -->

   <!-- fonts links -->

   <link rel="preconnect" href="https://fonts.googleapis.com">

   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

   <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@500&display=swap"
rel="stylesheet">

   <!-- fonts links -->

   <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">

   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

   <!-- Google Fonts Link for Poppins -->

   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap"
rel="stylesheet">

</head>



<body>

   <div class="all-content">

      <!-- navbar html start -->

      <nav class="navbar navbar-expand-lg" id="navbar">
```

```
        <div class="container-fluid">
          <a class="navbar-brand" href="#" id="logo">
            <img src="../static/images/Navbarlogo.png" alt="logo">
          </a>
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span><i class="fa-solid fa-bars" style="color: white; font-size: 23px;"></i></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
              <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="#home">Home</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="#about">About</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="#detect">Detect</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="#team">Team</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="#Resource">Resource</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="#contact">Contact</a>
              </li>
              <!-- Logout code html Link start -->
              <li class="nav-item">
                <a class="nav-link btn btn-circle btn-danger" href="/logout">Logout</a>
              </li>
            </ul>
          </div>
```

```
        </div>
    </nav>
    <!-- navbar html end -->
    <!-- Home Section HTML Start -->
    <section id="home">
        <div class="content">
            <h1>Identify Counterfeit Bangladeshi <br>banknotes in real time
            </h1>
            <p class="fade-in delay">Verify the genuineness of your banknotes using our advanced detection technology.
            </p>
            <button id="btn"><a href="#detect">Detect</a></button>
        </div>
    </section>
    <!-- Home Section HTML End -->

    <!-- about section html start -->
    <div class="about" id="about">
        <div class="container">
            <div class="heading3">ABOUT</div>
        </div>
        <div class="container" id="container2">
            <div class="row align-items-center">
                <div class="col-md-6">
                    <div class="card">
                        <img src="../static/images/bg3.jpg" alt="Bangladesh Banknote">
                    </div>
                </div>
                <div class="col-md-6">
                    <h3 class="about-title">Counterfeit Detection of Bangladesh Banknotes</h3>
                    <p class="about-text">The Counterfeit Detection of Bangladesh Banknote project allows users to verify the authenticity of 500 and 1000 taka
                    banknotes. When you upload an image, our system automatically assesses security elements like watermarks and holograms
                    using advanced deep learning techniques. Our goal is to provide a simple, user-friendly solution for combating
```

```
            counterfeit currency and raising awareness about financial fraud prevention.</p>
            <button id="about-btn">Learn More</button>
          </div>
        </div>
      </div>
    </div>
    <!-- about section html end -->

    <!-- Detect section html start -->
    <section class="detect" id="detect">
      <div class="container">
        <div class="heading3">Detect Banknote</div>
      </div>
      <div class="container" id="container2">
        <!-- Note Detection Section -->
        <section id="note-detection" class="py-5 text-center">
          <div class="container">
            <h1 class="detect-title">Note Detection</h1>
            <h3 class="detect-description">Upload an image of the banknote to check its
authenticity.</h3>
            <!-- this ic Image Upload code start -->
            <div class="row align-items-center justify-content-center">
              <div class="col-md-6">
                <form id="imageUploadForm" class="text-center" enctype="multipart/form-data">
                  <input type="file" class="form-control-file mb-3" id="uploadImage" name="image"
                    accept="image/*" required onchange="previewImage(event)">
                  <!-- This is Display Image Html code start -->
                  <div class="mt-3">
                    <img id="uploadedImage" src="" alt="Uploaded Banknote" class="img-fluid"
                      style="max-width: 100%; height: auto; display:none; border: 2px solid #b2744c;
border-radius: 10px;">
                  </div>
              </div>
              <div class="col-md-6">
                <!-- this is  Results show html code start-->
```

```
                    <button type="submit" class="btn-primary btn-detect mt-3">Check Authenticity</button>
                    </form>
                    <div id="results" class="mt-4" style="display:none;">
                      <h3 class="result-title">Prediction Result:</h3>
                      <p id="predictionText"></p>
                    </div>
                  </div>
                </div>
              </div>
            </section>
          </div>
        </section>
        <!-- Detect section html end -->


        <!-- Team Section HTML Start -->
        <section class="team my-5 text-center" id="team">
          <div class="container">
            <div class="heading3">Our Team</div>
          </div>
          <div class="container" id="container2">
            <div class="row justify-content-center">
              <!-- Team Member 1 HTML Start -->
              <div class="col-sm-12 col-md-6 col-lg-4">
                <div class="team-member card p-4 shadow-lg" data-aos="zoom-in-down">
                  <img src="../static/images/joy.jpeg" class="img-fluid rounded-circle team-img">
                  <h5 class="card-title mt-4">Mohammad Hossain</h5>
                  <p class="card-text">Computer Science and Engineering<br>North South University</p>
                  <a href="https://www.facebook.com/mh.joy.75457?mibextid=LQQJ4d" class="btn btn-
outline-primary mt-3">Read More</a>
                </div>
              </div>
               <!-- Team Member 2 HTML Start -->
              <div class="col-sm-12 col-md-6 col-lg-4">
                <div class="team-member card p-4 shadow-lg" data-aos="zoom-in-left">
```

```
                <img src="../static/images/asif.jpeg" class="img-fluid rounded-circle team-img">
                <h5 class="card-title mt-4">Asifur Rahman</h5>
                <p class="card-text">Computer Science and Engineering<br>North South University</p>
                <a href="#" class="btn btn-outline-primary mt-3">Read More</a>
              </div>
            </div>
              <!-- Team Member 3 HTML Start -->
            <div class="col-sm-12 col-md-6 col-lg-4">
              <div class="team-member card p-4 shadow-lg" data-aos="zoom-in-right">
                <img src="../static/images/pritom.jpeg" class="img-fluid rounded-circle team-img">
                <h5 class="card-title mt-4">Khandakar Anjuman</h5>
                <p class="card-text">Computer Science and Engineering<br>North South University</p>
                <a href="#" class="btn btn-outline-primary mt-3">Read More</a>
              </div>
            </div>
          </div>
        </div>
      </section>
      <!-- Team Section HTML End -->


<!-- Educational Resource HTML Start -->
<section class="Resource my-5" id="Resource">
  <div class="container">
    <div class="heading3">Educational Resource</div>
  </div>

  <div class="container">
    <div class="row">
      <div class="col-md-4">
        <div class="card shadow-sm" data-aos="fade-up">
          <img src="../static/images/banknote security feature.jpg" alt="Banknote Security Features"
            class="card-img-top">
          <div class="card-body">
            <h3 class="card-title">Understanding Banknote Security Features</h3>
```

```
                Banknotes incorporate watermarks, security threads, and microprinting as essential security
                elements to deter
                counterfeiting. These elements offer various levels verification...</p>
                <a href="https://www.bb.org.bd/currency/note.php" class="btn btn-primary">Read
More</a>
            </div>
          </div>
        </div>

        <div class="col-md-4">
          <div class="card shadow-sm" data-aos="fade-up" data-aos-delay="200">
            <img src="../static/images/Counterfeith banknote techniq.png" alt="Counterfeit Detection
Techniques"
              class="card-img-top">
            <div class="card-body">
              <h3 class="card-title">Counterfeit Detection Techniques</h3>
              The identification of counterfeit currency depends on the utilization of machine learning
models
              to examine the
              characteristics of banknotes, combined with UV/IR scanning to unveil concealed security...
              </p>
              <a href="https://ieeexplore.ieee.org/document/7760162/similar#similar"
                class="btn btn-primary">Read
                More</a>
            </div>
          </div>
        </div>

        <div class="col-md-4">
          <div class="card shadow-sm" data-aos="fade-up" data-aos-delay="400">
            <img src="../static/images/economicaly impact.png" alt="Economic Impact of Counterfeiting"
              class="card-img-top">
            <div class="card-body">
              <h3 class="card-title">The Impact of Counterfeiting on the Economy</h3>
              Counterfeiting undermines economic stability by weakening confidence in currency, resulting
in
```

```
                financial losses due to
                decreased tax income and increased security costs and hampers overall economic growth
and
                increases the burden on law enforcement and..</p>
                <a href="https://www.tandfonline.com/doi/abs/10.1080/01924036.2012.726320"
                  class="btn btn-primary">Read
                  More</a>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>
    <!-- Educational Resource HTML End -->

    <!-- Contact Us HTML Start -->
    <section class="contact" id="contact">
      <div class="container">
        <div class="row align-items-center">
          <div class="col-md-6">
            <div class="heading6">
              <h1><span>Contact Us</span></h1>
            </div>
            <p>Please provide your name, email, and a message, and we'll get back to you as soon as
              possible.</p>

            <form>
              <div class="mb-3">
                <input class="form-control" type="text" placeholder="Name" required>
              </div>
              <div class="mb-3">
                <input class="form-control" type="email" placeholder="Email" required>
              </div>
              <div class="mb-3">
                <input class="form-control" type="number" placeholder="Phone Number" required>
```

```
                </div>
                <div class="mb-3">
                  <textarea class="form-control" placeholder="How can we help you?" rows="4"
                    required></textarea>
                </div>
                <button id="contact-btn" type="submit" class="btn btn-primary">Send Message</button>
              </form>
            </div>

            <div class="col-md-6" id="col">
              <iframe
                src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3650.098092052639!2d90.42298167533794!3d23.81511067862623!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3755c64c103a8093%3A0xd660a4f50365294a!2sNorth%20South%20University!5e0!3m2!1sen!2sbd!4v1729506139796!5m2!1sen!2sbd"
                width="100%" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe>
            </div>
          </div>
        </div>
      </section>
      <!-- Contact Us HTML End -->

      <!-- Footer HTML Start -->
      <footer id="footer">
        <div class="container">
          <div class="social-links text-center">
            <a href="#" class="social-icon"><i class="fa-brands fa-twitter"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-facebook-f"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-instagram"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-youtube"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-pinterest-p"></i></a>
          </div>

          <div class="credit text-center">
            Designed By <a href="#">Team AM</a>
          </div>
```

```
            <div class="copyright text-center">
              &copy; <strong><span>2024 Copyright: Counterfeit Detection of Bangladesh Banknote</span></strong>
              All Rights
              Reserved
            </div>
          </div>
        </footer>
        <!-- Footer HTML End -->

    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
      crossorigin="anonymous"></script>
    <script src="https://smtpjs.com/v3/smtp.js"></script>

  <script>
    // This code is Preview image javascript code
    function previewImage(event) {
      const uploadedImage = document.getElementById("uploadedImage");
      const file = event.target.files[0];
      if (file) {
        const reader = new FileReader();
        reader.onload = function (e) {
          uploadedImage.src = e.target.result;
          uploadedImage.style.display = "block";
        };
        reader.readAsDataURL(file);
      }
    }

    // image Handle form  image submission and get result prediction javascript code
    document.getElementById("imageUploadForm").addEventListener("submit", function (e) {
```

```
        e.preventDefault();


        const formData = new FormData(this);
        const resultsDiv = document.getElementById("results");
        const predictionText = document.getElementById("predictionText");
        // This code sShow the prediction result
        resultsDiv.style.display = "block";
        fetch('/predict', {
            method: 'POST',
            body: formData
        })
            .then(response => response.json())
            .then(data => {
                predictionText.textContent = data.result; // this code Display result directly


                // This code is result color Change text color based on prediction prediction result green=original
and red=counterfeit
                if (data.result === 'Your Bank note is Original') {
                    predictionText.classList.remove('result-red');
                    predictionText.classList.add('result-green');
                } else if (data.result === 'Your Bank note is Counterfeit') {
                    predictionText.classList.remove('result-green');
                    predictionText.classList.add('result-red');
                }
            })
            .catch(error => {
                console.error('Error:', error);
                predictionText.textContent = 'Error occurred while processing the image.';
                predictionText.classList.remove('result-green', 'result-red'); // This code is Remove previous result
text color class
            });
    });
</script>
</body>
```

```
</html>
```

Login.html code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Counterfeit Detection of Bangladesh Banknotes</title>
   <link rel="stylesheet" href="style.css">
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css">
   <!-- This link for bootstrap -->
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
   <!-- fonts links -->
   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
   <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@500&display=swap"
rel="stylesheet">
   <!-- fonts links -->
   <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
   <!-- This links for Google Fonts Poppins -->
   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap"
rel="stylesheet">
</head>

<body>
   <div class="all-content">
      <!--navbar html code  start -->
      <nav class="navbar navbar-expand-lg" id="navbar">
         <div class="container-fluid">
            <a class="navbar-brand" href="#" id="logo">
               <img src="../static/images/Navbarlogo.png" alt="logo">
```

```
    </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
      data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
      aria-expanded="false" aria-label="Toggle navigation">
      <span><i class="fa-solid fa-bars" style="color: white; font-size: 23px;"></i></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#home">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#about">About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#detect">Detect</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#team">Team</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#Resource">Resource</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#contact">Contact</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{{ url_for('login') }}">Login</a>
        </li>

      </ul>
    </div>
  </div>
</nav>
```

```
    <!-- navbar html code end -->
    <!-- Home Section HTML Start -->
    <section id="home">
      <div class="content">
        <h1>Identify Counterfeit Bangladeshi <br>banknotes in real time
        </h1>
        <p class="fade-in delay">Verify the genuineness of your banknotes using our advanced detection technology.
        </p>
        <button id="btn"><a href="{{ url_for('login') }}" >Detect</a></button>
      </div>
    </section>
    <!-- Home Section HTML End -->
    <!-- about section html start -->
    <div class="about" id="about">
      <div class="container">
        <div class="heading3">ABOUT</div>
      </div>
      <div class="container" id="container2">
        <div class="row align-items-center">
          <div class="col-md-6">
            <div class="card">
              <img src="../static/images/bg3.jpg" alt="Bangladesh Banknote">
            </div>
          </div>
          <div class="col-md-6">
            <h3 class="about-title">Counterfeit Detection of Bangladesh Banknotes</h3>
            <p class="about-text">The Counterfeit Detection of Bangladesh Banknote project allows users to verify the authenticity of 500 and 1000 taka
            banknotes. When you upload an image, our system automatically assesses security elements like watermarks and holograms
            using advanced deep learning techniques. Our goal is to provide a simple, user-friendly solution for combating
            counterfeit currency and raising awareness about financial fraud prevention.</p>
            <button id="about-btn">Learn More</button>
          </div>
```

```
        </div>
      </div>
    </div>
    <!-- about section html end -->


    <!-- Detect section html code start -->
    <section class="detect" id="detect">
      <div class="container">
        <div class="heading3">Detect Banknote</div>
      </div>
      <div class="container" id="container2">
        <!-- This is Note Detection Section code start -->
        <section id="note-detection" class="py-5 text-center">
          <div class="container">
            <h1 class="detect-title">Note Detection</h1>
            <h3 class="detect-description">Upload an image of the banknote to check its
authenticity.</h3>
            <!-- This is Image Upload html code start -->
            <div class="row align-items-center justify-content-center">
              <div class="col-md-6">
                <a href="{{ url_for('login') }}" class="text-decoration-none">
                  <input type="file" class="form-control-file mb-3" id="uploadImage" name="image"
accept="image/*" required
                    style="display: none;">
                  <button type="button" class="btn btn-success">Upload Image</button>
                </a>
              </div>
              </div>
            </div>
          </div>
        </section>
      </div>
    </section>
    <!-- Detect section html code end -->
    <!-- Team Section HTML Start -->
```

```
<section class="team my-5 text-center" id="team">
  <div class="container">
    <div class="heading3">Our Team</div>
  </div>
  <div class="container" id="container2">
    <div class="row justify-content-center">
      <!-- Team Member 1 HTML Start -->
      <div class="col-sm-12 col-md-6 col-lg-4">
        <div class="team-member card p-4 shadow-lg" data-aos="zoom-in-down">
          <img src="../static/images/joy.jpeg" class="img-fluid rounded-circle team-img">
          <h5 class="card-title mt-4">Mohammad Hossain</h5>
          <p class="card-text">Computer Science and Engineering<br>North South University</p>
          <a href="https://www.facebook.com/mh.joy.75457?mibextid=LQQJ4d"
            class="btn btn-outline-primary mt-3">Read More</a>
        </div>
      </div>
      <!-- Team Member 2 HTML Start -->
      <div class="col-sm-12 col-md-6 col-lg-4">
        <div class="team-member card p-4 shadow-lg" data-aos="zoom-in-left">
          <img src="../static/images/asif.jpeg" class="img-fluid rounded-circle team-img">
          <h5 class="card-title mt-4">Asifur Rahman</h5>
          <p class="card-text">Computer Science and Engineering<br>North South University</p>
          <a href="#" class="btn btn-outline-primary mt-3">Read More</a>
        </div>
      </div>
      <!-- Team Member 3 HTML Start -->
      <div class="col-sm-12 col-md-6 col-lg-4">
        <div class="team-member card p-4 shadow-lg" data-aos="zoom-in-right">
          <img src="../static/images/pritom.jpeg" class="img-fluid rounded-circle team-img">
          <h5 class="card-title mt-4">Khandakar Anjuman</h5>
          <p class="card-text">Computer Science and Engineering<br>North South University</p>
          <a href="#" class="btn btn-outline-primary mt-3">Read More</a>
        </div>
      </div>
```

```
        </div>
      </div>
    </section>
    <!-- Team Section HTML End -->
    <!-- Educational Resource HTML Start -->
    <section class="Resource my-5" id="Resource">
      <div class="container">
        <div class="heading3">Educational Resource</div>
      </div>
      <div class="container">
        <div class="row">
          <div class="col-md-4">
            <div class="card shadow-sm" data-aos="fade-up">
              <img src="../static/images/banknote security feature.jpg" alt="Banknote Security Features"
                class="card-img-top">
              <div class="card-body">
                <h3 class="card-title">Understanding Banknote Security Features</h3>
                Banknotes incorporate watermarks, security threads, and microprinting as essential
security elements to deter
                counterfeiting. These elements offer various levels verification...</p>
                <a href="https://www.bb.org.bd/currency/note.php" class="btn btn-primary">Read
More</a>
              </div>
            </div>
          </div>

          <div class="col-md-4">
            <div class="card shadow-sm" data-aos="fade-up" data-aos-delay="200">
              <img src="../static/images/Counterfeith banknote techniq.png"
                alt="Counterfeit Detection Techniques" class="card-img-top">
              <div class="card-body">
                <h3 class="card-title">Counterfeit Detection Techniques</h3>
                The identification of counterfeit currency depends on the utilization of machine learning
models to examine the
                characteristics of banknotes, combined with UV/IR scanning to unveil concealed security...
                </p>
```

```
                    <a href="https://ieeexplore.ieee.org/document/7760162/similar#similar"
                      class="btn btn-primary">Read
                      More</a>
                  </div>
                </div>
              </div>

              <div class="col-md-4">
                <div class="card shadow-sm" data-aos="fade-up" data-aos-delay="400">
                  <img src="../static/images/economicaly impact.png" alt="Economic Impact of
Counterfeiting"
                    class="card-img-top">
                  <div class="card-body">
                    <h3 class="card-title">The Impact of Counterfeiting on the Economy</h3>
                    Counterfeiting undermines economic stability by weakening confidence in currency,
resulting in financial losses due to
                    decreased tax income and increased security costs and hampers overall economic growth
and increases the burden on law enforcement and..</p>
                    <a href="https://www.tandfonline.com/doi/abs/10.1080/01924036.2012.726320"
                      class="btn btn-primary">Read
                      More</a>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </section>
        <!-- Educational Resource HTML End -->

        <!-- Contact Us HTML Start -->
        <section class="contact" id="contact">
          <div class="container">
            <div class="row align-items-center">
              <div class="col-md-6">
                <div class="heading6">
                  <h1><span>Contact Us</span></h1>
```

```
            </div>
            <p>Please provide your name, email, and a message, and we'll get back to you as soon as
              possible.</p>

            <form>
              <div class="mb-3">
                <input class="form-control" type="text" placeholder="Name" required>
              </div>
              <div class="mb-3">
                <input class="form-control" type="email" placeholder="Email" required>
              </div>
              <div class="mb-3">
                <input class="form-control" type="number" placeholder="Phone Number" required>
              </div>
              <div class="mb-3">
                <textarea class="form-control" placeholder="How can we help you?" rows="4"
                  required></textarea>
              </div>
              <button id="contact-btn" type="submit" class="btn btn-primary">Send Message</button>
            </form>
          </div>

          <div class="col-md-6" id="col">
            <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3650.098092052639!2d90.42298167
533794!3d23.81511067862623!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3755c64c103a8093
%3A0xd660a4f50365294a!2sNorth%20South%20University!5e0!3m2!1sen!2sbd!4v1729506139796!5m2!1se
n!2sbd"
              width="100%" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe>
          </div>
        </div>
      </div>
    </section>
    <!-- Contact Us HTML End -->


    <!-- Footer HTML Start -->
```

```
      <footer id="footer">
        <div class="container">
          <div class="social-links text-center">
            <a href="#" class="social-icon"><i class="fa-brands fa-twitter"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-facebook-f"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-instagram"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-youtube"></i></a>
            <a href="#" class="social-icon"><i class="fa-brands fa-pinterest-p"></i></a>
          </div>
          <div class="credit text-center">
            Designed By <a href="#">Team AM</a>
          </div>

          <div class="copyright text-center">
            &copy; <strong><span>2024 Copyright: Counterfeit Detection of Bangladesh
Banknote</span></strong>
            All Rights
            Reserved
          </div>
        </div>
      </footer>
      <!-- Footer HTML End -->
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
      crossorigin="anonymous"></script>
    <script src="https://smtpjs.com/v3/smtp.js"></script>

</body>

</html>
```

Style.css code:

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Merriweather', serif;
}

.all-content {
    background: #f6eee4;
}
ul li a {
    text-decoration: none;
}
body {
    font-family: 'Poppins', sans-serif;
    font-size: 18px;
    line-height: 1.6;
    color: #333;
}

h1,h2,h3,h4,h5,h6 {
    font-family: 'Poppins', sans-serif;
    font-weight: 600;
    margin-bottom: 20px;

}
h1 {
    font-size: 60px;
}
h2 {
    font-size: 48px;
}
```

```
h3 {
    font-size: 36px;
}
h4 {
    font-size: 28px;
}
h5 {
    font-size: 24px;
}
h6 {
    font-size: 20px;
}
p {
    font-size: 18px;
    margin-bottom: 20px;
}
button,
a {
    font-size: 20px;
    font-weight: bold;
}
/* navbar css start*/
#navbar {
    background-color: #b2744c;
    font-family: 'Roboto', sans-serif;
}
#logo img {
    border-radius: 8px;
    width: 40%;
    margin-left: 50px;
}
.navbar-nav {
    margin-left: 50px;
    margin-right: 50px;
```

```
}

.navbar a {
    font-size: 20px;
    font-weight: 500;
}
.nav-item .nav-link {
    color: rgb(237, 235, 235);
    margin-left: 10px;
    font-weight: bold;
    transition: 0.5s;
}
.nav-item .nav-link:hover {
    background: #edba5c;
    border-radius: 50px;
    color: black;
}
#navbar form button {
    background: black;
    color: white;
    border: 1px solid white;
}
/* navbar css end*/
/* Home Section CSS Start */
#home {
    width: 100%;
    height: 100vh;
    background-image: linear-gradient(rgba(30, 28, 28, 0.5), rgba(12, 1, 1, 0.5)), url(./images/h1.jpg);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
    display: flex;
    justify-content: center;
    align-items: center;
```

```
}

.content {
    text-align: center;
    color: rgb(248, 243, 243);
    animation: fadeInUp 2s ease forwards;
}
.content h1 {
    font-family: fantasy;
    font-size: 45px;
    font-weight: bold;
    line-height: 1.2;
}
.content p {
    margin-top: 15px;
    font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
    font-size: 24px;
    font-weight: 500;
    color: #ffffff;
}

#btn {
    width: 180px;
    height: 50px;
    margin-top: 30px;
    background: #de773c;
    border-radius: 50px;
    font-weight: bold;
    font-size: 22px;
    color: white;
    border: none;
    transition: all 0.3s ease;
    cursor: pointer;
}
```

```
#btn:hover {
    background: #b2744c;
    color: white;
}
#btn a {
    color: white;
    text-decoration: none;
}
@keyframes fadeInUp {
    0% {
        opacity: 0;
        transform: translateY(50px);
    }
    100% {
        opacity: 1;
        transform: translateY(0);
    }
}
/* Media Queries for tablets Responsiveness */
@media screen and (max-width: 768px) {
    .content h1 {
        font-size: 50px;
    }
    .content p {
        font-size: 20px;
    }
    #btn {
        width: 150px;
        height: 40px;
        font-size: 18px;
    }
}
```

```
@media screen and (max-width: 576px) {
    .content h1 {
        font-size: 40px;
    }
    .content p {
        font-size: 18px;
    }
    #btn {
        width: 130px;
        height: 36px;
        font-size: 16px;
    }
}
/* Home Section CSS End */
/* about part css start */
#about {
    width: 100%;
    height: 100%;
    padding: 70px 0;
    font-family: 'Roboto', sans-serif;
}
.card {
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s ease;
}
.card img {
    width: 100%;
    height: auto;
    display: block;
}
.card:hover {
    transform: scale(1.05);
```

```
}

.about-title {
    font-size: 28px;
    font-weight: 700;
    color: #333;
    margin-bottom: 20px;
}
.about-text {
    font-size: 16px;
    color: #555;
    line-height: 1.8;
    margin-bottom: 30px;
}
#about-btn {
    padding: 12px 30px;
    background-color: #b2744c;
    color: white;
    border: none;
    border-radius: 5px;
    font-size: 16px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
#about-btn:hover {
    background-color: #a5623a;
}
@media screen and (max-width: 991px) {
    .about-text {
        font-size: 15px;
    }
    .about-title {
        font-size: 26px;
    }
```

```
}

@media screen and (max-width: 768px) {
    .about-text {
        font-size: 14px;
    }

    .about-title {
        font-size: 24px;
    }

    #about {
        padding: 60px 0;
    }
}

@media screen and (max-width: 576px) {
    .about-title {
        font-size: 22px;
    }
    .about-text {
        font-size: 13px;
    }
    #about-btn {
        font-size: 14px;
        padding: 10px 25px;
    }
    .card img {
        height: 250px;
        object-fit: cover;
    }
}
/* about part css end */
```

```
/* detect CSS start */
.detect {
    width: 100%;
    background-color: #e2dfda;
    font-family: 'Roboto', sans-serif;
    font-size: large;
    padding: 80px 0;
    margin-bottom: 0px;
}
.heading3 {
    border-bottom: 2px solid #b2744c;
}
.detect-title {
    font-size: 28px;
    font-weight: 700;
    color: #333;
}
.detect-description {
    font-size: 16px;
    color: #555;
    margin-bottom: 30px;
}
.btn-detect {
    background-color: #b2744c;
    color: rgb(206, 89, 89);
    padding: 12px 30px;
    border: 3;
    border-radius: 5px;
    font-size: 16px;
    transition: background-color 0.3s ease;
}
.btn-detect:hover {
```

```
    background-color: #23e805;
}
#uploadImage {
    padding: 8px;
    border: 2px solid #b2744c;
    border-radius: 5px;
    width: 100%;
    max-width: 400px;
    cursor: pointer;
    transition: border-color 0.3s ease;
}
#uploadImage:hover {
    border-color: #a5623a;
}
#uploadedImage {
    max-width: 480px;
    max-height: 150px;
    width: auto;
    height: auto;
    border: 2px solid #b2744c;
    border-radius: 10px;
}
.result-title {
    font-size: 24px;
    color: #333;
    font-weight: bold;
}
#results {
    text-align: center;
    font-size: 24px;
}
.result-green {
    color: green;
    font-size: 38px;
```

```
}
.result-red {
    color: red;
    font-size: 38px;

}
@media screen and (max-width: 991px) {
    .detect-title {
        font-size: 24px;
    }
    .detect-description {
        font-size: 15px;
    }
    .btn-detect {
        font-size: 14px;
    }
    .result-title {
        font-size: 22px;
    }
}
@media screen and (max-width: 768px) {
    .detect-title {
        font-size: 22px;
    }
    .detect-description {
        font-size: 14px;
    }
    .btn-detect {
        font-size: 13px;
    }
    .result-title {
        font-size: 20px;
    }
```

```
}

@media screen and (max-width: 576px) {
    .detect-title {
        font-size: 20px;
    }
    .detect-description {
        font-size: 13px;
    }
    .btn-detect {
        font-size: 12px;
        padding: 10px 20px;
    }
    #uploadImage {
        width: 100%;
    }
}
/* detect CSS end */
/* Team Section CSS Start */
.team {
    margin-top: 0;
    background-color: #f6f6f6;
    padding: 80px 0;
}
.team-member {
    background-color: #ffffff;
    border-radius: 10px;
    overflow: hidden;
    transition: transform 0.3s ease-in-out;
}
.team-member:hover {
    transform: translateY(-10px);
}
.team-img {
```

```
    width: 150px;
    height: 150px;
    border: 5px solid #b2744c;
    border-radius: 50%;
    margin-bottom: 20px;
    transition: border-color 0.3s;
}
.team-img:hover {
    border-color: #a5623a;
}
.card-title {
    font-size: 20px;
    font-weight: 600;
    color: #333;
}
.card-text {
    font-size: 14px;
    color: #555;
}
.btn-outline-primary {
    border-color: #b2744c;
    color: #b2744c;
    transition: background-color 0.3s, color 0.3s;
}

.btn-outline-primary:hover {
    background-color: #b2744c;
    color: #ffffff;
}
/* Responsive Design */
@media screen and (max-width: 991px) {
    .team-img {
        width: 120px;
        height: 120px;
```

```
    }

    .card-title {
        font-size: 18px;
    }
    .card-text {
        font-size: 13px;
    }
}
@media screen and (max-width: 768px) {
    .heading3 {
        font-size: 28px;
    }
    .team-img {
        width: 100px;
        height: 100px;
    }

    .card-title {
        font-size: 16px;
    }
    .card-text {
        font-size: 12px;
    }
}
@media screen and (max-width: 576px) {
    .heading3 {
        font-size: 26px;
    }
    .team-img {
        width: 80px;
        height: 80px;
    }
    .card-title {
```

```
        font-size: 14px;
    }
    .card-text {
        font-size: 11px;
    }
}
/* Team Section CSS End */
/* Educational Resource CSS Start */
.Resource {
    background-color: #f4f4f4;
    padding: 80px 0;
}
.heading3 {
    font-size: 32px;
    font-weight: bold;
    color: #b2744c;
    text-align: center;
    margin-bottom: 50px;
    position: relative;
}
.card {
    background: #fff;
    border-radius: 10px;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
}
.card:hover {
    transform: translateY(-10px);
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.1);
}
.card-img-top {
    border-radius: 10px 10px 0 0;
    height: 200px;
    object-fit: cover;
}
```

```
.card-title {
    font-size: 20px;
    color: #333;
    font-weight: bold;
}
.card-body p {
    color: #555;
    font-size: 14px;
}
.btn-primary {
    background-color: #b2744c;
    border-color: #b2744c;
    color: white;
    border-radius: 30px;
    padding: 10px 20px;
    transition: background-color 0.3s, color 0.3s;
}
.btn-primary:hover {
    background-color: #9e6438;
    color: white;
}

/* Responsive Design */
@media (max-width: 991px) {
    .card-title {
        font-size: 18px;
    }

    .card-body p {
        font-size: 13px;
    }
}
```

```
@media (max-width: 768px) {
    .card-img-top {
        height: 180px;
    }

    .btn-primary {
        padding: 8px 16px;
    }
}

@media (max-width: 576px) {
    .card-title {
        font-size: 16px;
    }

    .card-body p {
        font-size: 12px;
    }

    .card-img-top {
        height: 150px;
    }
}

/* Educational Resource CSS End */




/* Contact CSS Start */
.contact {
    background-color: #f7f5f2;
    padding-top: 80px;
```

```
    padding-bottom: 50px;
}

.contact .container {
    background: linear-gradient(95deg, white 60%, #b2744c 0%);
    padding: 50px;
    border-radius: 15px;
}

.heading6 h1 {
    font-size: 34px;
    font-weight: bold;
    color: #333;
    margin-bottom: 20px;
}

.heading6 h1 span {
    color: #b2744c;
}

.contact p {
    font-size: 16px;
    color: #333;
    margin-bottom: 30px;
}

.contact .form-control {
    border: 1px solid #544545;
    border-radius: 5px;
    padding: 10px;
    font-size: 14px;
}

#contact-btn {
```

```
    background-color: #b2744c;
    border: none;
    color: white;
    padding: 10px 30px;
    border-radius: 50px;
    font-size: 16px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

#contact-btn:hover {
    background-color: #5d114b;
}

#col iframe {
    width: 100%;
    border: none;
    border-radius: 15px;
}

/* Responsive Design */
@media (max-width: 768px) {
    .contact .container {
        background: white;
        padding: 30px;
    }

    #col {
        margin-top: 30px;
    }

    .contact {
        padding-top: 50px;
    }
```

```
    .heading6 h1 {
        font-size: 28px;
    }
}

@media (max-width: 576px) {
    .contact p {
        font-size: 14px;
    }

    .contact .form-control {
        font-size: 13px;
        padding: 8px;
    }

    #contact-btn {
        font-size: 14px;
        padding: 8px 20px;
    }
}

/* Contact CSS End */









/* Footer CSS Start */
#footer {
    background: #cdc5bf;
    padding: 40px 0;
```

```
    font-size: 18px;
}

#footer .social-links {
    margin-bottom: 20px;
}

#footer .social-links .social-icon {
    display: inline-block;
    margin: 0 10px;
    background-color: #3a3a3a;
    color: #fff;
    border-radius: 50%;
    width: 40px;
    height: 40px;
    line-height: 40px;
    text-align: center;
    transition: background-color 0.3s ease;
}

#footer .social-links .social-icon:hover {
    background-color: #b2744c;
}

.credit {
    margin-top: 20px;
    font-size: 16px;
    color: #333;
}
.credit a {
    color: #fffefd;
    text-decoration: none;
    font-weight: bold;
}
```

```
.credit a:hover {
    text-decoration: underline;
}
.copyright {
    margin-top: 15px;
    font-size: 20px;
    color: #222;
}
#footer span {
    color: #2c2c2c;
    font-weight: bold;
}
/* Responsive Design */
@media (max-width: 768px) {
    #footer .social-links .social-icon {
        width: 35px;
        height: 35px;
        line-height: 35px;
    }
    .credit {
        font-size: 14px;
    }
    .copyright {
        font-size: 12px;
    }
}

@media (max-width: 576px) {
    #footer .social-links {
        margin-bottom: 15px;
    }
    .credit {
        font-size: 13px;
    }
```

```
    .copyright {
        font-size: 11px;
    }
}
/* Footer CSS End */
```

App.py code:

```
#import library,package,framework,module
from flask import Flask, redirect, url_for, session, render_template, request, jsonify
from flask_oauthlib.client import OAuth
import numpy as np
import tensorflow as tf
import cv2
import os

#Flask App Start
app = Flask(__name__)
#secret key
app.secret_key = '5e884898da28047151d0e56f8dc62977'
oauth = OAuth(app)

# This Configure of Google OAuth
google = oauth.remote_app(
    'google',
    # Client id and client secret from google cloud console
    consumer_key='1006252504341-
ukl0s2m7l9ivttq5004a0na7h412mu71.apps.googleusercontent.com',
    consumer_secret='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',
    request_token_params={ 'scope': 'email',},
    base_url='https://www.googleapis.com/oauth2/v1/',
    request_token_url=None,
    access_token_method='POST',
    access_token_url='https://accounts.google.com/o/oauth2/token',
    authorize_url='https://accounts.google.com/o/oauth2/auth',
```

```
)

# This code is Tensorflow(TFLite) model load
model_path = 'mobilenet_transfer_learning_model_for_counterfeit_detection_quantized.tflite'
interpreter = tf.lite.Interpreter(model_path=model_path)
interpreter.allocate_tensors()

# this is Get input and output tensors
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

# uploaded image Pre-processing
def preprocess_image(image_path):
    img = cv2.imread(image_path)
    img = cv2.resize(img, (224, 224))  # Resize image
    img = img.astype('float32') / 255.0  # Normalize to [0, 1]
    img = np.expand_dims(img, axis=0)  # Add batch dimension
    return img

#page rendering start
@app.route('/')
def index():
    return render_template('login.html')

@app.route('/home')
def home():
    if 'user' in session:
        #  index.html page render if user is logged in
        return render_template('index.html', user=session['user'])
    # Redirect to user login but if not user  logged in
    return redirect(url_for('login'))
```

```
@app.route('/login')
def login():
    # Redirect to login for Google
    return google.authorize(callback=url_for('authorized', _external=True))

@app.route('/login/callback')
def authorized():
    response = google.authorized_response()
    if response is None or 'access_token' not in response:
        return 'Access denied: reason={} error={}'.format(
            request.args['error_reason'],
            request.args['error_description']
        )

    session['google_token'] = (response['access_token'], '')
    # user information fetch from google
    user_info = google.get('userinfo')
    # user information store in session
    session['user'] = user_info.data
    # Redirect to home after user successful login
    return redirect(url_for('home'))

@google.tokengetter
def get_google_oauth_token():
    # Retrieve the token from session
    return session.get('google_token')

@app.route('/logout')
def logout():
    # remove the user from session
    session.pop('user', None)
    # Remove google token from session
```

```
    session.pop('google_token', None)
    # Redirect to login page
    return redirect(url_for('index'))

#This code start for model result prediction
@app.route('/predict', methods=['POST'])
def predict():
    if 'image' not in request.files:
        return jsonify({'error': 'No file uploaded.'})

    file = request.files['image']
    if file.filename == '':
        return jsonify({'error': 'No selected file.'})

    # Save the uploaded image
    image_path = os.path.join('static', file.filename)
    file.save(image_path)

    # Image Preprocessing
    input_image = preprocess_image(image_path)

    # Set the tensor for input
    interpreter.set_tensor(input_details[0]['index'], input_image)

    # Run inference
    interpreter.invoke()

    # Get TensorFlow Lite models are lightweight and optimized for fast inference, especially in web
apps.
    output_data = interpreter.get_tensor(output_details[0]['index'])
    predicted_class = np.argmax(output_data)

    # Class mapping
```

```
    class_indices = {0: 'Your Bank note is Counterfeit', 1: 'Your Bank note is Original'}
    result = class_indices[predicted_class]

    # Remove the uploaded image after processing to save storage
    #Sends the prediction result back to the frontend as a JSON response.
    os.remove(image_path)

    return jsonify({'result': result})

if __name__ == '__main__':
    app.run(debug=True)
```

MobileNet Model Code:

```
#Model train using mobilenet transfer learning started from here
from tensorflow.keras.applications import MobileNet
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

train_directory = '/content/drive/MyDrive/Main Dataset/data_split/Train Data'
validation_directory = '/content/drive/MyDrive/Main Dataset/data_split/Validation Data'
test_directory = '/content/drive/MyDrive/Main Dataset/data_split/Test Data'

#Image size and batch size
image_size = (224, 224)
batch_size = 32

#Augmentation for training data required for model train
train_datagen = ImageDataGenerator(
    rescale=1.0/255.0,  # Normalize pixel values between 0 and 1
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
```

```
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

#Rescaling for validation and test data
validation_datagen = ImageDataGenerator(rescale=1.0/255.0)
test_datagen = ImageDataGenerator(rescale=1.0/255.0)

#Load training, validation, and test data
train_generator = train_datagen.flow_from_directory(
    train_directory,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

validation_generator = validation_datagen.flow_from_directory(
    validation_directory,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_directory,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)
#Model train using mobilenet transfer learning started from here
from tensorflow.keras.applications import MobileNet
```

```
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping

train_directory = '/content/drive/MyDrive/Main Dataset/data_split/Train Data'
validation_directory = '/content/drive/MyDrive/Main Dataset/data_split/Validation Data'
test_directory = '/content/drive/MyDrive/Main Dataset/data_split/Test Data'

#Image size and batch size
image_size = (224, 224)
batch_size = 32

#Augmentation for training data required for model train
train_datagen = ImageDataGenerator(
    rescale=1.0/255.0,  # Normalize pixel values between 0 and 1
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

#Rescaling for validation and test data
validation_datagen = ImageDataGenerator(rescale=1.0/255.0)
test_datagen = ImageDataGenerator(rescale=1.0/255.0)

#Load training, validation, and test data
train_generator = train_datagen.flow_from_directory(
    train_directory,
    target_size=image_size,
    batch_size=batch_size,
```

```
    class_mode='categorical'
)

validation_generator = validation_datagen.flow_from_directory(
    validation_directory,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_directory,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)
#Early stopping callback to prevent overfitting
early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

#Train the model
history = model.fit(
    train_generator,
    epochs=10,
    validation_data=validation_generator,
    callbacks=[early_stop]
)
#Evaluate the model on the test data
test_loss, test_acc = model.evaluate(test_generator)
print(f'Test accuracy: {test_acc*100:.2f}%')

model.save('/content/drive/MyDrive/Main
Dataset/models/mobilenet_transfer_learning_model_for_counterfeit_detection.h5')
import matplotlib.pyplot as plt
import seaborn as sns # Added import for seaborn
```

```
from sklearn.metrics import confusion_matrix, precision_score, recall_score

#Training & validation accuracy values
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

#Training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

y_true = test_generator.classes
y_pred_prob = model.predict(test_generator)
y_pred = np.argmax(y_pred_prob, axis=1)

#Confusion matrix
con_matrix = confusion_matrix(y_true, y_pred)

#Confusion matrixtest set
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()
```

```
#Precision and recall
precision = precision_score(y_true, y_pred)
recall = recall_score(y_true, y_pred)

print(f'Precision: {precision:.2f}')
print(f'Recall: {recall:.2f}')
from tensorflow import keras

# Load the saved model
model = load_model('/content/drive/MyDrive/Main
Dataset/models/mobilenet_transfer_learning_model_for_counterfeit_detection.h5')
# Print the model summary
model.summary()
#Testing the model after model train using images from our dataset
from tensorflow.keras.preprocessing import image
def preprocess_image(image_path, target_size=(224, 224)):
    image = image.load_img(image_path, target_size=target_size)
    image_array = image.img_to_array(image)
    image_array = np.expand_dims(img_array, axis=0)
    image_array /= 255.0
    return image_array

from tensorflow.keras.models import load_model
model = load_model('/content/drive/MyDrive/Main
Dataset/models/mobilenet_transfer_learning_model_for_counterfeit_detection.h5')

def predict_banknote(image_path):
    image = preprocess_image(image_path)
    prediction = model.predict(image)

    #[0] = 'fake', [1] = 'real'
    if prediction[0][1] > 0.5:
        print("The banknote is real.")
    else:
```

```
        print("The banknote is fake.")


predict_banknote('/content/drive/MyDrive/Main Dataset/All real 500 check/IMG_0612.jpg')

#Load .h5 model

keras_model_path = '/content/drive/MyDrive/Main
Dataset/models/mobilenet_transfer_learning_model_for_counterfeit_detection.h5'

model = tf.keras.models.load_model(keras_model_path)


#Convert the model to TFLite to connect with website

converter = tf.lite.TFLiteConverter.from_keras_model(model)

converter.optimizations = [tf.lite.Optimize.DEFAULT]

tflite_model = converter.convert()


#Save the TFLite model

tflite_model_path = '/content/drive/MyDrive/Main Dataset/TFlite
Model/mobilenet_transfer_learning_model_for_counterfeit_detection_quantized.tflite'

with open(tflite_model_path, 'wb') as f:

   f.write(tflite_model)
```