An Industry Oriented Mini Project(CS704PC)

on

"FAKE CURRENCY DETECTION USING IMAGE PROCESSING"

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

Bachelor of Technology

in

Computer Science and Engineering

by

Ms. CHENCHALA HASINI

(19261A05D1)

Under the Guidance of

VEDAVATHI. K

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

GANDIPET, HYDERABAD -500 075, INDIA

2022-2023

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
GANDIPET, HYDERABAD – 500 075, Telangana

CERTIFICATE

This is to certify that the Industrial Mini Project entitled "**FAKE CURRENCY DETECTION USING IMAGE PROCESSING"** is being submitted by **CHENCHALA HASINI** bearing **Roll No: 19261A05D1** in partial fulfillment for the award of **B. Tech** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** is a record of bonafide work carried out under the supervision of **Vedavathi. K, Assistant Professor, Department of CSE.**

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

Project Guide	Head of the Department
Vedavathi. K	**Dr. C. R. K Reddy**
Assistant Professor, Dept. of CSE	Professor, Dept. of CSE

External Examiner

DECLARATION

This is to certify that the work reported in this project titled "**FAKE CURRENCY DETECTION USING IMAGE PROCESSING"** is a record of work done by us in the Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad. No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The report is based on the work done entirely by me and copied from any other source.

CHENCHALA HASINI

(19261A05D1)

ACKNOWLEDGEMENT

I would like to express our sincere thanks to **Dr. G Chandramohan, Principal MGIT**, for providing the working facilities in college.

I wish to express our sincere thanks and gratitude to **Dr. C R K Reddy, Professor and HOD**, Department of CSE, MGIT, for all the timely support and valuable suggestions during the period of project.

I am extremely thankful to **Dr. M. Sreevani, Associate professor**, and **Dr .K. Mahesh Kumar, Assistant Professor,** Department of CSE, MGIT, Mini Project Coordinators for their encouragement and support throughout the project.

I am extremely thankful and indebted to my internal guide **Vedavathi K**, **Assistant Professor,** Department of CSE, for their constant guidance, encouragement and moral support throughout the project.

Finally, I would also like to thank all the faculty and staff of CSE Department who helped us directly or indirectly, for completing this project.

CHENCHALA HASINI

(19261A05D1)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In the market, fake currency is the most important problem that speaks a lot. Due to the growth of technology, fake currency production has increased which degraded the economy of our country. Counterfeit money notes are expanding step by step, to beat this we propose an exceptionally accommodating and productive framework to identify counterfeit notes. The existing works to detect a counterfeit note are mostly based on traditional image processing techniques. This project deals with Deep Learning in which a convolution neural network(CNN) model is built with the motive to identify counterfeit notes. Images are acquired using the smartphone camera and fed to the CNN network. The results obtained are encouraging and can be improvised by further research and improvements in the architecture of the Deep CNN model. A new approach of Convolution Neural Networks towards the identification of fake currency notes through their images is examined in this approach which is comparatively better than previous image processing techniques. This method is based on Deep Learning, which has seen tremendous success in image classification tasks in recent times. This technique can help both people and machines in identifying a fake currency note in real time through an image of the same. The Accuracy in the proposed system is evaluated using accuracy. The proposed system is implemented by building an architecture model from an existing CNN architecture(ResNet50). The model extracts the features of the training dataset images such as color, spatial information of bank notes, etc., I am using a weight file to train the model. The result of this process is used in validation and testing.

KEYWORDS: Image Processing, Feature Extraction, ResNet50 Architecture, Convolutional Neural Networks, Keras, TensorFlow

1. INTRODUCTION

Tensor Flow

Tensor Flow uses multi-layer neural networks to build complex applications with great accuracy. It can be used for image processing, video analysis, real-time object detection, decision-making, audio, manipulation, and the detection of anomalies in a dataset. It provides algorithms and structure to implement Machine Learning using ANN and decision trees to compute large numerical datasets while maintaining accuracy.

Keras

Keras is an open-source software library that provides a python interface for neural networks. It acts as an interface for the TensorFlow library. Keras supports utility layers like normalization, dropout, and pooling.

ResNet50

Resnet50 is a convolutional neural network (CNN) that is 50 layers deep. A Residual Neural Network (ResNet) is an Artificial Neural Network (ANN) of a kind that stacks residual blocks on top of each other to form a network[18]. A ResNet50 model was pretrained on a million images from the ImageNet database and can classify images into 1000 object categories. Based on this new dataset of CT images, a transfer learning model was adapted to significantly shorten the training time and improve the accuracy.

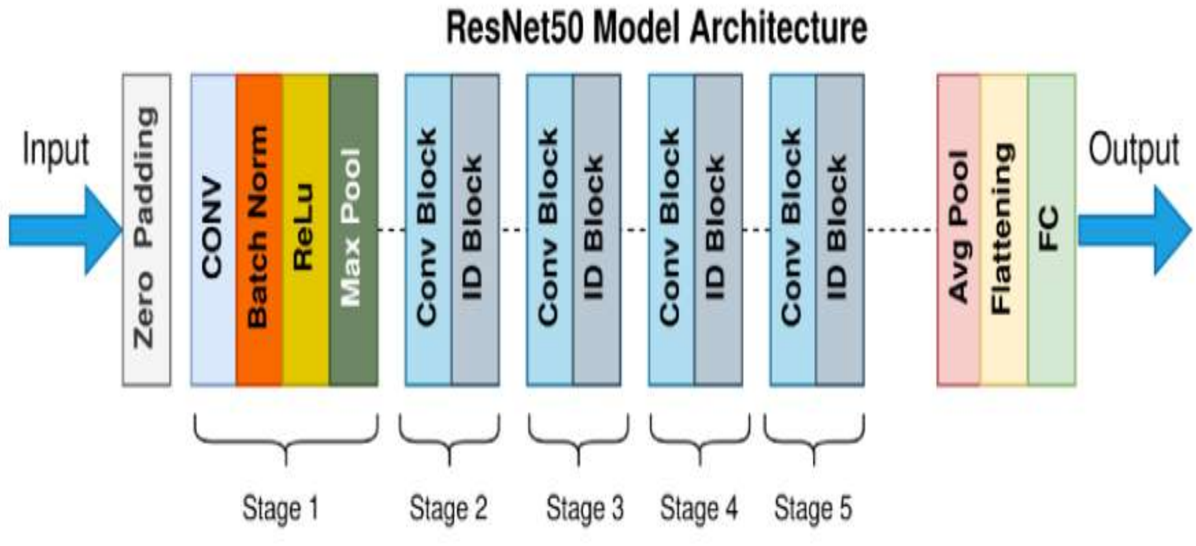

Figure 1: ResNet50 Architecture

CNN

A Convolutional Neural Network is a Deep Neural Network (DNN) widely used for the purposes of image recognition and processing and NLP. Also known as a ConvNet, a CNN has input and output layers, and multiple hidden layers, many of which are convolutional. In a way,CNN’ss are regularized multilayer perceptrons.

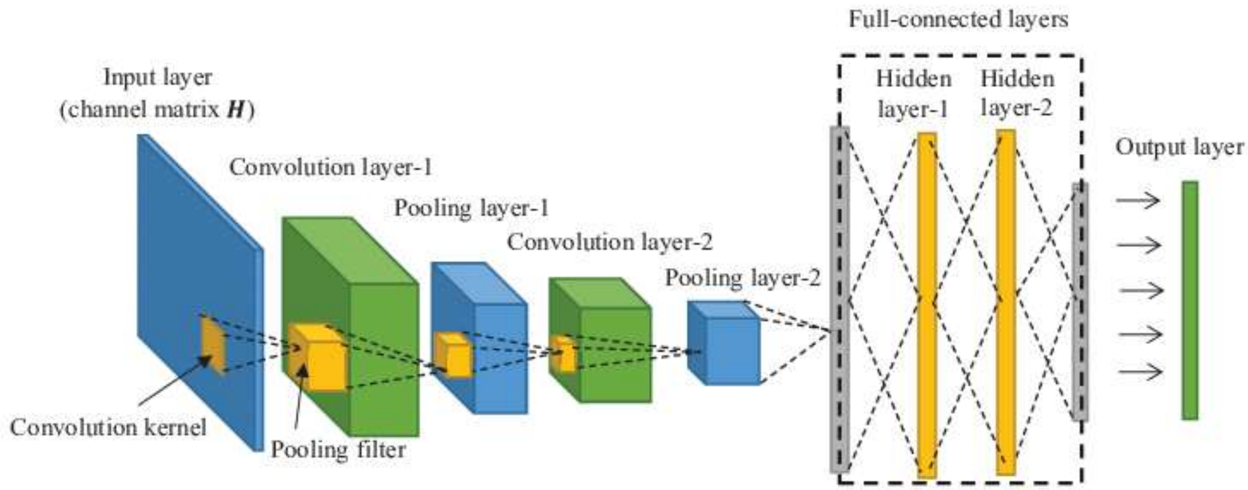

Figure 2: Basic Structure of CNN

1.1 Problem Statement:

Project currencies are the center of the target for counterfeiters. The counterfeit detection is mainly executed based on the Chemical or Physical properties of project currencies. Counterfeiters nowadays can evade the chemical property & physical feature-based counterfeit project currency detection system due to technological advancement. Moreover, the unavailability, high cost, poor accuracy, and lack of user-friendliness lead these fake detection tools to the least acceptance situation among the end-users. That is why a feature-based counterfeit detection system is now the focus of active research[12]. To overcome these drawbacks, this project proposes a detection technology for new banknotes and coins by applying faster region-based CNN, geometric constraints, and the residual network (ResNet).

1.2 Existing System

Currently, with the development of better image processing methods, new methods for the identification of currency is designed by analyzing specific safety information present in the currency. The safety features are watermarks, hidden images, security threads, and optically variable inks. Therefore, to determine the currency using image processing, extract the specific statistics from the currency image and select the correct recognition approach. The main methods for currency recognition is by characteristic geometric size [1] and by characteristic texture [2]. The general steps followed by the image processing approach are to acquire the image, detect the edge, convert the image to grayscale, feature extraction, image segmentation, and decision making [3].

Drawbacks of the existing system

1. The drawback of these approaches is detection efficiency is less since feature extraction is a challenging task.
2. In this existing architecture, the most effective front part of the note is to think about and now no longer the rear part.
3. Firstly we have the step known as image acquisition manner we have to take enter because the photograph is most effective via the scanner and on this, there may be little need for any virtual digital camera to size the photograph withinside the actual time gadget.

Proposed System

To overcome this problem now the trend is toward deep learning since it is a multilayer neural network. The deep neural network is effective for different applications in real time. The way we are going to achieve it is by training an artificial neural network on image data set of currency and making the NN (Neural Network) predict which class the image belongs to when it sees an image having a fake note or original note the next time. Convolutional neural networks (CNNs) are nowadays widely used in pattern recognition and image recognition problems. They have many advantages compared to other techniques. Typically, Convolution neural networks use approximately 5 to 25 distinct layers of pattern recognition. They take raw data, without the need for an initial separate pre-processing or feature extraction stage: in a CNN, the feature extraction and classification occur naturally within a single framework. This is a major advantage when compared to other image processing techniques, while they need a lot of computations only for pre-processing step. In general, to form a deep neural network, we need a large set of image data for the activity to be done. But thanks to the transfer learning technique, we only need a small number of data sets. What we do is take a model already trained in a large data set and use our weights to reconstruct the small data set we have. In this way, a large data set is not necessary and the model is also designed correctly. Hence in this project transferred learned ResNet50 is used by fine-tuning the last layer of this model to get the desired accuracy. Reset consists of convolutions, max pooling, dropout, ReLU activations, and fully-connected layers

Advantages:

1. We can obtain good accuracy.
2. The model takes less time and is less complex.
3. Provide cheaper and more accurate systems to the user which can be easily accessible and give accurate recognition of currency notes.

1.4 Requirements Specifications

1.4.1 Software Requirements

1.4.1.1 Operating System: Windows Environments

1.4.1.2 Platforms: Visual Studio Code

1.4.1.3 Language: Python 3.6

1.4.2 Hardware Requirements

1.4.2.1 Processor: Pentium IV Processor or higher

1.4.2.2 Hard disk: 256 GB

1.4.2.3 RAM: 512Mb or more

2. LITERATURE SURVEY

2.1. Previously Implemented Techniques for Fake Currency Detection

2.1.1 ETHIOPIAN BANK NOTE RECOGNITION AND FAKE DETECTION USING SUPPORT VECTOR MACHINE

In this research, we have implemented an Ethiopian banknote recognition system prototype that is able to classify denominations of Ethiopian banknotes and other countries. We collected around five hundred Ethiopian currency images and then divided them into two datasets so-called training (80%) and testing (20%) datasets. At last, the result was displayed in form of a confusion matrix with a total recognition accuracy rate of 98%. This prototype also tested some foreign banknotes like USA dollars$, Indian rupees, and others. This prototype only recognizes scanned or captured on the front side only. In the case of project currency verification validity, this prototype can detect counterfeit banknotes that compare the original stored intensity value of a thin strip line, a wide strip line, a watermark, and an identification mark with the input banknotes. In order to detect fake notes capture correctly the banknote images carefully against the background light effect and make sure hidden parameters are visible which are considered as basic features for currency verification using this technique working prototype able to detect fake notes with an average accuracy of 93% rate.

2.1.2 DESIGN AND IMPLEMENTATION OF PROJECT CURRENCY RECOGNITION WITH COUNTERFEIT DETECTION

Counterfeit currency recognition systems have become an important part of the banking sector. In the proposed system summation of non-masked pixel values in each banknote is computed and then fed to a Neural network. The performance of the proposed algorithm is evaluated on the Indian project currency system. The parameters with the most, influence on denomination classification are length, width, aspect ratio, hue, intensity, and then intensity standard deviation. If the values of the input note do not match the expected value of real notes, then the note is determined to be counterfeit. Using this algorithm, the success rate of counterfeit identification is 90%. As the techniques used have the advantage of low processing time, low intricacy, and reliability, it is suitable for real-time applications.

2.1.3 IDENTIFICATION OF FAKE NOTES AND DENOMINATION RECOGNITION

The Indian currency system has different denominations which are unique in one feature or the other. These features may be color, size or identification marks, etc. This system recognizes and transforms the framework in order to diminish human energy, and consequently perceive the measure of money esteem and to change it over it to different monetary forms without human supervision. Our proposed work differentiates white project and currency, by detecting the colors of different currencies. We have used a feature extraction mechanism to identify various identification patterns and also the denominations in the currency. We have developed an efficient system for the monetary standards 10,20,50,100,500 and 2000 which produces 98.8% accuracy.

2.1.4 INDIAN CURRENCY DETECTION USING IMAGE RECOGNITION TECHNIQUE

In this methodology, the hybrid algorithm based on PCA and LBP techniques here basically increases the recognition accuracy by giving 100% correct recognition. The database of the images should be enough large i.e., should contain samples of different forms of currency including the n notes, dirty notes, and torn notes. So that an increase in the accuracy. We can implement the currency recognition-based application for mobile users to increase the availability and to make it handy, in addition to the above-mentioned points, the added features i.e. GUI interface and portability. Neural networks can also be considered for the process of training the images. Because Artificial Neural Network-based currency classification is one of the most frequently used methods.

2.1.5 HYBRID DISCRIMINATIVE MODELS FOR BANKNOTE RECOGNITION AND ANTI-COUNTERFEIT

This project presents a solution for banknote recognition and authenticity based on a hybrid of deep learning and traditional SVM. The approach hybrid of machines with expected to produce good results in both situations of distinguished and confused face values from national currencies. The CNN model is used as a feature extraction instead of face value recognition and counterfeit detection. The color sample is classified into face value and national currency classes by using the SVM technique. The proposed approach is implemented on real datasets. In future works, we are focusing on reducing computational cost and improving accuracy for more national currencies and face values in order to apply for the real-time banknote inspection system.

2.1.6 INDIAN FAKE CURRENCY DETECTION USING COMPUTER VISION

In this project, we proposed ORB(Oriented FAST and Rotated BRIEF) and Brute-Force matcher in OpenCV for the Indian currency detection system and currency security feature; everybody has their own centrality. By using the said technique we have found that extraordinary results can be completed in so much less time. By using the said technique we have found that extraordinary results can be completed in so much less time. The project also includes the study of detailed information of about various Indian currency notes. This is an OpenCV-based using effective computer vision methods and algorithms which provide accurately and gives reliable results. At present we are having new MG series Indian currency note Rs.200 and we can also make experiment with notes Rs. 2000, Rs. 500, Rs. 100, Rs.50, Rs.20, and Rs.10. Our experiment shows that this is a low-cost system to detection the Indian banknote. We checked different notes on this system and then a result is 95.0% which means that the system is working efficiently. In the future, we will develop an android app for the detection of Indian currency.

2.2. Tabular Representation

S. No	YEAR	AUTHOR	TITLE	TECHNIQUES	ADVANTAGES	DISADVANTAGES
1.	2018	Ayalew Tessfaw, M.E.,Ramani,M.B, & KebedeBahiru, M. T.	Ethiopian Bank note Recognition and fake Detection Using SupportVectorMachine.	Image Processing, SVM	The result is displayed in the form of the confusion matrix with a total recognition accuracy rate of 98%.	This currency Recognition system Prototype only Recognizes banknotes scanned on the frontside only
2.	2016	Murthy,S.,Kurumathur, J.,& Reddy, B.R.	Design and implementation of project currency recognition with counterfeit detection.	Object Detection, Likelihood algorithm	To determine the unfit notes from fit notes with high accuracy.	Accuracy can be Improved by more than 90% but this process takes time.
3.	2018	Deepak M. P and Prajwala N. B	Identification of Fake Notes and Denominations-Recognition.	Probabilistic Neural Network, GLCM, Noise Removal Methods	Developed an efficient system for the monetary standards with high accuracy.	Software suggested here could not be utilized for various monetary standards.
4.	2020	Gautam,Kalpna	Indian Currency Detection using Image Recognition Technique.	Image Processing, PCA Algorithm LCB and Euclidean Distance	A hybrid algorithm based on PCA and also LBP Techniques. Here which basic-ally increased Recognition accuraacy by giving the 100% correct Recognition	The system doesn't recognize the hidden features like latent image and the watermark of the project currency

S. No	YEAR	AUTHOR	TITLE	TECHNIQUES	ADVANTAGES	DISADVANTAGES
5	2018	Van-dunghoang, Hoang-Thang-vo	Hybrid discriminator models for banknote recognition and anti counterfeit	Deep Learning, Support Vector Machine Algorithm	The hybrid approach of SVM and Deep Learning provided good results	Less accuracy, better algorithms could increase accuracy
6	2020	Devid Kumar, Surendra Chauhan	Fake currency detection using computer-vision	ORB (Oriented FAST and Rotated BRIEF) and Brute Force matcher inOpenCV, Contour Analysis, Face Recognition,Speeded UP Robust Features (SURF) and CannyEdge & Hough transformation algorithmOpenCV	Canny Edge & Hough transformation algorithm of high performance,softwarecore performs flawlessly and detection rate tends to 100%	Some challenges that persist involves in tight hardware requirements to create the environment for extracting featues of project currency.

Table 1: Tabular Representation of the Background Research

3. METHODOLOGY

In the coding part, I am going to use Keras deep learning library in python to build our CNN (Convolutional Neural Network). We should install TensorFlow and Theano which works on the back end of Keras. TensorFlow is an open-source software library for data flow programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. After installing the required libraries, we train our model as discussed above. After training and testing the model, we set an epoch value that increases the accuracy of the model upon increasing the value of epochs[10].

Convolution Neural Network (CNN) enhances image resolution and helps to increase the resolution of the old manuscript [15]. In the proposed model to identify the counterfeit India currency note, we trained the model using Deep CNN, with three Convolution layers as shown in figure . We use two fully connected layers to classify the note, to get the probability of the currency note either fake or original. The proposed scheme objective is to learn more features using fully connected layer. For this optimization hyper parameter is used to grow up the learning feature. The method is tested in real time using webcam, as soon as the image is captured the network starts learning the feature of the input currency note and compares with the learned features and gives the result as “Real Note” (or) “Fake Note”.

3.1 System Architecture

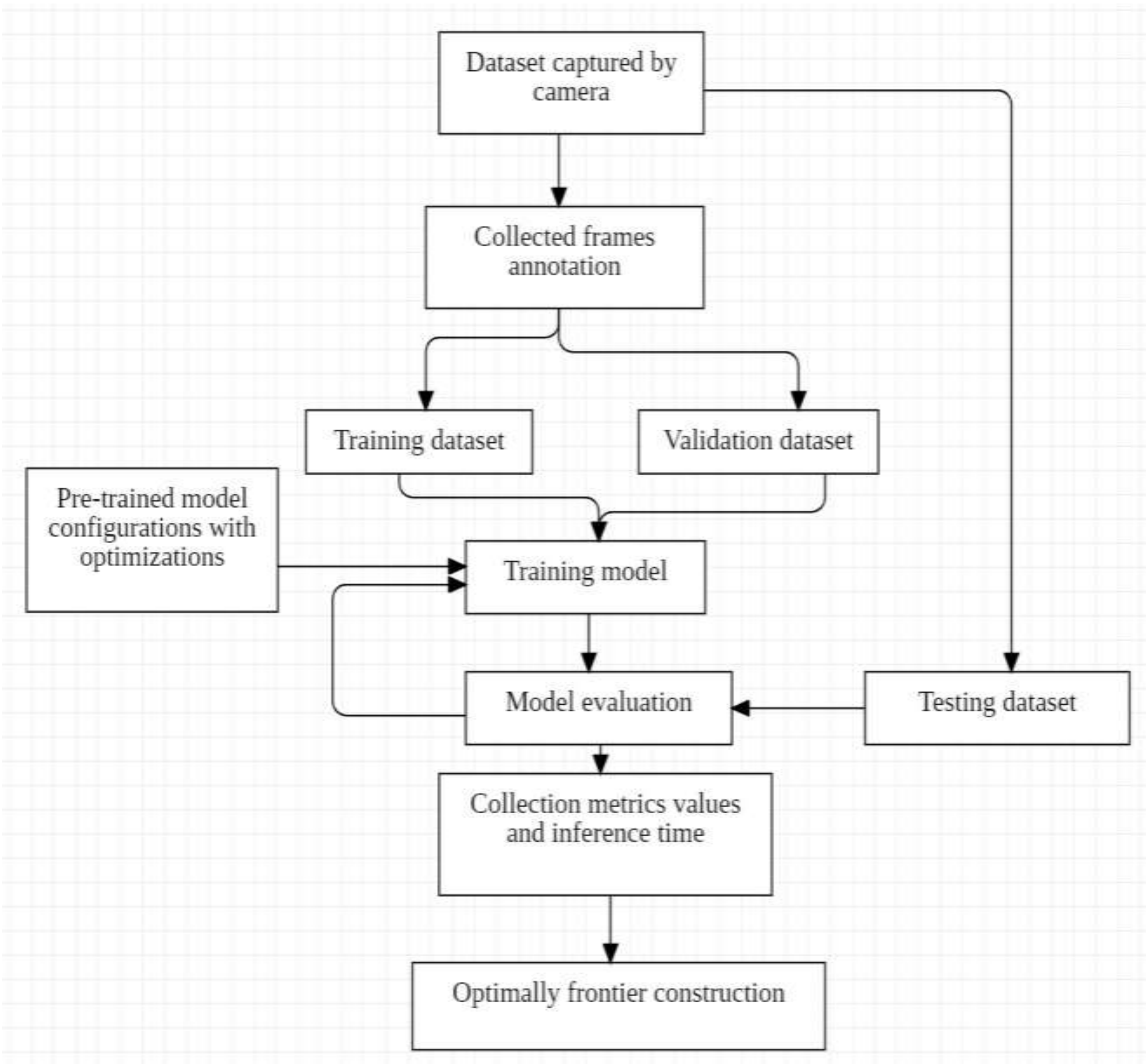

Figure 3.1: Architecture Diagram – Decision making and Implementation

This part is to propose the methodology to determine the best CNN model for the real-time Indian banknote classification by iteratively varying the pre-trained model configurations. Figure 3.1 demonstrates the block diagram of the proposed model procedure. First, the input images were applied to the pre-trained CNN model. The captured image is pre-processed, features are extracted and background noise is removed. The specific currency features such as The security features are watermarks, hidden images, security threads, optically variable inks, etc, extracted and are analyzed, and hoarded. The captured input image, after being pre-processed and the currency notes features being extracted. An approach with a backpropagation network and principal component analysis when used for feature extraction, the approach is successful with an additional investigation [12]. In convolutional-based neural networks when the number of layers is more the complexity is also more [13]. But convolution neural network is robust and detection speed is faster [14]. Then feature extraction prediction process is done. Here we get a class label as Fake or Real.

3.2 Modules

The entire proposed system of "Fake Currency Detection" can be defined by the following Models and Classifiers.

3.2.1 Image Processing Module:

Digital image processing consists of the manipulation of images using digital computers. Often, the raw image is not directly usable and requires processing. The processing of digital images can be divided into several classes: image enhancement, image restoration, image analysis, and image compression. Here, first, the fake currency is inputted into the model in the .jpg format. The features of the notes are extracted by the model and then compared against the features learned during the training of the dataset. And, then the note is classified as fake or real.

3.2.2 Convolutional Neural Network Module:

The convolution neural network(CNN) model is built with the motive to identify counterfeit notes on handy devices like smartphones, and tablets. The model built was trained and tested on a self-generated dataset. Images are acquired using the smartphone camera and fed to the CNN network. The image features are extracted by the CNN model and the features are extracted when the sample test input image is given the model compares the features with the analysis previously attained from the training dataset and finally the weights are saved in a file. The classification is finally made as Real or Fake.

3.2.3 ResNet50 Architecture Module:

ResNet-50 is also the kind of convolutional neural network that is trained on more than 1,000,000 pics from the ImageNet database [1]. This type is about 50 layers deep and can classify images into one thousand item classes, which incorporate keyboard, mouse, pencil, and masses of animals. As a result, the network has found wealthy characteristic representations for a big variety of pictures[12]. Deep neural networks start to converge we see another problem of the accuracy getting saturated and then degrading rapidly and this was not caused by overfitting as one may guess and adding more layers to a suitable deep model just increased the training error. This problem was further rectified by taking a shallower model and a deep model that was constructed with the layers from the shallow model and adding identity layers to it and accordingly the deeper model shouldn't have produced any higher training error than its counterpart as the added layers were just the identity layers.

3.3 Libraries

a Keras

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. Keras High-Level API handles the way we make models, define layers, or set up multiple input-output models. Image processing requires deep learning methods that use data to train the neural network algorithms to do various machine learning tasks. CNN’s are mainly used to classify different types of objects by analyzing images. Processing the image need many different techniques.

b TensorFlow

TensorFlow is an end-to-end open-source machine learning platform with a focus on deep neural networks. Deep learning is a subtype of machine learning that analyses massive amounts of unstructured data. Since it works with structured data, deep learning is different from normal machine learning. Keras is a high-level neural network library that runs on top of TensorFlow.

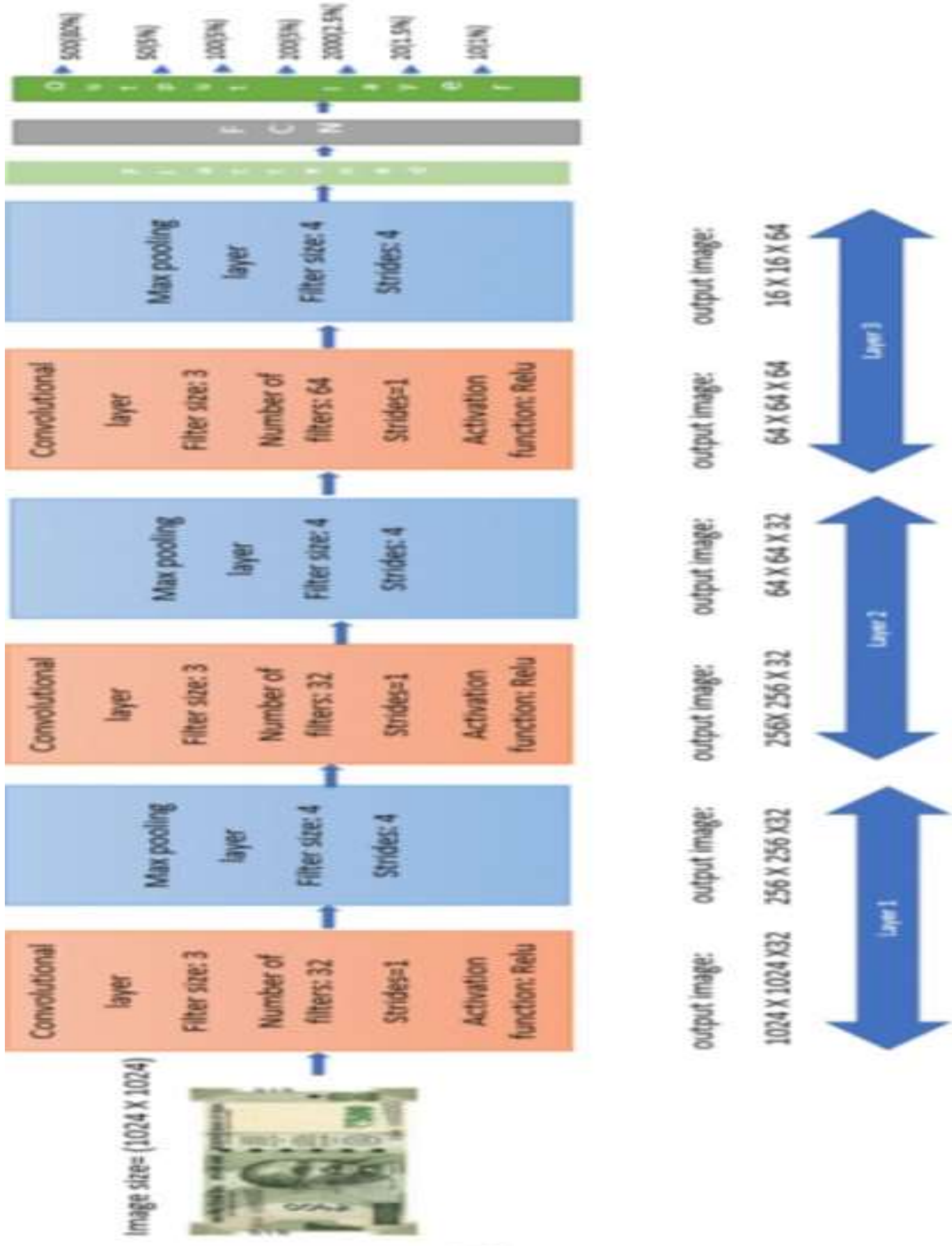

Figure 3.2: Pipeline representation for entire system

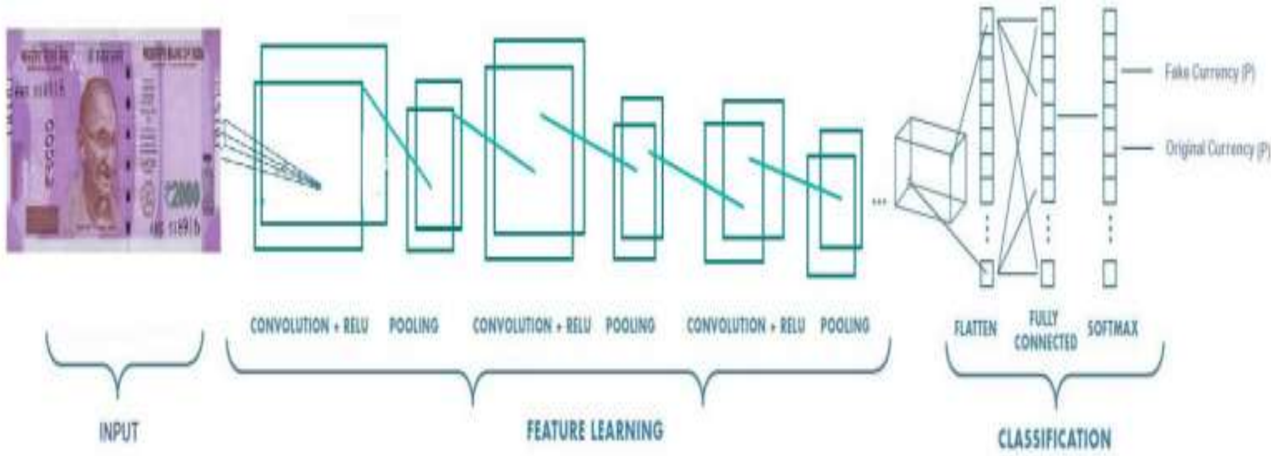

Figure 3.3: Identifying Counterfeit Currency note -2000 using ResNet50

3.4 Diagrammatic Representations

3.4.1 STATE CHART DIAGRAM

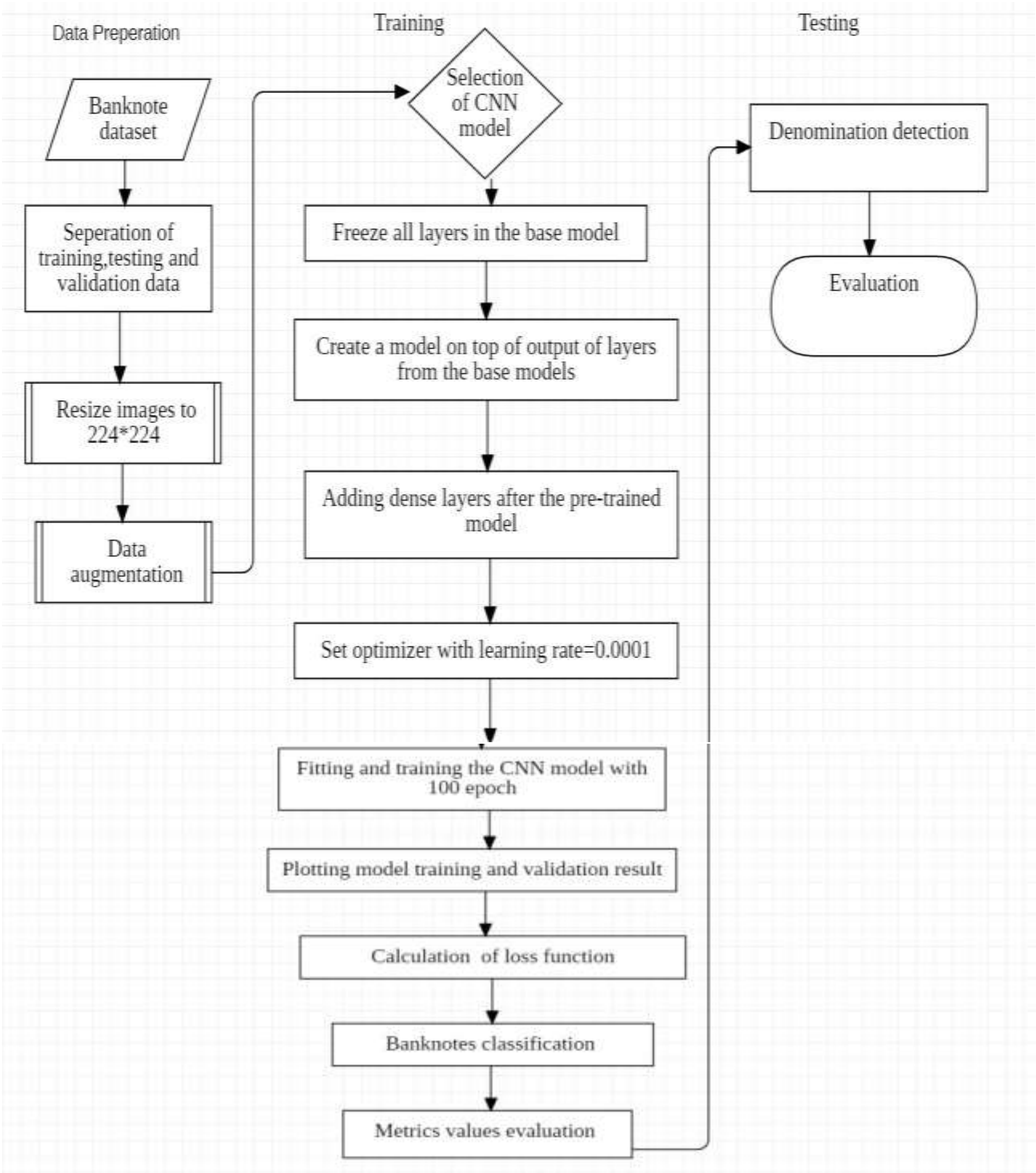

Figure 3.4: State chart Diagram- Representation of States and their transformation

Figure 3.2 defines the Representation of states and how their transitions occur. The initial state The methodology deals with capturing of an image through the cam of a mobile phone. It runs into a loop until an input image is captured successfully.

The next stage is pre-processing. In this state, extraneous details and background noise is eliminated from the captured input image. The security features are watermarks, hidden images, security threads, and optically variable inks, etc, extracted. After this, in the next state, feature extraction is done. etc. are analyzed and hoarded[8].

The state after this is the feature extraction. The captured input image, after being pre-processed and the facial features being extracted, using the weighted file.

Then it is sent to prediction. If the image is properly predicted a class label as Fake or Real is appeared on the Keras window.

3.4.2 Class Diagram

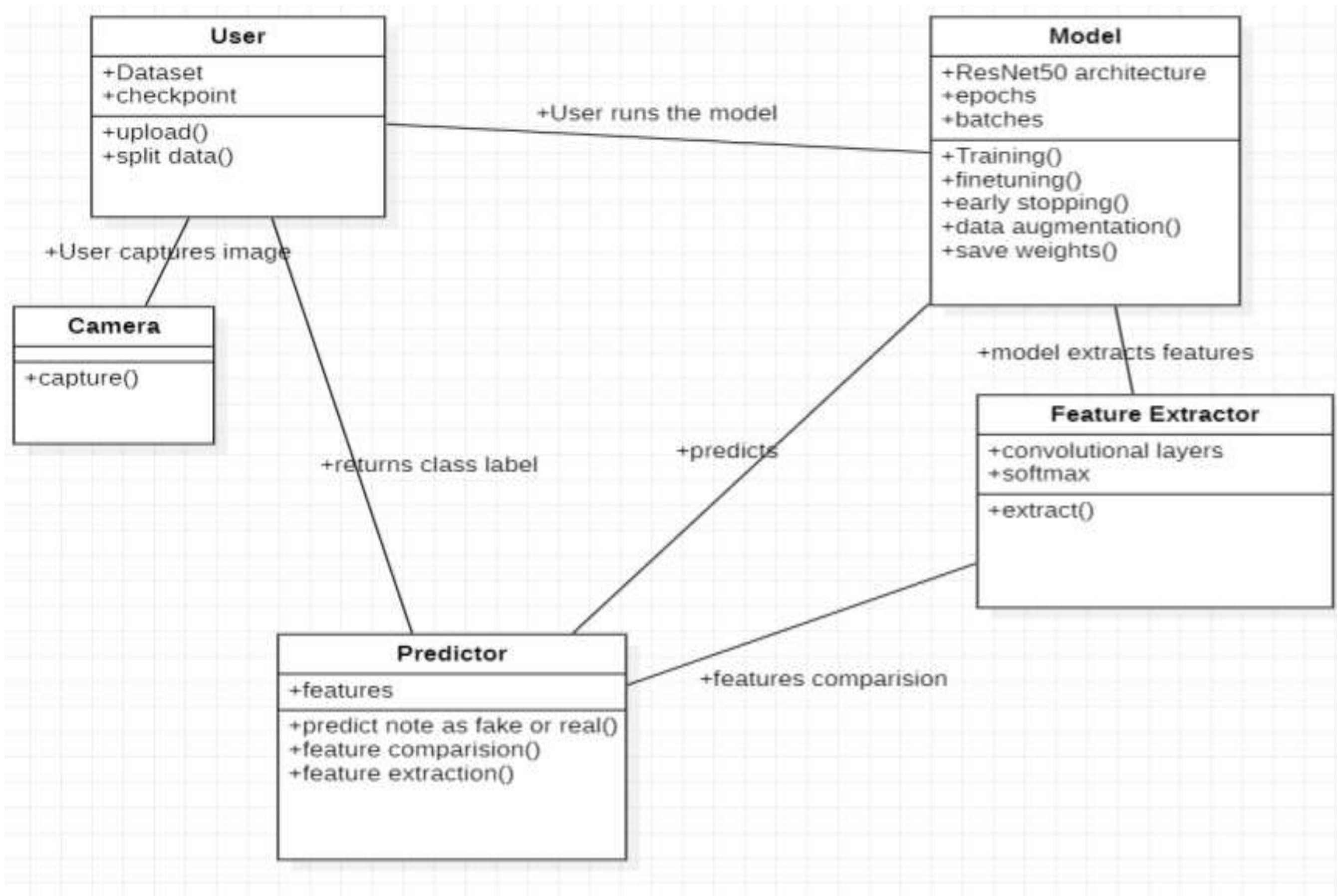

Figure 3.5: **Class Diagram** - Representation of classes and their relationships

The above figure provides the class diagram of the model. 5 classes are present which are User, Camera, Model, Feature Extractor, Predictor.

The user splits the dataset into the train, validation, and testing datasets, and builds the model on ResNet architecture.[2]

The camera is used to capture the image. Feature Extractor is used to extract all the features of the model using CNN layers. The model learns the features from the training images, Earlystops in the case of overfitting the model. The model also save the weights to Final_model.h5 file. The predictor is used to classify the image as real or fake.

3.4.3 Use Case Diagram

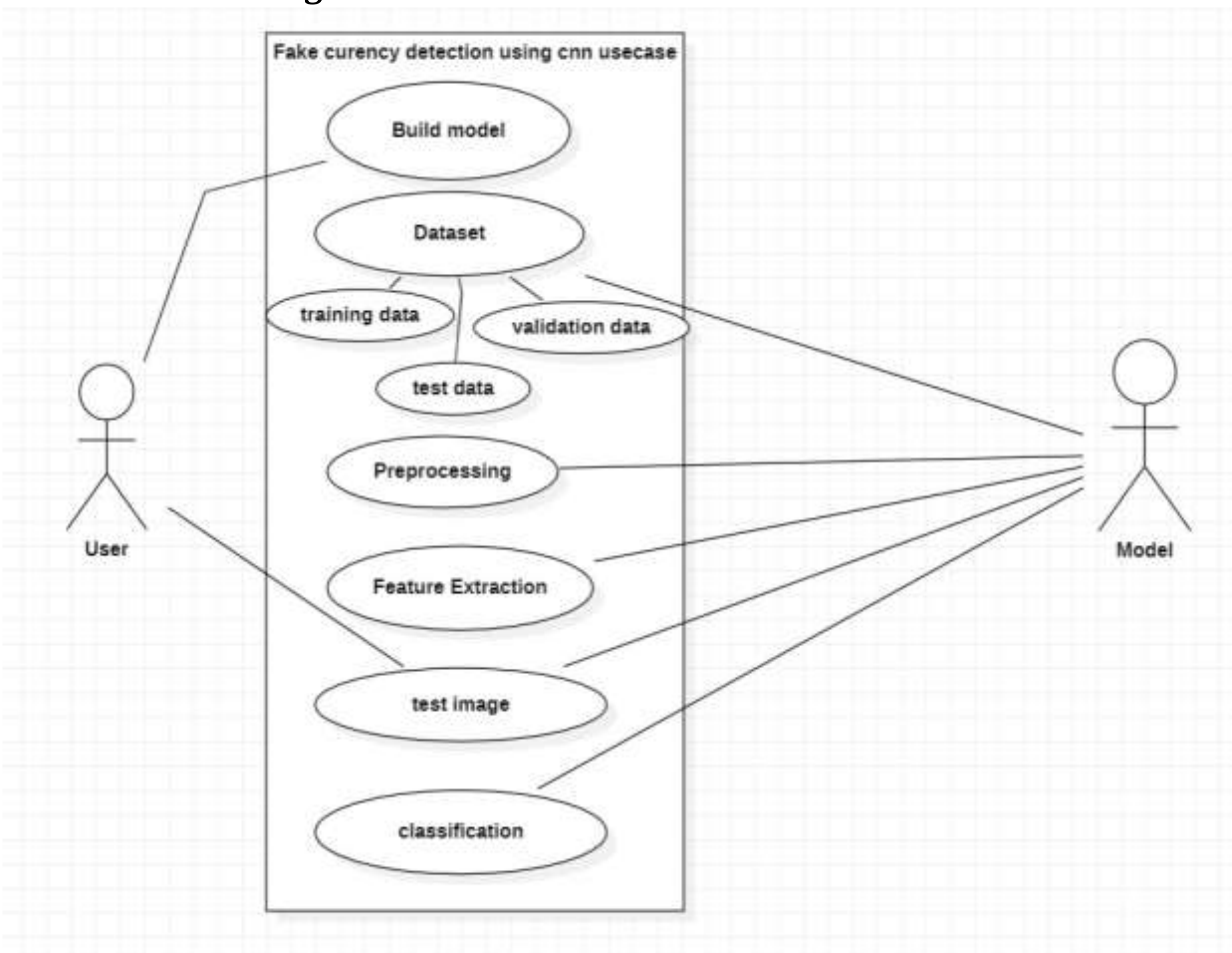

Figure 3.6: Use Case Diagram – Representation of the system's functionality by incorporating use cases, actors, and their relationships

Figure 3.4 is used for the representation of classes and their relationships. The Actors User and Model are present and use cases to build models, datasets, preprocessing, feature extraction, test image, and classification.[13] The user builds the model on pre-trained ResNet50 architecture. The data set is divided into train, test, and validation datasets. The datasets are used to train the model.

Feature Extractor is used to extract all the features of the model using CNN layers. Classification is done by the model when the input image is given as a test image to the model in .jpg format which is captured using the camera of the mobile phone. The output is given as a Fake or Real class label[17].

3.4.4 Sequence Diagram

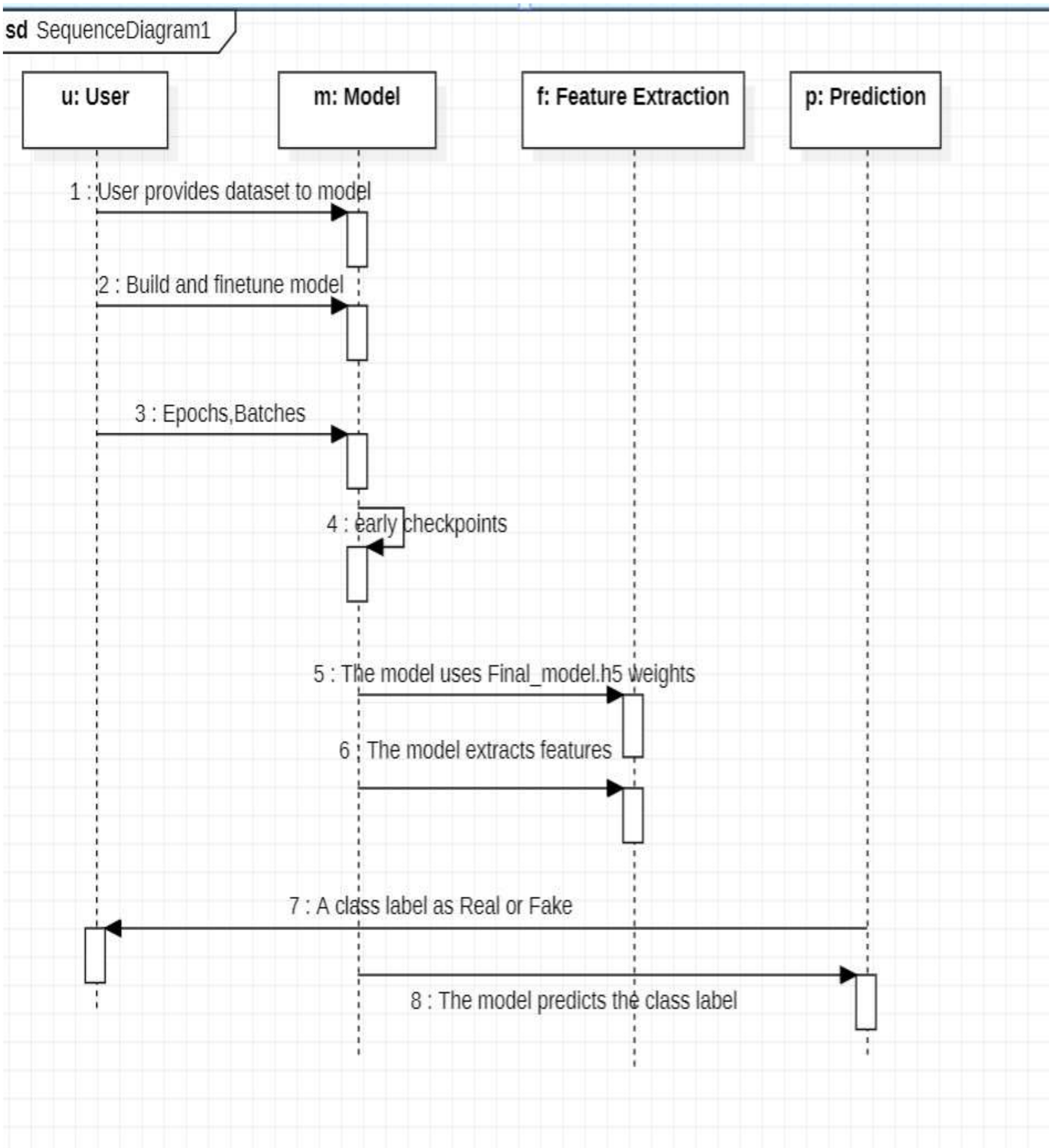

Figure 3.7: Sequence Diagram - Object Interaction and Process Execution

Figure 3.7 indicates the interaction among the application's objects. There are four objects, User, Haarcascade, Feature Extraction, and Prediction[5].

The 'User' himself/herself starts the model, and the Haarcascade detects the image of the user and sends it to the Feature Extraction. All the features of the user are extracted and the prediction process is started. If the Keras window cannot predict the face it returns a no faces label in the console window.

4. TESTING AND RESULTS

4.1 Results

1.When Main.ipynb is compiled and run
Firstly, the input image is given

Figure 4.1: Input 1 (Real.jpg)

Then the image is taken as input and trained using the weight file Final_model.h5 then produces the output as *Real*.

Figure 4.2: Output1(Real)

Output2: As the input image is real, The output is given as Real

Similarly, I have another input:

Figure 4.3: Input 2(Fake.jpg)

The output of Input 2 image:

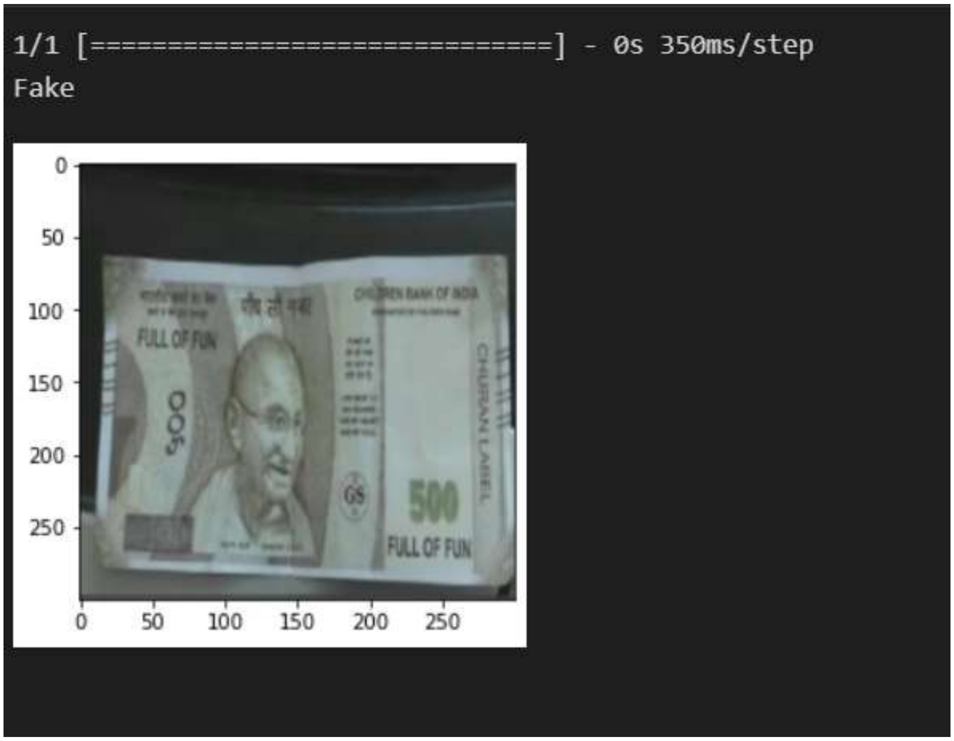

Figure 4.4: Output2(Fake)

Input2 is taken as input and trained using the weight file Final_model.h5 then produced the output as *Fakel.*

4.2 Results

Train Accuracy:

A train accuracy metric is used to measure the algorithm's performance in an interpretable way. It is the measure of how accurate the proposal model's prediction is compared to the true data. The banknote classification of training accuracy results is 93.2%. However, the Resnet50 model also shows a better accuracy value of 96.88% using batch sizes 32 and 128.

Optimsation	Batchsize32	Batchsize64	Batchsize128
SDG	91.2	92.5	93.2

Table 2: Training accuracy of batch sizes 32, 64, and 128 for ResNet Model

Test Accuracy:

The test accuracy, we must measure performance on unseen data, which are not used for training data. To measure the test accuracy, use the test dataset. The banknote classification test accuracy is 93.6%.

Optimsation	Batchsize32	Batchsize64	Batchsize128
SDG	93.1	93.3	93.6

Table 3: Training accuracy of batch sizes 32, 64, and 128 for ResNet Model

Train Loss:

In this project, we have employed the loss function as categorical cross-entropy to classify multiple classes. The ResNet50 showed good training and test accuracy, but it has not achieved zero loss.

Optimsation	Batchsize32	Batchsize64	Batchsize128
SDG	2.66	0.03	0.03

Table 4:Training Loss of batch sizes 32, 64, and 128 for ResNet Model

Test Loss:

When we increased the number of layers, the training and test error rate also increases. ResNet50 showed almost equal training and test accuracy, but the test loss is very high for ResNet50 because its gradient is too large, so this model is overfitting the data.

Optimsation	Batchsize32	Batchsize64	Batchsize128
SDG	1.06	1.30	1.4

Table 5: Testing Loss of batch sizes 32, 64, and 128 for ResNet Model

Table representation of obtained results

S.No	Training set	Validation set	Epoch	Training Accuracy	Input	Probability Fake Original	Output	Remark
1	200	55	100	93.2	2000_test.jpg (original)	0.559 0.441	Fail	Original note with stains
2	200	55	100	93.2	500_6.jpg (original)	0.001 0.999	Success	-
3	200	55	100	93.2	500_4.jpg (Fake)	0.835 0.165	Success	-
4	200	55	100	93.2	2000.jpg (original)	0.356 0.644	Success	-
5	200	55	100	93.2	2000-3.jpg (Fake)	0.755 0.245	Success	-
6	200	55	67	89.23	2000_test.jpg (original)	0.423 0.577	Fail	Original note with stains
7	200	55	45	85.7	500-6.jpg (original)	0.006 0.994	Success	-

8	200	55	45	85.7	500-4.jpg (original)	0.750 0.250	Success	-
9	200	55	45	85.7	2000-5.jpg (original)	0.411 0.589	Success	-

Table 6 : Displays the prediction accuracy value of Fake /Original currency note

Sample data set images of Indian Currency

5(a) Original Currency Rs 500

5(b) Original currency note Rs 200 with stain

5(c) Fake currency note Rs 2000

5(d) Fake currency of Rs 200

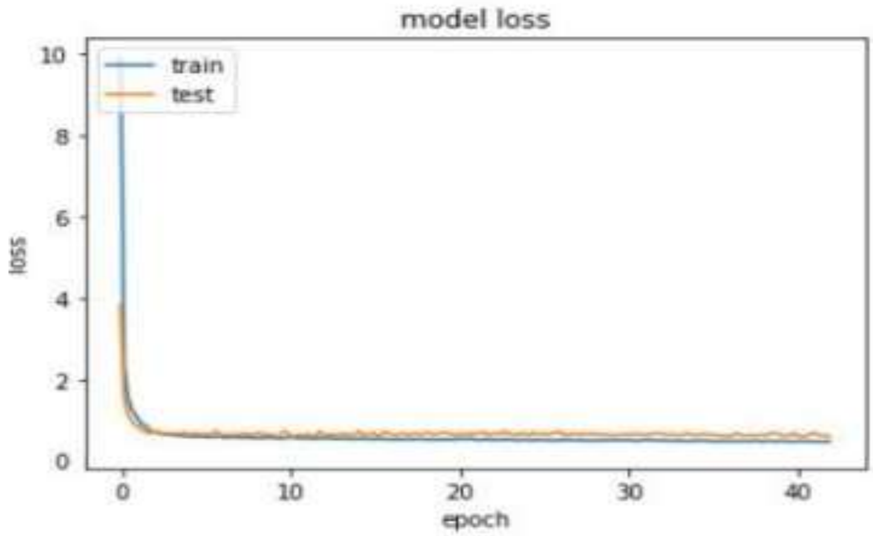

Figure 4.5: Display 40 epochs-model loss

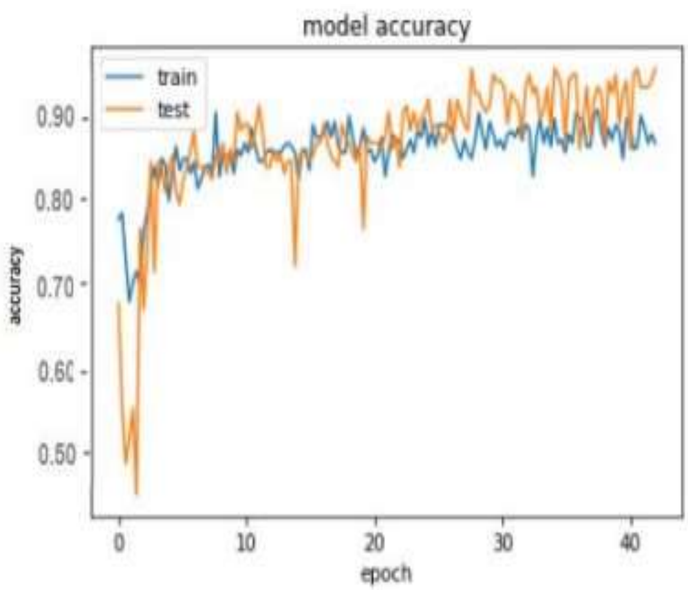

Figure 4.6: Display 40 epoch - model accuracy and loss

5. Conclusion

5.1 Conclusion

In this model, transfer learned ResNet50 as our model for performing the fake currency detection. The detection accuracy is most accurate since the currency characteristics features are learned layer by layer. Here we have considered the whole currency image, but in the future, we will try to include all the security features of currency by employing a suitable structural design and with suitable training data[20]. This model can also be compared with various architectures of CNN which may have a low error rate. The proposed model is considered with a minimal effort that can be utilized to recognize various features of a cash note and consummately works for Indian money Rs 200, 500, and 2000. In this system, 306 images are considered after augmentation out of which 80% were used for training and the remaining 20% were used for validation, the learning rate is 0.001, and the model finally achieved an accuracy of 96.6 with a success rate of 80%, the proposed model can separate the features even the note has scribbled on it. The testing accuracy obtained is about 93.2%, and training and validation accuracies were 93.57% and 92.55% respectively.

5.2 Future Scope

Further studies would be conducted on deep networks that can detect small objects such as coins in images more accurately. In addition, a shallow network-based detection method that can shorten the processing time would be examined.

Bibliography

[1] S. Arya and M. Sasikumar, "Fake Currency Detection," 2019 Int. Conf. Recent Adv. Energy Efficient Comput.Commun. ICRAECC 2019, pp. 2019–2022, 2019, doi: 10.1109/ICRAECC43874.2019.8994.

[2] A. Kumar and A. Kumar, "Dog Breed Classifier for Facial Recognition using Convolutional Neural Networks," pp. 508–513, 2020.

[3] M. Haider Ali, "Thesis Report on Fake Currency Detection using Image Processing Method," Akiful Mohaimin Rifat Islam Shahriar Chowdhury, no. 13301148, pp. 1–38, 1330.

[4] M. N. Shende and P. P. Patil, "A Review on Fake Currency Detection using Image Processing," Int. J. Futur. Revolut. Comput. Sci. Commun. Eng., vol. 4, no. 1, pp. 391–393, 2018.

[5] T. Agasti, G. Burand, P. Wade, and P. Chitra, "Fake currency detection using image processing," IOP Conf. Ser. Mater. Sci. Eng., vol. 263, no. 5, pp. 88– 93, 2017, doi: 10.1088/1757-899X/263/5/052047

[6] S. Adhikari, S. Thapa, and B. K. Shah, "Oversampling based Classifiers for Categorization of Radar Returns from the Ionosphere," Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020, no. Icesc, pp. 975–978, 2020, doi: 10.1109/ICESC48915.2020.9155833.

[7] A. Ghimire, S. Thapa, A. K. Jha, S. Adhikari, and A. Kumar, "Accelerating business growth with big data and artificial intelligence," Proc. 4th Int. Conf. IoT Soc. Mobile, Anal. Cloud, ISMAC 2020, pp. 441–448, 2020, doi: 10.1109/ISMAC49090.2020.9243318.

[8] P. Gidveer, S.R. Darade, Automatic Recognition of Fake Indian Currency Note, International Conference on Electrical Power and Energy Systems (ICEPES), Bhopal, India, pp. 290–294.

[9] D.A.K.S. Gunaratna, N.D. Kodikara, Premaratne, ANN Based Currency Recognition System using Compressed Gray Scale and Application for Sri Lankan Currency Notes - SLCRec, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering 2 (2008) 2957–2962.

[10] P. Viola and M. Jones, ''Rapid object detection using a boosted cascade of simple features,'' in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Kauai, HI, USA, Dec. 2001, pp. I-511–I-518.

[11] F. Grijalva, J. C. Rodriguez, J. Larco, and L. Orozco, ''Smartphone recognition of the U.S. Banknotes' denomination, for visually impaired people,'' in Proc. IEEE ANDESCON, Bogota, Colombia, Sep. 2010, pp. 1–6.

[12] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[13] I. A. Doush and S. AL-Btoush, ‘‘Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms,’’ J. King Saud Univ.-Comput. Inf. Sci., vol. 29, no. 4, pp. 484–492, Oct. 2017.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, ‘‘Grad-CAM: Visual explanations from deep networks via gradientbased localization,’’ in Proc. IEEE Int. Conf. Comput. Vis., Venice, Italy, Oct. 2017, pp. 618–626.

[15] T. M. Hoang, S. H. Nam, and K. R. Park, ‘‘Enhanced detection and recognition of road markings based on adaptive region of interest and deep learning,’’ IEEE Access, vol. 7, pp. 109817–109832, Aug. 2019.

[16] K. N. Oo and A. K. Gopalkrishnan, "Zernike moment based feature extraction for classification of Myanmar project currencies," in *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 1–6, Bangkok, Thailand, 2018.

[17] O. K. Oyedotun and A. Khashman, "Banknote recognition: investigating processing and cognition framework using competitive neural network," *Cognitive NeuroDynamics*, vol. 11, no. 1, pp. 67–79, 2017.

[18] P. Ponishjino, K. Antony, S. Kumar, and S. Jebakumar, "Bogus currency authorization using HSV techniques," Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2017, vol. 2017- January, pp. 179–183, 2017, doi: 10.1109/ICECA.2017.8203667.

[19] Q. Zhang and W. Q. Yan, "Currency Detection and Recognition Based on Deep Learning," Proc. AVSS 2018 - 2018 15th IEEE Int. Conf. Adv. Video SignalBased Surveill., pp. 0–5, 2019, doi: 10.1109/AVSS.2018.8639124.

[20] A. Upadhyaya, V. Shokeen, and G. Srivastava, "Analysis of counterfeit currency detection techniques for classification model," 2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018, pp.1–6, 2018, doi: 10.1109/CCAA.2018.8777704

APPENDIX

Main.py

“”””

Fake Currency Detection

“”””

#TenosorFlow -TensorFlow is an end-to-end open source platform for machine learning.

from tensorflow.keras.applications.resnet50 import ResNet50

ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks

#resnet. preprocess_input will convert the input images from RGB to BGR, then will zero-center each #color channel with respect to the ImageNet dataset, without scaling

from tensorflow.keras.applications.resnet50 import preprocess_input

#Keras and TensorFlow are open-source Python libraries for working with neural networks, creating #machine learning and performing deep learning. Because Keras is a high level API for TensorFlow, they are installed together.

from tensorflow import keras

#Image Processing with Keras in Python. Keras API is a deep learning library that provides methods to #load, prepare and process images.

from keras.preprocessing import image

#Keras ImageDataGenerator is used for getting the input of the original data and further, it makes the #transformation of this data on a random basis and gives the output resultant containing only the data that #is newly transformed.

from keras.preprocessing.image import ImageDataGenerator

from keras.preprocessing import image

Dropout is a way of cutting too much association among features by dropping the weights #at a probability Reducing associations can be applied among any layers which stops weight updation for the edge.

Flatten layers are used when you got a multidimensional output and you want to make it linear to pass it #onto a Dense layer

from keras.layers import Dense,Activation,Flatten,Dropout

A sequential model is appropriate for a plain stack of layers where each layer has exactly one #input tensor and one output tensor.

from keras.models import Sequential,Model,load_model

SGD is the default optimizer for the python Keras library as of this writing. SGD differs from regular #gradient descent in the way it calculates the gradient

from keras import optimizers

Too many epochs can lead to overfitting of the training dataset, whereas too few may result in an underfit #model. Early stopping is a method that allows you to specify an arbitrary large number of training epochs #and stop training once the model performance stops improving on a hold out validation dataset.

from keras.callbacks import ModelCheckpoint,EarlyStopping

When training deep learning models, the checkpoint is at the weights of the model. These weights #can be used to make predictions as is or as the basis for ongoing training.

NumPy is a Python library used for working with arrays. It also has functions for working in domain of #linear algebra, fourier transform, and matrices.

import numpy as np

We use Matplotlib to ease the analysis of statistical data. Matplotlib is a visualization tool and hence #provides a visual analysis of the data.

import matplotlib.pyplot as plt

#Define height and width of the image

height=300

width=300

#Create a ResNet model Instance without top layer as we will add out own top layer

base_model = ResNet(width=’imagenet’,include_top=False,input_shape=(height,width,3)

#define directory containing training and validation data

train_dir=”C://Users//hasin//OneDrive//Fake_currency_detection//FakeCurrencyDetectionSystem-20221018T194432Z- 001//FakeCurrencyDetectionSystem//Dataset//Training”

```
Validation_dir="C://Users//hasin//OneDrive//Fake_currency_detection//FakeCurrencyDetection
System-20221018T194432Z-001//FakeCurrencyDetectionSystem//Dataset//Training"
#number of batches data has been divided into
batch_size=8
#create datagen and generator to load the data from training directory
train_datagen=ImageDataGenetor(preprocessing_function=preprocess_input,rotation_range=90,
horizontal_flip=True,vertical_flip=True)
train_generator=train_datagen.flow_from_directory(train_dir,target_size=(height,width),batch_si
ze=batch_size)

#Create datagen generator to load from validation directory

validation_datagen=ImageDataGenerator(preprocessing_function=preprocess_input,rotation_ran
ge=90,horizontal_flip=True,vertical_flip=True)validation_generator=validation_datagen.flow_fr
om_directory(validation_dir,target_size=(height,width),batch_size=batch_size)
#our own model which will be added onto the ResNet50 model
def    build_finetune_model(base_model,dropout,fc_layers,num_classes):
        for layer in base_model.layers:
                layer.trainable=False
                x=base_model.output
                x=Flatten()(x)
        for fc in fc_layers:
                x=Dense(fc,activation='relu')(x)
                x=Dropout(dropout)(x)
       predictions=Dense(num_classes,activation='softmax')(x)
        finetune_model=Model(inputs=base_model.input,outputs=predictions)
        return finetune_model
class_list=['Real','Fake'] #the labels of our data
FC_Layers=[1024,1024]
dropout=0.5
```

```
finetune_model=build_finetune_model(base_model,dropout=dropout,fc_layers=FC_Layers,num
_classes=len(class_list))

#define number of epochs(the number of times the model will be trained) and number of training
#images
num_epochs=100
num_train_images=35

#checkpoint stops the process if anything goes wrong
checkpoint=ModelCheckpoint("Final_model.h5",monitor='val_acc',verbose=1,save_best_only=T
rue,save_weights_only=False,mode='auto',save_freq=1)
early=EarlyStopping(monitor='val_acc',min_delta=0,patience=40,verbose=1,mode="auto")
#compile the model before using
# we need to recompile the model for these modifications to take effect
# we use SGD with a low learning rate
finetune_model.compile(loss="categorical_crossentropy",optimizer=keras.optimizers.SGD(lr=0.
000001,momentum=0.9),metrics=['accuracy'])
#train the model
finetune_model.fit_generator(generator=train_generator,steps_per_epoch=num_train_images//ba
tch_size,epochs=num_epochs,validation_data=validation_generator,validation_steps=1,callbacks
=[checkpoint,early])
#save the model
finetune_model.save_weights("Final_model.h5")

#testing the model
from tensorflow.keras.preprocessing.image import load_img
img=load_img("Dataset/Testing/Real.jpg",target_size=(300,300))
#The path of the testing image,the pic taken from the phone should come here
img=np.asarray(img)
plt.imshow(img)
img=np.expand_dims(img,axis=0)
```

```
finetune_model.load_weights("Final_model.h5")
output=finetune_model.predict(img) #predicting the image using model created
if(output[0][0]>output[0][1]): #comparison
    print("Fake")
else:
    print("Real")

#testing the model
img=load_img("Dataset/Testing/Fake.jpeg",target_size=(300,300))
#The path of the testing image,the pic taken from the phone should come here
img=np.asarray(img)
plt.imshow(img)
img=np.expand_dims(img,axis=0)
finetune_model.load_weights("Final_model.h5")
output=finetune_model.predict(img) #predicting the image using model created
if(output[0][0]>output[0][1]): #comparison
        print("Fake")
else:
        print("Real")
```