International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Handwriting Forgery Detection using NN

Siya Philip[1], Shikha S Nambiar[2], Shreya J[3] , T V N Satya Pratyusha[4] and Sneha S Bagalkot[5]

[1]*Student, Dept. of Computer Science Engineering, Presidency University Bangalore, Karnataka, India*
[2] *Student, Dept. of Computer Science Engineering, Presidency University Bangalore, Karnataka, India*
[3]*Student, Dept. of Computer Science Engineering, Presidency University Bangalore, Karnataka, India*
[4]*Student, Dept. of Computer Science Engineering, Presidency University Bangalore, Karnataka, India*
[5]*Assistant Professor, Dept. of Computer Science Engineering, Presidency University, Bangalore, Karnataka, India*

Abstract – *Handwriting is unique to each person, much like a fingerprint. Since every handwriting is unique, it is also referred to as the brain's fingerprint. Criminals use handwriting forgery to fraudulently produce, change, or write a person's handwriting such that it appears similar to the real handwriting in most cases, with the intent of profiting from the innocent party. In this present study, a method has been proposed where the model is trained with a dataset of handwriting, and predictions are made as to whether a provided signature is genuine or forged based on the features like ratio, centroid, eccentricity, skew and kurtosis, and solidity of the words.*

Key Words: Handwriting Forgery Detection, Word Segmentation, Image Pre-processing, Feature Extraction, Multi-Layer Perceptron, Neural Network, Prediction.

1.INTRODUCTION

Handwriting is often used to assess a person's personality. It is very normal for certain experts to be able to predict a person's behavior based solely on their handwriting. In a similar vein, we're attempting to develop a framework that calculates some characteristics to determine the validity of a note.

The internet has altered how the environment operates. During this transition, we tend to exchange a lot of documents over the internet because we are still transitioning to a completely online mode where paper is seldom used.

Since we are sending a physically printed document that does not contain any digital signatures that can be checked, the documents can easily be forged during this process. In this way, we put ourselves in a vulnerable position, allowing the forger to easily alter the text. This could result in a financial loss or unauthorized changes to a legal document. That is the reason Forgery is considered a white-collar crime.

To avoid such situations, we should ensure the document's originality, which we can do by conducting a Handwriting forgery search. This will assist us in being safer.

Specific features of Handwriting are:

I. The roundness of the letters
II. Spacing between letters
III. The pressure put on paper while writing
IV. The average size of letters
V. The inclined angle of letters

The above are some characteristics that aid in determining the authenticity of a person's handwriting. Since they will be unique to each person, these characteristics can be used to make decisions.

We tried to assess the validity of a given input by considering variables and comparing them in this project because this is an essential consideration when considering the trueness of the text.

This paper is our take on reducing the chances of forgery by comparing the handwriting of multiple users and training the model to recognize the real and forged inputs.

2. PREVIOUS WORK

Using previous work on the subject as a source of inspiration [8] In this research paper, they proposed a text-independent handwriting forgery detection system based on branchlet features and GMMs. They divide the function data into groups and use an exhaustive method to create their own GMMs. They then compare each category's similarity to the other input data.

3. PROPOSED SYSTEM

3.1 Word Segmentation

We will use Scale Space Technique for Word Segmentation to segment words from a given input image [7] where a grey level image is used as the system's input. The image is processed to eliminate horizontal and vertical line segments that could cause problems during subsequent operations.
The page is then dissected into lines using grayscale image projection analysis methods that have been updated. The projection function is smoothed with a Gaussian filter (low

pass filtering) to eliminate false alarms, and the locations of the local maxima (white space between the lines) are determined. Line segmentation is useful for breaking up connected ascenders and descenders as well as deriving an automatic scale selection mechanism.

To build a scale space, the line images are smoothed and then convolved with second-order anisotropic Gaussian derivative filters, and the blob-like features that provide us with the focus of attention regions (i.e., words in the original document image). A connected component analysis of the blob image is used to extract words, which is accompanied by a reverse mapping of the bounding boxes. After that, the box is vertically extended to make room for the ascenders and descenders.

Fig -1: Genuine Text

Fig -2: Forged Text

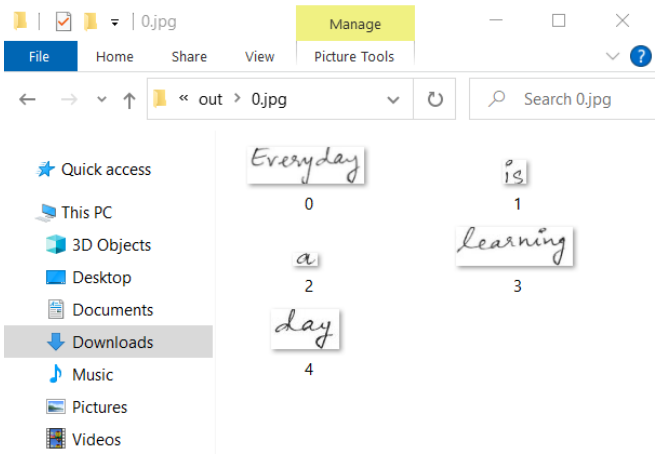

Fig -3: Word Segmentation and Dataset Formation

3.2 Pre-Processing

Pre-Processing is an essential step to perform before we dive into the comparisons as this helps in enhancing the feature of the images and increases the efficiency of the system. To increase the clarity of the image we have followed a set of steps which are as follows:

First, we convert the given input image into a greyscale format so that we now have the images only in black, white, and shades of grey colors which will provide us the raw image. This can be helpful to increase the efficiency of the as feature extraction will now be able to give more accurate values.

Second, we then remove any noise disturbances using Gaussian filter in the image and then convert it into the binary format as it will be very easy for us to consider and compare them later on.

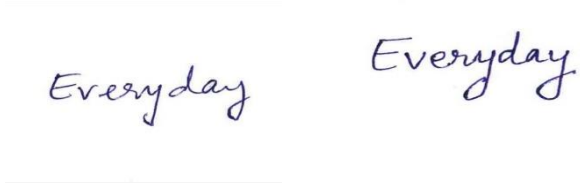

Fig -4: Genuine Image **Fig -5**: Forged Image

Images after Pre-Processing:

Fig -6: Genuine Image **Fig -7**: Forged Image

3.3 Feature Extraction

This process identifies important features of the image and gets their values and stores them in one place. Then we compare these values among the images to get the end result. Well, there are numerous features that can be considered we have listed out a few which according to us will give accurate values. This particular step defines the efficiency of the software.
The following are the features whose values will be extracted from the image [3]:

I. Ratio: This calculates the relationship between the height and width of the image. It basically describes the shape of the image.

ii. Centroid: The centroid of the image is often considered the intersection point of all the hyperplanes of symmetry within the image, by doing this we get the center point of the image.

iii. Eccentricity: The eccentricity of an ellipse is the ratio of the distance between its foci to the length of its main axis. The value is always in the range of 0 to 1.

iv. Skew: When transferring data to a digital format, skew detection is one of the first operations performed on scanned documents. Its aim is to align an image before it is processed because text segmentation and recognition methods depend on correctly aligned next lines.

$$skewness = \frac{\sum_{i=1}^{N}(x_i-\bar{x})^3}{(N-1)s^3}$$

where:

- σ is the standard deviation
- $\bar{x}$ is the mean of the distribution
- N is the number of observations of the sample

Fig -8: Skewness

kurtosis: Kurtosis is a statistical indicator of how often the tails of distribution vary from the tails of the regular distribution. It assesses the sharpness of a frequency distribution curve.

$$kurtosis = \frac{\sum_{i=1}^{N}(x_i-\bar{x})^4}{(N-1)s^4}$$

where:

- σ is the standard deviation
- $\bar{x}$ is the mean of the distribution
- N is the number of observations of the sample

Fig -9: Kurtosis

VI: Solidity: The area of an image object is divided by the area of its bounding rectangle to determine its extent. The area of an image object divided by the area of its convex hull determines its solidity. A fraction of the size of your actual picture.

3.4 MODEL

We have built a neural network (multi-layer perceptron) using TensorFlow and successfully train it to recognize if handwriting is genuine or forged.

3.4.1 Multi-Layer Perceptron

Multi-Layer Perceptron is the most complex Artificial Neural Network architecture, fundamentally made up of several layers of the perceptron. For supervised learning, MLP networks are used. It's a feedforward Artificial Neural Network that takes a set of inputs and produces a set of outputs. An MLP has several layers of input nodes that are linked as a directed graph between the input and output layers. The ability of a Neural Network to learn the representation of training data and how to relate it to the output variable that we need to predict is what gives it its strength. The hierarchical or multi-layered structure of a Neural Network contributes to its predictive performance.

Artificial neurons are the building blocks of a neural network. These are essentially computational units that take weighted input signals and use an activation function to generate an output signal.[5]

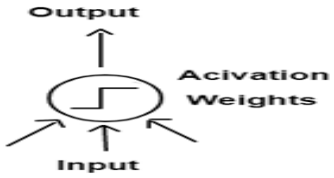

Fig -10: Weights and Activation

a) Weights

Weights are often set to small random values, such as those in the range of 0 to 0.3.

b) Activation

The weighted inputs are added together and passed through a transfer function, which is an activation function. It controls the frequency of the output signal as well as the threshold at which the neurons are activated.

3.4.2 Networks of neurons

There are several layers in a network.

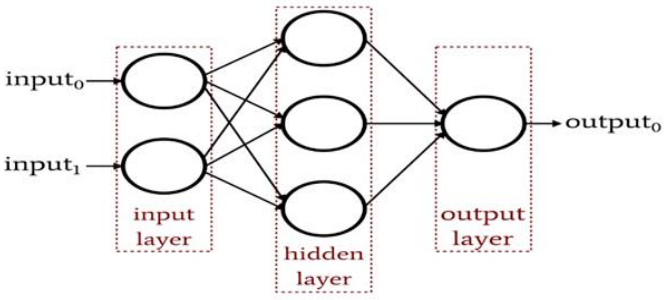

Fig -11: Layers of Neuron Network

a) Input/Visible Layer

Since it is the exposed component of the network, the bottom layer that takes input from the dataset is called the visible layer. Typically, a neural network is drawn with one neuron per input value or column in the dataset and one visible layer.

b) Hidden Layer

The hidden layer is the next layer after the input layer. They aren't exposed to the input directly. The most basic network structure is a single neuron in the hidden layer that outputs the value directly.

c) Output Layer

They are the final hidden layer, and they are in charge of the problem-related output values needed for the problem statement.

3.4.3 Training model

a) Data set

The first step is to read the train and test data. We begin by reading the training CSV file and performing operations on it, such as retrieving data from a column and storing it in an array using call values.

Then we use the astype() function, which creates a new copy of the training input with each value converted to a float (in our case). It doesn't change the training input, so you can check the value returned by astype() to get the converted array.

The method to categorical() can then be used to transform a NumPy array with data representing various categories into a NumPy array with binary values. It has the same number of rows as the input array and the same number of columns as the number of classes.

On the testing range, a similar function is performed. The model receives this result of binary values as data.

The default graph stack is then cleared, and the global default graph is reset.

b) Neuron Layers

To work with the Tensor, we define a few main parameters and variables, such as learning rate, training epochs, and display steps. The number of hidden layers and neurons for each layer is then determined. There are three layers in total: one input layer, two hidden layers, and one output layer.

c) Weights and Bias

Weights and biases are assigned to each of these layers. When inputs are transmitted between neurons, the weights and bias are applied to the input. Weights determine the strength of the interaction between neurons, or in other words, how much influence the input has on the output.

The previous layer has no impact on bias units, and they have no outgoing relations with their own weights. Before moving the data to the next layer, a single node multiplies the input data by an allocated weight and adds a bias.

The two parameters vary in the degree to which they affect the input data, with Bias responsible for the difference between the function output and the intended value. The weight aids in the connection of one layer to the next.

The first layer's input value and weight are multiplied and added to the bias of the first layer, and the output is used as the input for the next layer.

Fig -12: Weights and Bias

This is how we develop our model, and we build it by passing the input value.

To feed data into the tensor flow graph, we use a placeholder.

d) Loss and Optimizer

The next move is to locate the optimizer and loss. The aim of optimization is to reduce the loss function to the smallest possible value. If the loss is reduced to an appropriate amount, the model will learn an indirect function that maps the input to the output.

e) Accuracy

We'll now figure out how accurate each epoch is. The accuracy of our model is the percentage of predictions that were right.[6]

f) Prediction

The image prediction would then include an array of two columns, the first column containing the score of confidence against genuine and the second column containing the score of confidence against forged, based on the measurement accuracy. If the first column value exceeds the second column value, the document is genuine; otherwise, it is forged.

This model is used to determine whether or not a handwritten document is authentic.

4. IMPLEMENTATION AND RESULTS

Subjects were asked to write test samples in their natural handwriting style as well as forge handwriting samples from other subjects. Handwriting samples were scanned and saved digitally as a part of the process. The writing samples were then used to compute word-level features.[4]

When we get the dataset from the user, we let each image go through pre-processing. Once we get the raw images that would give us more accurate values, we perform feature extraction in which we extract values for ratio, centroid, solidity, eccentricity, skewness, and kurtosis.

In the given figure we have considered a dataset of 10 users taking their real handwriting and also made a 3rd

person write the same word which we will consider as the forged image. Each of these 10 users have given 5 images as in a user with id no 1 has given 5 samples of his real handwriting and also 5 of the forged ones. This is so that when performing comparison of values at the end we can get an efficient result if we have more values.

Now we transfer these to function extraction, which returns their values in CSV files. As can be seen in Figs 6 and 7, the device stores three of the values from five actual images in the Training CSV file and the other two in the Testing CSV file.

As can be seen in Figures 6 and 7, the situation for the 5 forged photos is identical, with three of them being sent to the Training file and the other two being sent to the Testing file.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ratio	cent_y	cent_x	eccentricit	solidity	skew_x	skew_y	kurt_x	kurt_y	output		
2	0.10793	0.484153	0.476339	0.977854	0.121271	0.136598	-0.1352	-1.53575	0.148335	1		
3	0.102177	0.485792	0.46773	0.966284	0.120314	0.242219	-0.18865	-1.51873	-0.13604	1		
4	0.099638	0.487985	0.469175	0.981397	0.109333	0.190346	-0.385	-1.54093	0.065313	1		
5	0.099336	0.570633	0.490918	0.981696	0.123256	0.085529	-0.96108	-1.41873	0.389648	0		
6	0.128324	0.608098	0.476115	0.985722	0.151386	0.144321	-0.95128	-1.41911	0.268592	0		
7	0.085526	0.602313	0.497622	0.982542	0.118069	0.042114	-0.85087	-1.41722	0.105634	0		
8												
9												
10												

Fig -13: Training Dataset

	A	B	C	D	E	F	G	H	I	J	K	L
1	ratio	cent_y	cent_x	eccentricit	solidity	skew_x	skew_y	kurt_x	kurt_y	output		
2	0.10664	0.47543	0.444786	0.96866	0.126933	0.308466	-0.17781	-1.48331	-0.02827	1		
3	0.101542	0.481809	0.493841	0.974536	0.11495	0.108399	-0.13627	-1.56974	-0.09078	1		
4	0.108076	0.598349	0.447133	0.979925	0.142562	0.192157	-0.76241	-1.4216	-0.03872	0		
5	0.105044	0.547544	0.446016	0.975917	0.125297	0.216097	-0.8596	-1.48769	-0.0579	0		
6												
7												
8												

Fig -14: Testing Dataset

To do the final verification on whether the handwriting is genuine or forged, we ask the user to enter a specific person's id for whom they want to check the handwriting.

We take a user's input image, perform feature extraction on it, and save it as a new csv file. The model now compares these values to the testing and training csv files, determining if the nearest value is genuine or forged.

It then tests the output value of the closest value of features and shows it as a real image if that value is 1, and as a forged image if that value is 0.

5. CONCLUSIONS

In this paper, we presented a handwriting forgery detection system using a multi-layer perceptron. This method assists in determining whether a handwritten document is authentic or forged. The system accomplishes this by measuring the precision and making an appropriate prediction.

We collected experimental handwriting data from subjects who wrote examples of their own handwriting and forgeries of the handwriting of other subjects. These handwritings were scanned digitally and saved in a folder.

As a result, we were able to distinguish between genuine and forged handwritten documents using pre-processing, feature extraction, and training the model with genuine and forged image datasets.

REFERENCES

[1] Navin Karanth , Vijay Desai and S. M. Kulkarni. 2011. Development of an automated handwriting analysis system

[2] Amr Megahed, Sondos M Fadl Harbin and Qi Han.2017. Handwriting forgery detection based on ink colour features

[3] S JeromeGiden, AnuragKandulna , Aron Abhishek Kujur , A diana and KumudhaRaimond.2018. Handwritten Signature Forgery Detection using Convolutional Neural Networks

[4] Sung-Hyuk Cha and Charles C. Tapper, Automatic Detection of Handwriting Forgery

[5] Agarwal, A., 2018. Multi-Layer perceptron using Tensorflow. [online] Medium. Available at: <https://towardsdatascience.com/multi-layer-perceptron-using-tensorflow-9f3e218a4809>

[6] Deb, S., 2017. Neural Network Tutorial —Multi Layer Perceptron. [online] Medium. Available at: <https://medium.com/edureka/neural-network-tutorial-2a46b22394c9>

[7] Manmatha R., Srimal N. (1999) Scale Space Technique for Word Segmentation in Handwritten Documents. In: Nielsen M., Johansen P., Olsen O.F., Weickert J. (eds) Scale-Space Theories in Computer Vision. Scale-Space 1999. Lecture Notes in Computer Science, vol 1682. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48236-9_3

[8] Chin-Shyumg Fahn,Chu-PingLee and Heng-I Chen .2016. A Text Independent Handwriting Forgery Detection System Based on Branchlet Features and Gaussian Mixture Models