International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

DETECTION OF FORGERY IN HANDWRITTEN SIGNATURES

Monu Agrawal
19BCE2531
School Of Computer Science and Engineering
VIT, Vellore
Tamil Nadu, India
monuagrawal1011@gmail.com

Riwaz Udas
19BCE2532
School Of Computer Science and Engineering
VIT, Vellore
Tamil Nadu, India
udasriwaz@gmail.com

Pranav Kumar Jha
19BCE2579
School Of Computer Science and Engineering
VIT, Vellore
Tamil Nadu, India
jhapranav6418@gmail.com

Abstract—In our life, handwritten signatures are essential. Signatures are used to identify people in organizations like banks and institutions. But there are several difficulties associated with signatures since even two written by the same individual may resemble one another very little. It becomes quite challenging to distinguish between authentic and fake signatures as a result. Forgery detection systems, in conjunction with the ideas of machine learning and CNN, are a solution to this issue to prevent any such identity frauds conducted in banks and many other businesses. The program is implemented using parallelization ideas for improved speed and time economy. This application may be used to verify signatures on a variety of platforms, including applying for loans, signing legal documents, applications, and much more.

Index Terms—parallelization, CNN, signatures, authentic

I. Introduction

A. Objective

The objective of this research paper is listed below:

- To verify if signature is forged or original.
- To ensure authorized use of confidential information

B. Background

The Convolutional Neural Network will be the central idea for this system. The CNN will be trained using a dataset with a large number of signatures so that it can learn to anticipate specific traits and determine whether or not a forgery has occurred. Our goal is to develop software that validates signatures and ensures it is more dependable, effective, and accurate than current solutions. As a person ages, their sig-natures change throughout time. The signature can alter for a variety of reasons that are indistinguishable to the average person. This program may be used to verify signatures on a variety of platforms, including applying for loans, signing legal documents, applications, and much more. Many businesses have suffered significant financial losses as a result of a single fake, therefore being able to spot one may help the business save money, time, and its reputation. As people of different skills are intended to use this system even people who have low technical skills or knowledge, the system will be easy to use and will not require complex tasks in order to use efficiently.

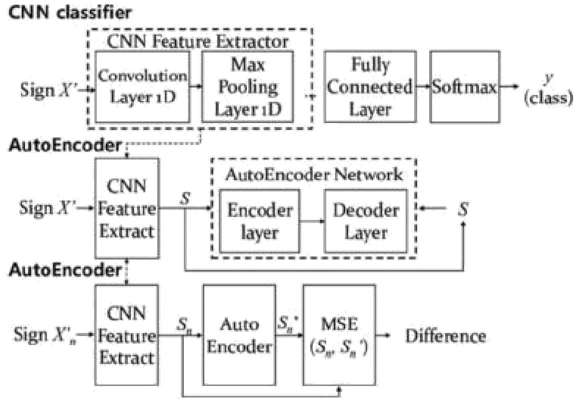
II. Proposed Methodology

The methodology that we are going to take in the project is by implementing a two-step method. The first step involves the use of convolutional neural network. We will first train the base model with convolutional neural network then further use ResNet to add on to the existing model by adding more detailed statistics into the final model. As CNN becomes more complex for images with more features we have use ResNet to supplement it in terms of helping in simplifying the process of training the model.

A. Convolutional Neural Network

The ability of a convolutional neural network (CNN), a multi-layer neural network with a deep supervised learning architecture, to extract features for classification on its own is well recognized. An autonomous feature extractor and a trainable classifier are the two components of CNN. The feature extractor uses convolution filtering and down sampling, two procedures, to extract features from the input data. The trainable classifier generates the classification results after being trained using a back-propagation method with a fully connected layer based on these characteristics.

A CNN is used in the suggested strategy as both a feature extractor and a classifier. The suggested CNN-AE model’s architecture is depicted in the figure below. The CNN extraction procedure is a black box, just like other deep neural networks, and the precise details of the features are yet unknown. We assume that a trained CNN may extract useful features for differentiating behavior traits of forgery, such as reluctance and delay before drawing the difficult section of a signature, if the CNN is taught for identifying forged and authentic signatures. Consequently, a feature vector known as the S-vector is created from the output of the CNN feature extractor (denoted by S in figures and equations). The input to an AE for creating the subject mode is the S- vector.

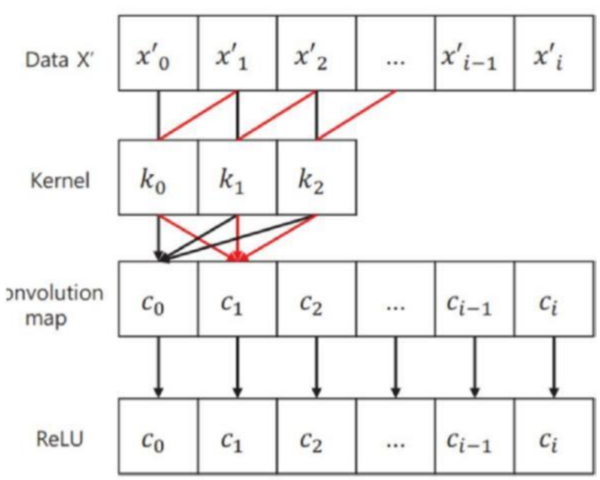


The convolution layer in the CNN feature extractor takes the value X' as an input. Equation below, where k is a kernel (also known as a filter) and l is the number of kernels employed, defines each node ci of the convolution layer.

$$c_i = \sum_{j=0}^{l} k_j \times x'_{i+j}$$

The activation function ReLU stated in the equation below takes the output of ci as input. A convolution map is created from the activation results. Below is a diagram illustrating these convolution layer processes.

$$ReLU(c_i) = max(0, c_i)$$



1) Drawbacks of Convoluted Neural Networks: Deep net-works have the advantage of representing numerous com-plicated functions and learning features at various levels of abstraction, moving from edges, which are often a component of much lower layers, to sophisticated features that exist in the deepest layers. However, a significant obstacle when employing deep networks is that when we back propagate from the final layer to the first layer, the gradient rapidly and exponentially declines to zero. In extremely unusual circumstances, it might abruptly and swiftly increase or erupt into very huge levels.

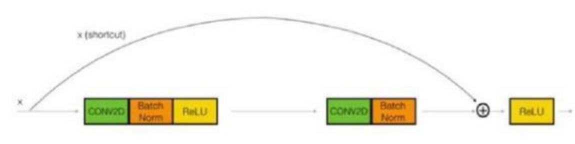
2) Solution to the Problem: ResNet, which functions as a shortcut or essentially omits some stages or connections, is ultimately necessary to get around this since it enables the gradient to be immediately backpropagated, minimizing the likelihood that it will fast collapse to a very tiny value.

B. ResNet

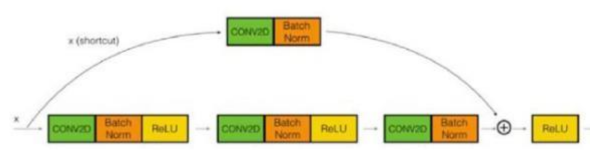
Implementing ResNet, a kind of neural network, can im-prove system functionality because of the phenomenon of gradient decrease.

Residual networks help build deeper neural networks which is important because it helps in avoiding the degradation of the accuracy and the error rate of the handwritten signatures. As you progress through the levels, the matrix's weights continue to multiply. Batch normalization is a technique for improving the speed and performance thus allowing the layer to train better. Based on how different or similar are the ResNet blocks, they are divided into two types:

- The Identity Block



- The Convolution Block



III. Literature Survey

A. Image splicing forgery detection combining coarse to re-fined convolutional neural network and adaptive clustering

The two-part splicing forgery detection approach is exam-ined in this research. A coarse-to-refined convolutional neural network (C2RNet) and a diluted adaptive clustering network make up the two components. The discrepancies in the image are discovered in the proposed model by cascading a coarse CNN and refined CNN (C-CNN and R-CNN respectively).

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Scales where the image has been altered are created because of the cascading, which reveals differences in their character-istics. Instead of using a patch level CNN into C2RNet, an image-level CNN reduces the computational cost of the entire model. Because the differences in qualities are compared, the outcomes are stabilized. Additionally, the CNN adaptive clustering is used to discover the forged regions following a preliminary detection. It was discovered that the suggested strategy, even under assault, outperforms the currently avail-able splicing techniques for forgery detection.

B. Offline signature verification and forgery detection using fuzzy modeling

This paper investigates an automatic signature verification that has many applications in the financial and business sec-tors. It is a vast field of study that is currently being researched. This model suggests a method for offline, automated forgery detection and signature verification. It is founded on a fuzzy model strategy that also makes use of the Takagi-Sugeno (TS) model. A box technique is utilized to extract the angle aspects of the sign to carry out the signature verification. There is a corresponding fuzzy set for each feature. The characteristics used in the TS model are then exponentially fuzzed. This produces a better model by accounting for the many structural variations that are present because of handwriting styles and signee's mood.

C. OFFLINE SIGNATURE FORGERY DETECTION USING CONVOLUTIONAL NEURAL NETWORK

The author of this paper put out a CNN-based system for signature identification and verification. Using a dataset of 150 people, each with 5 signatures, for a total of 750 signatures, this method constructs a knowledge base by extracting special features for signatures. The signatures go through geometric changes and are made into grayscale images. Using a CNN with numerous convolutional, pooling, and fully connected layers to obtain the output, the features of these pre-processed signatures are retrieved and compared with those in the system to ascertain if they are authentic or fraudulent.

D. Digital signature Forgery Detection using CNN

In this paper ,the author's goal is to give an analysis of the techniques for determining the authenticity of visual media, specifically the ways for identifying counterfeit signature images. Active and passive approaches are frequently used to detect fake digital images. The authors have investigated proactive strategies for spotting fake digital images. A key method for ensuring the validity of digital images utilizing ac-tive techniques is the use of digital signatures and watermarks. The most important way to confirm a signature's legitimacy is by its signature, which is made up of a variety of distinct qualities. It can operate via offline or online techniques. The signature of a current query is compared to reference signatures that have already been written in offline techniques to confirm a person's identification. They took the dataset of 2000 images which combine both the forged and the original image and RGB format was used. Then the CNN model was trained using the data with three input layers and with different weights and biases and 3 hidden layers with a set of neurons and an output layer to show the final output. They applied softax to find or calculate the accuracy. The highest accuracy found in that paper was 99.7 percent by the author and average accuracy was 97.8 percent.

E. HANDWRITTEN SIGNATURES FORGERY DETECTION

The author of this paper stated that the paper's goals were to understand the features of signatures, determine if a signature was genuine or fake, and create a system to do so. Even though highly qualified specialists can spot signature fraud, great precision cannot be obtained due to the numerous variances in handwriting styles and professionalism of forgers. A high-accuracy signature verification and distinction between a genuine and forgery signature can be greatly improved by automatic recognition systems. In the suggested approach, a CNN serves as both a feature extractor and a classifier. Con-volution filtering and down sampling are used by the feature extractor to extract features from the input data. According to the authors, if a CNN is trained to discriminate between fake and real signatures, it will be able to extract useful information that can be used to identify forgery-related behaviors, such as sketching intricate parts of a signature slowly or hesitantly. Even though deep networks may learn features at many levels of abstraction and represent complex functions, it will become problematic if the gradient rapidly approaches 0 while the backpropagation from the last to the first layer continues.

F. Writer-independent Offline Handwritten Signature Verifica-tion using Novel Feature Extraction Techniques

Their primary goal in this paper's signature verification is to distinguish between authentic signatures and fake ones. The work of identifying an original signature was difficult since even two signatures belonging to the same individual might differ in a number of ways, including their beginning and finishing locations, angle of inclination, relative spacing between characters, height, and breadth. The difficulty of offline signature verification is increased by the absence of dy-namic information about the signing process. Offline signature verification has been the subject of various research projects over the years, however the issue is still unresolved. Image preprocessing, feature extraction, and verification are the three stages that are frequently utilized in signature verification sys-tems. Two unique characteristics that may be recovered from preprocessed signature photos during the feature extraction stage are provided in this work. I) Stroke angle and average intersected spots are the features that are suggested. ii) The signature nucleus' pixel density. The purpose of this research is to improve the feature set with the suggested characteristics, which will aid in more precise signature verification.

G. Offline Signature Recognition and Forgery Detection using Deep Learning

The paper begins with a brief description to define the im-portance of signature verification and need for increased secu-rity when handling issues in regard with signature authentica-tion and integrity. The paper implies the use of Convolutional Neural Network with the use of Crest-Trough method along with SURF algorithm and Harris corner detection to perform necessary actions to perform verification. It is necessary to remove as many factors in a signature as possible to increase the efficiency and potency of the training algorithm to ensure the classification is properly trained to be later tested. CNN involves a series of assorted layer where the input images are fed into the classifier. Crest Trough is another important step in the process as it analyzes each range and magnitude of the crest and trough. The steps that are carried out in paper are grey scale removal followed scaling images to have consistent sizing then comes noise removal along with rotation, trimming and finally centralization and cropping. After all these steps are completed, the model is trained using which signature can be verified. The verification process is done by using SURF and Harris which compare the degree of sharpness of every curve and corner to determine the authenticity.

H. Performance evaluation of handwritten signature recognition in mobile environments

Biometric has been booming in use in mobile devices. With the added benefit of ease of use and functionality of touch in mobile device makes it more probable to be used in mobile devices. For this very reason the paper consists of evaluations conducted to determine the efficiency and secu-rity of handwritten signature in mobile devices. Handwritten signatures have an added drawback of changing over time which makes it increasingly difficult to verify mood or age can easily change the signature making it unable to be read easily. Human error increases the model’s efficiency as the signature can be different which may alter the final model. The paper describes different algorithms to verify signatures like HMM, NN, DTW, etc. The paper includes five different devices used in performing the evaluation of the signatures. To remove the number of external factors affecting the results necessary preprocessing measures were taken. The environment was kept constants for all the tests. Post collection of necessary signatures, each signature was evaluated after it was fed into the DTW model. After the model was completed, another signature was taken later to compare the results. Finally, all the results were complied to get a full evaluation of test.

I. Signature Verification: A Study

Signature is an important form of identification. It is one of the most basic and foremost forms of identification along with fingerprint and DNA. Similarly, verification is the process of identifying whether a signature belongs to a person or not. The process involves four basic steps namely Data Ac-quisition, Preprocessing, Feature Extraction, Enrollment, and verification. First, the paper describes the different ways in which signature can be forged. They are random, simple, and skilled. Random forgery is performed without any idea of how the signer does it nor do they know the name. Simple is the type of forgery in a simple copy is done and skilled is done by proper replication which makes it incredibly difficult to detect. Data Acquisition is the step in which signature is collected to train a model which is collected with scanners and cameras. Preprocessing is the step where all pictures are edited to remove noise and make them of the same size to make it easier to analyze. Feature extraction is the phase where each characteristic is analyzed and recorded to be used as model training source. The final step is done using the collected data where the data is passed through a Neural Network to be trained. The trained model is then used to test the final signature to determine the authenticity.

J. Machine Learning for Signature Verification

Signature Verification is the most common task that is con-sidered in the field of forensic document analysis. Signature presents itself as one of the most basic forms of identification and it can be used to track any person. A signature is unique to any person and hence can be considered a sustainable form of verification. But today even something as sustainable as a signature has fallen victim to forgery and duplication. For this very reason the paper introduces a machine learning based forgery detection system. The proposed system utilizes a quasi-multiresolution based idea to determine a signatures authenticity. It uses a mixture of Gradient, GSC as well as concavity features to design a suitable model on which a signature can be tested upon. The GSC feature works by designing a grid partition approach where the signature is spilt into multiple smaller images against which other feature extraction is done. The final model is made using person independent classification which compares a signature to be verified with one collected signature sample.

K. Handwritten Signature Forgery Detection using Convolutional Neural Networks

The paper defines the need of signatures in one’s life calling it a means to which a person is identified. A signature being same has a probability that is negligible making it one of the most advanced and trustworthy forms of identification. But even so the technology nowadays has made it possible for such signatures to be copied making the identity of people at risk. The paper has also defined three methods through which a signature can be properly verified namely form, movement and variation. Using these three features the paper defines a system which involves taking samples from a user which is then analyzed by the CNN algorithm after preprocessing which uses a grid-based approach to convert the sample signature to a model that can be used to compare future signatures to verify it.

L. Off-line signature verification and forgery detection using fuzzy modelling

The suggested method is based on fuzzy modelling that uses the Takagi-Sugeno (TS) model. Using angle characteristics recovered from the box technique, signature verification and fraud detection are performed. The TS model, which has been extended to add structural parameters, uses an exponential

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

membership function to fuzzify the features. The structural characteristics are designed to convey moods and adjust for any variances that may occur owing to handwriting styles. In the TS model, the membership functions serve as weights. The solution for the parameters is obtained by optimizing the TS model's output in relation to the structural parameters. Both aspects for detecting forgeries and verifying signatures are included in the suggested system.

M. Signature Verification using Geometrical Features and Artificial Neural Network Classifier

Since signature photos lack a lot of texture, they include a lot of important geometrical data. In this, the authors have put forth a straightforward yet efficient technique of signature verification. The method described in this study uses an Artificial Neural Network (ANN) classifier to leverage the geometrical characteristics of a signature picture, such as its Centre, isolated points, connected components, etc., and classifies the signature image according to these characteris-tics. It uses a clustering-based network for signature verifi-cation. These characteristics have been used to apply fuzzy c mean and Kohonen Clustering Network (KCN) clustering techniques. To obtain the ideal learning rate, the learning rate has been updated using EFKCN. Decision tree, Naive Bayes, and Euclidean distance are three different statistical classifiers that have been applied.

N. Improving Presentation Attack Detection in Dynamic Handwritten Signature Biometrics

In this paper, it investigates the previous findings and accordingly two Presentation Attack Detection approaches have been put into practice. A new evaluation has been done with these implementations, and it shows an improvement in performance. Under operational conditions, error rates have dropped from around 20percent to under 3percent. All current international standards were followed in developing the evaluation platform. A Dynamic Time Warping (DTW) based dynamic handwritten signature verification system was assessed using such a platform. With all these factors consid-ered, adding dynamic information to the comparison process may be a technique to decrease the success of forgeries. To reject forgeries, this section suggests the addition of two extra metrics to the comparison process.

O. Real Time Signature Forgery Detection Using Machine Learning

The suggested approach is based on off-line signature ver-ification utilizing deep learning model including Convolution Neural Network (CNN) and new method for extracting local characteristics. This technique can be applied in a variety of settings where the number of people is limited so that the model can train the users before spotting future forgeries. It obtains vital and pertinent information for classification, suitable for recognition activities. This CNN can have up to 250 layers. Large amounts of labelled data and neural network designs that learn features from the input are used to create deep learning models without the need for human feature extraction. The datasets go through a few pre-processing steps. The model is then trained with the use of CNN and the VGG16 architecture, which categorises whether the image is forged or not.

P. GMM For Offline Signature Forgery Detection

This study introduces an offline signature verification method that extracts new local and geometric characteristics, such as the quadratic surface feature, the area-to-distance ratio, and others. For this, some real signatures were collected from five different people, and after carrying out the necessary preprocessing processes, information was retrieved and the characteristics from each sample. To create a reference model for each sample of a specific user's signature during the training phase, the Gaussian Mixture Model (GMM) approach is used. Acceptance range is determined by determining the Euclidian distance between the reference signature and each training batch of signatures. A query signature is recognized as an authenticated signature if its Euclidian distance falls within the acceptable range; otherwise, it is recognized as a forgery.

IV. Modules Description

- Image Input The testing images are given to the system as an image input.
- Pre-Processing The image is converted into a NumPy array for further processing. Data augmentation is done where the images are randomized to train the model to generate efficient and accurate results.
- Standardization Standardization makes the data like each other and gives them a common ground to work on. It involves re-scaling of the data.
- Feature Extraction The CNN is used to identify certain characteristics and patterns in pictures. As we move over an image, we effectively look for trends in that area. This works because of filters, which are stacks of weights expressed as vectors and multiplied by the results produced by the convolution. The pooling layer divides the input picture into a collection of rectangles and produces a value for each such sub-region. The idea behind this is that a feature's precise placement is less significant than its general location in relation to other features. Consequently, CNN is trained on a dataset and has the capacity to intelligently recognise certain characteristics for a test image.
- Passing Through Layers Below diagram shows the vari-ous layers present in the deep neural network stage wise. It includes the convolutional block and the ResNet. This creates a ResNet50 function which is used to establish the deep neural networks.
- Classification Two classes have been established. One class is called the forged and the other class is called the original. The test image will be classified under one of these classes.
- Output The output of the system will be displayed as given in the below diagram. The input image shape is

given and the class prediction vector to identify if the signature is forged or original.

V. Implementation

```python
import numpy as np
import os
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import matplotlib.cm as cm
from scipy import ndimage
from skimage.measure import regionprops
from skimage import io
from skimage.filters import threshold_otsu
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
import pandas as pd
import numpy as np
from time import time
import keras
```

Fig. 1. Imported Libraries

The modules we have primarily used are Keras, Numpy, Pandas, Tensorflow and Matplotlib. Keras is used to implement the modelling portion while we have used Pandas and Numpy to do complex arithmetic operations on a larger set like the dataset that we have. Finally TensorFlow is used in the pixel data extraction where we have extracted the data from the individual image files having the signature data within them.

```python
def rgbgrey(img):
    # Converts rgb to grayscale
    greyimg = np.zeros((img.shape[0], img.shape[1]))
    for row in range(len(img)):
        for col in range(len(img[row])):
            greyimg[row][col] = np.average(img[row][col])
    return greyimg
```

```python
def greybin(img):
    # Converts grayscale to binary
    blur_radius = 0.8
    img = ndimage.gaussian_filter(img, blur_radius)  # to
#     img = ndimage.binary_erosion(img).astype(img.dtype)
    thres = threshold_otsu(img)
    binimg = img > thres
    binimg = np.logical_not(binimg)
    return binimg
```

Fig. 2. Pre-processing of Images

After the model is created using CNN the model is then tested using out test dataset. The testing is done by giving the image as input to the program which will then preprocessed the image as done before when making the model. Finally the preprocessed image goes through a series of feature extraction which will then be tested against the existing model.

Feature Extraction

```python
def Ratio(img):
    a = 0
    for row in range(len(img)):
        for col in range(len(img[0])):
            if img[row][col]==True:
                a = a+1
    total = img.shape[0] * img.shape[1]
    return a/total
```

Fig. 3. Extracting image x and y coordinate ratios

```python
def Centroid(img):
    numOfWhites = 0
    a = np.array([0,0])
    for row in range(len(img)):
        for col in range(len(img[0])):
            if img[row][col]==True:
                b = np.array([row,col])
                a = np.add(a,b)
```

Fig. 4. Calculating the centroid value for the signature

```python
def EccentricitySolidity(img):
    r = regionprops(img.astype("int8"))
    return r[0].eccentricity, r[0].solidity
```

```python
def SkewKurtosis(img):
    h,w = img.shape
    x = range(w)
    y = range(h)
    #plt.plot(x,y)
    #plt.show()
    #calculate projections along the x and y axes
    xp = np.sum(img,axis=0)
    yp = np.sum(img,axis=1)
```

Fig. 5. Getting Skew and Kurtosis values from the image

```python
    #centroid
    cx = np.sum(x*xp)/np.sum(xp)
    cy = np.sum(y*yp)/np.sum(yp)
    #standard deviation
    x2 = (x-cx)**2
    y2 = (y-cy)**2
    sx = np.sqrt(np.sum(x2*xp)/np.sum(img))
    sy = np.sqrt(np.sum(y2*yp)/np.sum(img))

    #skewness
    x3 = (x-cx)**3
    y3 = (y-cy)**3
    skewx = np.sum(xp*x3)/(np.sum(img) * sx**3)
    skewy = np.sum(yp*y3)/(np.sum(img) * sy**3)
```

TF Model

```
def testing(path):
    feature = getCSVFeatures(path)
    if not(os.path.exists('E:\\project\\Dataset/TestFeatures')):
        os.mkdir('E:\\project\\Dataset/TestFeatures')
    with open('E:\\project\\Dataset\\TestFeatures/testcsv.csv', 'w') as handle:
        handle.write('ratio,cent_y,cent_x,eccentricity,solidity,skew_x,skew_y,kurt_x,kurt_y\n')
        handle.write(','.join(map(str, feature))+'\n')
```

```
n_input = 9
train_person_id = input("Enter person's id : ")
test_image_path = input("Enter path of signature image : ")
train_path = 'E:\\project\\Dataset\\Features\\Training/training_'+train_person_id+'.csv'
testing(test_image_path)
test_path = 'E:\\project\\Dataset\\TestFeatures/testcsv.csv'

def readCSV(train_path, test_path, type2=False):
    # Reading train data
    df = pd.read_csv(train_path, usecols=range(n_input))
    train_input = np.array(df.values)
    train_input = train_input.astype(np.float32, copy=False)  # Converting input to float_32
    df = pd.read_csv(train_path, usecols=(n_input,))
    temp = [elem[0] for elem in df.values]
    correct = np.array(temp)
    corr_train = keras.utils.to_categorical(correct,2)      # Converting to one hot
    # Reading test data
    df = pd.read_csv(test_path, usecols=range(n_input))
    test_input = np.array(df.values)
    test_input = test_input.astype(np.float32, copy=False)
    if not(type2):
        df = pd.read_csv(test_path, usecols=(n_input,))
        temp = [elem[0] for elem in df.values]
```

Fig. 6. Storing extracted values in CSV file and retrieving them to make mode using CNN and ResNet

```
    # Network Parameters
    n_hidden_1 = neurons # 1st layer number of neurons
    n_hidden_2 = 7 # 2nd layer number of neurons
    n_hidden_3 = 30 # 3rd layer

    train_avg, test_avg = 0, 0
    n = 10
    for i in range(1,n+1):
        if display:
            print("Running for Person id",i)
        temp = ('0'+str(i))[-2:]
        train_score, test_score = evaluate(train_path.replace('01',temp), test_path.replace('01',temp))
        train_avg += train_score
        test_avg += test_score
    if display:
#           print("Number of neurons in Hidden layer-", n_hidden_1)
        print("Training average-", train_avg/n)
        print("Testing average-", test_avg/n)
        print("Time taken-", time()-start)
    return train_avg/n, test_avg/n, (time()-start)/n


evaluate(train_path, test_path, type2=True)

Enter person's id : 001
Enter path of signature image : E:\fastrack\CSE1901 TARP\project\forged\021001_001.png
Forged Image

False
```

Fig. 7. Final Output

VI. Result and Discussion

The implementation above shows a brief working of the signature forgery detection system. The system first takes in the file path of the training set. First, each image is preprocessed to remove any form of noise, and color. Then it It then proceeds to calculate the values of ratio, centroid, kurtosis, skewness, and eccentricity for each of the images. Each person has a total of 20 signatures in the data set which it uses to design a suitable model using CNN. A total of 12 different user signatures have been modeled by the program which is then used to test sample test cases. If a signature that was of the user is added it results saying that the image is not forged, else it returns that the signature has been forged. Each of the data items is stored in a CSV file after pre-processing and feature extraction. The output of the program successfully shows the implementation of the signature forgery detection successfully.

Acknowledgment

The completion of this undertaking could not have been possible without the guidance and participation of a lot of people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. However, we would like to express our deep appreciation and indebtedness to Prof. Jayakumar K (dr.jayakumar1979@gmail.com) for her endless support and guidance throughout the semester without which the comple-tion of this project would be difficult.

References

[1] Xiao, Bin, et al. “Image Splicing Forgery Detection Combining Coarse to Refined Convolutional Neural Network and Adaptive Clus-tering.” Information Sciences, vol. 511, 2020, pp. 172–191., DOI: 10.1016/j.ins.2019.09.038.
[2] Hanmandlu, M., Yusof, M. H. M., & Madasu, V. K. (2005). Off-line sig-nature verification and forgery detection using fuzzy modelling. Pattern Recognition, 38 (3), 341–356. DOI: 10.1016/j.patcog.2004.05.015.
[3] Raj Balsekar, et al. ”OFFLINE SIGNATURE FORGERY DETEC-TION USING CONVOLUTIONAL NEURAL NETWORK”, Interna-tional Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.7, Issue 5, page no.13-18, May-2020.
[4] Digital signature Forgery Detection using CNN Lakkoju Chandra Ki-ran1, Gorantla Akhil Chowdary2, Manchala Shalem Raju3, Kondaveeti Gopi Krishna4
[5] HANDWRITTEN SIGNATURES FORGERY DETECTION Kshitij Swapnil Jain1, Udit Amit Patel2, Rushab Kheni3
[6] International Journal of Computer Applications (0975 – 8887) Volume 177 – No. 14, October 2019 21 Writer-independent Offline Handwritten Signature Verification using Novel Feature Extraction Techniques
[7] Offline Signature Recognition and Forgery Detection using Deep Learn-ing Jivesh Poddara, Vinanti Parikha, Santosh Kumar Bharti
[8] Performance evaluation of handwritten signature recognition in mo-bile environments: Ramon Blanco-Gonzalo,Raul Sanchez-Reillo,Oscar Miguel-Hurtado,Judith Liu-Jimenez
[9] Signature Verification: A Study; Saba Mushtaq, A.H.Mir
[10] Machine Learning for Signature Verification; Harish Srinivasan, Sargur N. Srihari† and Matthew J. Beal
[11] Handwritten Signature Forgery Detection using Convolutional Neural Networks; Jerone Gideon S, Anurag Kandulna, Aron Abhishek Kujur, Diana A, Kumudha Raimond
[12] https://citeseerx.ist.psu.edu/viewdoc/download?doi= 10.1.1.515.7879 & rep=rep1 & type=pdf
[13] https://arxiv.org/pdf/2108.02029.pdf
[14] https://ieeexplore.ieee.org.egateway.vit.ac.in/stamp/stamp.jsp?tp= & ar-number=9807905
[15] https://ieeexplore.ieee.org.egateway.vit.ac.in/stamp/stamp.jsp?tp= & ar-number=6949044