COUNTERFEIT CURRENCY DETECTION USING RESOURCE EFFICIENT NEURAL NETWORKS

A PROJECT REPORT

Submitted by

ARUNABHA MITTRA [RA1611003010781]
INDRANIL PAUL [RA1611003011225]

Under the guidance of

Mrs. S. SHARANYA
(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

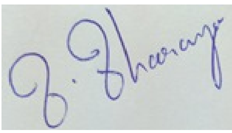
JUNE 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**COUNTERFEIT CURRENCY DETECTION USING RESOURCE EFFICIENT NEURAL NETWORKS**" is the bonafide work of "**ARUNABHA MITTRA [RA1611003010781], INDRANIL PAUL [RA1611003011225], , ,** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.



SIGNATURE

Mrs. S. SHARANYA
GUIDE
Assistant Professor
Dept. of Computer Science & Engineering

SIGNATURE

Dr. B.AMUTHA
HEAD OF THE DEPARTMENT
Dept. of Computer Science And Engineering

Signature of the Internal Examiner

Signature of the External Examiner

Own Work Declaration

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **BTech / Computer Science and Engineering**

Student Name : **Arunabha Mittra ; Indranil Paul**

Registration Number : **RA1611003010781 ; RA1611003011225**

Title of Work : **Counterfeit Currency Detection Using Resource Efficient Neural Network**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

Arunabha Mittra

Arunabha Mittra
RA1611003010781
24 May, 2020

Indranil Paul

Indranil Paul
RA1611003011225
24 May, 2020

ABSTRACT

Recent media reports have clearly listed the major causes of economic instability in India, of which counterfeiting of paper currency notes is one which demands immediate attention. Several cases have been filed and traced by reports to show the danger that this issue poses. The Government with deft hands have managed to take actions against such atrocities. In spite of upgrading to stricter rules and regulations, the counterfeiters manage to find loopholes and continue to con lay mans. The field of counterfeit currency has seen massive improvements coupled with new age digital technology, which helped improve the economic development. However, it is impractical to assess and evaluate all counterfeit notes in a short period of time. Existing system requires a long procedure of filing a case, verifying the documents and a delayed outcome. This method aka Deep Learning has been successful in image classification. The amount of the computation resource used by CPU and storage that the traditional Deep Learning algorithm demands makes it an very big process. The objective is to make an efficient and user friendly way to see bad note with easy implementation at any location convenient to the user.

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. Sandeep Sancheti**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. C. Muthamizhchelvan**, Director, Faculty of Engineering and Technology, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. B. Amutha**, Professor Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor **Dr. A. Jeyasekar**, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his great support at all the stages of project work.

We would like to convey our thanks to our Panel Head, **Dr. Revathi Venkatraman**, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor, **Dr. G.Manju** and **Mrs J.V.Vidhya**, Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Mrs. S. Sharanya**, Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under her mentorship. She provided me the freedom and support to explore the research topics of my interest. Her passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Arunabha Mittra

Indranil Paul

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

India RBI, is only bank which has full authority rights to issue currency notes. Counterfeit money is fake money made illegally . Over the recent few years, as a result of the great technological advances in color printing, duplicating, and scanning adversely affects the fake money printing, which may destabilize indian economy. To prevent and detect the circulation of counterfeit notes, a system must be developed. The project is grounded on detecting counterfeit currency notes using AI algorithms. The objectives of the project are:

- To improve existing currency detection process and make it less expensive.
- The intended beneficiaries of the work done is common man as they won't have to go through the long hectic process.
- Can be used as the back end for web application and android/ios application.
- Neural network and image processing using Tensor flow is used in carrying out the project.
- Data use to train model is authentic.
- Certain features based on the texture and print pattern of the nodes will be studied to predict whether the particular note is counterfeit or not.

1.2 Existing System

Detection of Indian Counterfeit notes using Neural Network using Image Capture Under UV light, RGB to Grey-scale conversion, Edge detection and feature comparison and NN efficiency calculation but cannot classify Rs. 2000 note as it was too complex. Recognition of fake currency notes using Convolutional Neural Networks using Convoluted Neural Network and the performance metrics used where Pre-processing, Image Re-scaling Using CNN but where having high efficiency loss in basic filter.

```
net.Layers

ans =
  47x1 Layer array with layers:

     1   'input'     Image Input     224x224x3 images with 'zerocenter' normalization
     2   'conv1_1'   Convolution     64 3x3x3 convolutions with stride [1  1] and padding [1  1  1  1]
     3   'relu1_1'   ReLU            ReLU
     4   'conv1_2'   Convolution     64 3x3x64 convolutions with stride [1  1] and padding [1  1  1  1]
     5   'relu1_2'   ReLU            ReLU
     6   'pool1'     Max Pooling     2x2 max pooling with stride [2  2] and padding [0  0  0  0]
     7   'conv2_1'   Convolution     128 3x3x64 convolutions with stride [1  1] and padding [1  1  1  1]
     8   'relu2_1'   ReLU            ReLU
     9   'conv2_2'   Convolution     128 3x3x128 convolutions with stride [1  1] and padding [1  1  1  1]
    10   'relu2_2'   ReLU            ReLU
    11   'pool2'     Max Pooling     2x2 max pooling with stride [2  2] and padding [0  0  0  0]
    12   'conv3_1'   Convolution     256 3x3x128 convolutions with stride [1  1] and padding [1  1  1  1]
    13   'relu3_1'   ReLU            ReLU
    14   'conv3_2'   Convolution     256 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    15   'relu3_2'   ReLU            ReLU
    16   'conv3_3'   Convolution     256 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    17   'relu3_3'   ReLU            ReLU
    18   'conv3_4'   Convolution     256 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    19   'relu3_4'   ReLU            ReLU
    20   'pool3'     Max Pooling     2x2 max pooling with stride [2  2] and padding [0  0  0  0]
    21   'conv4_1'   Convolution     512 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    22   'relu4_1'   ReLU            ReLU
```

Figure 1.1: Array with layers

Figure 1.2: 2000 note

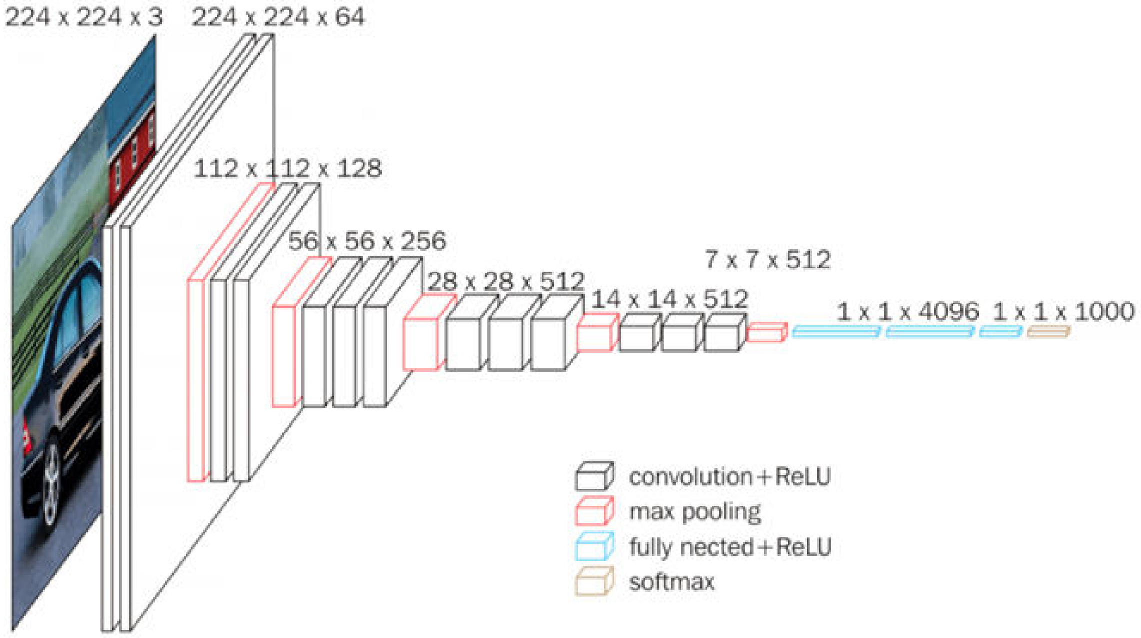


Figure 1.3: VGG 16

1.3 Proposed System

1.3.1 One Shot Classification

One shot classification using Siamese Neural Network we require only one training example for each class. Yes you got that right, just one. Hence the name.

- Using Neural Networks elimination of discoloration mark and training the network.
- Any food mark or discoloration will not be treated as counterfeit and any fold marks will not hinder image detection.
- Using limited feature points we must determine whether the currency note is counter -feit or not.

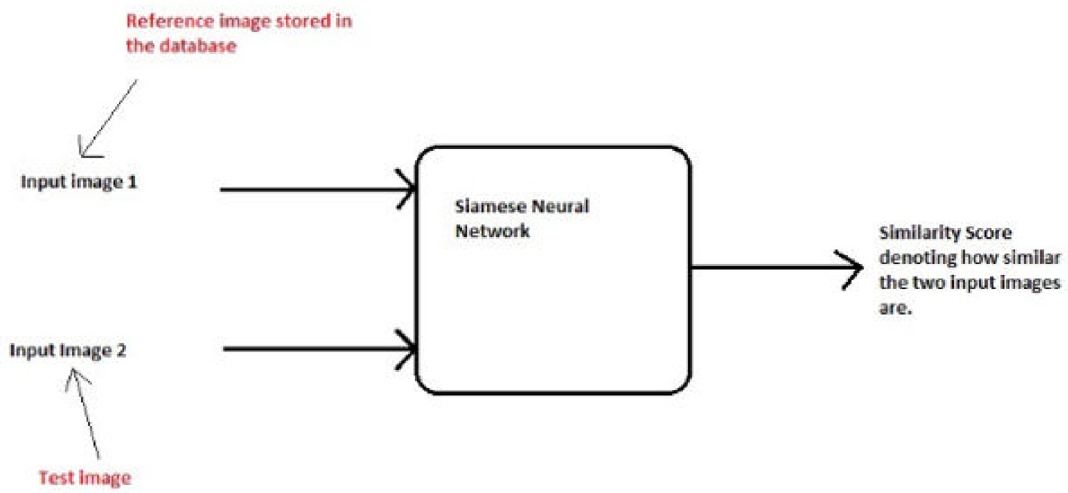


Figure 1.4: One Shot Classification

CHAPTER 2

LITERATURE SURVEY

Various papers were distributed to discover new creative approaches to identify forge notes. Few have utilized ordinary picture recognition methods utilizing MATLAB others have utilized fundamental AI procedures to check for the photos of the picture overall. The rest have attempted to prepare the model for singular notes, and have it settled on forge or bona fide dependent on a solitary component. "Acknowledgment of Forge Note utilizing CNN by Nava K, Sukh Raviteja, Nagapremi B. "A Neural Network based Model for Paper Currency Recognition and Verification" taken a stab at utilizing calculation for directed learning of twofold classifiers and have not been particularly productive. The administrative work has referenced its personal inconveniences for moderate yet their strategy is least expensive . Paper dependent on "Counterfeit Currency Detection utilizing Image Processing" have proposed utilization of S.V.M K.N.N AI methods to prepare a managed what's more, a performance realizing approach to manage make non-parametrical similarly as regressive examination to have done on the instructive file picked. The outcomes have been contrasted and one another and discovered that S.V.M is a superior calculation and it is in reality multiple times the precision of K.N.N yet none of them have had the option to arrange a picture having a food colour or an overlap scar that is never indications of forge money. Having Mean mistake rate 42% . "Recognition of fake notes Using Neural Network" has suggested serving the recent picture using a procedure known as levelling to expel each and every commotion and turn into gray-scale from R.G.B. Subsequent to changing over it is relied upon to remove highlights as dark steak spotting imprints to scan real/forge with proficiency about 72%. "Paper Currency acknowledgment technique by a Small Size Neural Network with Optimized Matrices" , have utilized the idea of spaces to isolate the entire note into a lot of pixels like a grid and spread pieces of the framework to make pictures and afterward the preparation pictures for a compact Size N.N and have proficiency about 71%. "Counterfeit Currency Detection utilizing picture handling" , have presented utilizing basic MATLAB to group notes dependent on their category and furthermore identufy forge. Their work depends on Edge Spotting, Image Classification, Feature Extraction. Their work makes reference to identify just recognize pictures with a base force edge about 70% and more. Paper 7 from the table below proposes using a previous sifted through channel by finding neighborhood mean and vacillation and a short time later going for Paper 4 similarly proposes Neural Networks distinguishing the phony notes yet uses C.Y.M.K. extraction as opposed to R.G.B. because C.Y.M.K. is the parameter for the printing of anything on earth and simply sensitive copy pictures and gathered by R.G.B. He moreover proposes a non-Machine Learning system by considering the light refracted from the sparkling segment of the note. The work area work has shown a mean precision level about 54%. Paper 9 from the table have proposed relative component extraction of features like printing methodology, ink properties and crafted by craftsmanship used in the closer view and establishment to check for counterfeit notes and making them as parameters on which phony is picked. The work area work have has attested an accuracy about 72% on their model. A larger piece of these techniques rely upon separating and getting the note with everything taken into account and a while later applying channels to it. Subsequently, the food steins and overlay marks have been in like manner considered during the treatment of the image and are figured in as un-removable noise which impacts the outcome of the model. Moreover, there have been no confinements on process power and space availability on any of the models. A model will be made which will give most perfect results in obliged and reasonable proportion of advantages for such a degree, that it might be solidified into rapid and fundamental systems of ordinary day by day presence.

2.1 Literature Survey Table

REF NO.	OBJECTIVE	ALGORITHM USED	DATA SETS	METRICS	REMARK
1	Fake Currency detection using Image Processing	Image Capture Under UV light. RGB to Greyscale conversion Edge detection Segmentation Intensity of feature calculation	Standard RBI Guidelines Indian Currency	Handles live data	Cannot classify damaged authentic notes
2	A paper currency recognition method by a Little Size Neural Networks with Closed matrices method	Image Processing Matrix Calculation Neural networks	Japanese Currency	Usage of pixel masking and Analysis	Does not work efficiently on the US Dollar
3	Indian fake note detection using AI	Image Capture Under UV light RGB to Greyscale conversion Edge detection Feature comparison and NN efficiency calculation	Standard RBI Guidelines Indian Currency	Mean squared error rate Root mean square error rate	Cannot classify Rs 2000 note. Too complex

4	Fake Currency detection using image processing	Image Processing K-NN SVM	Indian Currency	Edge Detection based data point mapping	SVM having better performance than K-NN
5	Recognition of fake currency notes using Convolutional Neural Networks	Convoluted Neural Network	Standard Google Net/Inception Data set Le Net data set	Pre-processing Image re scaling Using CNN	Having high efficiency loss in basic filter
6	Machine Dating of Hand-written manuscripts	Multinomial Chi-square test	National Library Kolkata	Probabilistic model	Sometimes practical value differs from actual value
7	Fake Currency detection using Hidden Markov Model	Feature Extraction Color Histogram Texture based feature extraction	American Dollar Euro	Probabilistic model	Different values for different currencies

Table 2.1: Previous System

2.2 Inference from survey

The following are the important research gaps identified from the survey:

- Huge amount of compute time.
- Huge space required for data storage.
- Wear and tear marks and food steins cannot be ignored.
- Some currency notes too complex to train as too many features are being considered.
- Low efficiency value of the model

CHAPTER 3

PROPOSED APPROACH

3.1 Architectural Diagram

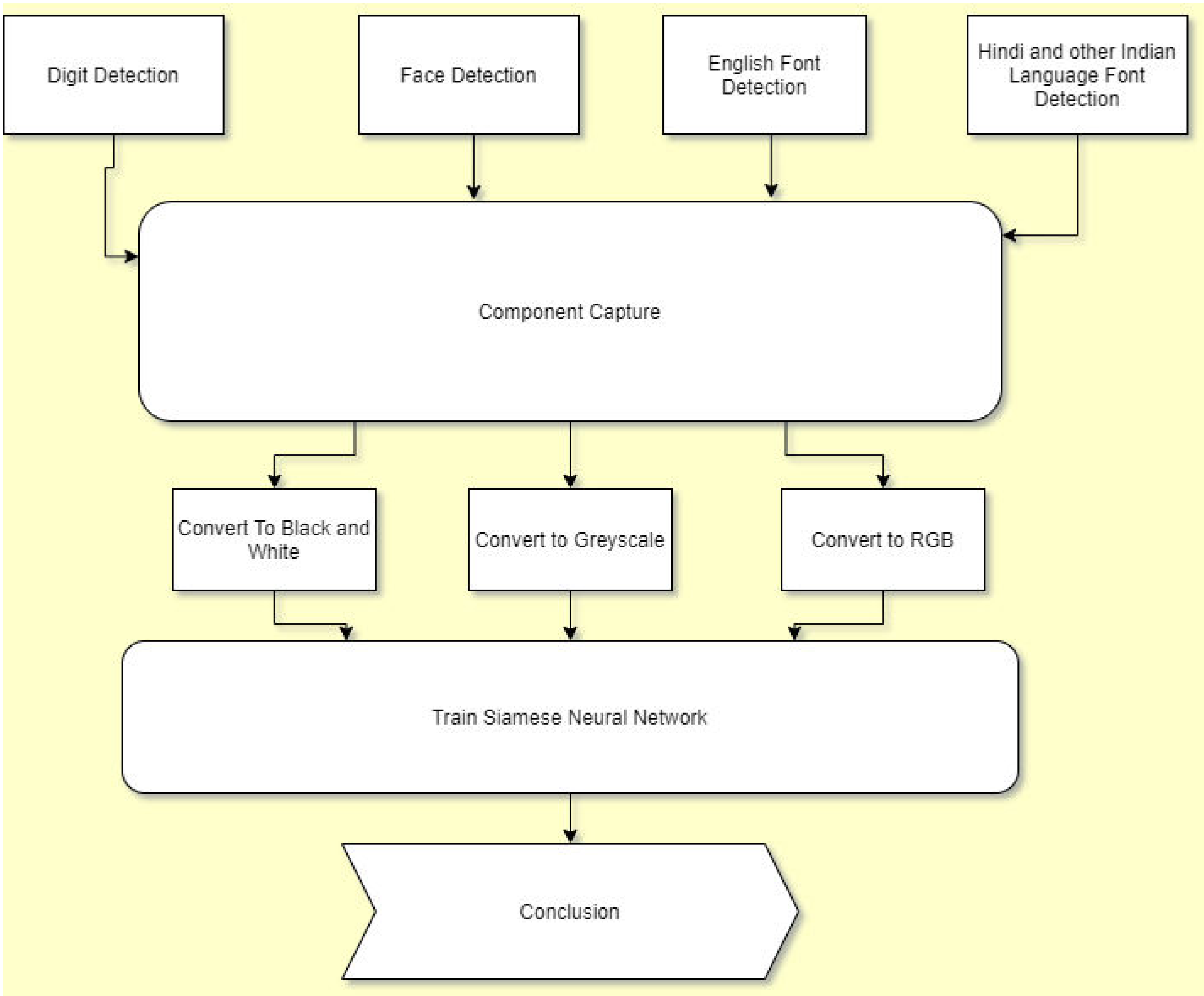


Figure 3.1: Architectural Diagram

.

The initial modules involves implementing Digit Detection using the the Digit Detection harrcascade, Face Detection using the Face Detection Harrascade using the Face Detection Algorithm trained for only Gandhis face in Open CV. English Font Detection is done using the same harrcascades for the font detection trained against the given font template in the Rs 100 note. Hindi and other language font detection are done the same way as English font detection but changing the appropriate language font templates as standard RBI guidelines. All these features are captured from the dataset of regular photos of the currency. After the detected features are extracted, they are multiplied by adding a number of filters like RGB, and Greyscale and Black and White filters and those are used as basic filters to train the Siamese Neural Network. The end-result of the trained model shows the detection accuracy with increasing epochs.

3.2 Novelty in Methodology

- The approach is to divide the whole note into features and detecting them individually and then checking for the authenticity of the features.
- Reducing the size of the data set and limiting it to the smallest possible space.
- Trying to discard the fold and discoloration marks before checking the counterfeit note.
- Earlier even the food stains or any text on notes were considered counterfeit which won't no longer be a problem using a Resource Efficient Neural Network.

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

- Intel or AMD processor 1.5 GHz Dual Core or Quad Core Processor with 4GB or 2GB RAM minimum.
- Maximum 2-3 GB of Hard Disk Space.
- A consistent Internet Connection
- Basic VGA Camera Sci-kit Learn,
- Python Runtime Environment.
- Jupyter Notebook, TensorBoard, Keras packages
- Xampp Server and phpMyAdmin

CHAPTER 5

MODULES

5.1 Face Detection

In face detention , the undertaking is to find the zones and sizes of all articles in an image that have a spot with a given class.. Face-detention calculations ranges around the recognition of face features. It is undifferentiated from picture discovery in which the live image of someone is correlated a little bit by bit. Any changes from the actual trained data will invalidate the planning procedure. A face-recognition approach reliant on the genetic count and the eigen-face procedure:First, the eye regions are distinguished by testing all the valley territories in the diminished tier picture. By then the inherited count is used to make all the possible face regions which join the eyebrows, the iris, the nostril and the mouth areas.Each conceivable face feature is standardized to minimize the glowing impacts, which is brought about by lopsided brightening; and the showing impact,which is a result of head advancement. The evaluation of all applicant is rated based on its prognosis of the eigen-faces.

5.2 Digit Detection

Digit Localization is done using Maximally Stable Extremal Regions (MSER) method which serves as a stable feature detector. MSER is mainly used for blob detection within images. The blobs are continuous sets of pixels whose external limit pixel forces are higher (by a given edge) than the inward limit pixel powers. Such areas are supposed to be maximally steady on the off chance that they don't change a lot over a shifting measure of powers. MSER has a lighter run-time unpredictability of O(nlog(log(n))) where n is the complete number of pixels on the picture. The calculation is additionally strong to obscure and scale. This makes it a good candidate for extraction of text / digits.

5.3 Text Detection

In present day coming of computerized write, "text style" is often interchangeable with "typeface". Each style is in a different "text style document"— for example, the typeface "Bulmer" may incorporate the textual styles "Bulmer roman", "Bulmer italic", "Bulmer striking" and "Bulmer broadened"— however the expression "textual style" may be applied either to one of these by themselves or to the entire typeface.

In both customary typesetting and current use, "text style" alludes to the conveyance system of the typeface plan. In customary typesetting, the text style would be produced using metal or wood. Today, the textual style is an advanced record.

5.4 Siamese Neural Network(SNN)

At first the entire money note has been isolated into highlights. Utilizing OpenCV to at first distinguish the element from note, the typical full-scale picture has been downsized to a 20 X 20 picture to make it simpler to prepare the harrcascades. After the harrcascades have been prepared the live picture distinguishes the highlights from the note and have been put away in a storehouse. These go about as the info esteems for the information work in the SNN. The picture of course has been changed over to greyscale to make it simpler for the model to learn.

After the highlights have been identified by means of OpenCV these are taken care of in to the SNN autonomously. There will be a different SNN for English text style, a different SNN for Hindi textual style, correspondingly for Gandhi's face of the Governor and the shading shade of one proportion of the note too. Each model takes at least 10 minutes or less or a limit of an hour to prepare and can be effectively utilized them from now on. Contingent upon the consequence of each SNN, an end emerges whether the last note is phony or true. Therefore, any possibility of a frail train on any element can decide to preclude that highlight and decide the outcome through different highlights.

- Non-negativity: $\delta(x, y) \geq 0$
- Identity of Discernible: $\delta(x, y) = 0 \iff x = y$
- Symmetry: $\delta(x, y) = \delta(y, x)$
- Triangle inequality: $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$

$$\delta(x^{(i)}, x^{(j)}) = \begin{cases} min \parallel \mathrm{f}\left(x^{(i)}\right) - \mathrm{f}\left(x^{(j)}\right) \parallel, i = j \\ max \parallel \mathrm{f}\left(x^{(i)}\right) - \mathrm{f}\left(x^{(j)}\right) \parallel, i \neq j \end{cases}$$

This form also allows the siamese network to be more of a half-twin, implementing a slightly different functions

$$\text{if } i = j \text{ then } \delta\left[\mathrm{f}\left(x^{(i)}\right), \mathrm{g}\left(x^{(j)}\right)\right] \text{ is small}$$

$$\text{otherwise } \delta\left[\mathrm{f}\left(x^{(i)}\right), \mathrm{g}\left(x^{(j)}\right)\right] \text{ is large}$$

i, j are indexes into a set of vectors

$\mathrm{f}(\cdot), \mathrm{g}(\cdot)$ function implemented by the half-twin network

$\delta(\cdot)$ function implemented by the network joining outputs from the siamese network

5.5 Convolutional Neural Network(CNN)

The traditional neural network can take information in the form of any image format, assign important and varied perspectives / questions to the image, and distinguish it from one another. In contrast to any other calibration calculation, the pre-requisite handling of a cannet is minimal. Although raw technology is hand-crafted in channels, with proper preparation, ConNets can master these channels / features.

The convention operation aims to remove advanced-level headlines, for example, edges from the information picture. It does not limit yourself to just one meeting layer. Regularly, the first converter is accountable for capturing low-level headlines, for example, edges, shading, tilt direction, and so forth. With the addition of layers, engineering also adjusts us to high-level headlines, giving the datasets a healthy understanding of how we look at images.

5.6 Comparing results between CNN and SNN

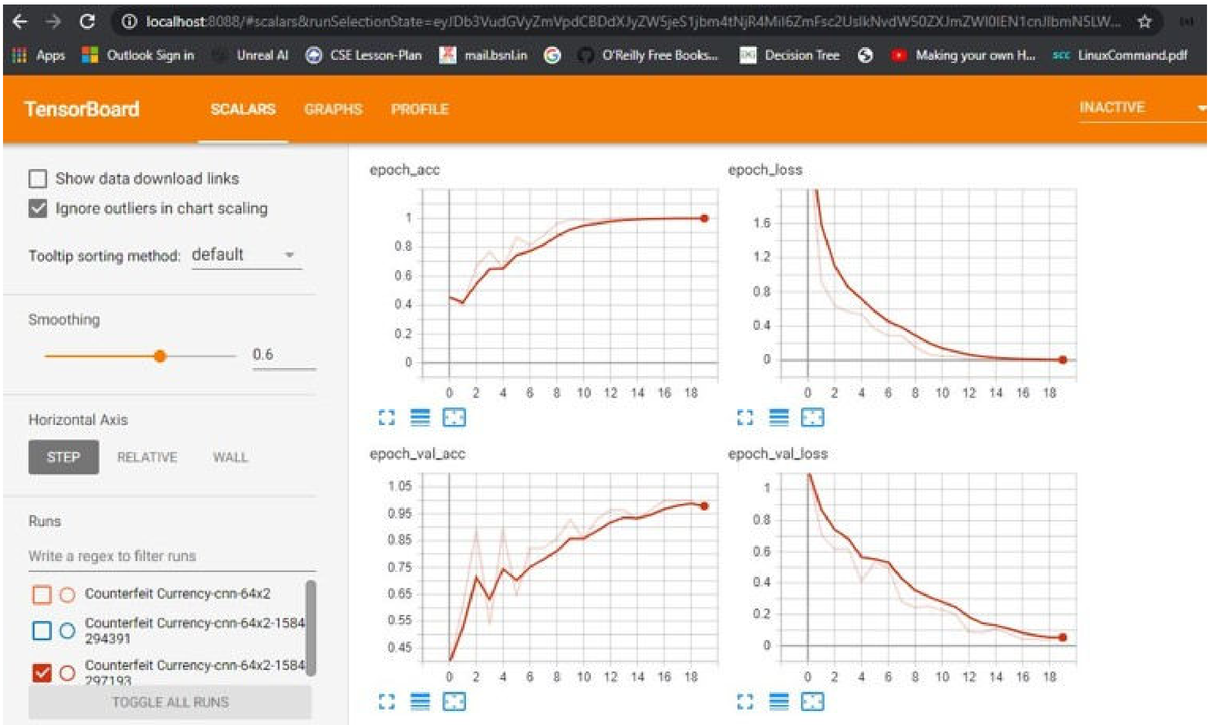


Figure 5.1: CNN

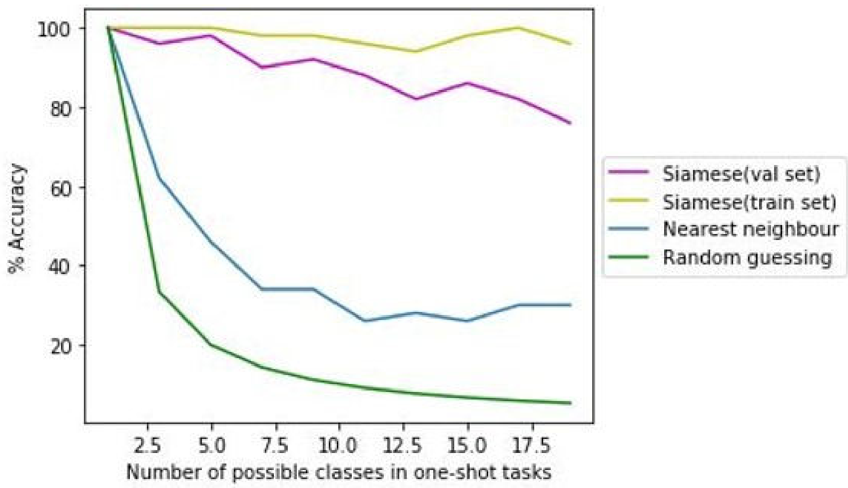


Figure 5.2: SNN

CHAPTER 6

CODING, TESTING

6.1 Face Detection

Face1 = vc.CascadeClssifier('harcascade$_{f}rontl.xml')$

$Eye1 = vc.CascadeClsifier('harcascade_{y}.xml')$

$cv2.CascadeClssifier('harcascade_{.}xml')$

$cap = vc.VideoCapture(1)$

$while2 :$

$rt, ig = cp.read()$

$gry = vc.cvtColor(img, vc.CLR_{B}GR2GY)$

$face = casc_{f}ace.tMultiScle(gray, 1, 1)$

$for(x1, y1, w1, h1)inface :$

$vc2.rect(ig, (x1, y1), (x1 + w1, y1 + h1), (0, 255, 0), 2)$

$gray = gray[y1 : y1 + h1, x : 1x1 + w1]$

$color = ig[y1 : y1 + h1, x1 : x1 + w1]$

$ey = ey_{c}ascade.detectMultScal(gray)$

$for(ex, ey, ew, eh)iney :$

$vc.rect(roi, (ex, ey), (ex + ew, ey + eh), (0, 245, 0), 4)$

$vc.imshow('imgur', imgur)$

$k2 = vc2.waitKey(20)$

$ifk2 == 29 :$

$break$

$cam.release()$

$vc2.destroyAllWindows()$

6.2 Text Detection

def code$_{p}reds(score, geo):$

$(nRows, nCols) = score.shape[2:7]$

$rect = []$

$confid = []$

$foryinrange(0, nRows):$

$scData = score[0, 0, yy]$

$aData = geometry[0, 4, yy]$

$forxinrange(0, nCols):$

$ifscData[x] < ags["min_{c}onf"]:$

$(offsX, offsY) = (xx * 4.0, yy * 4.0)$

$angle = anglesData[x]$

$return(rects, confidences)$

$ap1 = arg.ArgumentParser()$

$ap1.add_{a}rgument("-v", "\breve{}video_{c}odec", type = str,$

$ap1.add_{a}rgument("-c", "\breve{}min_{c}onfdence", type = float, default = 0.6,$

$net_{1} = cv2.dnn.readNet(args["east_{1}"])$

$ifnotargs.get("video_{1}", False):$

$print("[INFO]startingvideostream...")$

$else:$

$vs = cv2.VideoCapture(args["video_{2}"])$

$fps_{s}t = FPS.start()$

$frame = vs1.read()$

$frame = frame[1]ifargs.get("video_{2}", False)$

$else$

$changethesize$

$frame_{O}TF = img.imutils.resize(frame, width = 500)$

$original = frame.copy()$

$(H1, W1) = frame.shape[:4]$

$rW1 = W1/float(newW1)$

$rH1 = H1/float(newH1)$

$frame_new = cv2.resize(frame, (newW1, newH1))$

$startX_plot = int(startX_plotrW1)$

$startY_plot = int(startY_plotrH1)$

$endX_plot = int(endX_plotrW1)$

$endY_plot = int(endY_plotrH1)$

$vc2.rectangle(orig, (startX_plot, startY_plot), (endX_plot, endY_plot), (0, 255, 207), 2)$

$fps.update()$

$vc2.imshow(\jmath DetectingtheText\jmath, original)key_press = vc2.waitKey(1)0xFF$

$ifthe`q`killprocess, breakfromtheloop$

$ifkey_press == ord(\jmath q\jmath) :$

$break//breaktheexcecution$

$vc2.destroyAllWindows()$

6.3 Colour Gradient Filter

import cv2

cap = vc2.VideoCapture(0)

while True:

$\mathrm{frame}_I NF = capture.read()$

$hsv_r ead = vc2.cvtColor(frame_I NF, cv2.COLORBGR)$

$lowerred = np1.arry([120, 50, 50])$

$upperred = np1.arry([255, 207, 180])$

$mask1 = vc2.inRange(hsv_r ead, lowered, uppered)$

$res_O TF = vc2.bitwiseand(frame, frame_I NF, mask = mask1)$

$vc2.imshow(0frme0, frme1)$

$vc2.imshow(0msk0, msk1)$

$k_O TF = cv2.waitKey(5)0xFF$

$ifk == 27 :$

$break$

$vc2.destroyAllWindows()$

6.4 Feature Mapping

import numpy as np

import cv2

import matplotlib.pyplot as plt

img1 = cv2.imread('train$_{f}eature.jpeg',0)$

img2 = cv2.imread('test$_{f}eature.jpeg',0)$

orb = cv2.ORB$_{c}reate()$

kp1, des1 = orb.detectAndCompute(img1,None)

kp2, des2 = orb.detectAndCompute(img2,None)

bf = cv2.BFMatcher(cv2.NORM$_{H}AMMING, crossCheck = True)$

matches = bf.match(des1,des2)

matches = sorted(matches, key = lambda x:x.distance)

img3 = cv2.drawMatches(img1,kp1,img2,kp2,matches[:10],None, flags=2)

plt.imshow(img3)

plt.show()

6.5 Currency Classifier

path = "C:/Program Files/Python36/dataset/indian$_{c}urrency_{n}ew/training$"

$CATEGORY = ["100", "200"]$

$forcategoryinCATEGORY:$

$path1 = os.path.join(path, category)$

$forimginos.listdir(path1):$

$img_{a}rry = vc2.imread(os.path.join(path1, img), cv2.IMREAD.GRAYSCALE)$

$NEWIMG_{S}IZE = 400$

$new_{a}rray = vc2.resize(img_{a}rray, (IMG_{S}IZE, IMG_{S}IZE))$

$training_{d}ata()$

$len(training_{d}ata)$

$importpickleaspp$

$pp_{o}ut = open("X1.pp", "wb")$

$pp.dump(X, pp_{o}ut)$

$pp_{o}ut.close()$

$pp_{o}ut = open("y1.pp", "wb")$

$pp.dump(y1, pp_{o}ut)$

$pp_{o}ut.close()$

$pp_{i}n = open("X1.pp", "rb")$

$X1 = pp.load(pp_{i}n)$

$Y1 = pp.load(pp_{i}n)$

$fromtensorflow.keras.modelsimportSequential$

$X2 = pp.load(open("X1.pp", "rb"))$

$Y2 = pp.load(open("y1.pp", "rb"))$

$X2 = X2/255.0$

$Model1 = Sequential()$

$Mode1l.add(Conv2D(64, (4, 4), input_{s}hape = X.shape[1 :]))$

$Model1.add(Activation('relu'))hello1$

$Model1.Add(MaxPooling2D(pool_{s}ize = (2, 2)))$

$Model1.add(Conv2D(64, (3, 3)))$

$Model1.add(MaxPooling2D(pool_size = (2, 2)))$

$Model1.add(Flatten())$

$Mode1l.add(Dense(64))$

$Model1.add(Dense(1))$

$Model1.add(Activation('sigmoid'))$

$Mode1l.compile(loss =' binary_crossentropy',$

$optimzer =' adam',$

$metrics_fac = ['accuracy'])$

$model1.fit(X, y, batch_size = 32, epochs = 10, validation_split = 0.1)$

$X2 = pickle.load(open("X.pickle", "rb"))$

$Y2 = pickle.load(open("y.pickle", "rb"))$

$X2 = X2/255.0$

$Model2 = Sequential()$

$Model2.add(Conv2D(64, (3, 3), input_shape = X.shape[1 :]))$

$model.2add(Activation('relu'))hello1$

$model2.add(MaxPooling2D(pool_size = (2, 2)))$

$model2.add(Conv2D(64, (3, 3)))$

$model2.add(Activation('relu'))hello2$

$model2.add(MaxPooling2D(pool_size = (2, 2)))$

$model2.add(Flatten())$

$model.2add(Dense(64))$

$model2.add(Dense(1))$

$model2.add(Activation('sigmoid'))$

$model2.compile(loss =' bnarycrossentropy)$

$optimizer =' adam'$

$, metrics = [0accuracy'$

$])$

$Model2.fit(X, y, batch_size = 32, epochs = 10, validation_split = 0.2)$

$model.fit(X, y, batch_size = 32, epochs = 20, validation_split = 0.2, callbacks =$

$fromsklearn.metricsimportclassification_report, confusion_matrix$

$train_data_path2 = 0C : /ProgramFiles/Python36/dataset/indiancurrencynew/training0$

$test_data_path1 = 0C : /ProgramFiles/Python36/dataset/indiancurrencynew/validation0$

$img_rows = 160$

$img_cols = 160$

$epochs = 20$

$batch_size = 32$

$num_of_train_samples = 840$

$num_of_test_samples = 200$

$train_datagen = ImageDataGenerator(rescale = 1./255,$

$rotation = 40,$

$t_ra = 0.2,$

$hshif = 0.2, ill_mode = 0nearest0$

$)$

$test_gen = ImageDataGenerator(rescale = 1./255)$

$train_gen = train_gen.flow_from_directory(trainpath,$

$tsize = (imrows, imcols),$

$bsize = batchsize,$

$)$

$validgenerator = test_gen.flow_directory(testpath,$

$tsize = (imrows, imcol$

$bsize = batch_size,)$

Model3.add(Activation(model3.add(M axP ooling2D(pool$_size = (2, 2)))$

$Model3.add(Flatten())$

$Model3.add(Dense(64))$

$Model3.add(Activation($

0))

$4.add(Dropout(0.5))$

$Model4.add(Dense(5))$

$model.4add(Activation($

0

$softmax0$

))

$Model4.compile(loss = 0$

$categorical_c rosentropy'$

$optimizer = 0$

$rmsprop0$

,

$= [0accuracy0$

])

$importnumpyasnp$

$Ypred = model3.predict_generator(valgen, numtestsamples//batc$

$ypred = np.argmax(Ypred, axis = 1)$

6.6 SNN

train_folder = "../images_background/"

val_folder = ’../images_evaluation/’

save_path = ’../data/’

def loadings(path,n = 0):

’’’

path => Path of train directory or test directory

’’’

X=[]

y = []

cat_dict =

lang_dict =

curr_y = n

for alphabetlist in os.listdir(path):

print("loading character: " + alphabet)

langdict[alphabet] = [curr_y, None]

alphabetpath = os.path.join(path, alphabetlist)

forletterinos.listdir(alphabetpath) :

catdict[curry$_y] = (alphabetlist, letter_charac)$

$catimages = []$

$letPath = os.path.join(alphapath, letter_charac)$

$forfilenameinos.listdir(letpath) :$

$imgpath = os.path.join(letpath, filename)$

$img = imread(imgpath)$

$catimages.append(img)$

$yarray.append(curryy)$

$try :$

$Xarray.append(np.stack(catimages))$

$exceptValueErrorase :$

$langdict[alphabetlist][1] = curry1$

$y = np.vstack(y_old)$

$X = np.stack(X_old)$

$returnX, y, langdict$

$X, y, c = loadings(trainfolder)$

$withopen(os.pathdiir.join(savepath, \jmath train.pp\jmath), \jmath wb\jmath)asf :$

$pp.dump((X, c), f)$

$Xval, yval, cval = loadings(val_folder)$

$withopen(os.path.join(save_path, \jmath val.pickle\jmath), \jmath wb\jmath)asf :$

$pp.dump((Xval, cval), f1)$

$definitwts(shape, name = None) :$

$returnnp.random.normal(loc = 0.0, scale = 1e2, size = shape)$

$definitialize_bias(shape, name = None) :$

$returnnp.random.normal(loc = 0.5, scale = 1e2, size = shape)$

$siamesemodel(input_shape) :$

$left_inpt = Input(shape)$

$rht_inpt = Input(shape)$

$model1 = Sequential()$

$model1.add(Conv2D(64, (10, 10), activation = 0$

$relu0$

$, input_shape = input_shape, kernel_initializer =$

$initialize_weights, kernel_regularizer = l2(2e4)))$

$model1.add(MaxPooling2D())$

$model1.add(Conv2D(128, (7, 7), activation = 0$

$relu0,$

$kernel_=initialize,$

$bias_initializer = initialize_bias, kernel_regularizer = l2(2e4)))$

$model1.add(MaxPooling2D())$

$mode1l.add(Conv2D(128, (4, 4), activation = 0$

$relu0$

$, kernel = initialize_weights$

$bias = initialize_bias, kernel_regularizer = l2(2e4)))$

$model1.add(MaxPooling2D())$

$Model1.add(Conv2D(256,(4,4), activation = 0$

$Model1.add(Flatten())$

$Model1.add(Dense(4096, activation = 0$

$sigmoid0,$

$prediction = Dense(1, activation = 0$

$sigmoid0$

$, bias_initializer = initialize_bias)(L1_distan$

$siamese_net = Model(inputs = [left_input, right_input], outputs = prediction)$

$returnsiamese_net$

$model2 = get_siamese_model((105, 105, 1))$

$model2.summary()//fromIPython.displayimportImageImage(retina = True, filename =$

$0model.png0)$

$optimizer1 = adam(lradd = 0.00008)model2.compile(loss = \text{ʝ}binary_crossentropy\text{ʝ}, optimizer =$

$optimizer1)$

$ifi >= batch_size//2 :$

$category_2 = category$

$else :$

$category_2 = (category + rng.randint(1, n_classes))$

$pairs[1][i, :, :, :] = X[category_2, idx_2].reshape(w, h, 1)$

$returnpairs, targets$

$defgenerate(batch_size, s = \text{ʝ}train\text{ʝ}) :$

$whileTrue :$

$pairs, targets = get_batch(batch_size, s)$

$yield(pairs, targets)$

$defmake_oneshot_task(N, s = \text{ʝ}val\text{ʝ}, language = None) :$

$ifs == 0$

$train0 :$

$X = Xtrain$

$categories = train_classes$

$else :$

$X = Xval$

$categories = val_classes$

$n_classes, n_examples, w, h = X.shape$

$indices = rng.randint(0, n_examples, size = (N,))$

$if language is not None :$

$lowend, highend = categories[language]$

$if N > higheendlowend :$

$raise ValueError(\jmath This language () has less than letters \jmath .format(languag$

$categories = randomng.choice(range(lowend, highend), size = (N,), replace = False)$

$else :$

$categories = randomng.choice(range(n_classes), size = (N,), replace = False)$

$true_category = categories[0]$

$ex1, ex2 = rng.choice(nexamples, replace = False, size = (2,))$

$test_image = np.asarray([X[true\_category, ex1, \ , \ ]]N).reshape(N, w, h, 1)$

$support_set = X[categories, indices, :, :]$

$support_set[0, :, :] = X[true_category, ex2]$

$support_set = supportset.reshape(N, w, h, 1)$

$Targets_vals = np.zeros((Number,))$

$Targets_vals[0] = 1$

$Targets_vals, test_images, suppset = shuffle(targets_valss, test_images, supset)$

$pairs = [test_images, suppset]$

$return pairs, targets_vals$

$def test_oneshot(model, N, k, s = \jmath val \jmath, verbose = 0) :$

$ncorrect = 0$

$if verbose :$

$print(\jmath Evaluating model on random way one shot learning tasks ... \jmath .forma$

$for i in range(k) :$

$inputs_vals, targets_vals = make_oneshotLearn(N, s)$

$Prob_stat = model.predict(inputs_vals)$

$percent_correct = (100.0 n_correct/k)$

$if verbose :$

$print(\jmath Getiing a mean of \% of ways n n acc shot learning Val_acc \jmath .format(percent$

$Return perc_acc$

$$
\begin{aligned}
&evaluate_{e}very = 200intervalforevaluatingononeshottasks\\
&batch_{s}ize = 32\\
&n_{i}ter = 20000No.oftrainingiterations\\
&N_{w}ay = 20howmanyclassesfortestingoneshottasks\\
&n_{v}al = 250howmanyoneshottaskstovalidateon
\end{aligned}
$$

$$
\begin{aligned}
&model_{p}ath = 0./weights/0\\
&print(\jmath Startingtrainingprocess!\jmath)\\
&print(\jmath\jmath)\\
&t_{s}tart = time.time()\\
&foriinrange(1, n_{i}ter1) :\\
&(inputs, targets) = getbatch(batch_{s}ize)\\
&loss = model.train_{o}nbatch(inputs, targets)\\
&ifi\\
&print(\jmath\jmath)\\
&print(\jmath Timefor0iterations : 1mins\jmath.format(i, (time.time()tstart)/60.0))\\
&print(\jmath TrainLoss : 0\jmath.format(loss))\\
&val_{a}cc = test_{o}neshot(model, Nway, nval, verbose = True)\\
&model1.saveWeights(os.path.join(mode1l_{p}ath, 0weights..h50.format(iterator)))\\
&ifval_{a}cc >= best :\\
&print(\jmath Currentbest : 0, previousbest : 1\jmath.format(val_{a}cc, best))\\
&best = val_{a}cc\\
&Mode1l.load_{w}eights(os.path.join(model_{s}ave_{p}ath, \jmath weights.30000.h5\jmath))\\
&defcnn(N_{w}ays, n_{t}rials) :\\
&print(\jmath Evaluatingnearestneighbouronuniquewayoneshotlearningtasks...\jmath.\\
&n_{r}ight = 0\\
&foriinrange(n_{t}rials) :\\
&pairs, targets = make_{o}neshot_{t}ask(Nways, \jmath val\jmath)\\
&correct = nearest_{n}eighbour_{c}orrect(pairs, targets)\\
&n_{r}ight+ = correct\\
&return100.0n_{r}ight/n_{t}rials\\
&ways = np.arange(1, 20, 2)
\end{aligned}
$$

$resume = False$

$trials = 50$

$val_accs, train_accs, nn_accs = [], [], []$

$forNinways :$

$val_accs.append(test_oneshot(model, N, trials, \jmath val \jmath, verbose = True))$

$train_accs.append(test_oneshot(model, N, trials, \jmath train \jmath, verbose = True))$

$nn_acc = test_nn_accuracy(N, trials)$

$nn_accs.append(nn_acc)$

$print(\jmath NNAccuracy = \jmath, nn_acc)$

$withopen(os.path.join(save_path, \jmath accuracies.pickle \jmath), \jmath wb \jmath)asf :$

$pp.dump((val_accs, train_accs, nn_accs), f)$

$withopen(os.path.join(save_path, \jmath accuracies.pickle \jmath), \jmath rb \jmath)asf :$

$(val_accs, train_accs, nn_accs) = pickle.load(f)$

$defconcat_images(X) :$

$nc, h, w,_= X.shape$

$X = X.reshape(nc, h, w)$

$n = np.ceil(np.sqrt(nc)).astype(\jmath int8 \jmath)$

$img = np.zeros((nw, nh))$

$x = 0$

$y = 0$

$forexampleinrange(nc) :$

$img[xw : (x + 1)w, yh : (y + 1)h] = X[example]$

$y+ = 1$

$ify >= n :$

$y = 0$

$x+ = 1$

$returnimg$

$defplot_oneshot_task(pairs) :$

$ax1.matshow(pairs[0][0].reshape(105, 105), cmap = 0$

$gray0$

$)$

$img = concat_images(pairs[1])$

$ax1.get_yaxis().setvisible(False)$

$ax1.get_xaxis().set_visible(False)$

$ax2.matshow(img, cmap = 0$

$gray0$

$)$

$plt.xticks([])$

$plt.yticks([])$

$plt.show()$

$ax.plot(ways, val_accs, \jmath m\jmath, label = \jmath Siamese(valset)\jmath)$

$ax.plot(ways, train_accs, \jmath y\jmath, label = \jmath Siamese(trainset)\jmath)$

$plt.plot(ways, nn_accs, label = \jmath Nearestneighbour\jmath)$

$ax.plot(ways, 100.0/ways, \jmath g\jmath, label = \jmath Randomguessing\jmath)$

$plt.xlabel(\jmath Numberofpossibleclassesinoneshottasks\jmath)$

$plt.ylabel(\jmath\%Accuracy\jmath)$

$plt.title(\jmath OmiglotOneShotLearningPerformanceofaSiameseNetwork\jmath)$

$box_plotter_Fig = axInput_params.get_position()$

$inputs, targets = make_oneshot_task(50, \jmath val\jmath,$

$plot\_oneshot\_task(inputs)$

CHAPTER 7

RESULTS

7.1 Data Sets

Entire thought of utilizing this sort of matrix is to diminish the proportion of the informational collection and abbreviate process time massively. The images in the dataset are being multiplied in many ways. A portion of the images are being rescaled into 50 x 50 size and being saved as a separate image. A certain degree of white noise is being added to some of the images and saved separately using OpenCV. Some Images have the same currency note taken at a different angle and amount of light falling on the note. Images have been digitally rotated at increments of very small angles to multiply the data set. The advantages of having a moderately sized data set is to make the sure the model performs well in the training set as well as the validation set.

7.2 Tools Used

OpenCV is an Open Source library which contain functions and packages are that can be used to configure and calibrate an optical device and use it as an interface for the computer. It is a product made by Intel at first and was eventually released to public for free development. It the most versatile Computer Vision platform because it is cross-compatible to so many platforms like C++, C, Java and Python which are so different in their structure and yet they can be used along with each other. The libraries are mostly used to read and analyse the image captured by the camera and analyse the pixels mapped singularly or collectively. This procedure has fundamentally been hyarcascades on a Linux Based Server on P.U.T.T.Y. Client Service, N.S.C.P. to get to the File System of the server and Tensor stream for the most part to prepare and run the SNN. Using harrcascades all the images are uploaded, and the program is run. The target harrcascade file gets built and they can be used for basic image detection.

7.3 Result and Discusion

The final model detects forge notes using Siamese with ranging from 10 to 100 epochs. The values go down for about 20 to 30 for validation accuracy and then go up while increasing to further epochs. The results show the difference in different models like KNN and random guessing having accuracy below 50 but the One-Shot Learning Tasks report accuracies above 90. To compare the One-Shot Learning Tasks with Convoluted Neural Networks we are Looping the One-Shot learning tasks wit gradual increments from 10 One-Shot Learning Tasks to 90 One Shot Learning Tasks. This way the accuracies can have fair ground in comparing to each other. The One-Shot Learning Tasks with accuracy around 87- 91% are not remarkably better than the CNN accuracy of 95%+ but with the data set minimization and compute minimization with 50MB in One Shot Learning Tasks and 3GB in traditional CNN the overall efficiency of the One-Shot Learning Tasks is much better.

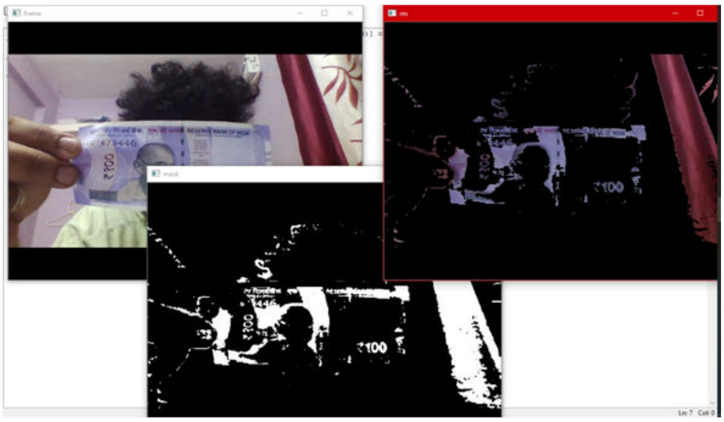


Figure 7.1: Color Gradient Filtering

In Figure 7.1 above shows a colour gradient filtering mechanism which performs live capture of the images and uses two important filters. The one below is a boolean filter which filters out the exact shade or gradient hex value of the Lavender colour gradient of the Hundred Rupee note which is fed into the code and displays it as white object and the rest is black. Any devitaion from that shade implying a counterfeit note will show up as 0 or black and the exact colour will show up as 1 or white.

In Figure 7.2 above shows an authentic Hundred Rupee note which has a high resolution image

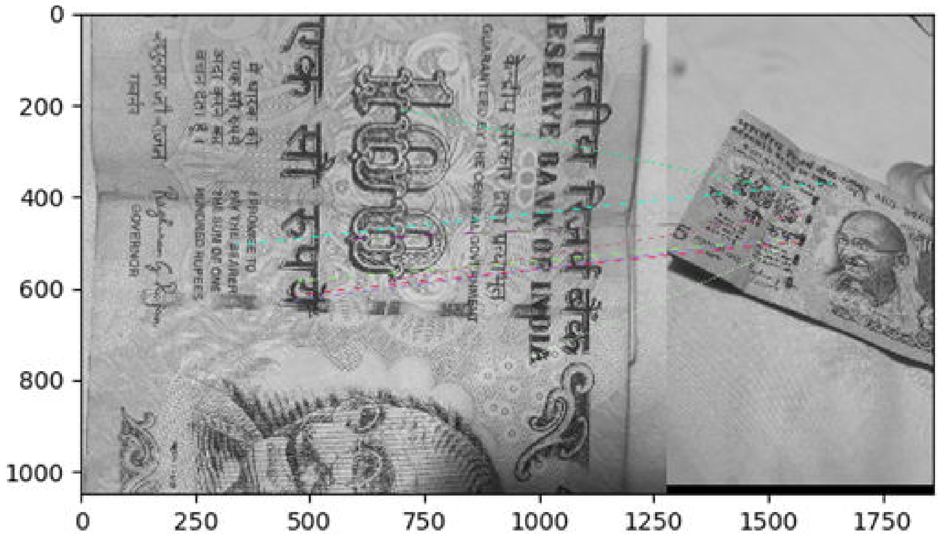


Figure 7.2: Feature Mapping

of the note in the left and another note with is low resolution and is taken at an angle with change in the light exposure. The code is supposed to compare and match the features from the image in the left and connect them against the image on the right. This is one of the preliminary procedures to determine whether the image placed in the right is at all a currency note. If features are matching then only we will go for detailed analysis using Deep Learning.

Figure 7.3: Training Images

The above Figures 7.3 and 7.4 are images of a 100 Rupee note. The first image is a full scale image of 400 x 200 high resolution. The second image is re-scaled to 250 x 250 medium resolution. The higher the resolution the longer it takes for the model to learn and more com-

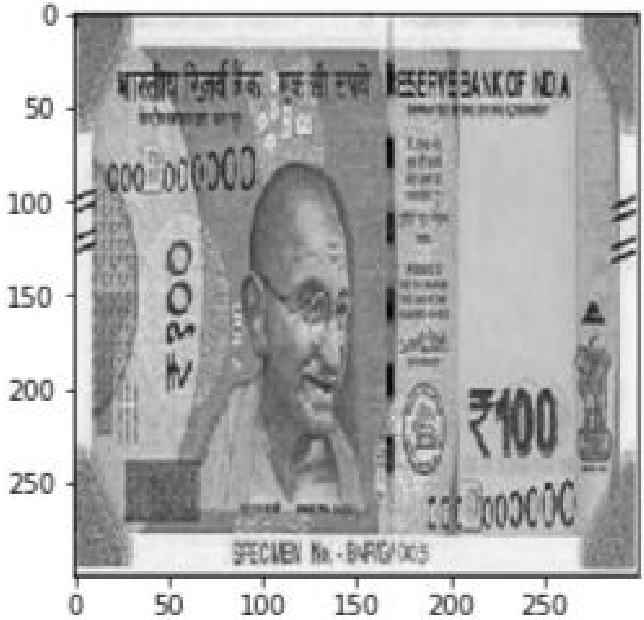


Figure 7.4: Training Images Rescaled

pute power is demanded. The goal is to re-scale it to a suitable value so that the image does not become hard to learn by the model resulting in a bad training model but to make it just enough for the model to learn well as well as use space and compute power efficiently.

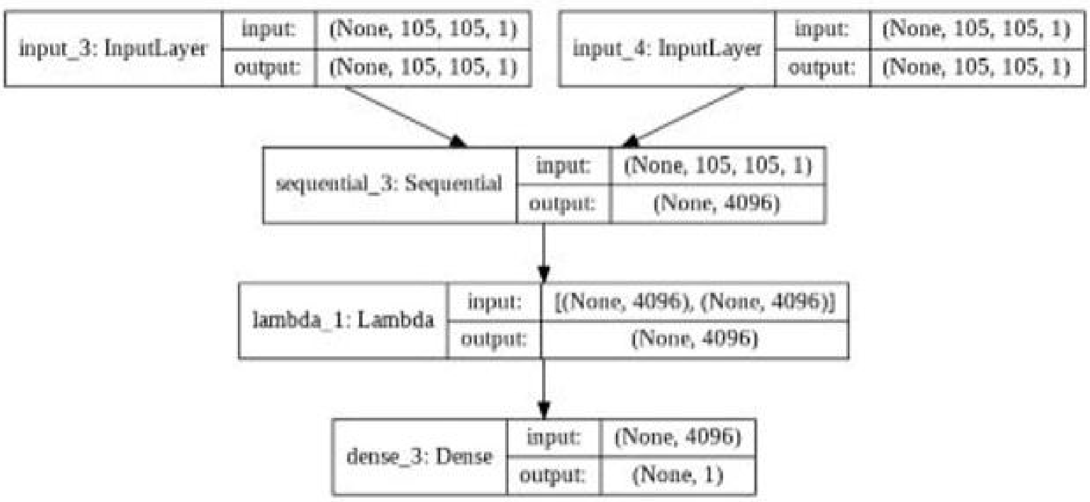


Figure 7.5: Layers

The next procedure is to scale all the images in the same way and then add layers of the Neural Network one by one. There are 2 input layers here. The first one is the left input layer and the second one is the right input layer. The input and output shape is the same for both the layers. After the input layers it is being fed into a Sequential Model which is the most method of handling multi-layer models. It then passes into a Lambda Layer which is the most important layer as it helps us define the function exclusively for the Siamese Neural Networks. Lambda functions are always important when the functions are not pre-built. After that comes the most common hidden layer called the Dense Layer consisting of neurons which involve matrix vector multiplications. The values of the matrix are parameters which are trainable. The parameters which are trainable get updated during back-propagation. The output shape of this layer changes to a single dimensional output which is the final result of the model. The mode of learning rate optimization which has been used here is Adam which stands for adaptive learning rate method. This method help increase speed of training

7.4 Validation Charts and Images

```
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|  0.99875|    0.785|
+----+---------+---------+
|   4|  0.99875|    0.685|
+----+---------+---------+
|   5|  0.99875|   0.4725|
+----+---------+---------+
END>
Training until now has taken 0 days 0 hours 0 minutes 42 seconds.

===== TRAINING 3-stage =====
<BEGIN
POS count : consumed   800 : 806
NEG count : acceptanceRatio    400 : 0.150602
Precalculation time: 4
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|   0.9975|    0.785|
+----+---------+---------+
|   4|  0.99625|    0.515|
+----+---------+---------+
|   5|   0.9975|    0.665|
+----+---------+---------+
|   6|   0.9975|     0.51|
+----+---------+---------+
|   7|   0.9975|     0.39|
+----+---------+---------+
END>
Training until now has taken 0 days 0 hours 1 minutes 1 seconds.

===== TRAINING 4-stage =====
<BEGIN
POS count : consumed   800 : 808
NEG count : acceptanceRatio    400 : 0.06628
Precalculation time: 4
```

```
===== TRAINING 9-stage =====
<BEGIN
POS count : consumed   800 : 822
NEG count : acceptanceRatio    400 : 0.00228986
Precalculation time: 4
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|  0.99625|   0.8775|
+----+---------+---------+
|   4|  0.99875|   0.8125|
+----+---------+---------+
|   5|  0.99625|    0.765|
+----+---------+---------+
|   6|  0.99625|   0.6025|
+----+---------+---------+
|   7|   0.9975|    0.635|
+----+---------+---------+
|   8|   0.9975|    0.525|
+----+---------+---------+
|   9|  0.99625|   0.3075|
+----+---------+---------+
END>
Training until now has taken 0 days 0 hours 2 minutes 51 seconds.
root@openvpnas2:~/opencv_workspace#
```

. **Figure 7.1**:Training on linux server

Epochs	SNN Accuracy (in percentage)	CNN Accuracy (in percentage)	KNN Accuracy (in percentage)	Random Guessing
4	80	60	50	25
6	85	80	46	20
8	92	82	40	18
10	96	90	42	12

Table 7.1: Comparing Validation Accuracy

Figure 7.1 shows the harrcascade training on the Linux Based server, this part is done well before any deep learning algorithms are applied. The table in the terminal shows HR stands for Hit Rate meaning what percent of true images it has classified correctly, where 1 means 100 percent, and consequently FA stands for False Alarm, meaning how many false images it has classified wrongly. The pictures show training done in stages from stage 0 through stage 9 done sequentially.

Table 7.1 compares the results obtained from Deep Learning algorithms used after that. Depending upon the applications of the algorithm, the performance is different in different situations and the type and magnitude of data that has been provided. The main objective is to extract the best possible accuracies in the minimal possible dataset size and compute time. The table lists a brief summary for different AI techniques. The table concludes that with random guessing and KNN, the size of the minimal dataset is affecting the results deeply as accuracy is unable to build. With massive increase in the size of the dataset it might show some improvement in KNN model. Traditional CNN model still tries to build accuracy but is unable to be at par with the Siamese Neural Network owing the size of the dataset that is being used. The table has been drawn from **Figure 5.1-5.2** where both the results have been plotted on graphs using TensorBoard and Jupyter Environment with matplotlib libraries respectively.

CHAPTER 8

FUTURE ENHANCEMENT

There are many ways in which this research can be enhanced. The ones having practical results and meaningful outcomes are the best choices. One such choice can be the addition of many more filters to the Neural Network to find out more features and give more depth to the learning. The model can also be trained across other currencies and other objects such as fake Chinese products. With more available compute power and camera quality the detection speeds can be also enhanced. The immediate future enhancement can be a mobile based application deployed by Android and to query a cloud-based server and detect the counterfeit currency live. As we are aiming at fast and efficient detection, a mobile or handheld device is most relevant in this situation.

CHAPTER 9

CONCLUSION

The single-shot profound learning way to deal with difficult proclamation is special since it diminishes cp-time and figure prerequisite recognize fake note also assist spares with timing and the dreary procedure that is generally taken in an ordinary bank to decide it through careful assessment and furthermore in an enormous confounded N.N. Thus, the fake location will support the thrift and furthermore prevent fear mongers from subsidizing their activities. The next prospect parts of this task is to actualize it in the mobile Device with the goal that notes can be minded the go and the administrative work can keep misrepresentation from happening immediately and sparing numerous organizations and employments from monetary danger.

REFERENCES

[1] Tushar Agasti, Gajanan Burand, Pratik Wade and P Chitra," Fake Currency Detection Using Image Processing," IOP Conf. Series: Materials Science and Engineering 263 (2017) 052047 doi:10.1088/1757-899X/263/5/052047.

[2] F.Takeda, S. Omatu,T. Inoue, and S. Onami, "Bill Money Recognition Using Neural Network with FFT as Pre-processor, " Trans. on ISCIE of Japan, Vo1.5, No.7, pp.265-273, 2010.

[3] E.Haripriya, K. Anusdha, "Detection of Indian Countefeit Banknotes Using Neural Networks" Volume 04, Issue 12; December- 2018 [ISSN: 2455-1457].

[4] Ms. Monali Patil, Jayant Adhikari, Rajesh Babu, "Fake Currency Detection using Image Processing," International Journal on Future Revolution in Computer Science Communication Engineering ISSN: 2454-4248 Volume: 4 Issue: 4(2017).

[5] Navya Krishna G, Sai Pooja G, Naga Sri Ram B, Yamini Radha V, Rajarajeswari P, "Recognition of Fake Currency Note using Convolutional Neural Networks," International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-5 March 2019"

[6] Angelo Frosini, Marco Gori, Member, IEEE, and Paolo Priami, "A Neural Network-Based Model for Paper Currency Recognition and Verification," IEEE transactions on Neural Networks, Vol. I, No 6, November 2006.

[7] UtpalGarain, S. K Parui, T. Paquet, L. Heutte, "Machine Dating of Handwritten Manuscripts,Vol. III, No 7, November 2009."

[8] Hamid Hassanpour , Payam M. Farahabadi, "Using Hidden Markov Models for paper currency recognition", Expert Systems with Applications 36 (2009) 10105–10111.

[9] Soo-Hyeon Lee and Hae-Yeoun Lee," Counterfeit Currency Note Detection Using Deep

Learning," International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 304-310.

[10] Murdoch, S.J Laurie," Machine Assisted authentication of paper currency: An experiment on Indian Banknotes" IEEE Trans. Pattern Anal. Mach. Intell. 10(6), 978–983 (2008).

[11] N. Akamatsu, M. Fukumi, "A method to design a neural pattern recognition system by using a genetic algorithm with partial fitness and a deterministic mutation" 11(4), 922–984 (2009).

[12] A. Chowdhury, N. Jahangir, Bangladeshi banknote recognition by neural network with axis symmetrical masks in masks, in: 10th international conference on computer and information technology. IJTEE 2010.

[13] Da-costa, Multiview banknote recognition with component and shape analysis IJTEE 2014.

[14] S. Das, CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more, IJTEE 2017.

[15] P.R. Nagpure, T.Ghotkar, S. Shetty, Recognition and Fake Note Detection, International Journal of Innovative Research in Computer and Communication Engineering Vol 4 (March 2016).

[16] T. Nishikage, F. Takeda, Multiple kinds of paper cur- rency recognition using neural network and applica- tion for Euro currency (2000) 27–47. 11. S. Omatu, F. Takeda, High speed paper currency recognition by neural networks (1995) 73–77.

[17] Paisios, Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms (2012).

[18] J. S.Dewan, Extended Local Binary Pattern for Face Recognition, Technology ICAET,23–41(2015).

[19] V.Vashishtha, M.Sadim, A Paper Currency Recogni- tion System Using Image Processing To Improve the Reliability with PCA Method, International Journal of Engineering Science (2015).

[20] Gonzalez, R. C., Woods, R. E. Digital image processing (2nd ed.). Prentice Hall, 34(19), 2710–2800. (2002).

[21] Hassanpour, H., Yaseri, A., Ardeshir, G. Feature Extraction for Paper currency Recognition. In International symposium on signal processing and its applications (ISSPA), Sharjah, UAE. (2007).

[22] Kim, M., Kim, D., Lee, S. Face recognition using the embedded HMM with second-order block-specific observations. Pattern Recognition, 36(11), 2723–2733. (2003).

[23] Takeda, F., Nishikage, T Multiple kinds of paper currency recognition using neural network a3nd application for Euro currency. In IEEE International Joint Conference on Neural Networks (pp. 143–147).(2000).

[24] Vila, A., Ferrer, N., Mantecon, J., Breton, Garcia, J. F. Development of a fast and non-destructive procedure original and fake euro notes. Analytica Chimica Acta, 559, 257–263.(2006).

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(**To be attached in the dissertation/ project report**)

1	Name of the Candidate **(IN BLOCK LETTERS)**	**Indranil Paul**
2	Address of the Candidate	**paulmicky619@gmail.com** **Mobile Number :** **9830875293**
3	Registration Number	**RA1611003011225**
4	Date of Birth	**06/12/1997**
5	Department	**Computer Science and Engineering**
6	Faculty	**Engineering and Technology**
7	Title of the Dissertation/Project	**Counterfeit Currency Detection Using Resource Efficient Neural Network**
8	Whether the above project/dissertation is done by	~~Individual~~ or group : (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : **2** b) Mention the Name & Register number of other candidates : **Arunabha Mittra** **RA1611003010781**
9	Name and address of the Supervisor / Guide	**Mrs. S. Sharanya** **Mail ID : Mobile Number :**
10	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID : Mobile Number :**

11	Software Used	Turnitin
12	Date of Verification	24/05/20
13	**Plagiarism Details: (to attach the final report from the software)**	

Chapter	Title of the Chapter	Percentage of similarity index (including self citation)	Percentage of similarity index (Excluding self citation)	% of plagiarism after excluding Quotes, Bibliography, etc.,
1	Literature Survey	6		
2				
3				
4				
5				
6				
7				
8				
9				
10				
Appendices				

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Indranil Paul

Signature of the Candidate

Name & Signature of the Staff (Who uses the plagiarism check software)

Name & Signature of the Supervisor/Guide

Name & Signature of the Co-Supervisor/Co-Guide

Name & Signature of the HOD

Format - I

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES

(To be attached in the dissertation/ project report)

1	Name of the Candidate **(IN BLOCK LETTERS)**	Arunabha Mittra
2	Address of the Candidate	arunabha.mitra97@outlook.com **Mobile Number :** 9007273283
3	Registration Number	RA1611003010781
4	Date of Birth	01/05/1997
5	Department	Computer Science and Engineering
6	Faculty	Engineering and Technology
7	Title of the Dissertation/Project	Counterfeit Currency Detection Using Resource Efficient Neural Network
8	Whether the above project/dissertation is done by	~~Individual~~ or group : (Strike whichever is not applicable) a) If the project/ dissertation is done in group, then how many students together completed the project : 2 b) Mention the Name & Register number of other candidates : Indranil Paul RA1611003011225
9	Name and address of the Supervisor / Guide	Mrs. S. Sharanya **Mail ID : Mobile Number :**
10	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID : Mobile Number :**

11	Software Used	**Turnitin**		
12	Date of Verification	**24/05/20**		
13	**Plagiarism Details: (to attach the final report from the software)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	Literature Survey	6		
2				
3				
4				
5				
6				
7				
8				
9				
10				
Appendices				

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Arunabha Mitra **Signature of the Candidate**	**Name & Signature of the Staff (Who uses the plagiarism check software)**
Name & Signature of the Supervisor/Guide	**Name & Signature of the Co-Supervisor/Co-Guide**

Name & Signature of the HOD

counterfiet

ORIGINALITY REPORT

6%	0%	6%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Counterfeit Currency Detection using Resource Efficient Neural Networks", International Journal of Innovative Technology and Exploring Engineering, 2020 Publication	6%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

Counterfeit Currency Detection using Resource Efficient Neural Networks

Arunabha Mittra, Indranil Paul, S Sharanya

Abstract*:** ***One of the leading causes of economic instability is the large-scale counterfeiting of the paper currency notes. Several media reports bring to light the alarming cases and the humungous scales of currency counterfeiting and how this issue has become very serious now. A report on how the Government is coping with these threats with new and stricter rules however counterfeiters adapt to the new rules in an alarmingly fast pace. Criminals continue to find a loophole in the system despite such strict security features. There have been impressive discoveries in the field of counterfeit currency, and this coupled with new age digital technology, counterfeiting is being fought well. However, it is impossible to track all counterfeit notes and impossible to have them checked at a short amount of time. Existing systems involve filing a case with the police, sending the documents for verification and waiting for the results to come. 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 machine in identifying a fake currency note in real time through an image of the same. Traditional Deep Learning algorithms require tremendous amount of compute power and storage and hence it is an expensive and elaborate process. The main goal is to make a faster and simpler mechanism to detect a counterfeit note that can be implemented in any random place like an ATM dispenser or an android application. The success of this application will greatly help the quick identification of the threat and help law enforcement in finding the source of the threat faster.

Keywords: Counterfeit Currency, One-Shot Neural Networks, Deep Learning.

I. INTRODUCTION

Currency is the backbone of a functioning society. Although the same society is moving towards cashless electronic transactions, India being a country of vast Socio-Economic backgrounds depend on cash majorly. India is also considered as a farmer intensive country where the farmers rely on cash to make their trade and business and runt their livelihood. Therefore, curbing the circulation of counterfeit notes by fast detection and blocking them at the source is very important. Also, both the paper authors being from the state of West Bengal which is a border state which has a considerable number of terrorists crossing over from neighboring countries and using counterfeit notes to fund their terror operations in the country and hence the check for counterfeit is a dire need for the welfare of this country in the broader aspect. Previous approaches have mainly relied solely on Image detection techniques and deep learning algorithms individually to classify the currency notes and has made it a computation intensive or space intensive process. Some approaches are unable to remove some features associated which get associated with the authentic note as a result of wear and tear and hence get declared as counterfeit and hence it is not the case. The whole idea of making it a faster and less resource hungry process is by applying Neural Networks which use minimal data sets and hence easier to build and modify on the go. This is done by dividing a single note into many features such as the Face of Gandhi, the font of the English letters and the font of the Hindi letters, the shade of the Numeric note in the corner and the texture and various shapes made in the background. After detecting these features, various filter will be applied to the nodes and detect feature points from it and train the neural network. The neural network after training will learn the features of an authentic note and on introduction of a new unknown note will be able to check for counterfeit or authentic.

II. STATE OF THE ART (LITERATURE SURVEY)

Various papers have been published to find new innovative ways to check for counterfeit. Some have used normal image detection techniques using MATLAB others have used basic machine learning techniques to check for the pictures of the image as a whole. Others have tried to train the model for single note, and have it decided on counterfeit or authentic based on a single feature. "Recognition of Fake Currency Note using Convolutional Neural Networks by Navya Krishna G, Sai Pooja G, Naga Shri Ram B. The CNN helps detect the most common patterns in the said authentic notes from the VGG-16 model dataset and using preprocessing, image rescaling, image shearing and perspective transformation to train the model and to have it detect the counterfeit notes [1]. "A Neural Network based Model for Paper Currency Recognition and Verification" by Angelo Frosni, Marco Gori and Paolo Priami have tried using multi layered perceptrons and have not been very successful with it. Their second approach has been an auto-associator model and the third has been a Pyramidal Multi-layered Neural Network. The paperwork has mentioned its own disadvantages of being slow but their technique is the cheapest so far [2]. Paper based on "Fake Currency Detection using Image Processing" by Ms Monali Patil, Jayant Adhikari, Rajesh Babu have proposed usage of SVM and KNN machine learning techniques to make a supervised and an unsupervised learning approach to make non-parametric as well as regressional analysis to have done on the data set chosen. The results have been compared with each other and found out that SVM is a better algorithm and it is actually two times the accuracy of KNN but none of them have been able to classify an image having a food stein or a fold mark which are not signs of counterfeit currency.

Revised Manuscript Received on April 1, 2020.
Arunabha Mittra, SRM univercity, Kattankulathur, Chennai Tamilnadu.
Indranil Paul, SRM univercity, Kattankulathur, Chennai Tamilnadu.
S Sharanya, SRM univercity, Kattankulathur, Chennai Tamilnadu.

Having Mean error rate as 42% [3]. “Detection of Indian Counterfeit Banknotes Using Neural Network” by E. Haripriya and K. Anusdha has proposed treating the new image through a process called Smoothening to remove all the noise and convert it to grayscale from RGB. After converting it is expected to extract features like Black stripes and Identification marks to check for genuine or counterfeit with efficiency around 72% [4]. A paper based on “Paper Currency recognition method by a Small Size Neural Network with Optimized Masks by GA” by Fumiaki Takeda, Sigeru Omatu, Salzo Onami, have used the concept of slots to divide the whole note into a set of pixels like a matrix and cover parts of the matrix to make images and then make them as the training images for a Small Size Neural Network and have efficiency around 71% [5]. “Fake Currency Detection using image processing” by Tushar Agasti, Gajanan Burand, Pratik Wade and P Chitra from VIT Vellore, have proposed using simple MATLAB to classify notes based on their denomination and also check for counterfeit. Their work is based on Edge Detection, Image Segmentation, Characteristic Extraction, Calculation Intensity and finally concluding whether the note is fake or not. Their work mentions that classifier can only detect images having a minimum intensity threshold of 70% and more [6]. A research paper on “Machine Dating of Hand Written Manuscripts” by Utpal Garain, SK Parui, T Paquet and L Heutte has also been chosen as one of our reference papers to consider the sign of the RBI governor as a feature to be possibly extract and have the model decide whether the note as a whole is authentic or not as the printer finds it difficult to print the signature the exact manner found in the original note. Construction of fundamental hypothesis of the training model and tried out tests like Hypothesis Test and Multinomial Chi-Square Test to detect the authentic handwritten image. The paper has reported a mean efficiency of 62% [7]. “Using Hidden Markov Models for Paper Currency Recognition” By Hamid Hassanpur and Payam M Farahbadi proposes using a preprocessing filter by finding local mean and variance and then going for feature extraction like Size, Colour Histogram and Texture Based Feature Extraction and ending it with Grayscale quantization and ultimately having the accuracy of 82% of detecting the counterfeit note [8]. A paper on “Counterfeit Currency Note Detection Using Deep Learning” by Soo Hyeon and Hae Lee also proposes Neural Networks detecting the counterfeit notes but uses CYMK extraction instead of RGB because CYMK is the parameter for the printing of anything in the world and only soft copy images and classified by RGB. He also proposes a non-Machine Learning technique by studying the light refracted from the shiny strip of the note. The paperwork has shown a mean accuracy level of 54% [9]. “Machine Assisted authentication of paper currency: An experiment on Indian Banknotes” by Murdoch, S.J Laurie have proposed similar feature extraction of features like printing technique, ink properties and the artwork used in the foreground and background to check for counterfeit notes and making them as parameters on which counterfeit is decided. The paperwork have has claimed an accuracy of 72% on their model [10]. A majority of these processes are based on analyzing and capturing the note as a whole and then applying filters to it. As a result, the food steins and fold marks have been also taken into account during the processing of the image and are factored in as un-removable noise which affects the result of the model. Additionally, there have been no restrictions on compute power and space availability on any of the models. A model will be developed which will provide best possible results in limited and reasonable amount of resources such that it can be incorporated into quick and essential processes of everyday life.

III. PRPOSED WORK

A. Abbreviations and Acronyms

ANN – Artificial Neural Network, CNN – Convoluted Neural Network, SNN – Siamese Neural Network, ReLU – Rectified Linear Unit, LBS – Linux Based Server

B. Equations

- Non-negativity: $\delta(x,y) \geq 0$
- Identity of Discernible: $\delta(x,y) = 0 \iff x = y$
- Symmetry: $\delta(x,y) = \delta(y,x)$
- Triangle inequality: $\delta(x,z) \leq \delta(x,y) + \delta(y,z)$

$$\delta(x^{(i)}, x^{(j)}) = \begin{cases} min \ \| \mathrm{f}(x^{(i)}) - \mathrm{f}(x^{(j)}) \| , i = j \\ max \ \| \mathrm{f}(x^{(i)}) - \mathrm{f}(x^{(j)}) \| , i \neq j \end{cases}$$

This form also allows the Siamese network to be more of a half-twin, implementing slightly different functions.

$$\text{if } i = j \text{ then } \delta\left[\mathrm{f}\left(x^{(i)}\right), \mathrm{g}\left(x^{(j)}\right)\right] \text{ is small}$$
$$\text{otherwise } \delta\left[\mathrm{f}\left(x^{(i)}\right), \mathrm{g}\left(x^{(j)}\right)\right] \text{ is large}$$

The letter i and j are indexes into a set of vectors. The functions f(.) and g(.) are implemented by the half twin network. δ(.) function implemented by the network joining outputs from the One-shot Neural Network.

IV. IMPLEMENTATION(ONE SHOT NEURAL NETWORK)

The neural networks used before have been very large neural networks and have taken days of training and good amount of compute power to train them and a huge amount of data has been collected and stored in servers. The approaches that have not used CNN have been unable to ignore food stein marks, fold marks and wear and tear marks on a note and is using that as a measure to regard it as a counterfeit note. So, using a CNN is very crucial. The platform we will be using is Python and the Libraries we will be focusing on are OpenCV, Matplotlib and most importantly TensorFlow. Our OpenCV will mainly have implement native and custom harrcasscades trained via a Linux Based Server with positive and negative images which make the harrcasscades to have them detect certain features. A separate classifier has to be made for making each feature and then the model will feed each feature into our Neural Network. Before these go into the NN they have to be passed through various custom filers which will selectively take them in combination and return with multiple results. The fact of the matter is the neural networks used before have been very large neural networks and have taken days of training and good amount of compute power to train them and a huge amount of data has been collected and stored in servers. The approaches that have not used CNN have been unable to ignore food stein marks, fold marks and wear and tear marks on a note and is using that as a measure to regard it as a counterfeit note. So, using a CNN is very crucial.

A. Existing Work

All the existing work in this domain have either used MATLAB only or python only to determine the authenticity of the note and have mention facing trouble during old notes detection as The paperwork have food marks or fold marks which is also being detected as a feature which the paperwork is unable to remove. The other researchers have directly gone for neural networks where the paperwork is training from the note as a whole after image filtering and pre-processing. Our goal is to define our own features and detect only them from the note so that the neural network can ignore the unnecessary food marks and fold marks

B. Feature Extraction

Initially the whole currency note has been divided into features. Using OpenCV to initially detect the feature off the note, the normal full-scale image has been scaled down to a 20 X 20 image to make it easier to train the harrcascades. After the harrcascades have been trained the live image detects the features from the note and have been stored in a repository. These act as the input values for the input function in the SNN. The image by default has been converted to greyscale to make it easier for the model to learn.

C. Siamese Neural Networks

After the features have been detected via OpenCV these are fed in to the SNN independently. There will be a separate SNN for English font, a separate SNN for Hindi font, similarly for Gandhi's face signature of the Governor and the color-shade of one side of the note as well. Each model takes a minimum of 10 minutes or less or a maximum of an hour to train and can be successfully used them henceforth. Depending on the result of each SNN, a conclusion arises whether the final note is fake or authentic. As a result, any chance of a weak train on any feature can choose to omit that feature and determine the result via the other features.

V. RESULTS DISCUSSION

A. Data sets

As the whole idea of using this type of Network is to reduce the size of the data set and shorten compute time immensely. The data-set being used is roughly not bigger than 2 or 3 images.

B. Tools Used

Tools used in this process has primarily been OpenCV on Python platform on a LBS run by PUTTY Client Service, NSCP to access the File System of the server and Tensorflow mainly to train and run the SNN.

C. Result

The final model is being able to detect a counterfeit note by 75% efficiency but on using a note which has undergone extreme wear and tear efficiency is down to 40%, although such notes are often taken out of circulation by the Government itself and hence it can be ignored.

D. Validation Charts and Screen-shots

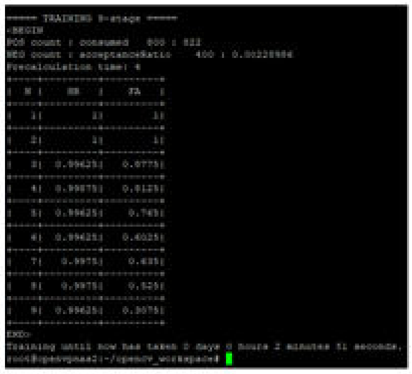


Figure 1: Training Harrcascades on LBS

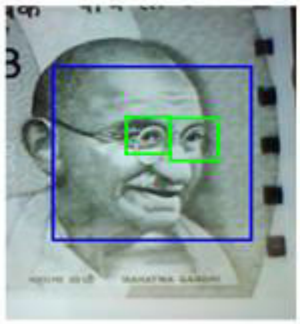


Figure 2: Feature Detection

Figure 3: Colour Extraction

VI. CONCLUSION

The one-shot deep learning approach to the problem statement is unique because it reduces the time and compute power requirement to detect the counterfeit note and help saves time and the tedious process that is usually taken in a conventional bank to determine it through thorough examination and also in a large complicated Neural Network Thus, the counterfeit detection will help boost the economy and also stop terrorists from funding their own operations. The future aspects of this project can be to implement it in the Android Device or a phone so that notes can be checked on the go and the paperwork can prevent fraud from happening instantly and saving many businesses and livelihoods from financial threat.

REFERENCES

1. Tushar Agasti, Gajanan Burand, Pratik Wade and P Chitra," Fake Currency Detection Using Image Processing," IOP Conf. Series: Materials Science and Engineering 263 (2017) 052047 doi:10.1088/1757-899X/263/5/052047.
2. F.Takeda, S. Omatu,T. Inoue, and S. Onami, "Bill Money Recognition Using Neural Network with FFT as Pre-processor, " Trans. on ISCIE of Japan, Vo1.5, No.7, pp.265-273, 2010.
3. E.Haripriya, K. Anusdha, "Detection of Indian Countefeit Banknotes Using Neural Networks" Volume 04, Issue 12; December- 2018 [ISSN: 2455-1457].
4. Ms. Monali Patil, Jayant Adhikari, Rajesh Babu, "Fake Currency Detection using Image Processing," International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248 Volume: 4 Issue: 4.
5. Navya Krishna G, Sai Pooja G, Naga Sri Ram B, Yamini Radha V, Rajarajeswari P, "Recognition of Fake Currency Note using Convolutional Neural Networks," International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-5 March 2019"
6. Angelo Frosini, Marco Gori, Member, IEEE, and Paolo Priami, "A Neural Network-Based Model for Paper Currency Recognition and Verification," IEEE transactions on Neural Networks, Vol. I, No 6, November 2006.
7. Utpal Garain, S. K Parui, T. Paquet, L. Heutte, "Machine Dating of Handwritten Manuscripts,"
8. Hamid Hassanpour , Payam M. Farahabadi, "Using Hidden Markov Models for paper currency recognition", Expert Systems with Applications 36 (2009) 10105–10111.
9. Soo-Hyeon Lee and Hae-Yeoun Lee," Counterfeit Currency Note Detection Using Deep Learning," International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 304-310.
10. Murdoch, S.J Laurie," Machine Assisted authentication of paper currency: An experiment on Indian Banknotes" IEEE Trans. Pattern Anal. Mach. Intell. 10(6), 978–983 (2008).
11. N. Akamatsu, M. Fukumi, "A method to design a neural pattern recognition system by using a genetic algorithm with partial fitness and a deterministic mutation" 11(4), 922–984 (2009).
12. A. Chowdhury, N. Jahangir, Bangladeshi banknote recognition by neural network with axis symmetrical masks in masks, in: 10th international conference on computer and information technology. IJTEE 2010
13. Da-costa, Multiview banknote recognition with component and shape analysis IJTEE 2014
14. S. Das, CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more, IJTEE 2017.
15. P.R. Nagpure, T.Ghotkar, S. Shetty, Recognition and Fake Note Detection, International Journal of Innovative Research in Computer and Communication Engineering Vol 4 (March 2016).
16. T. Nishikage, F. Takeda, Multiple kinds of paper cur- rency recognition using neural network and applica- tion for Euro currency (2000) 27–47. 11. S. Omatu, F. Takeda, High speed paper currency recognition by neural networks (1995) 73–77.
17. Paisios, Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms (2012)
18. J. S.Dewan, Extended Local Binary Pattern for Face Recognition, Technology ICAET, 2015
19. . V.Vashishtha, M.Sadim, A Paper Currency Recogni- tion System Using Image Processing To Improve the Reliability with PCA Method, International Journal of Engineering Science & (2015).
20. Gonzalez, R. C., & Woods, R. E. Digital image processing (2nd ed.). Prentice Hall. (2002).
21. Hassanpour, H., Yaseri, A., Ardeshir, G. Feature Extraction for Paper currency Recognition. In International symposium on signal processing and its applications (ISSPA), Sharjah, UAE. (2007).
22. Kim, M., Kim, D., & Lee, S. Face recognition using the embedded HMM with second-order block-specific observations. Pattern Recognition, 36(11), 2723–2733. (2003).
23. Takeda, F., Nishikage, T Multiple kinds of paper currency recognition using neural network and application for Euro currency. In IEEE International Joint Conference on Neural Networks (pp. 143–147).(2000).
24. Vila, A., Ferrer, N., Mantecon, J., Breton, D., & Garcia, J. F. Development of a fast and non-destructive procedure original and fake euro notes. Analytica Chimica Acta, 559, 257–263. (2006).
25. Hassanpour, H., Mesbah M. Neonatal EEG seizure detection using spike signatures in the time-frequency domain. In IEEE (2013).

AUTHORS PROFILE

Arunabha Mittra, SRM univercity, Kattankulathur, Chennai Tamilnadu.

Indranil Paul, SRM univercity, Kattankulathur, Chennai Tamilnadu.

S Sharanya,SRM univercity, Kattankulathur, Chennai Tamilnadu.

International Journal of Innovative Technology and Exploring Engineering

ISSN: 2278-3075 (Online) | Exploring Innovation | A Key for Dedicated Services

Published by: Blue Eyes Intelligence Engineering and Sciences Publication

G18-19-20, Block-B, Tirupati Abhinav Homes, Damkheda, Bhopal (Madhya Pradesh)-462037, India

Website: www.ijitee.org **Email:** submit2@ijitee.org

CERTIFICATE

This certifies that the research paper entitled '**Counterfeit Currency Detection using Resource Efficient Neural Networks**' authored by '**Arunabha Mittra, Indranil Paul, S Sharanya**' was reviewed by experts in this research area and accepted by the board of 'Blue Eyes Intelligence Engineering and Sciences Publication' which has published in **'International Journal of Innovative Technology and Exploring Engineering (IJITEE)'**, ISSN: 2278–3075 (Online), Volume-9 Issue-6, April 2020, Page No. 220-223.

The Value of Citation (VoC): of IJITEE is 6.03 for year 2019. Your published paper and Souvenir are available at: https://www.ijitee.org/download/volume-9-issue-6/

Blue Eyes Intelligence Engineering and Sciences Publication BHOPAL

Jitendra Kumar Sen
(Manager)

Dr. Shiv Kumar
(CEO)