PERSONALITY PREDICTION BASED ON HANDWRITING ANALYSIS(GRAPHOLOGY) USING SUPPORT VECTOR MACHINE

Tejashree A J, Swathi Ghosh P , Thriveni V, Sohara Banu A R

School of Computer Science and Engineering,
REVA University, Bengaluru.

Abstract.

Personality is a term that refers to a person's strengths and flaws. Handwriting, among all the distinctive traits of a human being, has the most information for gaining insights into the writer's physical, emotional, and mental state. Graphology can be used to determine a person's personality. Graphology is the science of examining and analyzing handwriting. It is a scientific approach for determining a person's personality traits such as fear, honesty, and defensiveness by evaluating numerous traits in the handwriting. The page margin, the baseline, letter size, pen pressure, the slant of the alphabets, line spacing, and word spacing are the most important aspects of handwriting to consider. All of these characteristics can reveal a lot about a person. Emotional stability, Will Power, Modesty, Lack of discipline, Personal Harmony, Non-communicativeness, Social Isolation, and poor concentration are some of the personality qualities that our project aims to predict.

We decided to use a supervised learning model called a support vector machine. The advantages listed above influenced our decision to use a support vector machine. It's effective in high-dimensional spaces, memory-efficient, and versatile: the decision function can be set with multiple Kernel functions. Custom kernels can be supplied in addition to the standard kernels.

Hiring a graphologist could be expensive, time consuming and accuracy is unpredictable, hence we intend to build a machine learning model which could do the same job as a graphologist but cost effective, time efficient and gives us more accuracy.

Keywords—SVM, Graphology, Personality prediction

1. INTRODUCTION

Handwriting, often known as brain-writing, is a powerful tool for determining an individual's personality characteristics. The handwriting of a person can reveal a lot about them. Personality traits can be discovered and understood using patterns and strokes in handwriting. Graphology is the name of this technique. Fear, honesty, a defensive personality, and many other personality traits are confirmed by handwriting. Graphology is a discipline that uses spaces, strokes, and curves in a drawing to determine, research, and interpret a person's personality using handwriting. Handwriting is nothing more than a subconscious mental expression, according to Graphology. It can either help the person in overcoming his/her personality crisis. A Graphologist is someone who analyses people's handwriting and if the graphology test is performed manually, it may take a lot of time due

to the number of factors that are examined in graphology. Furthermore, handwriting analysis accuracy is dependent on the analyst's ability level. Handwriting analysis may now be done automatically, thanks to advancements in image processing and pattern recognition. Because engaging a graphologist can be costly, this technology ensures economic feasibility and efficacy. It can also analyze more documents in less time. This application aids in the identification of personality on a wide scale in a short period of time, such as for mass recruiting in a firm, hiring for a specific position in a company, or matrimony sites.

2. LITERATURE SURVEY

AUTHOR	**SYSTEM**	**METHODOLOGY**	**PERFORMANCE**	**LIMITATIONS / (DRAWBACKS)**
Evi Septiana Pane and Adhi Dharma Wibawa, Harris Teguh Laga	Personality Classification from Online Handwritten Signature using KNN	k-Nearest Neighbors (KNN)	87.5% accuracy	Due to the digitization process, digital devices cause a shift in people's signatures from their original version.
Shitala Prasad, Vivek Kumar Singh, Akshay Sapre.	Personality Predictor	Support Vector Machine	90.3% accuracy	Inaccurate feature selection
Bharti W. Gawali and Vaishali R. Lokhande	Analysis of Sign for the Prediction of Personality Traits	ANN and Structural identification algorithms	95.4% accuracy	Only used five features to come to a conclusion

Krish Shah; Rajas Rade; Dharmil Shah;Nikita Lemos;	Personality Prediction based on Handwriting using Machine Learning	CNN (Convolutional Neural Network)	The total of each percentage of personality attribute equals 100 percent.	For the CNN model to be developed, the system requires a lot of processing power.

The rationale for utilizing Support Vector Machine is to tackle a problem that other models have failed to handle, such as not selecting crucial basic features or selecting a model that does not produce accurate results.

3. REQUIREMENTS

- Software Components

a) Open CV

OpenCV is an amazing image processing and computer vision software. It's an open-source library that may be used for things like facial recognition, object tracking, and landmark recognition, among other things. Python, Java, and C++ are just a few of the programming languages available. [19]

b) Python 3.7

Python is used over MATLAB because it is more straightforward to set up and use. Python is a good choice for our research because it includes a huge number of free image processing and machine learning libraries.

c) NumPy

Numpy is mainly used to deal with the matrix of any dimensions, and perform standard mathematical and scientific computations which helps in calculating variety of tasks.Matlab, A prominent technology computing platform is replaced with this combination .Numpy also has an advantage of being free and open-source.

d) Scikit-learn

Scikit-learn is the most accessible and machine learning library. It uses a Python consistency interface to deliver a set of efficient machine learning and statistical model capabilities, such as classification, regression, clustering, and dimensionality reduction. NumPy, SciPy, and Matplotlib are the foundations of this Python-based toolkit.

e) Streamlit

Streamlit is used to build web apps of data science and machine learning ,it takes very less time in building and deploying the apps .It allows to write streamlit app code which is the same like that we do write in python .The outcomes of streamlit are pretty straightforward.

b) Support Vector Machine-SVM

SVM is a supervised ML approach for classifying and predicting data and attempts to find a hyperplane in an N-dimensional space that categorizes data points clearly.

Two independent variables: X1 and X2

One dependent variable: A blue or red circle.

As shown in the diagram,

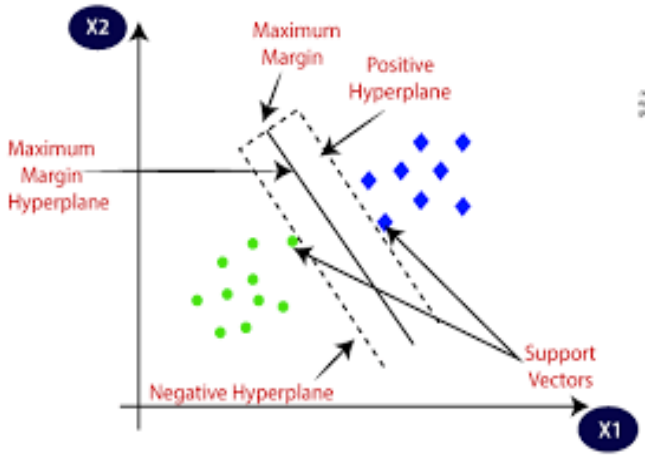

Fig -1: Separating two classes of data points with the SVM algorithm.

As you can see in the graphic above, there are multiple lines that split or categorize our data points into red and blue circles.

Optimal hyperplane, Choose the one with the greatest difference. If classes are completely linearly separable, a hard-margin can be used. A soft-margin is required otherwise. SVM overcomes this by using a kernel to produce a new variable if the data isn't linearly separable.

When there is a non-linear separation, the SVM kernel does some extremely sophisticated data transformations before deciding on the best approach for separating the data based on the labels.

Radial Basis Function is a commonly used kernel in SVC:

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

Where $||x-y||^2$ is the squared Euclidean distance between two data points x and y.

There are two parameters in the RBF kernel: gamma and C.

Gamma is an RBF kernel parameter that represents the spread of the kernel and thus the decision region. The 'curve' of the choice border is very low when gamma is low, resulting in a reasonably broad decision zone. The decision boundary's 'curve' is high when gamma is large, resulting in decision-boundary islands encircling data points.

C: The penalty for incorrectly categorizing a data point is C, which is an SVC learner parameter. The classifier doesn't mind if data points are misclassified when C is tiny (high bias, low variance). Because misclassified data is heavily penalized when C is large, the classifier goes to great lengths to avoid misclassified data (low bias, high variance).

4. FLOW OF SYSTEM

Fig -2: Flow of system

5. IMPLEMENTATION METHODOLOGY

a) Data Acquisition

We used the dataset from The IAM Handwriting Database which is avaliable on Computer Vision and Artificial Intelligence INF. The data is easily downloaded and can be used for non-commercial research. 657 writers provided samples of their handwriting to the collection, which has 1538 pages of scanned text. By physically studying each paper, each handwriting sample is identified with the associated psychological qualities. With the help of an automatic action script, these photos are trimmed and saved as PNG images. The image now has a height dictated by the handwriting content and a width of 850 pixels.

b) Pre-processing

Unwanted noise, printed phrases, and lines can be found in the handwriting photos we collected. By default, the source photographs have a very high resolution. Pre-processing is used to prepare picture data for feature extraction by removing unnecessary properties, improving quality, and applying transformations. This section discusses the procedures used in pre-processing.

i.Image Resolution and Cropping

An action script in Adobe Photoshop is used to crop out the left and right margins, resize all the photos to 850 pixels width and perspective height, and save them in PNG format.

ii.Image noise removal

Image noise is an electrical noise that is characterized as a random variation in image brightness or color information. The use of a bilateral filter to remove these disruptions is beneficial since it keeps the edges of the image's elements, which is desirable. A bilateral filter is a nonlinear image smoothing filter that keeps edges while reducing noise. It replaces each pixel's intensity with a weighted average of intensity data from nearby pixels.

iii.Grayscale and Binarization

Conversion to grayscale and binarization are crucial aspects of the pipeline for obtaining handwriting information. Inverted global thresholding is used to transform the image instances to grayscale and binarize them. A pixel in a binary image can be either 0 (black) or 255 (white) (white). A simple threshold can be used to divide all pixels in the picture plane into foreground and background pixels, i.e. the handwriting itself and the white background of the paper, in order to form the two-valued binary image. Then, using an inverted binary image function, pixels above a certain threshold (foreground) are converted to 255, while pixels below the threshold (background) are converted to 0. This operation of thresholding can be written as:

$$\texttt{dst}(x,y) = \begin{cases} 0 & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ \texttt{maxVal} & \text{otherwise} \end{cases}$$

The new pixel intensity = 0, when src(x,y) > threshold.

else it is set to maxVal

iv.Contour and Warp Affine Transformation

Following noise removal and picture conversion to grayscale and inverse binarization, the lines of the handwriting are straightened using the OpenCV library's dilation, contour, and warp affine transformations. Further processes using horizontal projection of the image to extract these handwriting lines will produce better results.

Before you can see the contours of a picture, you must first dilate it. Combining an image A with a kernel B, which is usually circular or square, is a dilation operation. Kernel B's anchor point is typically the kernel's core.

We compute the highest pixel value overlapped by B as the kernel B scans and updates the image.

Bright areas of an image grow as a result of this maximizing technique (therefore the name dilation).

The limits of an object in an image are represented by a contour, which is a closed curve of points or line segments. Correct contour extraction will yield more accurate features, increasing the likelihood of accurately identifying a given pattern.

The warp affine transformation rotates the contours on a picture so that the handwriting's baseline is strictly horizontal. One of the seven features afterwards is the angle of rotational transformation returned by the procedure.

c) Feature Extraction

i) Extraction of Baseline

For baseline extraction, follow the steps listed below.

1. The image is converted to a binary image using inverted binary thresholding with a threshold of 120. The foreground (handwriting) pixels are now black while the background pixels are white.

2. A 5x100 kernel is used to dilate the image from step 1 such that each line becomes a thick horizontal segment.

3. On the image from step 2, contours can be found. Contours with a height of less than 20 pixels are eliminated since they are not a handwritten line. The remaining contours now depict each line of handwriting or a group of congested lines.

4. The OpenCV minimum area rectangle function takes a contour object and returns as one of the returned values the angle that the contour makes with a hypothetical vertical line. The angle formed by these contours with an imaginary horizontal line is then calculated.

5. The baseline angle, which is our initial feature, is calculated by taking the average of all the contour angles.

6. Around a contour, a rotation matrix is created. The rotation matrix is used to make the contour precisely horizontal by rotating it by its baseline angle in the opposite direction. This procedure aids in the maximization of horizontal projection operations' effectiveness.

ii) Extraction of Individual Lines

Below is the algorithm for extracting individual lines: 1. The horizontal projection of the straightened images captured from baseline extraction is found in a python list called hpList.

2. The hpList gets scanned in reverse order. Some horizontal projection values, such as 0, depict blank space rows. A pixel row with a non-zero hpList value is one that encounters at least one foreground pixel (contour). From the commencement of a non-zero value through the next zero value, each contour is identified.

3. If there is a group of congested lines, the contour is scanned again to extract individual lines. At the intersection of the line above and the line below, a contour with a dense set of lines will have an extremely low hpList value. To detect such rows, a threshold is set. As it moves from the top of each line to the bottom, the hpList will progressively climb, then decrease. The hpList value is less than the barrier around the bottom region, indicating that

this line and the following line are overlapping. This line's end index and the next line's start index comes out from the index of the previous line.

4. Repeat step 4 for each of the contours discovered in step 2. We will have the starting and ending indexes of all the individual lines by the end.

iii) Extraction of Letter Size

To estimate letter size, the horizontal projection of each line extracted in the previous procedure is scanned. The number of consecutive rows with a projection value greater than a threshold is counted. The average letter size of all the lines will be used to establish our letter size. This only takes into account the size of the midzone, ignoring the upper and lower zones.

iv) Extraction of Line Spacing

1. Except for the top margin, the entire number of rows with horizontal projection 0 is counted. Let's just call it that.

2. In the retrieved lines, the total number of rows with horizontal projection less than a threshold is counted. Let's name it b for now. These are made up of the lines' upper and lower zones. (Finding the letter size required this step as well.)

3. The number of extracted lines in the manuscript is n. The entity x = a + b/n calculates the average handwriting line spacing.

4. To make the final line spacing proportional to the size of the handwriting, divide x by the letter size.

v) Extraction of Word Spacing

Word spacing is defined as follows

1. Each line of the image's vertical projection is generated and stored in a Python list (array).

2. Except for the left and right margins, the number of columns in the list with a value of 0 (column with all pixel values of 0, it is vacant area) is tallied. Let's assume it to be a.

3. The number of words or disconnected letters is determined by the number of non-zero column runs. Let's keep it as b.

4. Now, x = a+b shows the average word spacing while writing.

5. Find the average of these x's across all lines. Let it be y. The word spacing is calculated by dividing the letter size by y, so it is proportional to the handwriting size.

vi)Extraction of Top Margin

To extract the top margin, we simply scan the horizontal projection of the image from top to bottom for its first run of 0's. The amount of 0s determines the height of the top margin, which is then divided by the letter size to make it proportional to the handwriting size.

vii)Extraction of Pen Pressure

Pen Pressure is determined as follows:

1. The image is inverted using the formula: dst[x][y] = 255src[x][y]. This step is computationally very costly.

2. If src(x; y) is less than threshold=100, an inverted binary threshold (THRESH TOZERO) is used, and the new pixel value dst(x; y) is set to 0, otherwise it is kept unchanged.

3. The average value of all the non-zero pixels is taken as the pen pressure. The value is not inverted again (to reverse the effect of step 1) so that higher value would mean higher pen pressure.

viii)Extraction of Slant of Letters

The concept that when the number of columns carrying a continuous stroke reaches its maximum, the word becomes deslanted is used to determine the slant of letters in handwriting. The tilt is determined using the algorithm below.

1. A shear transformation is done to 9 different angles (-45, -30, -15, -5, 0, 5, 15, 30, and 45 degrees).

We get the following histogram.

H(m)=h(m) / Y(m),

where,

The distance between the highest and lowest pixel in the same column is given by y(m).

H(m) is the vertical density. H(m)=1 when column m has a continuous stroke. else, H(m) [0,1].

2. The following function is calculated for each shear converted image.

S = h(i) ^2

3. The slant of the handwriting is the angle that produces the maximum value of S.

d) Classification- Support Vector Machine

The seven raw features obtained from the handwriting samples are normalized into discrete values according to experimentally determined threshold values.

Feature	Normalized Value
Baseline	0 = descending 1 = ascending 2 = straight
Top Margin	0 = medium or bigger 1 = narrow
Letter Size	0 = big 1 = small 2 = medium
Line Spacing	0 = big 1 = small 2 = medium
Word Spacing	0 = big 1 = small 2 = medium
Pen Pressure	0 = heavy 1 = light 2 = medium
Slant Angle	0 = very reclined 1 = a little of moderately reclined 2 = a little inclined 3 = moderately inclined 4 = extremely inclined 5 = straight 6 = irregular

Fig -3: Normalization of Features.

The eight personality traits will be predicted by the combinations of these seven features. Hence, there will be eight separate labels for each personality trait and eight SVM classifiers. The images are labeled by studying each handwriting sample and its corresponding normalized features.

The SVM implementation of Sci-kit Learn Library is used and the eight classifiers are trained with radial basis function (RBF) kernel. Two third of all the images are randomly chosen for training and the remaining is used to find accuracy.

6. RESULT

The eight SVM classifiers are trained with randomly chosen two third of all the images. The remaining images are used to test the accuracy score. The following table shows the accuracy of each classifier.

Classifier	Personality Trait	Accuracy
1	Emotional Stability	100%
2	Mental Energy or Will Power	100%
3	Modesty	100%
4	Personal Harmony and Flexibility	100%
5	Lack of Discipline	100%
6	Poor Concentration	100%
7	Non-communicativeness	100%
8	Social Isolation	100%

Fig -4: Accuracy of the Classifiers

We are able to achieve hundred percent accuracy by using the RBF kernel

7. CONCLUSION

The use of machine learning to analyze an individual's handwriting patterns has been proposed as a tool for predicting some personality traits. We examined extracting seven handwriting variables and predicting eight personality traits using different combinations of them. Each SVM classifier is trained for each of the personality traits. We can estimate personality traits on new handwriting picture samples with remarkable accuracy and efficiency after a reasonable amount of training.

8. REFERENCES

[1] Z. Mohd Zam, "Handwriting Analysis for Employee Selection Using Neural Network", Thesis of Intelligent System Faculty of Information Technology and Quantitative Science of University Technology MARA, 2006.

[2] Kishan Mehrotra, Chilukuri K Mohan, Sanjay Ranka, "Elements of Artificial Neural Networks", the MIT Press, Cambridge, Massachusetts, USA 1997.

[3] B.Yagnanarayana ,"Artificial Neural Network" PHI Learning private Limited, New Delhi 1999.

[4] P.K.Grewal, D Prashar, "Behavior Prediction Through Handwriting Analysis", IJCST Vol. 3, Issue 2, April - June 2012

[5]] B. Ludvianto, "Handwriting Analysis", Gramedia Pustaka Utama, 2011

[6] EC. Djamal, "Recognition of Human Personality Based on Handwriting Using Multi Structures Algorithm and Artificial Neural Networks", 2nd IEEE Conference on Control, Systems & Industrial Informatics Bandung, Indonesia June 23-26, 2013.

[7] BuzzFeedYellow, "What Your Handwriting Says About You", Youtube, 31 May 2014, [online]. Available: https://www.youtube.com/watch?v=eurGvShP0T8 [Accessed: 16 July 2016]

[8] H.N Champa, K.R AnandaKumar, "Artificial Neural Network for Human Behavior Prediction through Handwriting Analysis", International Journal of Computer Applications (0975 – 8887) Volume 2 – No.2, May 2010.

[9] "HandwritingResearch Corporation" http://www.handwriting.com/facts/history.html G. Sheikholeslami, S. N. Srihari, V. Govindaraju, "COMPUTER AIDED GRAPHOLOGY", Center of Excellence for Document Analysis and

Recognition.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[10] A. McNichol, “Handwriting Analysis Putting It to Work for You”, Contemporary Books, 1994.

[11] B. Ludvianto, “Handwriting Analysis”, Gramedia Pustaka Utama, 2011.

[12] P.D. Sunar, “Read the full review Graphology Personality Through Writing His Handwriting”, Yogyakarta : Diva Press, 2010.

[13] Cleber Zanchettin, Byron Leite Dantas Bezerra and Washington W. Azevedo,“A KNN-SVM hybrid model for cursive handwriting recognition”,Neural Networks (IJCNN), The 2012 International Joint Conference.

[14] “Artificial Neural Network “ ,Available at: http://cogsci.stackexchange.com/questions/8509/dynamic al-systems-theory-as-a-metaphor-in-psychology-is-ituseful-or-not Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender

[15] Rashi Kacker, Hima Bindu Maringanti, “Personality analysis through handwriting” , GSTF Journal on Computing (JoC) Vol.2 No.1, April 2012.

[16] G. Sheikholeslami, S. N. Srihari, V. Govindaraju, “COMPUTER AIDED GRAPHOLOGY ", Center of Excellence for Document Analysis and Recognition.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[17] A. Varshney and S. Puri, "A survey on human personality identification on the basis of handwriting using ANN," International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2017, pp. 1-6.

[18] Lemos N, Shah K, Rade R, Shah D. Personality prediction based on handwriting using machine learning. In2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS) 2018 Dec 21 (pp. 110-113). IEEE.

[19] Basha, Syed Muzamil, Ravi Kumar Poluru, and Syed Thouheed Ahmed. "A Comprehensive Study on Learning Strategies of Optimization Algorithms and its Applications." *2022 8th International Conference on Smart Structures and Systems (ICSSS)*. IEEE, 2022.