TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

Minor Project Report

On

NEPALESE CURRENCY RECOGNITION SYSTEM

Submitted By:

Ashma Rai (THA075BCT011)

Niruta Shrestha (THA075BCT030)

Shikshya Shiwakoti (THA075BCT041)

Swostika Basukala (THA075BCT048)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

May, 2022

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS

Minor Project Report

On

NEPALESE CURRENCY RECOGNITION SYSTEM

Submitted By:

Ashma Rai	(THA075BCT011)
Niruta Shrestha	(THA075BCT030)
Shikshya Shiwakoti	(THA075BCT041)
Swostika Basukala	(THA075BCT048)

Submitted To:
Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the award of the
Bachelor's Degree in Computer Engineering

Under the Supervision of
Shikhar Bhattarai

May, 2022

DECLARATION

We hereby declare that the report of the project entitled "Nepalese Currency Recognition" which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus,** in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Engineering,** is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Ashma Rai (THA075BCT011) ________________________

Niruta Shrestha (THA075BCT030) ________________________

Shikshya Shiwakoti (THA075BCT041) ________________________

Swostika Basukala (THA075BCT048) ________________________

Date: May, 2022

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus,** a minor project work entitled "**Nepalese Currency Recognition System**" submitted by **Swostika Basukala, Shikshya Shiwakoti, Niruta Shrestha** and **Ashma Rai** in partial fulfillment for the award of Bachelor's Degree in Computer Engineering. The project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Computer Engineering.

____

Project Supervisor

Mr. Shikhar Bhattarai

Department of Electronics and Computer Engineering, Thapathali Campus

____

External Examiner

Mr. Sudeep Shakya

Associate Professor

Kathmandu Engineering College

____

Project Co-ordinator

Mr. Umesh Kanta Ghimire

Department of Electronics and Computer Engineering, Thapathali Campus

____

Mr. Kiran Chandra Dahal

Head of the Department,

Department of Electronics and Computer Engineering, Thapathali Campus

May, 2022

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to Er. Kiran Chandra Dahal, HOD of Department of Electronics and Computer Engineering, Thapathali Campus for providing this opportunity. We would also like to thank Er. Umesh Kanta Ghimire, Er. Rama Bastola, DHOD of Department of Electronics and Computer Engineering and Er. Shikhar Bhattarai, our supervisor under Thapathali Campus for their continuous and constructive critique in research and study for the preparation of this project. Without their guidance, this project would not have been successful.

This journey would also not have been possible without the sheer support of our classmates, friends and family members. We would like to acknowledge their encouragements and ever-lasting support.

Ashma Rai (THA075BCT011)

Niruta Shrestha (THA075BCT030)

Shikshya Shiwakoti (THA075BCT041)

Swostika Basukala (THA075BCT048)

ABSTRACT

This report represents the details on a project entitled "NEPALESE CURRENCY RECOGNITON SYSTEM" as a part of the curriculum for the third year minor project of B.E. in Computer Engineering. This report discusses various fundamentals and an implementation technique to build the offline system that should recognize the paper notes of Nepalese Currency. Our project mainly focuses on recognizing Nepalese Currency; thus developed system shall be a way to digitize the information in an image for the purpose of convenient retrieval and efficient processing of data. Dataset Preparation, Image Processing, Convolutional Neural Network and Image Classification are the few main areas, in which our project shall be based on. The prime focus of this project is firstly to train the model with a dataset of our own and hence predicting the classes in an input image by processing the image into a desired format which shall be then fed into the trained neural network.

Keywords: CNN, Currency Recognition, Dataset Collection, Image Classification, Nepalese Currency.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Network
CRSFVI	Currency Recognition System for Visually Impaired
FAST	Features from Accelerated and Segments Test
IKA	Inverse Kinematic Animation
LBP	Local Binary Pattern
LVQ	Learning Vector Quantization
ORB	Oriented FAST and rotated BRIEF
RBF	Radial Basis Function
SIFT	Scale-Invariant Features
UV	Ultraviolet

1. INTRODUCTION

1.1 Background

With the arrival of the new age of digitalization, many problems of the past were solved using the modern technologies and software. This has made the lives of differently-abled people easy and has made it possible for them to live as a normal person. Various software applications have been developed to help handicapped people such as Be My Eyes, TapTapSee, Supersense, Audible, Blind Square, Color ID, KNFB Reader and more.

1.1.1 Deep Learning

Deep learning is a type of machine learning in which different algorithms are considered to learn by machine itself. It is incorporated with artificial neural networks, which have been designed to imitate a human brain's capability of learning and thinking. This project uses Convolution Neural Network (CNN) because it is mostly preferred for Image Classification problems. Practical examples of deep learning are virtual assistants, face recognition, vision for driverless car and others.

1.1.2 Currency Recognition System

Every year there are many cases of counterfeit currency in Nepal. Large scale businesses use currency recognition device which verifies the authenticity using UV lights, security threads and more. But such devices is not feasible for small businesses. Currency recognition is even more difficult for individuals who are differently-abled and this is what the project aims to aid.

Currency Recognition System is an artificial intelligence (AI) based project whose main theme is to identify the paper notes. There are around 1.3 billion people estimated to be living with visual imparity worldwide, out of which 36 million are blind. Almost 220 million people have moderate to severe vision impairment [1]. In Nepal, even though the paper notes have some special markings in these notes to let the visually impaired know what the value of the currency is, blind people with years of experience finally learn to recognize them while few still have to seek help of others. With the help of our system we aim to shed some lights to this problem.

1.2 Motivation

Upon analyzing the data received through the survey conducted by Anne Jarry and her colleagues on 'Blind Adults' Perspectives on Technical Problems When Using Technology' [2], we deduced a conclusion. The age distribution included working-age adults and among these candidates surveyed, most were visually impaired since their birth. These people had been familiar with different types of technologies and operating systems, so it is safe to presume these people were quite experienced on the concept about how a simple mobile phone works. However, it was observed that major challenges were faced due to the accessibility barriers and usability issues and so external help was required. After having learnt the extent of difficulties that the visually impaired people have to go through on a daily basis, it inspired a concept within our team to create a user friendly system, focusing on the visually impaired people. The concept of this project has been initiated as a stand to bring about focus on the development of systems which could be usable by even handicapped people.

1.3 Problem Definition

This problem is to train datasets for Nepalese Currency (Rs. 5, Rs. 10, Rs. 20, Rs. 50, Rs. 100, Rs. 500 and Rs. 1000), develop a model with accuracy greater than 90 %, deploy the model through web application and finally recognize Nepalese Currency efficiently. The project is made with the vision of developing blind-friendly system in recognizing currency in future, aiming to empower and aid them.

1.4 Objectives

The main objectives of our project are listed below:

1. To train seven classes of dataset for Nepalese Currency using Convolutional Neural Network.
2. To recognize Nepalese Currencies through web application.

1.5 Scope and Applications

This project aims to recognize Nepalese Currencies efficiently and accurately. The system would be fed with a data to be recognized and an accurate denomination is to be expected.

According to Nepalese Blind Survey conducted in 2011, about 0.84% Nepalese are blind. While some might say, only minor people may benefit from this project, we shall not forget the inclusiveness that is direly needed if we are aiming for a sustainable development of country. With the advancement of technology and realization of Nepalese parents to send their children to school irrespective of gender, caste, race and any conventional pseudo-walls minorities have put up with, we have seen immense involvement of children in education field. Blind people are no exception. It is no longer surprising that blind people have excelled in technologies, mastered in various fields and given a new light to their identity. According to an informal interview conducted with one of the blind sisters, who is exceptionally well-versed, has completed her under-graduation and currently pursuing Master’s degree, she spilled her trouble of inability to recognize money, and how often she felt helpless. So, with the vision of helping the entire blind community in future, we aim to develop Nepalese Currency Recognition system.

1.6 Report Organization

The following report first discusses on the historical background of the technological development regarding recognition of currencies worldwide, and elaborates on the milestones achieved on Currency Recognition System. The required tools in order to complete our work has been discussed within Chapter 3. In Chapter 4, it has been further elaborated about the actual methodology we have proposed after the analysis of different approaches already applied so far. In Chapter 5, we have included the actual system architecture that we have used for the project. On the course of completion of our project, the detours that had to be conducted within the architecture has been explained in Chapter 6 along with its result analysis. The limitations and further future enhancements that shall be done have been explained in Chapter 7. Analyzing the results of different models, we have deduced our final approach that we have chosen on the Conclusion Chapter.

2. LITERATURE REVIEW

2.1 History of Paper Note Recognition System

In 2003,'Euro Recognition System' was proposed by M. Aoba et al. The main features of the method are: [3]

1. Three Layered perception
2. Radial Basis Function for accuracy check

In 2003, to remove the nonlinear dependencies and to extract important geographies of data, A. Ahmadi et al. assumed a method. In this approach, organizing map model has been used; the area has been divided into pieces with IKA implemented in each. The noticeable features of the method are [4]:

1. Applies the Learning Vector Quantization network (LVQ)
2. Simple Linear Mode
3. Accuracy Rate = 100%

"Celeric" system got proposed by D. Gunnaratna et al. with a shift operation in 2008. Noise patterns were removed without disturbing the coin paper's unique pictures. The prominent features of this method are [5]:

1. Neural Networks trained with color, brightness, noise, dust, etc.
2. Many layers of back propagation used.
3. Canny Algorithm for edge recognition

The side of the precious paper recognition method was proposed by B. V. Chetan et al. in 2012, with two-stages:

1. Match all database notes
2. Use correlation of the input edges.

Accuracy of 65% was achieved when applied to "Gabor Muweijeh". Resolution of 51% was achieved by the method of subtraction whereas 52.5% accuracy was achieved by the Local Binary Pattern (LBP) method. The proposed technique output 99.5 % of accuracy for a particular dataset [6].

A technique proposed by Manzoor and Ali in 2013 used image processing, which was an economical cost-efficient method to identify the Pakistani currency, and the efficiency was almost a hundred percent [7].

In 2012, a system was proposed by F. Lamont, et al. Different lighting change was the main emphasis of image classification. The main objective of this approach was the identification of Mexican Currencies. Local binary model was used to extract color and texture and characterized accordingly [8].

Nayak and Danti in 2014 based the recognition on prominent characteristics such as the denomination and print date; the research was done on Indian currency. The efficiency was also based on geometrical shape [9].

Oriented FAST and rotated BRIEF (ORB) algorithm was suggested by Ahmed & Taha, Mohamed & Selim, Mazen in 2018 [10], in which the FAST detector and the visual descriptor BRIEF (Binary Robust Independent Elementary Features) was used successfully becoming a more optimal alternative compared to Local Scale-Invariant Features (SIFT). This approach was tested on six kinds of Egyptian currencies. Important features were extracted from the background after image preprocessing, using ORB, followed by Hamming Distance used for matching binary descriptors. Accuracy of 96 % was achieved and runtime of 0.682s was observed, which was shorter compared to CRSFVI system.

In the article of Kshitiz Rimal, an AI enthusiast, technologist and Intel Student Ambassador, ‘Cash Recognition for the Visually Impaired Using Deep Learning’ was proposed [11]. A VGG16 model that is pre-trained in a huge dataset was taken and it’s learned weights were used to re-train on a small dataset. By using a fine-tuning, only the last layer of the model was needed to be retrained to achieve the right degree of accuracy.

Our ‘Nepalese Currency Recognition System’ has used Convolutional Neural Network because they are very good at picking up on patterns (lines, eyes, faces, colors, etc.) in the input images. CNN can operate directly on a raw image without any preprocessing. This can make training computationally heavy and might not be feasible at all times since tuning many parameters can be a complex task, which is made more convenient by the use of CNNs. Also, the images will be pre-processed and reshaped, alongside with augmentation of them to improve the quality and performance of neural networks.

3. REQUIREMENT ANALYSIS

3.1 Software Requirement

3.1.1 Python

Python is a high level, general purpose, interpreted, dynamic programming language in which multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles are supported.

3.1.2 Numpy

Numpy is a high performance numeric programming extension for python. It is the core library for scientific computing in python.

3.1.3 Tensorflow

TensorFlow is an open source software library for numerical computation using dataflow graphs, developed for the purpose of conducting machine learning and deep neural networks research.

3.1.4 Keras

Keras is a high-level neural networks API developed to make implementation of deep learning models optimal. Keras is Python based and capable of running on top of TensorFlow.

3.1.5 Flask

Flask is a micro web framework written in Python that speeds up application development by providing essential back end components for programmers to build upon. Flask is simple and lightweight, one of the most manageable frameworks around and contains only the vital necessities for web development.

3.2 Hardware Requirement

Since the project is based on using software applications, the only hardware required is a PC with at least 4 GB RAM and dedicated GPU.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 System Design

The main concept behind Currency Recognition System is to render a medium to recognize the paper notes issued by Nepal Rastriya Bank and display the value of the note. Many approaches exist to achieve this goal based on which a model has been built to identify paper notes. Success in paper notes recognition depends on extracting and modeling the large variety of datasets. The system developed through this project can be viewed with four cases:

1. Dataset Collection
2. Image Processing
3. CNN Training
4. Recognition

In the first case, the data of sample images have been collected by capturing images from cellphone and converted into trainable dataset. This involves series of image processing steps after collecting the samples and then extracting the features from it. Thus, extracted features of each image are properly labelled and used as dataset to train and validate the network.

After this, the user provides the images and the system perform processing, and display output from the provided image. Image processing is used while preparing dataset and also during the recognition phase. Every real-world image that are to be recognized are passed through the exact same image processing steps that were applied to the images while preparing the dataset.

After the datasets are prepared, the CNN is built using appropriate algorithms and then trained until we obtain a best hypothesis that can recognize images with highest accuracy, greater than 90%. The model is taught about the various classes and the way they are represented.

The trained classifier, in this case the CNN is used in recognition to get the approximations about paper note currency that are present in the image to be recognized.

4.2 Detailed Explanation

4.2.1 Dataset Collection

Considering the project to be deeply rooted on Image Classification, huge amount of training data incorporating the different varieties of images are expected to be collected. At least 2000 sample images of each currency from Rs. 5, Rs. 10, Rs. 20, Rs. 50, Rs. 100, Rs. 500 and Rs. 1000, i.e. of 7 classes has to be taken, which falls under supervised multiclass classification. Since the model has to train images, CNN is the best option. These data would be collected by capturing images in various orientations and backgrounds.

4.2.2 Image Processing

Image Pre-processing and Normalization

Since, most of the neural network models train assuming a square shaped input image, the datasets would be first resized to certain pixels. In order to ensure faster convergence during training, dataset would be first converted to grayscale, the histogram would be equalized followed by normalization. The mean is subtracted from each pixel during data normalization. Thus obtained result is then divided by the standard deviation. Thus, each input parameter has been maintained in a similar distribution.

Image Augmentation

Since for limited training datasets, the neural network tends to over-fit as the number of epochs increases, image augmentation techniques would be implemented. Image augmentation parameters include zoom, shear, rotation and pre-processing functions. Furthermore, contrast stretching, histogram equalization and adaptive histogram equalization can be used to produce augmented images. During training, these parameters results in output images having these attributes.

4.2.3 One-Hot Encoding

One-Hot Encoding is a method applied to converge data and thus prepare it for further algorithm processing. With one-hot encoding, a new categorical column is used to represent each categorical value. A binary vector is used to represent each categorical column i. e. 1 or 0 such that the indices are marked with one.

4.2.4 Convolution Neural Network (CNN)

Convolutional Neural Network (ConvNet or CNN) is a category of neural networks that has long been renowned as the optimal choice for image recognition and classification, such as identifying faces, object identification in traffics, wall detection for robots, self-driving cars. The basis behind these topics have been realized in this project.

1. Convolution Layer
2. Non-Linear Activation Functions (ReLU & Softmax)
3. Batch Normalization
4. Pooling
5. Cost Function and Optimizer
6. Classification
7. Performance Metrics

Above operations showcase the basic building blocks of any Convolutional Neural Network, the main important step to develop a proper CNN based model, is to first understand how the blocks work.

1. Convolution Layer

The primary purpose of Convolution in case of a ConvNet, is extracting prominent input image features. CNN preserves the spatial relationship between pixels and so the prominent features are learned using small input data squares. A matrix of numbers (kernels) is taken that is passed over the image and the image is then transformed based on the filter array parameters. The dimensions of the tensors are visualized using the following equation.

$$z^l = w^l . A^{l-1} + b^l \quad \ldots 4.1$$

$A^l = g^l . z^l$ …4.2

where,

z^l= output of the neurons located in layer l

w^l= weights of neurons in layer l

g^l= activation function

b^l= bias in layer l

Taking padding and stride into account, the output matrix dimensions can be calculated by the formula below.

$[n, n, n_c] * [f, f, n_c] = \left[\left[\frac{n+2p-f}{s} + 1\right], \left[\frac{n+2p-f}{s} + 1\right], n_f\right]$ …4.3

where,

n = image size

f = filter size

n_c = number of channels in the image

p = padding used

s = stride used

n_f = number of filters.

2. Activation Functions

Activation functions are mathematical equations that determine the output of a neural network model. Since two linear functions' composition is a linear function itself, all the layers behave in the same way regardless of the number of hidden layers attached. A non-linear activation function is required to learn the difference between desired output and generated output. Thus the following activation functions have been decided to be used.

a. Softmax Function

The Softmax function is sometimes called the soft argmax function, or multi-class logistic regression since it is a generalization of logistic regression that can be used for multi-class classification. The softmax function is ideally used in the output layer of the classifier where the probabilities are required to define the input images' class.

$$Softmax\ \sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \ldots 4.4$$

where,

$\vec{z}$ = input vector to the softmax function

z_i = elements of the input vector

e^{z_i}= standard exponential function applied to each element of the input vector

K = number of classes in the multi-class classifier

The Softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the values are converted to values between 0 and 1 by softmax. Thus now they are interpreted as probability. Small or negative inputs are converted to a small probability value.

b. ReLU Function (Rectified Linear)

ReLU is the most commonly used activation function in neural networks, whose mathematical equation is:

$$ReLu\ f(x) = \max(0, x) \qquad \ldots 4.5$$

So if the input is negative, the output of ReLU is 0 and for positive values, it is x.

$$ReLu\ Derivative\ f'(x) = \begin{Bmatrix} 1 \quad for\ x > 0 \\ 0\ for\ x \leq 0 \end{Bmatrix} \qquad \ldots 4.6$$

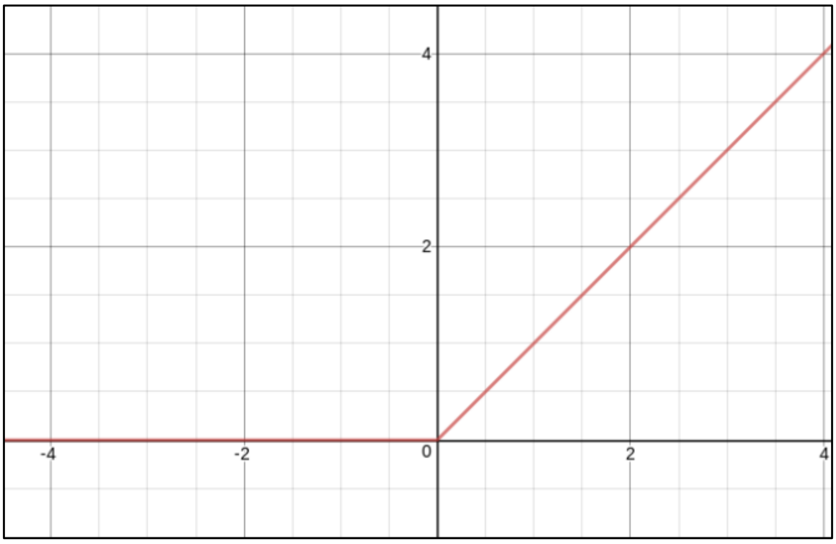


Figure 4-1: ReLU Function

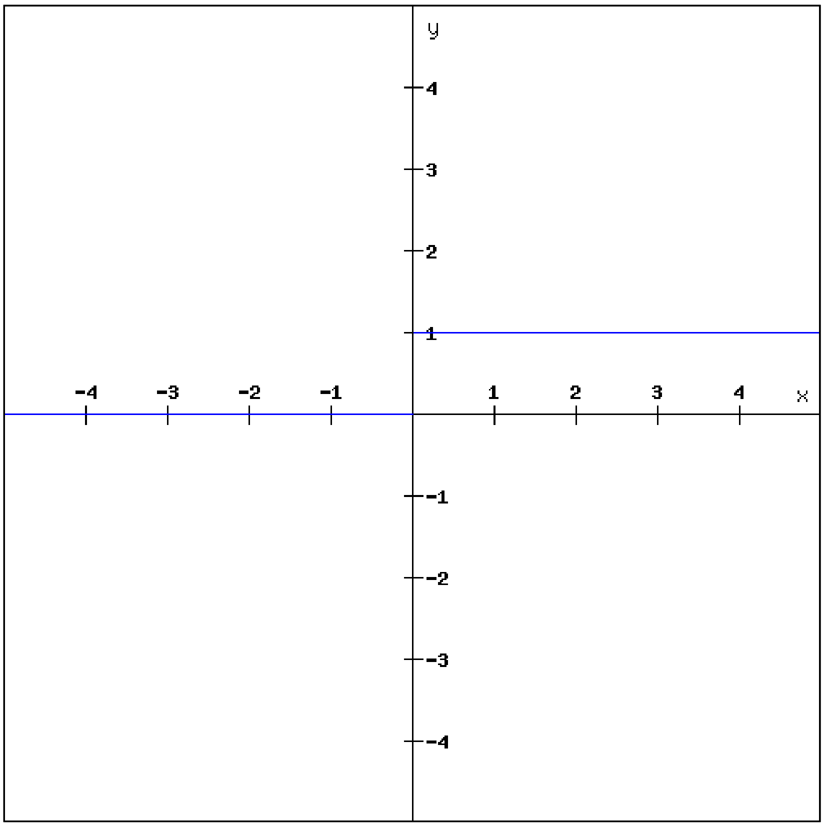


Figure 4-2: Derivative of ReLU

3. Batch Normalization

Batch Normalization is a normalization technique done between the layers of a neural network instead of in the raw data. Mni-batches are normalized rather than the full data set. By doing so, the training is sped up with higher learning rates hence ensuring easier learning. The normalization formula of Batch Normalization can be defined as:

$$z^N = \left(\frac{z-m_z}{s_z}\right) \quad \ldots 4.7$$

where,

m_z = mean of the neurons' output

s_z = standard deviation of the neurons' output.

4. Pooling

The pooling operation involves sliding a two-dimensional filter over each channel of feature map and summarizing the features lying within the region covered by the filter.

a. Max Pooling

Max pooling is a pooling operation in which the maximum element gets chosen from the pool of the feature map generated after running it through a filter. Therefore, a feature map containing the most prominent features is generated from the preceding feature map.

5. Cost Function and Optimizer

a. Categorical Cross-entropy

Categorical cross entropy is a cost function which is aimed for classifications for datasets having more than two classes. The difference between two probability distributions is quantified using these functions. The loss is calculated by the following formula:

$$Loss = -\sum_{i=1}^{\substack{output\\size}} y_i . \log \hat{y}_i \quad \ldots 4.8$$

where,

$\hat{y}_i$ = i^{th} scalar value in the model output

y_i = Corresponding target value

output size = the number of scalar values in the model output.

b. Adam Optimizer

Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient.

The formula used are as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[\frac{\partial L}{\partial w_t}\right] \quad \ldots 4.9$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[\frac{\partial L}{\partial w_t}\right]^2 \quad \ldots 4.10$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \quad \ldots 4.11$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad \ldots 4.12$$

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\propto}{\sqrt{\widehat{v_t}}+\epsilon}\right) \quad \ldots 4.13$$

Where,

w_t= weight at time t

w_{t+1}= weight at time t+1

β = moving average parameter

β_1 , β_2 = decay rates of average of gradients ($\beta_1 = 0.9,\ \beta_2 = 0.999$)

m_t = aggregate of gradients at time t

$\widehat{m}_t$ = bias-corrected aggregate of gradients at time t

v_t = sum of square of past gradients

$\hat{v}_t$ = bias-corrected sum of square of past gradients

$\propto$ = step-size parameter/ learning rate (0.001)

$\in$ = a small +ve constant to avoid 'division by 0' error when $\hat{v}_t$ -> 0, (10^{-8})

6. Performance Metrics

a. Confusion Matrix

To find the correctness and model accuracy, confusion matrix is a suitable approach since it is used in classification problem for classes of two or more types.

Terms associated with Confusion matrix

Following are the name of the cases, when the result of the input data are found as follows:

True Positives (TP)

- Actual Class = 1(True)
- Predicted Class = 1(True)

True Negatives (TN)

- Actual Class = 0(False)
- Predicted Class = 0(False)

False Positives (FP)

- Actual Class = 0(False)
- Predicted Class = 1(True)

False Negatives (FN)

- Actual Class = 1(True)
- Predicted Class = 0(False)

b. Accuracy

Accuracy is the number of correct predictions made by the model over all types of predictions made. Accuracy is a good measure when the target variable classes in the data are nearly balanced.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots 4.14$$

c. Precision

It is the ratio of positive predictions among which are the actual positives.

$$Precision = \frac{TP}{TP+FP} \quad \ldots 4.15$$

d. Recall

It is the ratio of actual positives which are predicted accurately.

$$Recall = \frac{TP}{TP+FN} \qquad \ldots 4.16$$

e. F1-Score

It is weighted harmonic mean between precision and recall.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad \ldots 4.17$$

4.3 Flowcharts

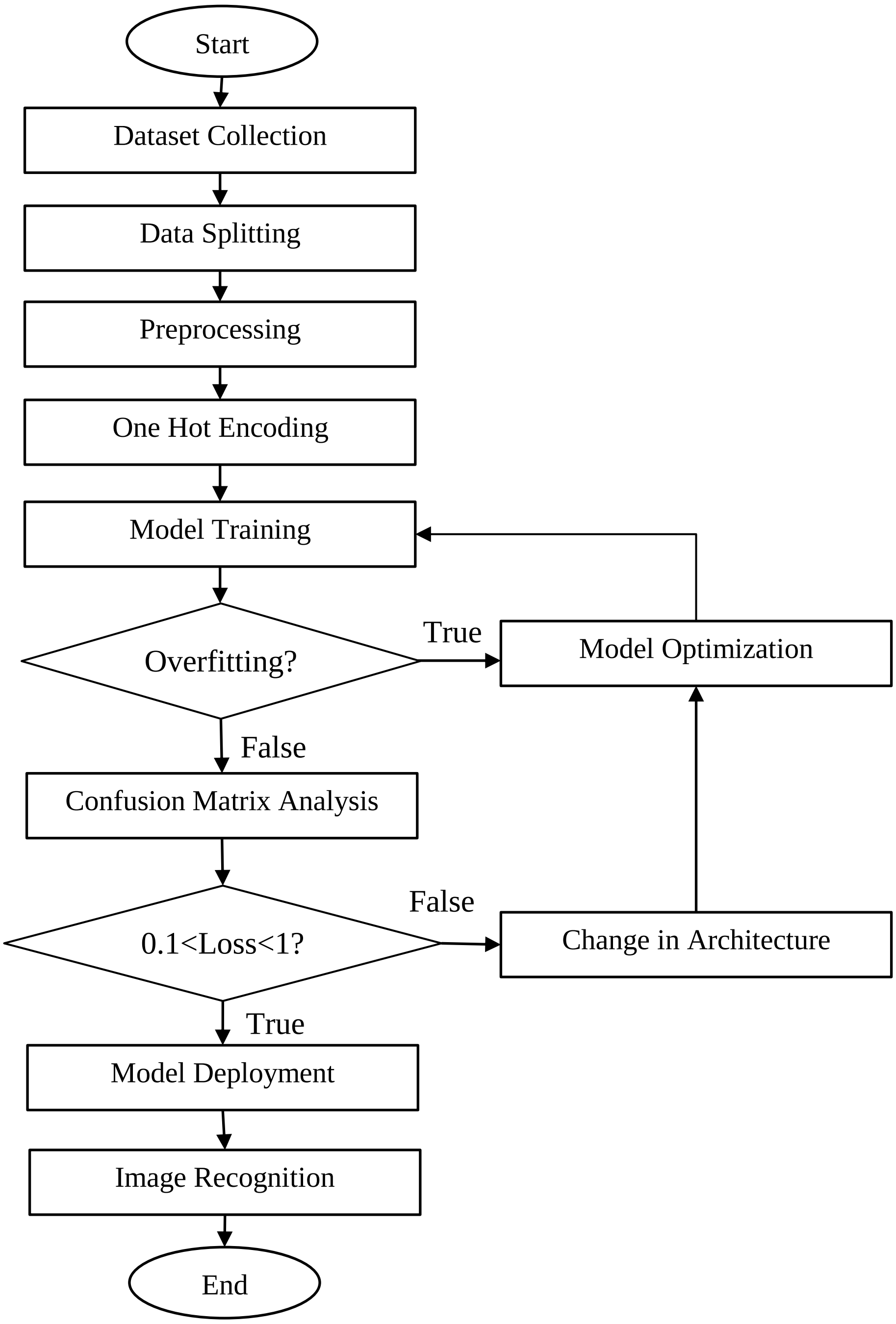


Figure 4-3: Flowchart of the Algorithm for System Development

5. IMPLEMENTATION DETAILS

5.1 Data-Set Preparation

The dataset has been collected from Kaggle website, from the author Gaurav Neupane which was last updated 2 years ago. [13] After inspecting through the dataset, it was found that the quantity for Rupees 5 was inadequate, and so we were firsthand involved in collection of images of five rupee notes. The team members and the classmates captured images of five rupee notes published in different years having different orientations on different backgrounds. And thus, the resultant dataset had equal distribution for each classes.

The distribution of training dataset for our project is shown below:

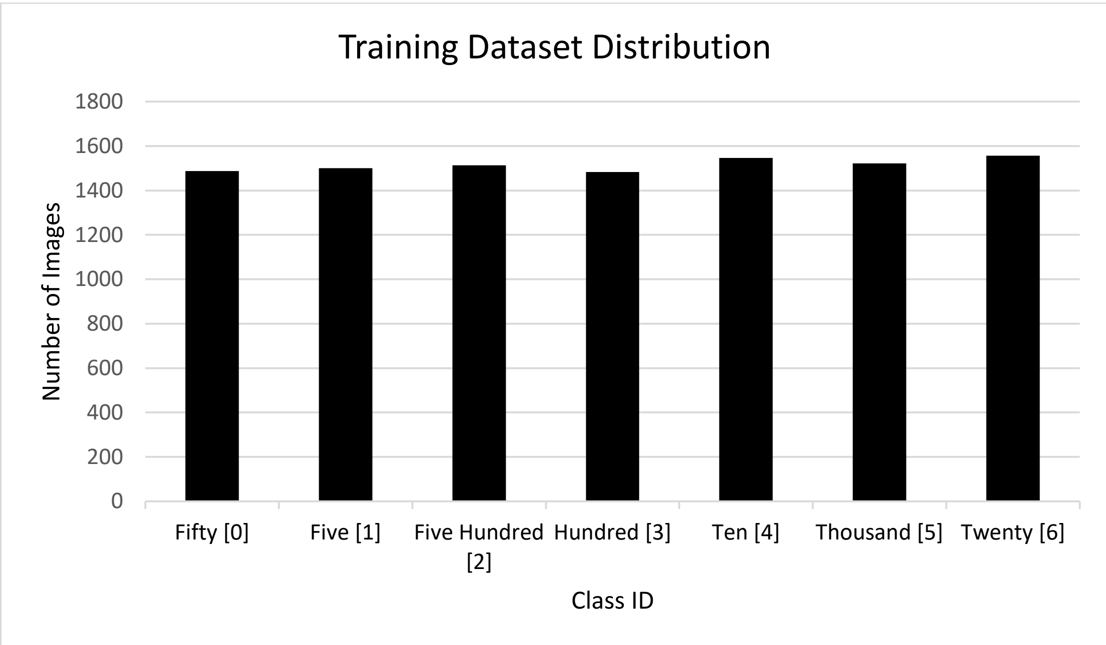


Figure 5-1: Training Dataset Distribution Chart

From our total dataset 17,229 we have used 10,609 data for training and 3537 data for validation and the remaining 3334 dataset was used for testing. The number of training samples of Rupees 50, 5, 500, 100, 1, 1000 and 20 are 1487, 1500, 1513, 1483, 1547, 1522 and 1557 respectively.

5.2 Image Pre-Processing

There has been no preprocessing done so as to extract the color feature. In this model, we have decided to augment the datasets before training the model. The difference of accuracy and loss while implementing models with preprocessing and without preprocessing shall be further discussed on the Result and Analysis chapter.

5.3 One-Hot Encoding

Since the categorical cross entropy produces a one-hot array containing the probable match for each category, the target labels of training, testing and validation datasets were also one-hot encoded.

5.4 Architecture

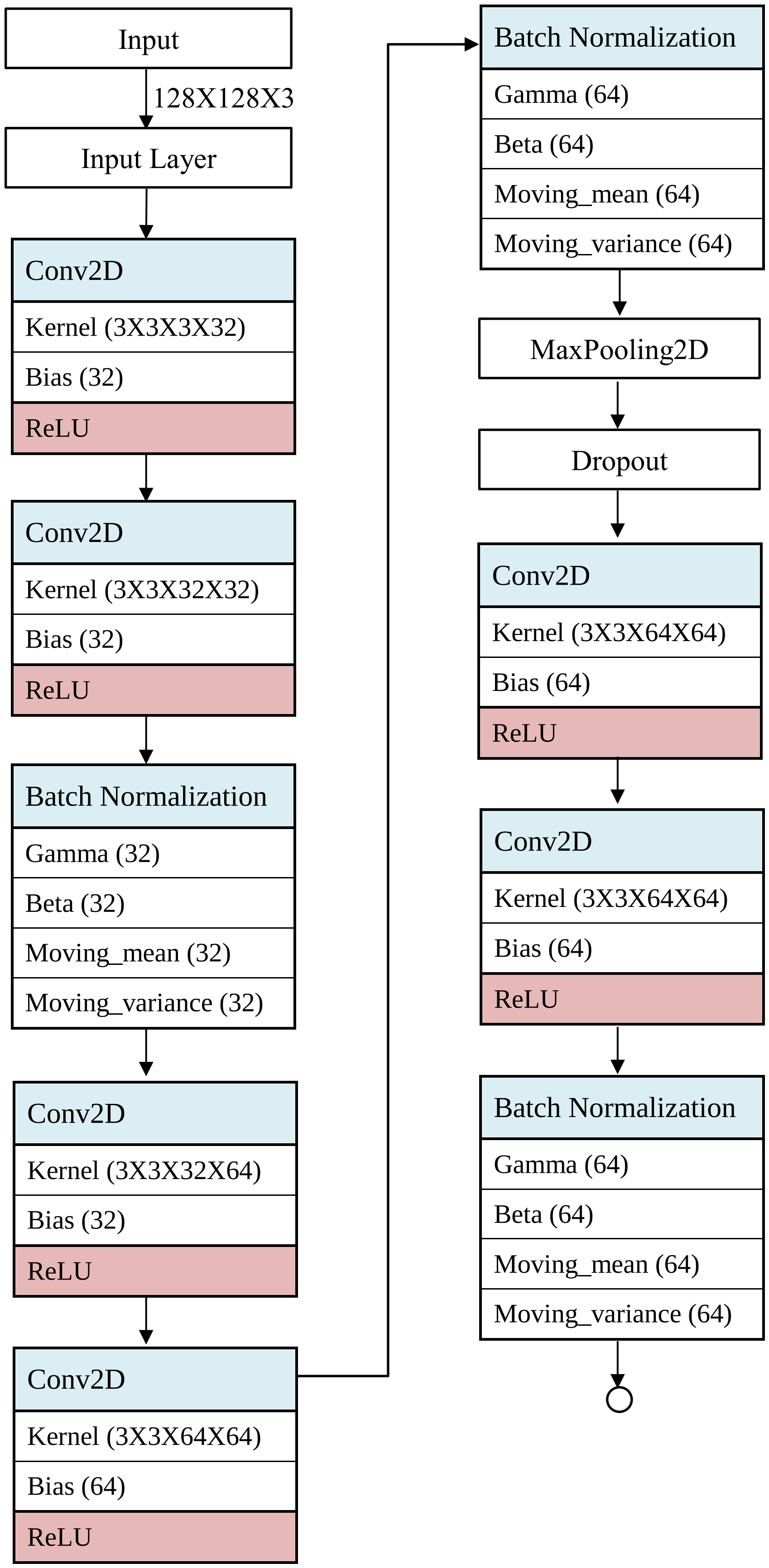


Figure 5-2: Model Architecture (contd.)

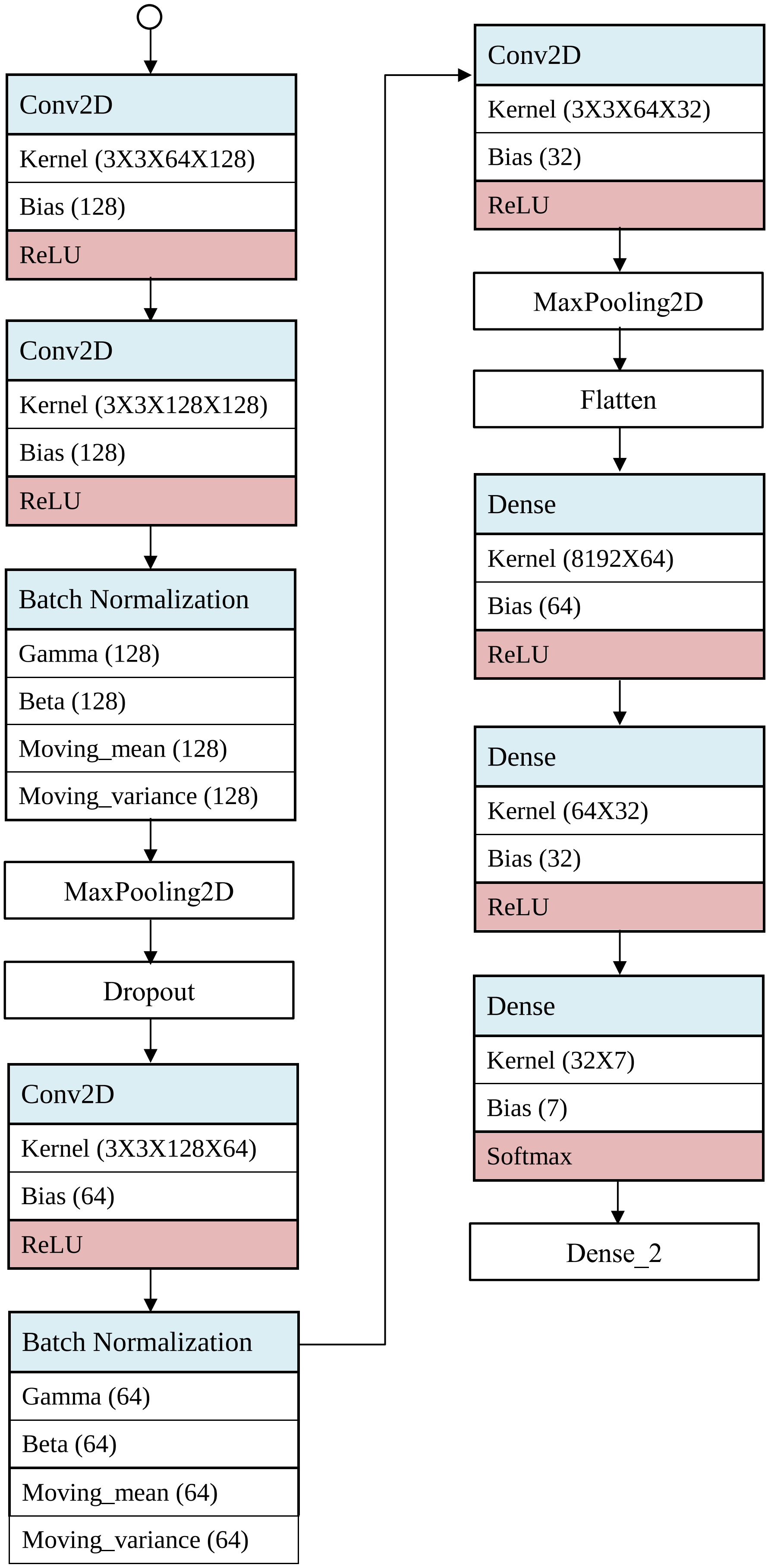


Figure 5-3: Model Architecture

5.4.1 Model Summary

The model is trained on datasets of seven classes, after resizing them in (128, 128) pixels using Convolutional Neural Network. The sequential class is used which implicitly constructs the forward method.

The input image of shape (128, 128, 3) is fed into two convolution layers where the size of filter is (3, 3) with 32 filter nodes. 'ReLU' activation function is used and the padding is assigned 'Same'. Followed by convolution, batches are normalized using Batch Normalization. The output shape after these layers was found to be (128, 128, 32).

Similarly, the features are further extracted by feeding the output images at above layers into two convolution layers where the filter nodes is 64, keeping remaining parameters constant. Batches are again, normalized using Batch Normalization. Size of images are reduced using 'MaxPooling' of pool size (2, 2) and model is regularized using dropout of 0.2. The model is now once again trained on datasets by feeding the extracted layers, maintaining the same filter size, filter nodes, activation function and padding. The output shape after these layers was found to be (64, 64, 64).

The images are then, fed into two convolutional layers where the nodes of filter is now incremented to 128, keeping remaining parameters constant. This process is followed by Batch Normalization, 'MaxPooling' of pool size (2, 2) and Dropout of 0.2. The output shape after these layers was found to be (64, 64,128).

The features are further extracted by decreasing the nodes of filter to 64 and 32, sequentially with the remaining parameters retaining their original assigned values. Finally, 'MaxPooling' of pool size (2, 2) is implemented. The output shape before sending them to dense layer is (16, 16, 32).

The loss of each training, testing and validating layers are computed using categorical cross entropy. The model uses Adam Optimizer to update weights during the back propagation as optimization technique for gradient descent.

We have flattened above generated feature map to one dimensional matrix which is then converted into a fully connected dense layer. The probability of the image classification of the input image is saved in a list and hence the correct output is displayed by calculating the ClassID that corresponds to the index of the maximum probability value.

After executing the above architecture, thus created model had been saved which has then been used henceforth to predict the test dataset. The result thus achieved further shall be discussed on the next chapter.

Image after being resized to 128X128 pixel is as shown below.

Input Image

After Resized to 128x128 pixels

Figure 5-4: Resized Image

5.4.2 Visualization of each layers

```
visualize_conv_layer("conv2d_1")
```

```
(1, 128, 128, 32)
```

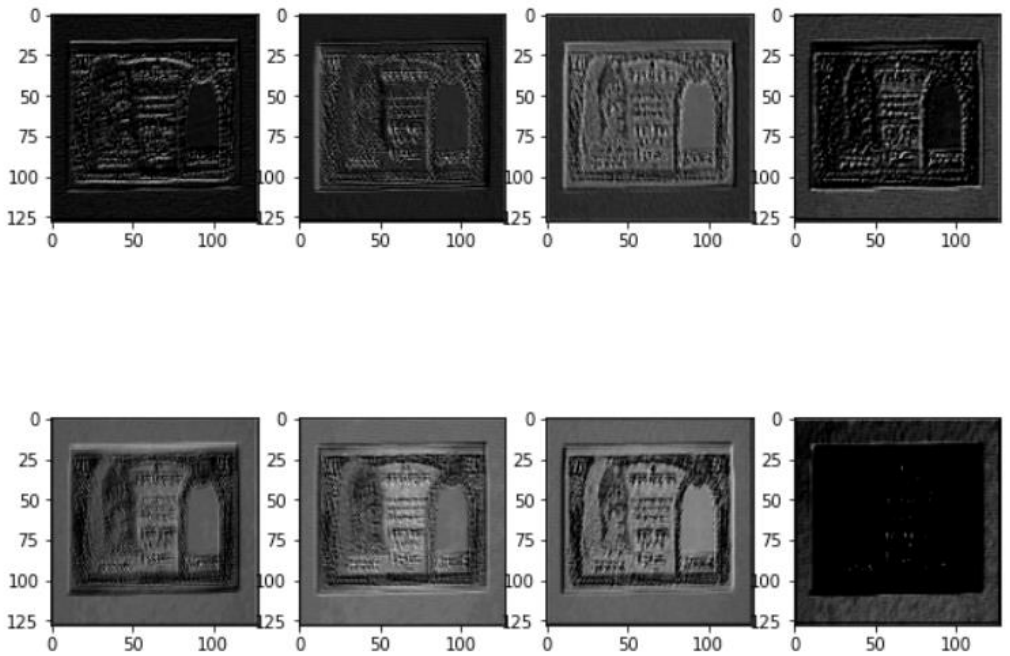


Figure 5-5: Convolutional_Layer_1

```
(1, 128, 128, 32)
```

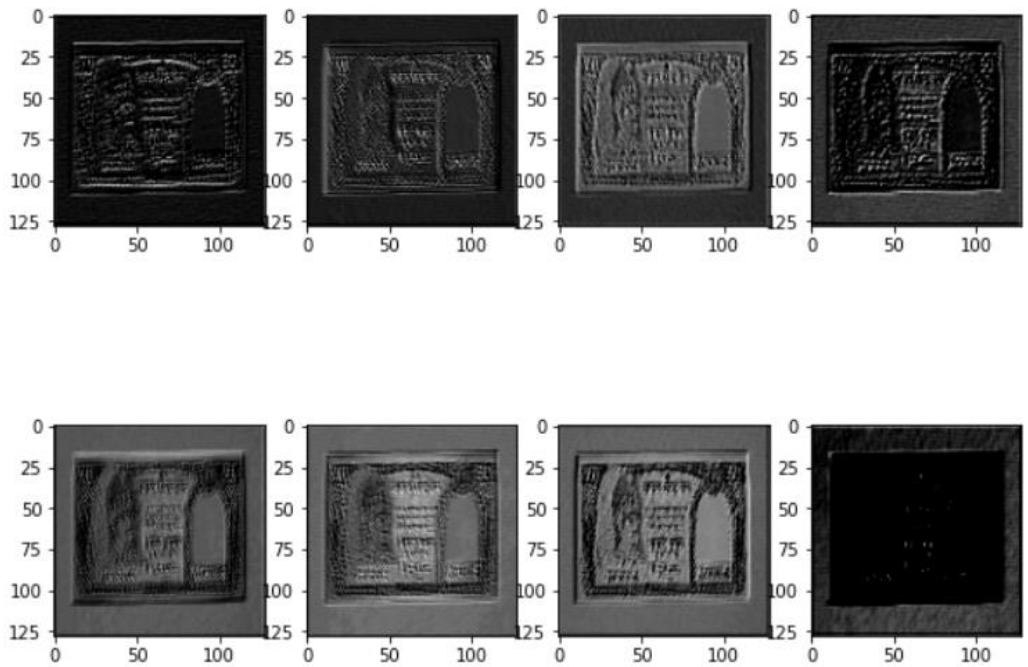


Figure 5-6: Batch_Normalization_0

```
(1, 128, 128, 64)
```

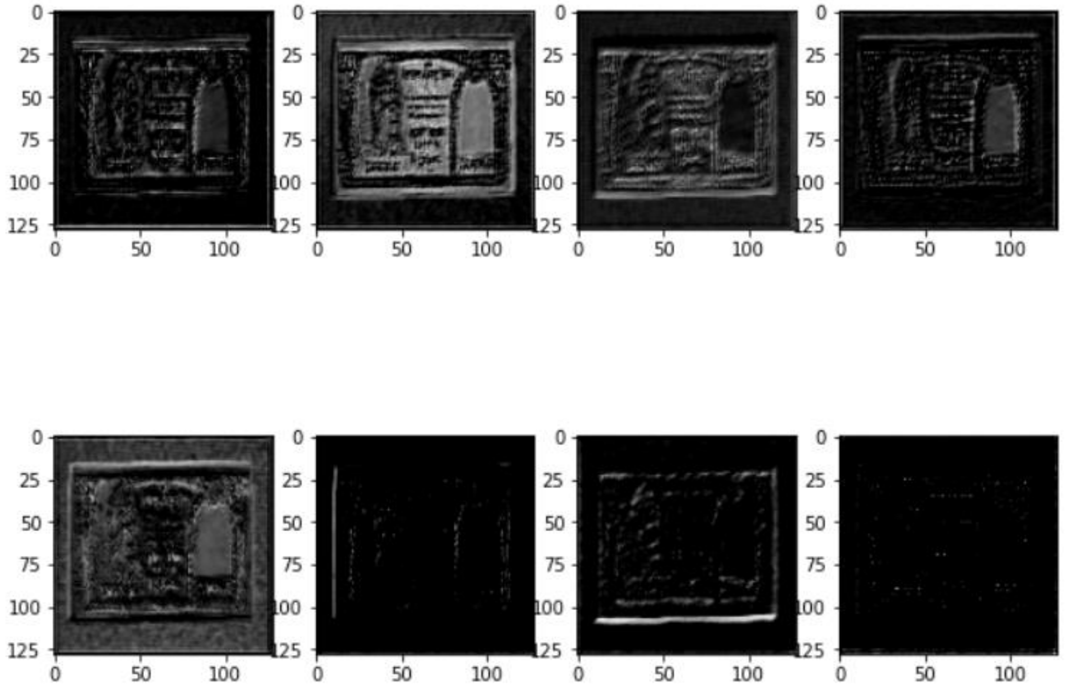


Figure 5-7: Convolution_Layer_3

```
visualize_conv_layer("batch_normalization_1")
```

```
(1, 128, 128, 64)
```

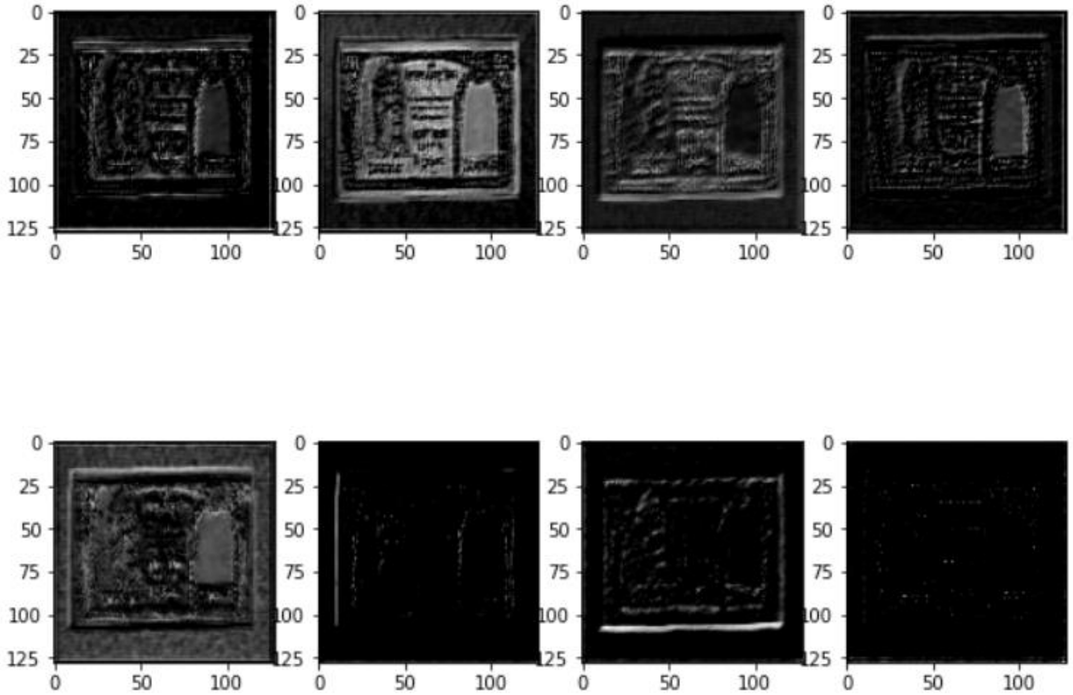


Figure 5-8: Batch_Normalization_1

```
visualize_conv_layer("max_pooling2d")
```

```
(1, 64, 64, 64)
```

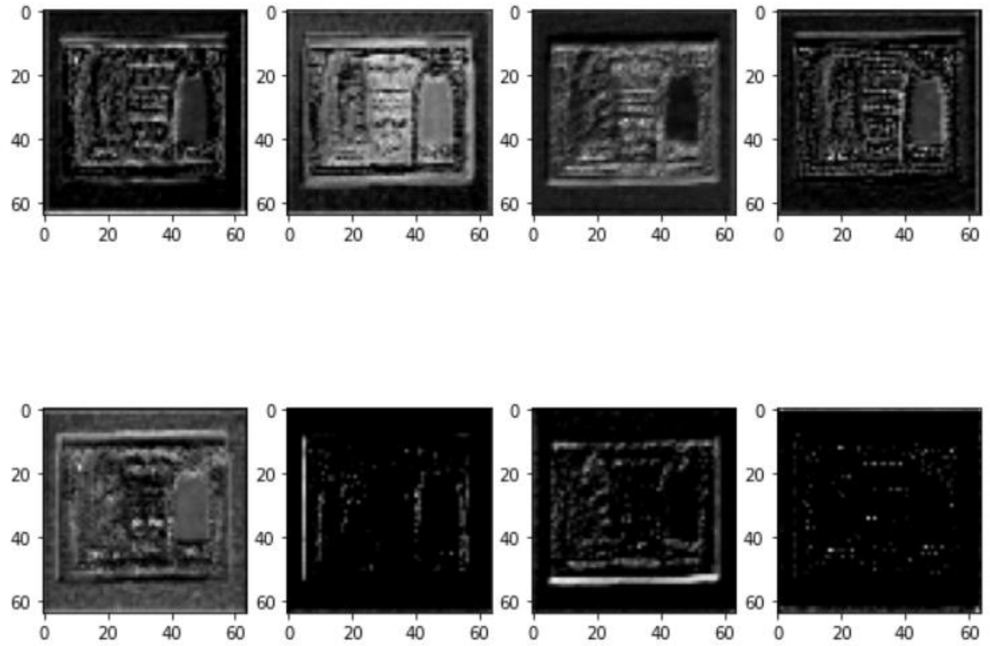


Figure 5-9: Max_Pooling_0

```
visualize_conv_layer("dropout")
```

```
(1, 64, 64, 64)
```

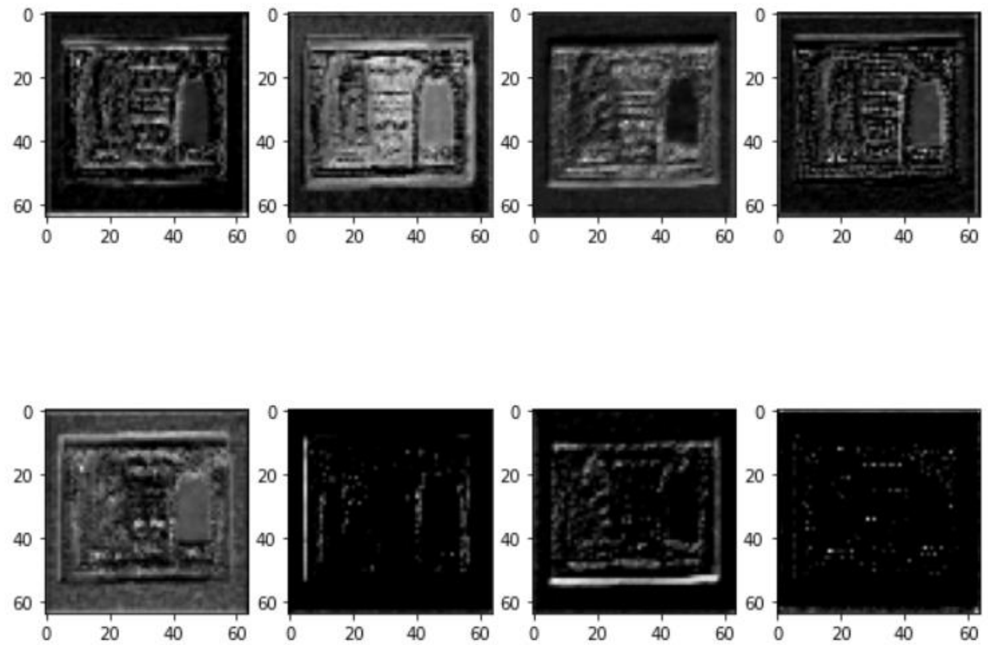


Figure 5-10: Dropout_0

```
visualize_conv_layer("conv2d_5")
```

(1, 64, 64, 64)

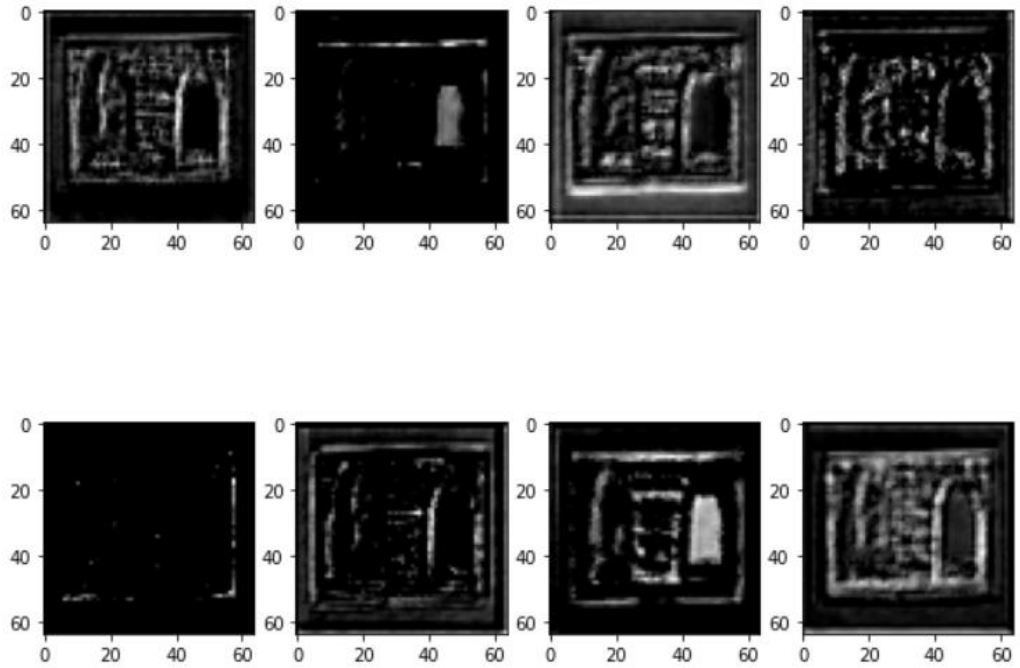


Figure 5-11: Convolution_Layer_5

```
visualize_conv_layer("batch_normalization_2")
```

(1, 64, 64, 64)

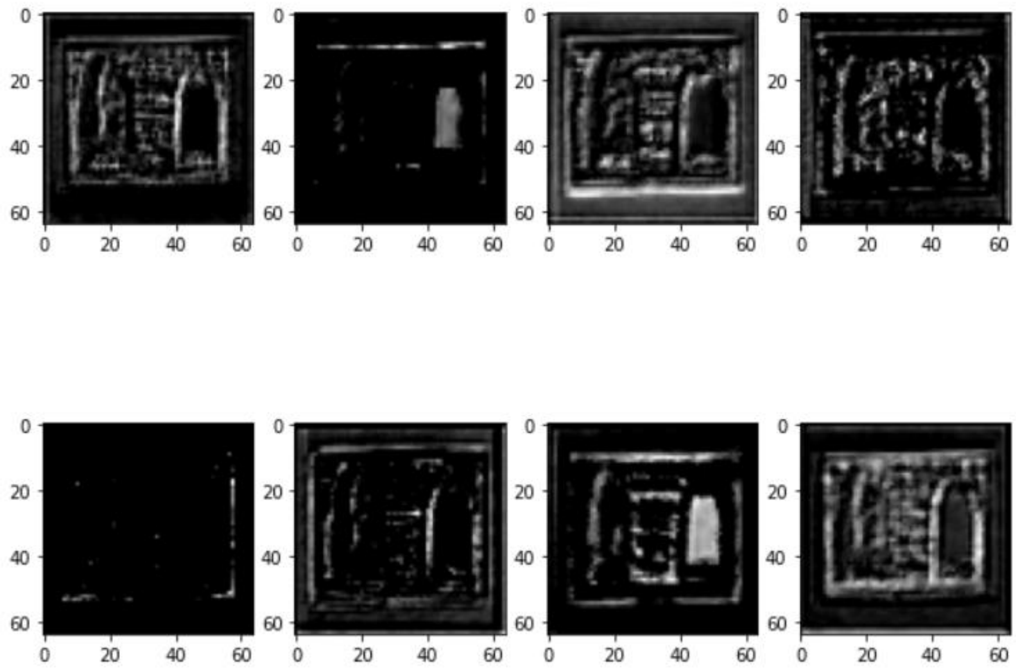


Figure 5-12: Batch_Normalization_2

```
visualize_conv_layer("conv2d_7")
```

```
(1, 64, 64, 128)
```

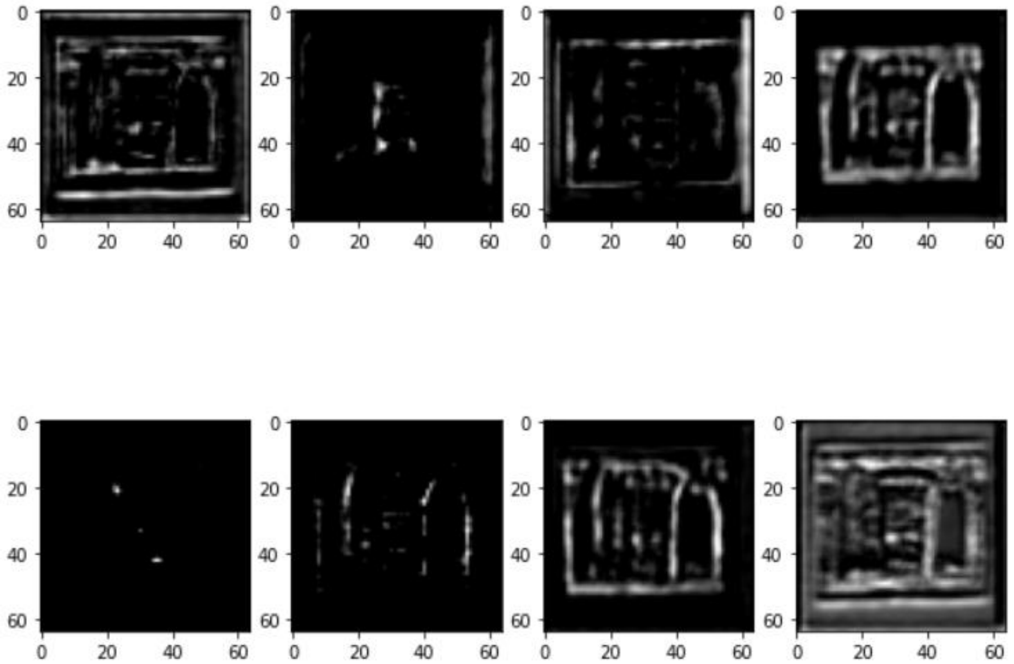


Figure 5-13: Convolution_Layer_7

```
visualize_conv_layer("max_pooling2d_1")
```

```
(1, 32, 32, 128)
```

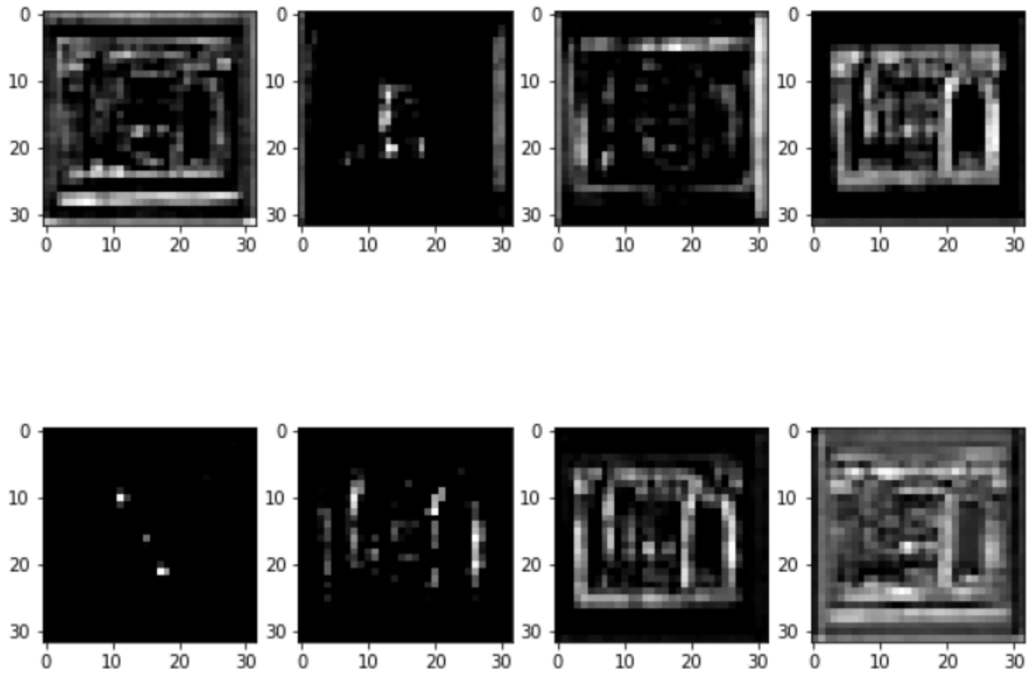


Figure 5-14: Max_Pooling_1

```
visualize_conv_layer("dropout_1")
```

```
(1, 32, 32, 128)
```

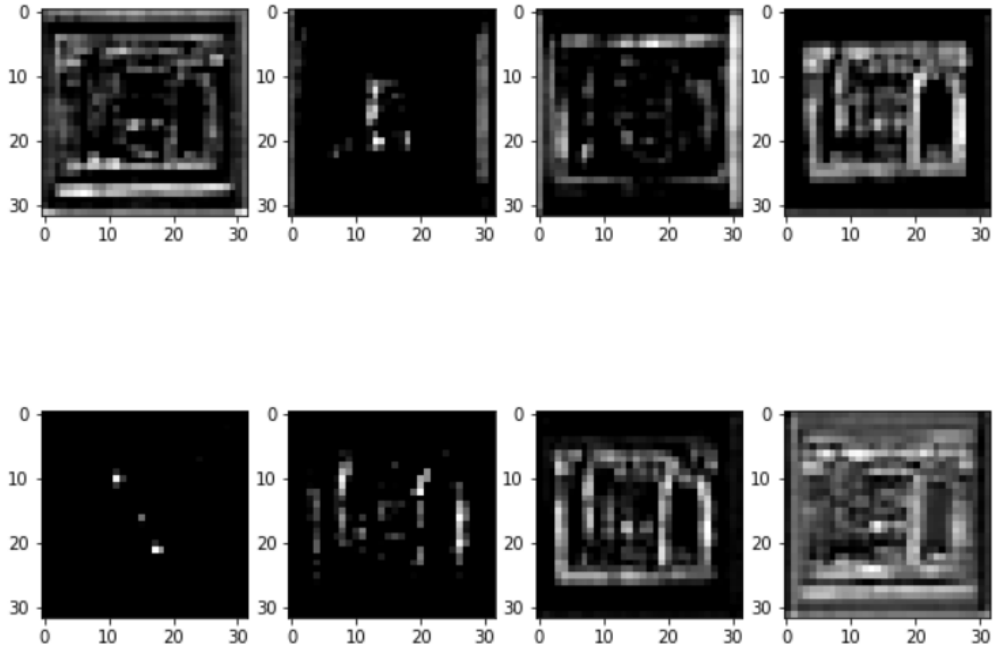


Figure 5-15: Dropout _1

```
visualize_conv_layer("conv2d_8")
```

```
(1, 32, 32, 64)
```

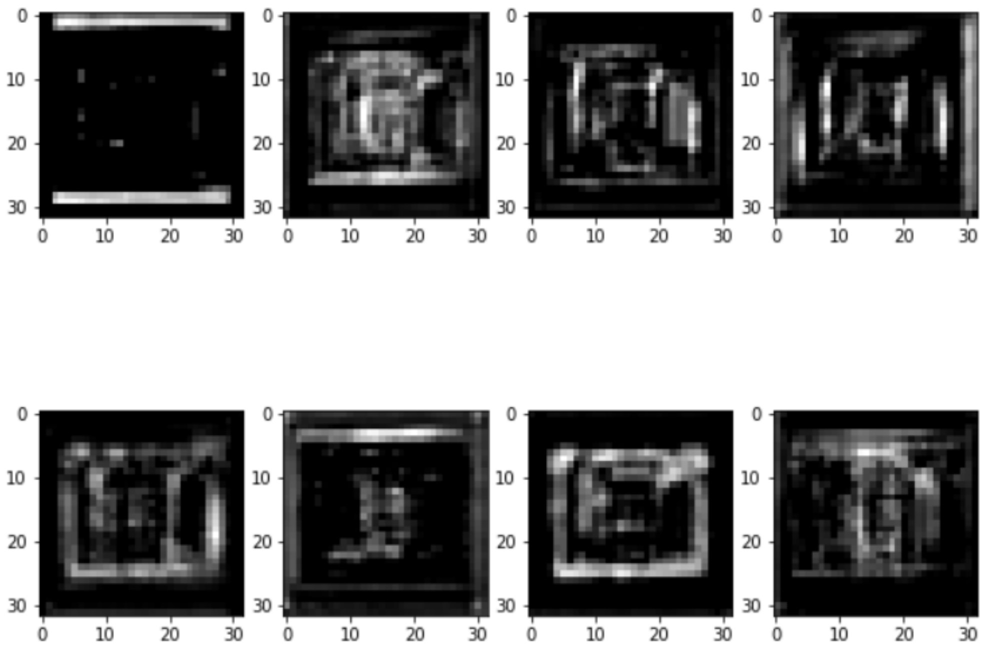


Figure 5-16: Convolution_Layer_8

```
visualize_conv_layer("batch_normalization_3")
```

(1, 64, 64, 128)

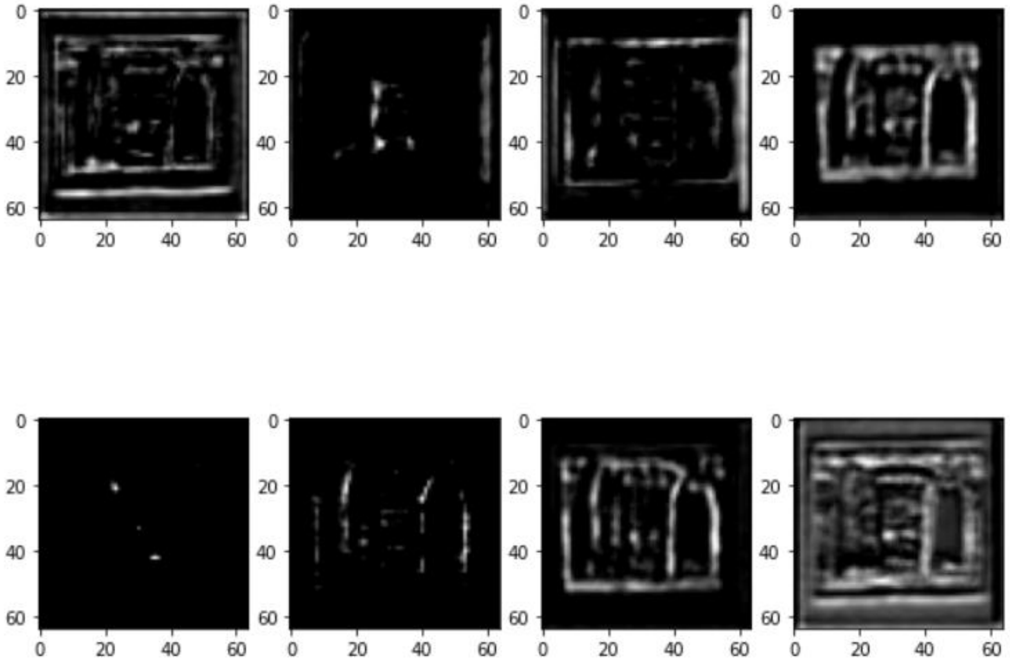


Figure 5-17: Batch_Normalization_3

```
visualize_conv_layer("conv2d_9")
```

(1, 32, 32, 32)

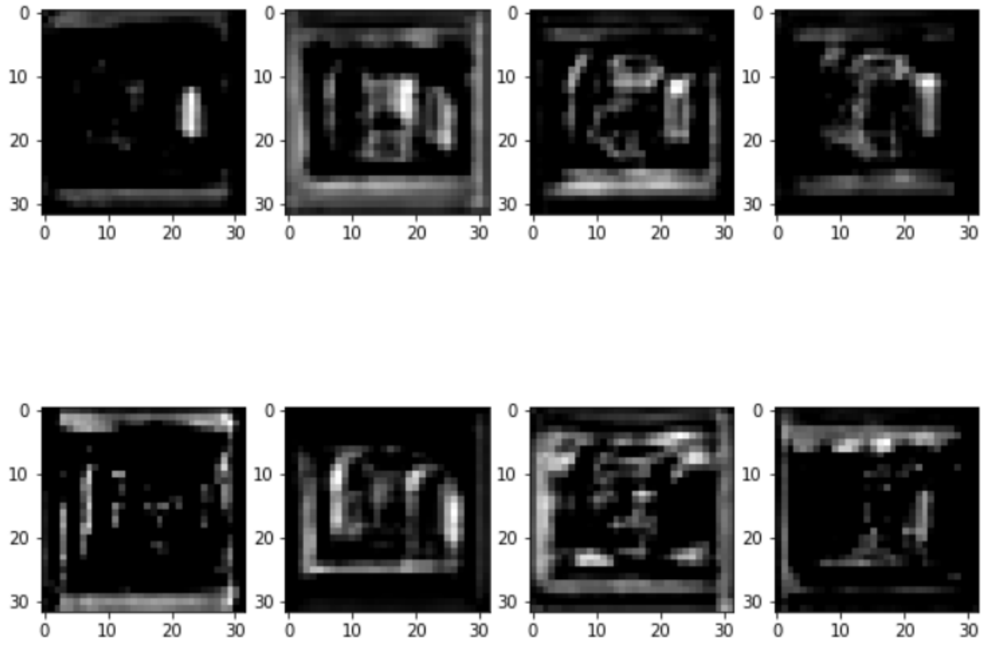


Figure 5-18: Convolution_Layer_9

```
visualize_conv_layer("max_pooling2d_2")
```

```
(1, 16, 16, 32)
```

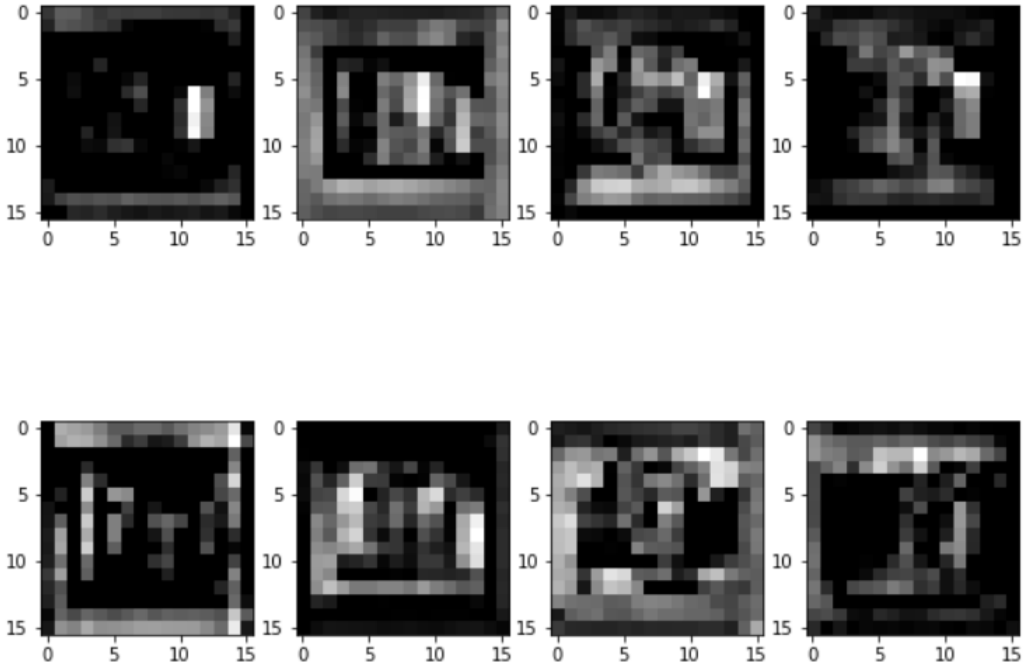


Figure 5-19: Max_Pooling_2

5.5 Model Deployment

In order to implement thus created model, different functions from keras, PIL, flask, tensorflow have been used to build a minimal localhost website which can be seen below.

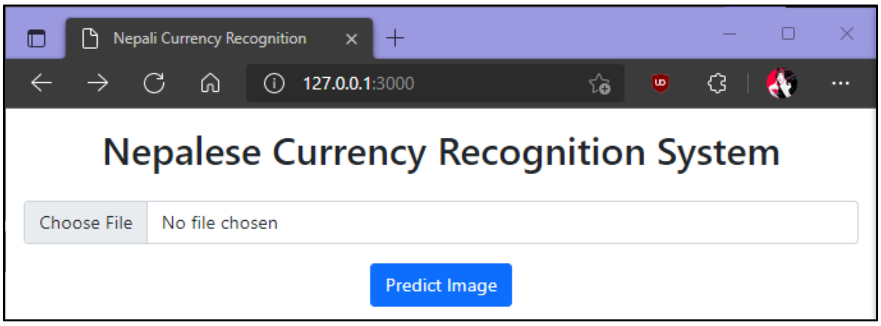


Figure 5-20: Initial Layout

Upon choosing an image for testing, the image is saved as variable 'imagefile' in the index.html, which is accessed using a python script in the backend. This python file uses functions from flask to save the image file, which is saved in a path and then loaded after executing necessary preprocessing techniques according to the requirement to send it under the model. Thus tested image is then shown on the screen along with an audio playback which is played on loop automatically. This audio declares the denomination of the currency in both Nepali and English language.

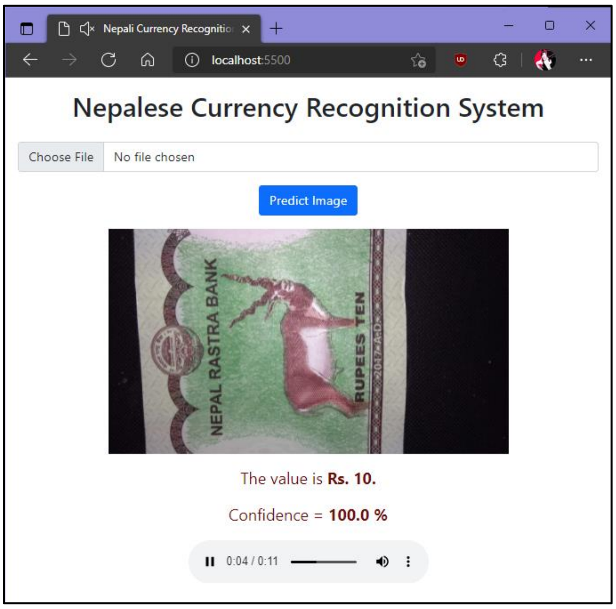


Figure 5-21: Output Layout

6. RESULTS AND ANALYSIS

Upon analyzing different architectures for our project, this chapter shows the results achieved from three models: Model 1, Model 2 and Model 3, using the mentioned architecture.

6.1 Model 1

In this model we have used the same architecture as the model discussed in our project but the only difference is that, the training dataset is first preprocessed in which, the images are resized to 32X32 pixel, converted to grayscale and then the histogram has been equalized followed by normalization.

6.1.1 Accuracy and Loss

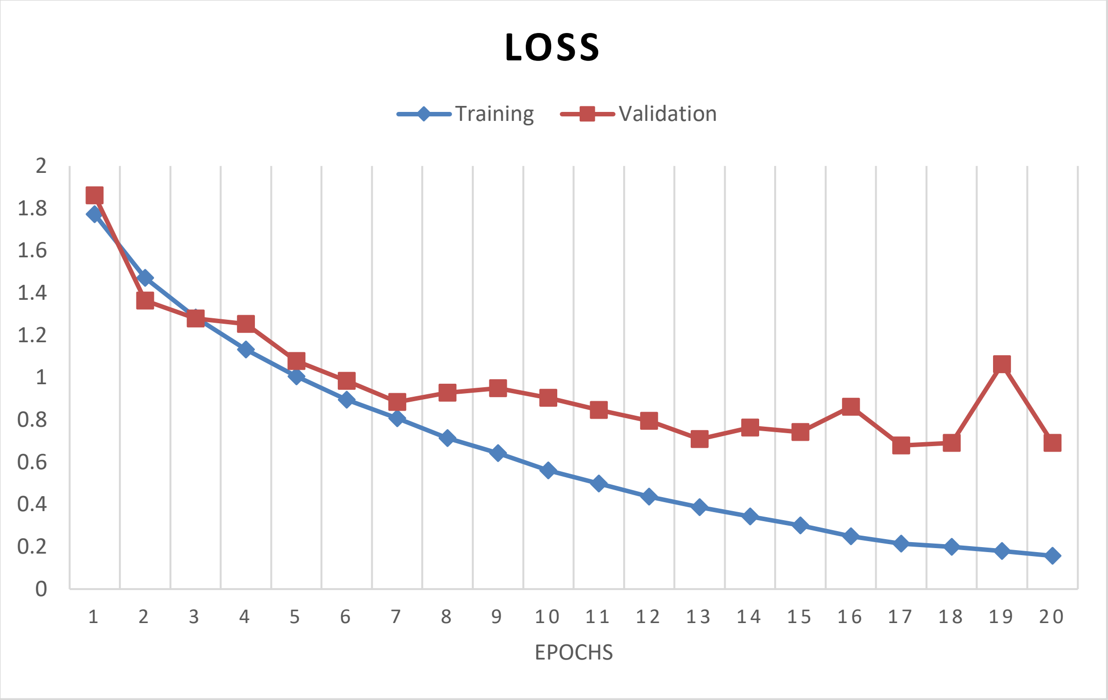


Figure 6-1: Loss Graph for Model 1

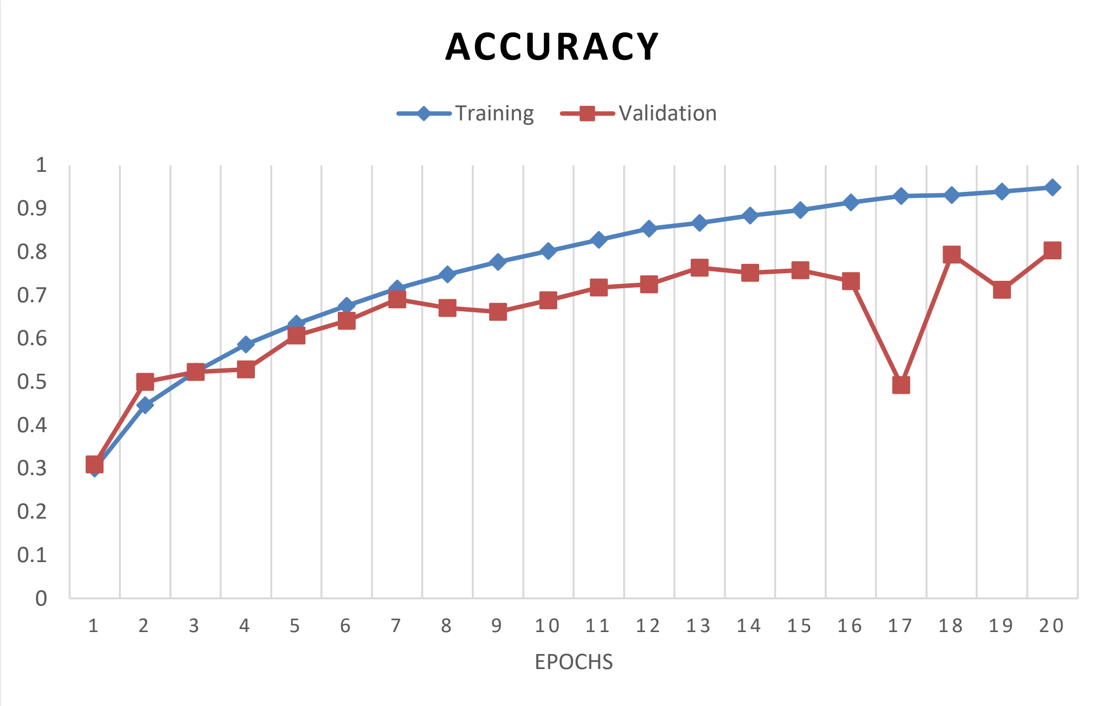


Figure 6-2: Accuracy Graph for Model 1

The model was over-fitted after the 7th epoch after which the validation graph (accuracy and loss) started diverging from the training graph. The overall test accuracy and score is as given below:

1. Accuracy = 46.01%
2. Loss = 2.2964

6.1.2 Confusion Matrix

Table 6-1: Table of Classification Report for Model 1

Class	Precision	Recall	F1-Score	Support
0	0.38	0.34	0.36	515
1	0.65	0.50	0.56	501
2	0.61	0.57	0.59	501
3	0.40	0.50	0.45	521
4	0.49	0.46	0.47	445
5	0.47	0.51	0.49	501
6	0.24	0.29	0.26	350

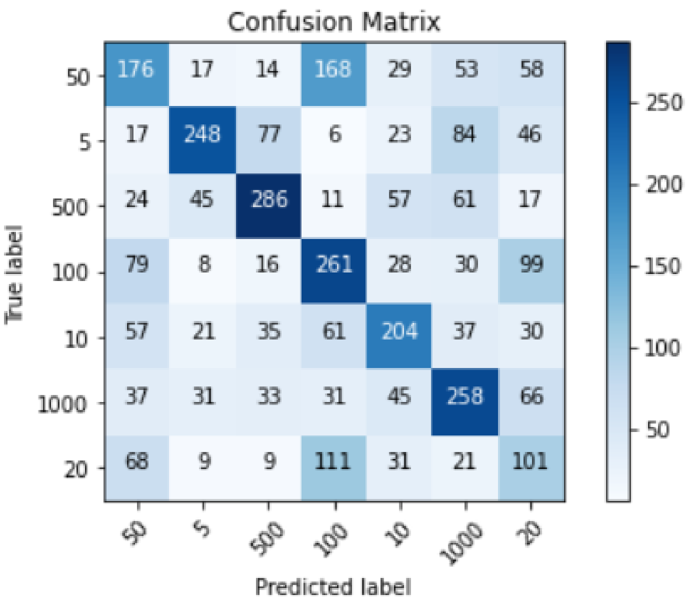


Figure 6-3: Confusion Matrix of Model 1

From the table, the lowest value of F1-Score was found for Class [6], i.e. Rs 20 in which the value of precision and recall was 0.24 and 0.29 respectively. This implies that only 24% of the images which were predicted Rs 20 are actually Rs. 20 whereas 29% of the true total Rs. 20 was correctly predicted.

The highest value of F1-Score was found for Class [2], i.e. Rs 500 in which the value of precision and recall was 0.61 and 0.57 respectively. This implies that 64% of the images which were predicted Rs 500 are actually Rs. 500 whereas 57% of the true total Rs. 500 was correctly predicted.

6.2 Model 2

Since the validation graph did not converge with the training graph, we decided to exclude the preprocessing from Model 1. This model uses the same architecture but it classifies the images using the color feature as well.

6.2.1 Accuracy and Loss

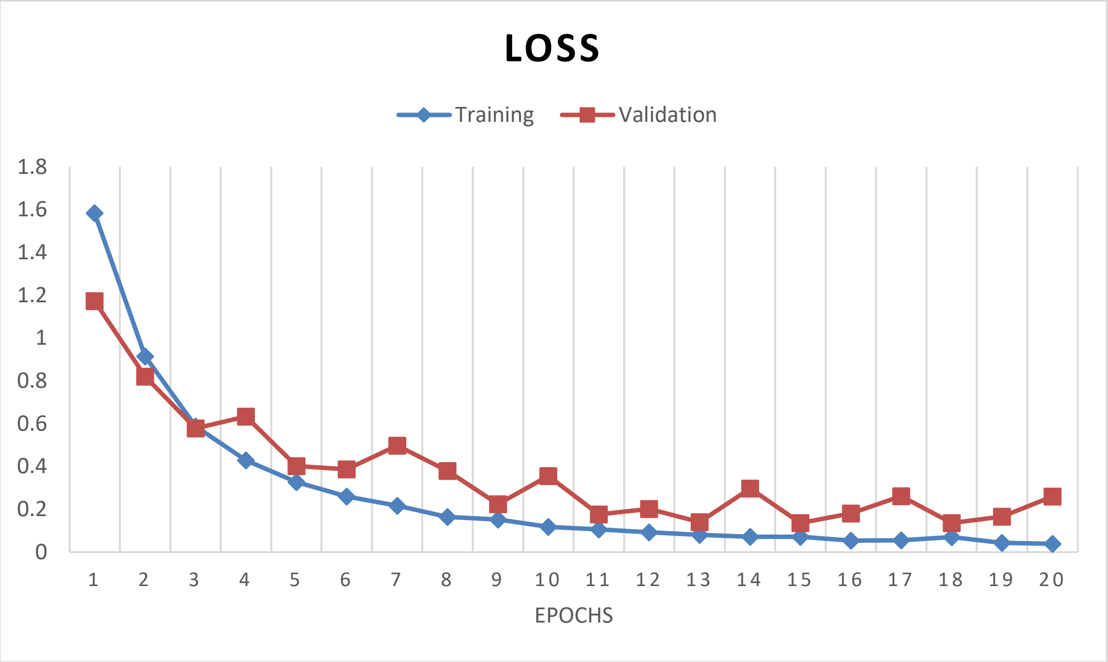


Figure 6-4: Loss Graph for Model 2

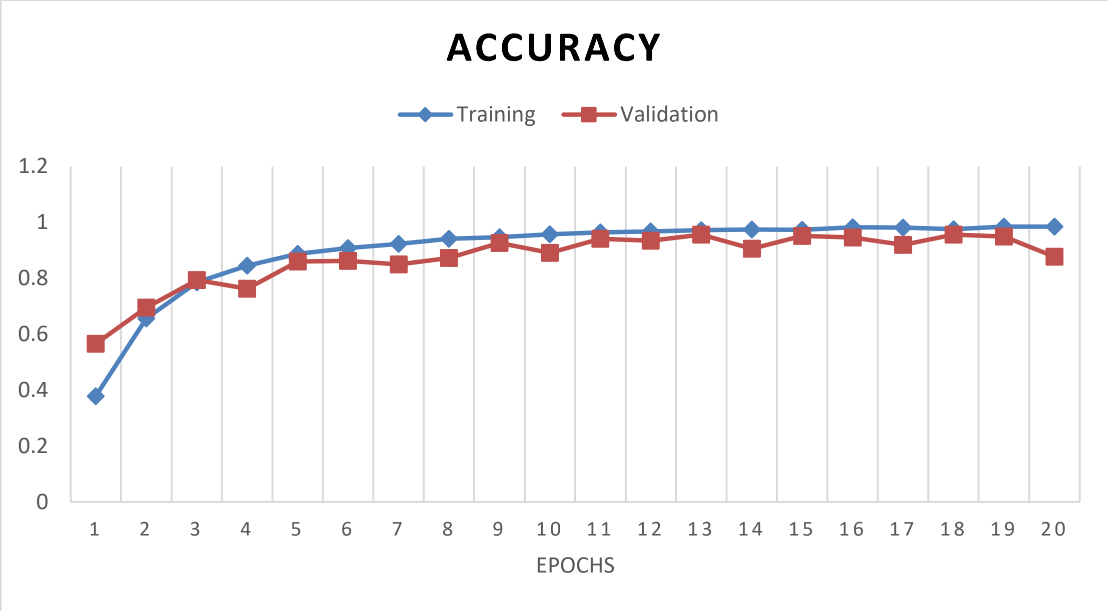


Figure 6-5: Accuracy Graph for Model 2

The model seemed to converge better than the first model but the overall performance score could not be increased more than 68.9% accuracy no matter how much the model was optimized. The overall test accuracy and score is as given below.

1. Accuracy =75.9%
2. Loss =1.5005

6.2.2 Confusion Matrix

Table 6-2: Table of Classification Report for Model 2

Class	Precision	Recall	F1-Score	Support
0	0.97	0.52	0.67	515
1	0.94	0.89	0.91	501
2	0.67	0.94	0.78	501
3	0.63	0.86	0.73	521
4	0.64	0.62	0.63	445
5	0.95	0.57	0.72	501
6	0.71	0.90	0.79	350

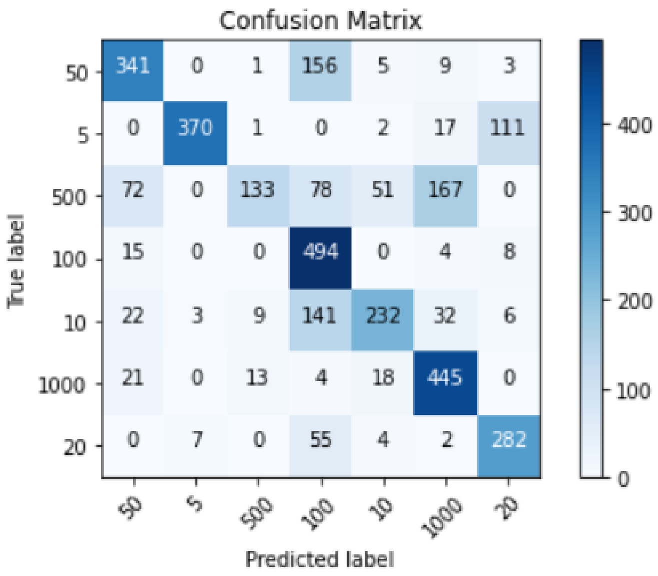


Figure 6-6: Confusion Matrix for Model 2

From the table, the lowest value of F1-Score was found for Class [4], i.e. Rs 10 in which the value of precision and recall was 0.64 and 0.62 respectively. This implies that only 64% of the predicted Rs 10 are actually Rs. 10 whereas 62% of the total Rs. 10 was correctly predicted.

The highest value of F1-Score was found for Class [1], i.e. Rs 5 in which the value of precision and recall was 0.94 and 0.89 respectively. This implies that 94% of the predicted Rs 5 are actually Rs. 5 whereas 89% of the total Rs. 5 was correctly predicted.

6.3 Model 3

Since in Model 1 and Model 2, the input images were resized to 32X32 pixels, the model seemed to lose a lot of information due to distortion which is why the images have been resized to higher dimensions. And thus the final model that has been discussed in our report has been represented as Model 3, where the images are first converted to 128X128 pixel and are not preprocessed. The architecture remains the same.

6.3.1 Accuracy and Loss

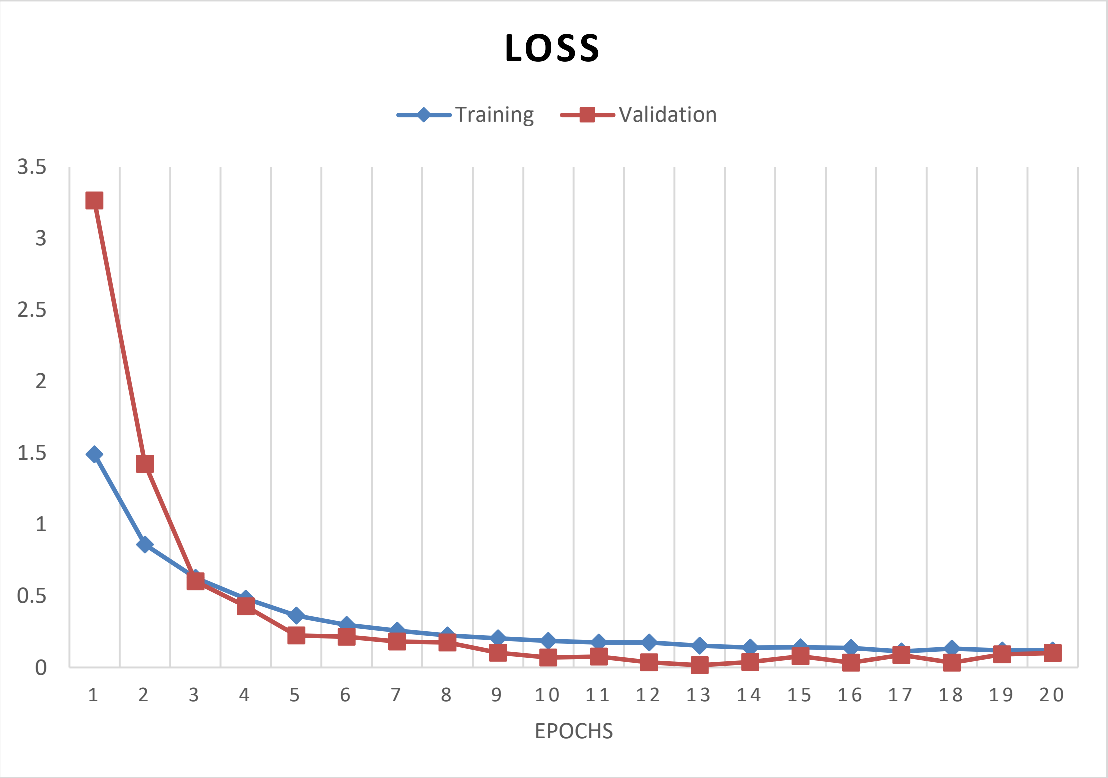


Figure 6-7: Loss Graph for Model 3

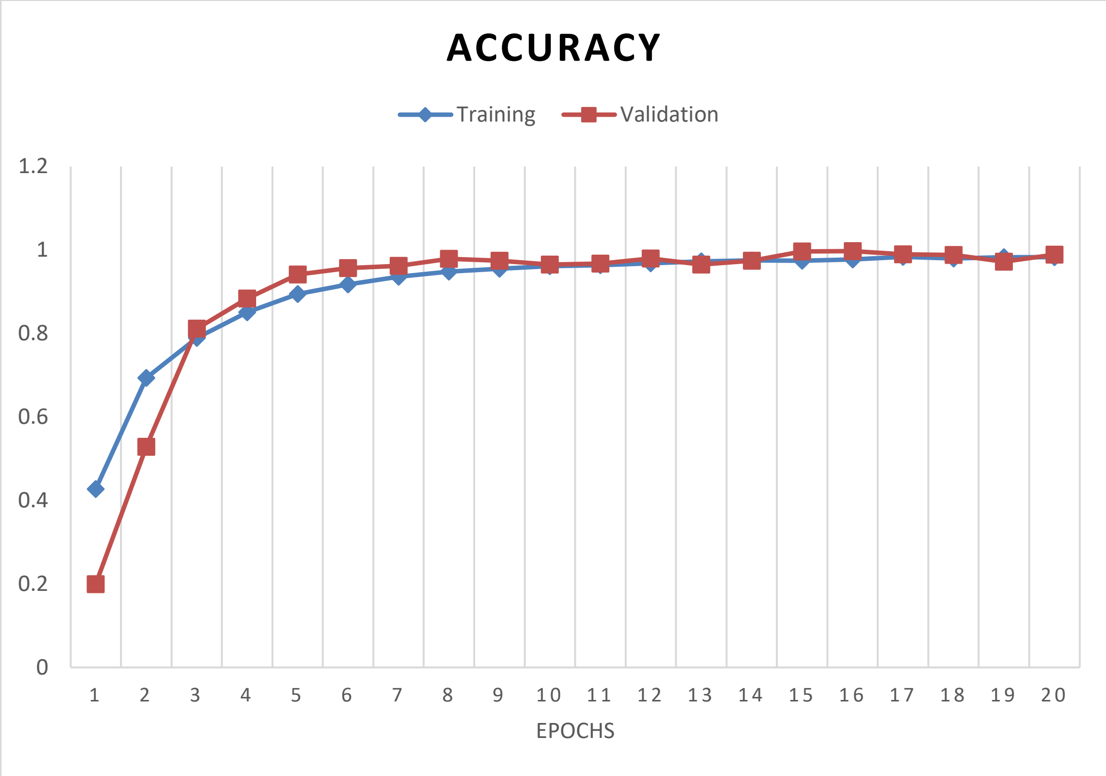


Figure 6-8: Accuracy Graph for Model 3

Since this model exhibited higher accuracy of 93.25%, this model was finalized as our project model. The overall test accuracy and score is as given below.

1. Accuracy = 93.25%
2. Loss = 0.26

6.3.2 Confusion Matrix

Table 6-3: Table of Classification Report of Model 3

Class	Precision	Recall	F1-Score	Support
0	0.98	0.94	0.96	515
1	0.96	0.97	0.96	501
2	0.95	0.97	0.96	501
3	0.89	0.93	0.91	521
4	0.95	0.78	0.86	445
5	0.88	0.98	0.93	501
6	0.93	0.94	0.93	350

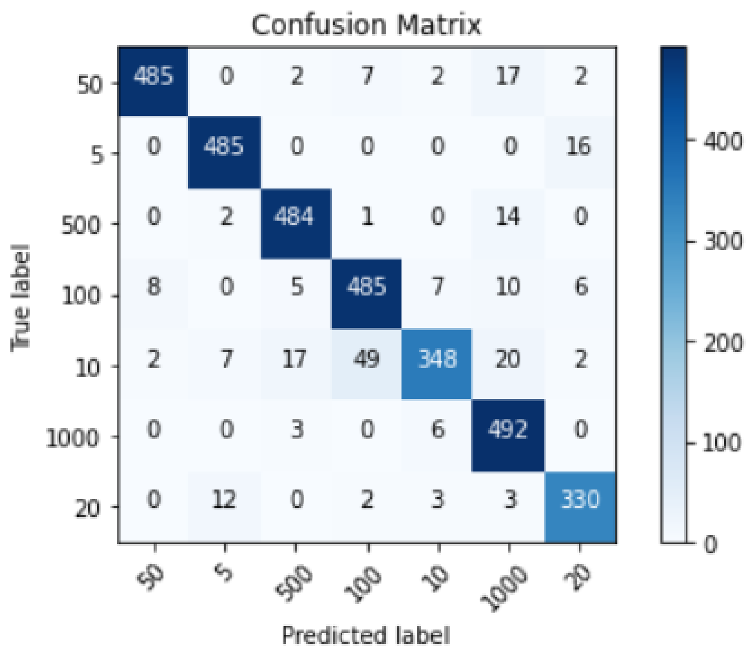


Figure 6-9: Confusion Matrix for Model 3

From the table, the lowest value of F1-Score was found for Class [4], i.e. Rs 10 in which the value of precision and recall was 0.95 and 0.78 respectively. This implies that only 95% of the predicted Rs 10 are actually Rs. 10 whereas 78% of the total Rs. 10 was correctly predicted.

The highest value of F1-Score was found for Class [1], i.e. Rs 5 in which the value of precision and recall was 0.96 and 0.97 respectively. This implies that 96% of the predicted Rs 5 are actually Rs. 5 whereas 97% of the total Rs. 5 was correctly predicted.

6.4 Output of Final Model

For each rupee note, we have chosen the best case and the worst cases. The best case refers to the output when 100% of the note is visible whereas the worst case refers to about (50-70) % of the note being visible on the image.

6.4.1 Output for Rs. 5

Best Case

Worst Case

Figure 6-10: Output for Rupees 5

6.4.2 Output for Rs. 10

Best Case

Worst Case

Figure 6-11: Output for Rupees 10

6.4.3 Output for Rs. 20

Best Case Worst Case

Figure 6-12: Output for Rupees 20

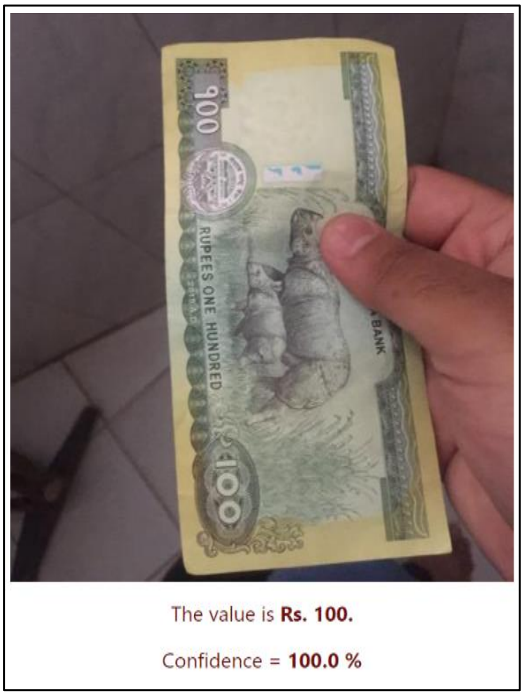
6.4.4 Output for Rs. 50

Best Case Worst Case

Figure 6-13: Output for Rupees 50

6.4.5 Output for Rs. 100

Best Case



Worst Case

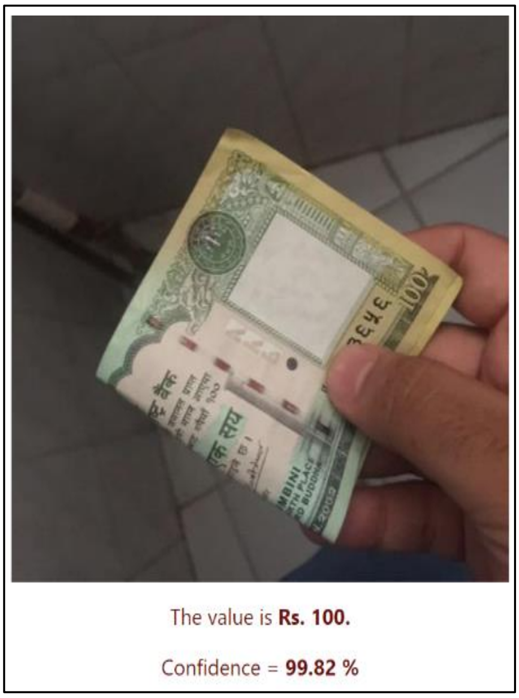


Figure 6-14: Output for Rupees 100

6.4.6 Output for Rs. 500

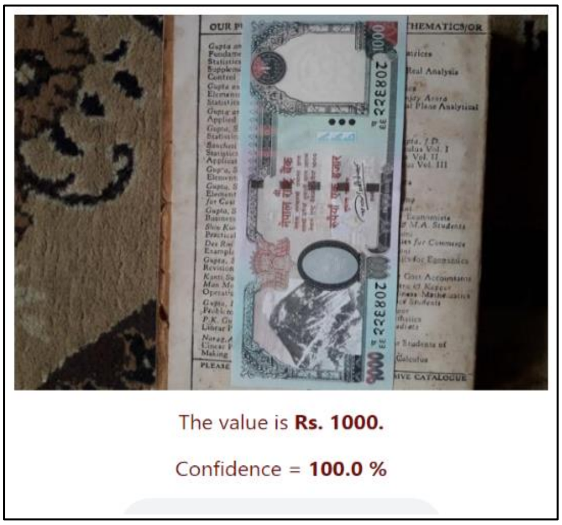
Best Case

Worst Case

Figure 6-15: Output for Rupees 500

6.4.7 Output for Rs. 1000

Best Case



Worst Case

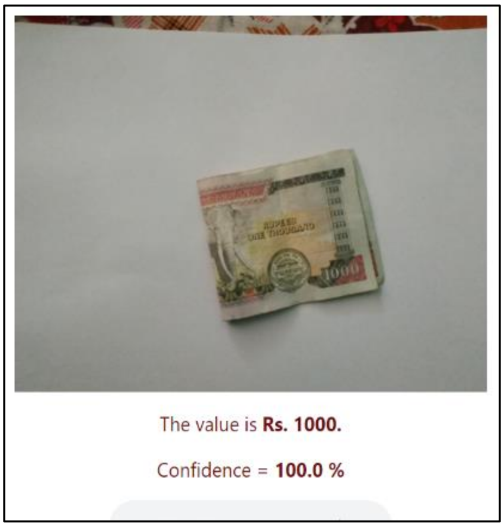


Figure 6-16: Output for Rupees 1000

7. FUTURE ENHANCEMENTS

7.1 Limitations

Some of the particular limitations of our project are:

1. The system cannot differentiate between legit and fake notes.
2. Does not support coins.
3. Negative images are not distinguished as non-currency.
4. Does not detect images in real time.

7.2 Further Works

The main objective of this project was to obtain a model with highest accuracy that can predict the real-world images. But yet the model we have obtained is not perfectly intelligent.

So, the enhancements that can be done in the future are as follows:

1. Currencies shall be detected on real time.
2. This system shall be deployed as a mobile application.
3. Negative images shall be distinguished from actual currency classes.
4. This system architecture can be reused for foreign currencies by updating the corresponding dataset.
5. Images received from user can be treated as datasets for further training.

8. CONCLUSION

Among the three models we have discussed before, the validation graph of Model 3 converged more with the training graph with the highest test accuracy 93.25% and the lowest loss value 0.26, this model has been chosen to be deployed. For the time being, we have deployed the system on a localhost server. The system was implemented in Python programming language and its performance was further tested on fresh images.

Despite the unaccounted limitations that exist in our project work, this project can be remarked as a great achievement and opportunity for us to explore applications of a CNN model. The main objective of the project was to recognize the Paper Currency Issued by Nepal Rastriya Bank and provide its value to the user, which we have rightfully succeeded in doing so.

Hence, this project work has exhibited a successful output for the course ‘Minor Project’ in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Engineering.

9. APPENDICES

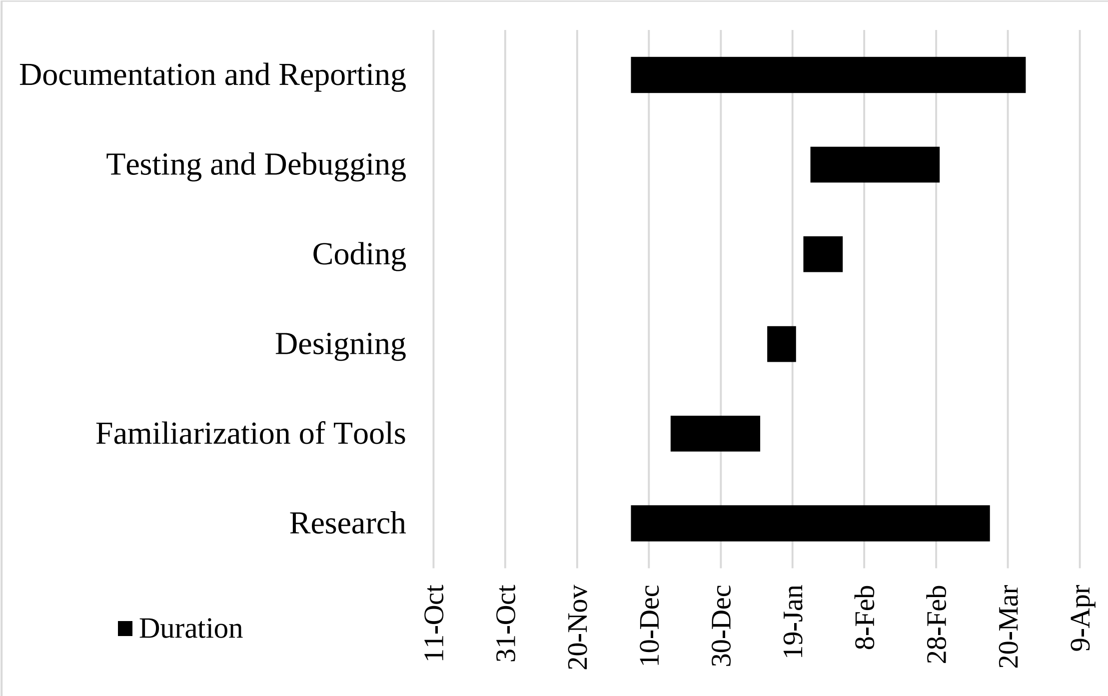
Appendix A: Headers and Commands

In order to train our model, some of the headers and functions that have been used are as follows:

- 'os' library: This library has been used in order to handle all the directory paths for datasets which includes testing dataset and validation dataset. The 'train' dataset has been saved in the list variable named path.
- Np.array(): This function has been then used to create a numpy array of images that are traversed one by one from the path defined above such that 'classNo' stores the Class ID for each currency folder and 'images' stores the corresponding images inside each folder.
- np.load(savepath3, allow_pickle=True): Above made numpy arrays are saved using pickle which makes it easier to load these pickle files later on.
- Train_test_split: In order to classify the dataset folder 'train' into train and validation dataset, we have used train_test_split function from sklearn library and split the data in the ratio of 0.25.
- Matplotlib.plyplot: This function has been imported in order to create a sample graph for the distribution of number of images inside every ClassID folder,i.e. for each denomination of currency, a separate ClassID with certain number of images.
- Cv2.cvtColor(), cv2.equalizeHist(): Since, we cannot train the images as it is in its raw form, we have used a separate 'preProcessing' function under which we have used cv2 library to convert the image into grayscale, to equalize the histogram and to normalize the image.
- X_train = np.array (list (map (preProcessing, X_train))): In order to preprocess each images under train, test and validation dataset, we have used a map function.
- X_train.reshape(X_train.shape[0],X_train.shape[1],X_train.shape[2],1)The shape of each test, train, valid data has been set with channel 1 using reshape() function.
- to_categorical(): Using the method to_categorical() from, a numpy array has been converted into a matrix which has binary values and has columns equal to the number of categories in the data.

Appendix B: Project Timeline

Table 9-1: Gantt Chart



Appendix C: Similarity Index

NEPALESE CURRENCY RECOGNITION SYSTEM

ORIGINALITY REPORT

19%

SIMILARITY INDEX

References

[1] A. S. M Naor, "Visual Cryptography," *Lecture Notes in Computer Science,* vol. 950, pp. 1-12, 1994.

[2] C. C. S. K. a. W. W. Anne Jarry, "Blind Adults’ Perspectives on Technical Problems and Solutions When Using Technology," *Journal of Blindness Innovation and Research,* vol. 7, 2017.

[3] T. K. Y. T. M. Aoba, "Euro Banknote Recognition System Using a Three-Layered Perceptron and RBF Networks," *IPSJ Transactions on Mathematical Modeling and its Applications,* p. 11, 2003.

[4] A. Ahmadi, S. Omatu and T. Kosaka, "A reliable method for recognition of paper currency by approach to local PCA," 2003.

[5] N. D. K. a. H. L. P. D. A. K. S. Gunaratna1, "ANN Based Currency Recognition System," *International Journal of Computer and Information Engineering,* 2008.

[6] D. P. V. Chetan B.V, "A Robust Side Invariant Technique of Indian Paper Currency Recognition," *International Journal of Engineering Research & Technology (IJERT),* vol. 1, no. 3, May 2012.

[7] J. C. A. L. Farid García-Lamont, "Recognition of Mexican banknotes via their color and texture features," *Expert Systems with Applications,* vol. 39, no. 10, pp. 9651-9660, 2012.

[8] M. M. Ahmed Ali, "Recognition System for Pakistani Paper Currency," *Research Journal of Applied Sciences, Engineering and Technology* , pp. 3078-3085, 2013.

[9] K. N. Ajit Danti, "Grid Based Feature Extraction for the," *International Journal of Latest Trends in Engineering and Technology (IJLTET),* vol. 4, no. 1, 2014.

[10] M. T. a. M. M. Ahmed Yousry, "Currency Recognition System for Blind people using," *International Arab Journal of e-Technology, Vol. 5, No. 1,* p. 7, 2018.

[11] K. Rimal, "Cash Recognition," *Cash Recognition for Visually Impaired*, 2018.

[12] G. Neupane, "Nepali Rupees Collection," 2020. [Online]. Available: www.kaggle.com. [Accessed Jan 2022].

[13] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed., O'Reilly Media, Inc., p. 851.

[14] M. Learning, "GeeksForGeeks," [Online]. Available: https://www.geeksforgeeks.org/machine-learning/#int. [Accessed Feb 2022].