HIGH SCHOOL EDITION
Journal of Student Research

Volume 9 Issue 2 (2020)
Research Article

Improving Signature Forgery Analysis through Deep Learning Classification Algorithms

Kashvi Gupta[1] and Shashank Gupta[1]

[1]Tesla STEM High School, Redmond, WA, USA

ABSTRACT

This project seeks to develop a machine learning algorithm to identify a forgery from a legitimate signature, for use in signature verification in low profile financial crimes. This model will be trained on data collected on specific handwriting characteristics used by professional document analysis experts. Signature forgery in financial institutions was recently brought to light in the Wells Fargo fake accounts scandal, where employees opened 3.5 million unauthorized accounts, for which 190,000 customers were unwittingly charged fees. This product will help protect individuals from exploitation by providing a verification tool for company managers and executives. The hypothesis predicted if characteristics such as length-to-height ratio and relative slant were taken into account, then the accuracy of the model would be greater than 90%. Each data point consisted of 6 signatures, of which 5 were 'true' (produced by the same person) and the final was forged. The data was fed to a processing program that used mathematical formulae (standard deviation from the mean) to account for, and negate, human error. Finally, using the python Scikit machine learning library, multiple models were trained on the data sample, using k-fold analysis. The most successful model, XGBoost Classifier, had an accuracy rate of 94.55%.

Introduction

Signature forgery is especially prevalent in financial industries where employees were asked to forge signatures on insurance documents of their clients (Johnson, 2017). Between 2014 and 2016, recorded cases of signature forgeries increased threefold (Mutual Fund Dealers Association of Canada, 2017). In addition to this, the forensics questioned document analysis field is lacking in scientific standards for handwriting and signature comparisons (Committee on Identifying the Needs, 2009). This protocol has discriminatory issues, as it assumes that no two people write the same way, though this has not been proven, and oftentimes, forged signatures have too little variability to confidently confirm a forgery over regular variation in writing. Such normal variation is extremely prevalent in signature. This project seeks to use computer analysis to identify forgery from legitimate signatures with a high accuracy using characteristics such as length and height.

Engineering Goals

Train a machine model to recognize if two signatures originated from the same individual with a minimum accuracy of 90%.

1. Analyze data using detailed, operationally defined characteristics used by professional question document experts.

2. Use a graphics tool to more accurately define the measurements of a signature to minimize human bias and error.
3. Adjust for human error in collecting data through use of mathematical algorithms.

Methods

Data Collection

Each signature sample consists of 6 signatures in total, as seen in Figure 1.

Figure 1. Signature sample. The first 5 signatures are 'true,' and the final signature is a forgery.

The first 5 signatures are true signatures, all signed by the same individual. The final signature is a forgery. It is important to have a wide variety of forgers produce the false signatures across the samples. By ensuring a wide variety of people produce the falsifications, the model will learn the characteristics that differentiate any forgery, instead of the semantics of a single individual's writing. Variation in signature is also very high, even those produced by the same person. By collecting 5 true signatures, the program will learn how to account for this variation.

Data Analysis and Quantification

To reduce potential for human error, signature samples were scanned and opened in ImageJ. Using the ruler tool, 4 (of 5 total) characteristics were measured and recorded for each signature in the data point.

a. Length: the furthest leftward mark to the furthest rightward mark made by the individual
b. Height: the highest mark to the lowest mark made by the individual
c. Baseline: the closest distance to the baseline that a letter (excluding lower case f, g, j, p, q, y, z) where signatures that dip below the baseline have a negative value and signatures that sit above the baseline have a positive value (See Figure 2). Record the maximum displacement from the baseline.

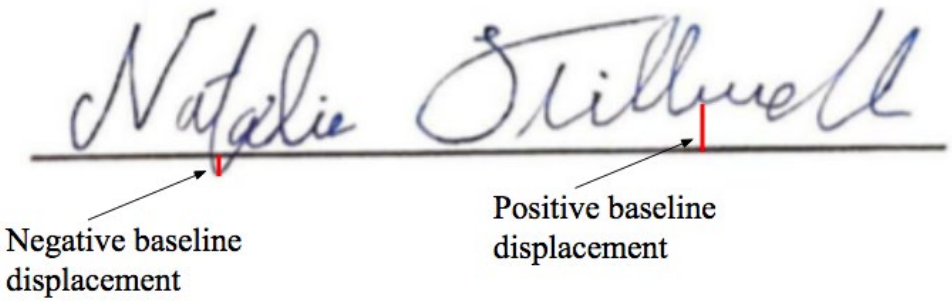

Figure 2. Indicating where to measure positive and negative baseline displacement on a signature.

d. Slant: the angle at which the same letter character sits against the baseline. This characteristic is unique, because it is a measurement of relative variance between signatures. Choose a line in a signature from the datapoint, such as the backbone of the letter T (as seen in Figure 3). Using ImageJ's angle tool, determine the angle between that line and the baseline.

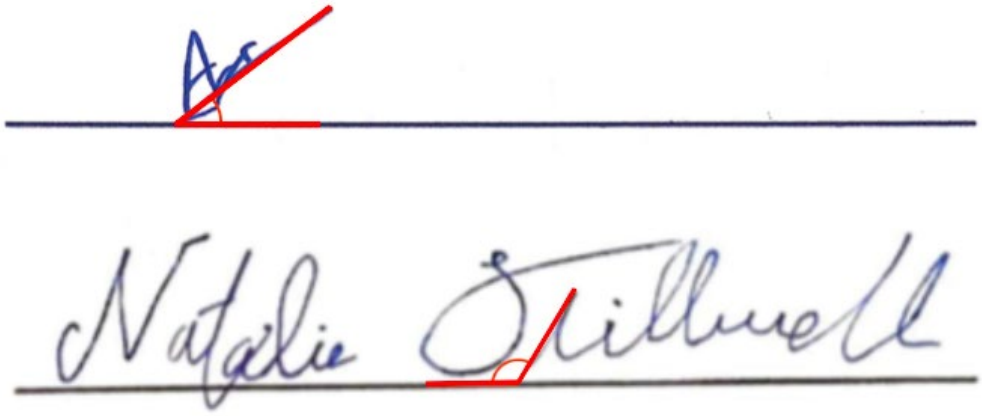

Figure 3: Indicating how slant allows for the measurement of unique characteristics in letter formation that vary from data point to data point.

Data Processing

The data needed to train the machine learning model must be formatted.
Each signature sample contains information about 6 signatures, each of which contains data about 4 characteristics (length, height, baseline displacement and slant). A fifth characteristic, the l-h ratio, is calculated for each signature using the following formula:

$$\text{L-H Ratio} = \frac{length\ of\ signature}{height\ of\ signature}$$

To additionally reduce human error in measurement, standard deviations from the mean for each characteristic were determined. These deviation-adjusted values were calculated across the multiple signatures of a signature sample, for each characteristic. By obtaining an average across a single characteristic, any inconsistencies in the forgery will be

doubly obvious. This process standardized signature samples that were collected in pixels, inches or centimeters, and by different people. Deviation-adjusted values are calculated using the following formula:

Deviation-adjusted value = $\frac{a - mean}{standard\ deviation}$

Where the mean and the standard deviation are calculated the same characteristic in a signature sample.

Sample Calculation:

Table 1. Raw Data Collected from a Signature Sample

Signature #	1	2	3	4	5	F
Length	160	209	227	229	200	213
Height	66	82	73	62	72	45
Baseline	7	13	2	3	2	1
Slant	60	53	37	50	53	67

Note: Data collected in pixels

1. Calculate the mean for the length characteristic across a single signature sample. See the first row of data in Table 1.

$\frac{160 + 209 + 227 + 229 + 200 + 213}{6} = 206.333$

m = 206.333

2. Calculate the standard deviation length characteristic across a single signature sample.

$(160 - m)^2 + (209 - m)^2 + (227 - m)^2 + (229 - m)^2 + (200 - m)^2 + (213 - m)^2 = 3179.333$

$\frac{3179.333}{6} = 529.889$

$\sqrt{529.889} = 23.019$

s = 23.019

3. Use the deviation-adjusted value formula to determine variance for each signature's length characteristic.

$\frac{160 - m}{s} = -2.013$

-2.013 is the deviation-adjusted value for the length of the first signature in this signature sample.

These standardized values were then reformatted to generate the data used to train the machine model. Each signature sample contains 6 signatures, but by reorganizing the data, such that each data point contains 2 signatures, 30 data points can be generated from a single signature sample. Each datapoint that contained a forgery was assigned a 1, indicating that the two signatures were not generated by the same person.

Training the Model

7 classification models were trained on the data with k-fold analysis, which allows for a more accurate set of metrics about the performance of a model. Using 10 folds, accuracy, precision and recall were calculated for each model. The average of each array returned by k-fold was calculated to determine the final values of each of the three metrics.

Data Analysis

Accuracy is the number of correctly categorized data points. Accuracy is calculated using the formula:

$$\text{Accuracy} = \frac{\#\ of\ correctly\ identified\ data\ points}{total\ \#\ of\ data\ points}$$

Precision determines the portion of correct positive identifications from the total number of positive identifications. Precision is calculated using the formula:

$$\text{Precision} = \frac{\#\ of\ true\ positives}{\#\ of\ true\ positives + \#\ of\ false\ positives}$$

Recall determines the portion of positives that were identified correctly. Recall is calculated using the formula:

$$\text{Recall} = \frac{\#\ of\ true\ positives}{\#\ of\ true\ positives + \#\ of\ false\ negatives}$$

Accuracy, precision and recall were calculated for each fold during k-fold analysis, and the array of final scores was averaged to get the final number for each model.

The F2 metric for each model was calculated from the precision and recall values derived from k-fold analysis. F2 is calculated as follows:

$$F2 = \frac{5 * precision * recall}{4 * precision + recall}$$

F2 places higher weight on recall than precision, which is ideal in this case, because only ⅙ of the total data consists of forgeries. The rest are true signatures. By evaluating using the F2 metric and emphasizing recall, this inconsistency in the data is acknowledged.

Results

The XGBoost Classifier performed the best of all the models tested. It had the highest accuracy, of 94.55%, and the highest F2 score of 89.09%. XGBoost also had the highest recall and accuracy scores of all the models tested, as can be seen in Tables 2 and 3.

Table 2. Accuracy, Recall and Precision Scores Obtained through K-Fold Analysis

	Accuracy	**Precision**	**Recall**
Logistic Regression	0.6710225140712947	0.26666666666666666	0.06593406593406592
RandomForestClassifier	0.8909474671669795	0.9387723387723389	0.7197802197802197
SVC	0.6536991869918699	0.03333333333333333	0.007142857142857143
GaussianNB	0.6408880550343964	0.5319179894179894	0.4461538461538462
DecisionTreeClassifier	0.9157676672920576	0.8743040293040293	0.8901098901098902
KNeighborsClassifier	0.718733583489681	0.6151839826839827	0.45054945054945056
XGBoost Classifier	0.9455331457160725	0.9596153846153846	0.8752747252747254

Table 3. F2 Scores Calculated from Precision and Recall

	F2 Scores
Logistic Regression	0.07761966364812417
RandomForestClassifier	0.8043787262605125
SVC	0.00847457627118644
GaussianNB	0.4610204320442007
DecisionTreeClassifier	0.8898114551121572
KNeighborsClassifier	0.47602822994754895

Conclusion

This project was successful, in the regards to the engineering goal, which was achieved. The best performing model was XGBoost Classifier with an accuracy of 94.55% and an F2 score of 0.891. These numbers are approaching the statistics of professional question document experts, whose accuracy hovers around 96-97%. This model provides the technical basis for a tool that can be used to help combat signature forgery in both financial and criminal investigations, by creating an objective, accessible evaluation to determine if a signature is genuine. In the future, developing an image recognition model will provide more opportunity to identify fakes, especially in terms of letter formation.

Acknowledgements

Samarjit Kaushik and Karen Song for project ideation and data collection.

References

Committee on Identifying the Needs of the Forensic Sciences Community, National Research Council. "QUESTIONED DOCUMENT EXAMINATION." *Strengthening Forensic Science in the United States: a Path Forward*, Bibliogov, 2012, pp. 163–167.

Egan, M. (2017, August 31). Wells Fargo uncovers up to 1.4 million more fake accounts. *CNN*. Retrieved from https://money.cnn.com/2017/08/31/investing/wells-fargo-fake-accounts/index.html?iid=EL

Gideon, S. J., Kandulna, A., Kujur, A. A., Diana, A., & Raimond, K. (2018, November 19). Handwritten signature forgery detection using convolutional neural networks. 978 - 987. 10.1016/j.procs.2018.10.336

Iranmanesh, V., Ahmad, S. M., Adnan, W. A., Yussof, S., Arigbabu, O. A., & Malallah, F. L. (2014). Online hand-written signature verification using neural network classifier based on principal component analysis. TheScientific-WorldJournal, 2014, 381469. doi:10.1155/2014/381469

Leasure, P., & Zhang, G. (2017). "THAT'S HOW THEY TAUGHT US TO DO IT": Learned Deviance and Inade-quate Deterrents in Retail Banking. *Deviant Behavior*, *39*(5), 603–616. doi: 10.1080/01639625.2017.1286179

Parizeau, Marc & Plamondon, Réjean. (1992). A handwriting model for syntactic recognition of cursive script. 308 - 312. 10.1109/ICPR.1992.201779.

S. Dutta Chowdhury, U. Bhattacharya, S.K. Parui, "Online Handwriting Recognition Using Levenshtein Distance Metric", *Document Analysis and Recognition (ICDAR) 2013 12th International Conference on*, pp. 79-83, 2013.

Srihari, S. N., & Bozinovic, R. (1981). *A string correction algorithm for cursive script recognition*. Buffalo, NY: State University of New York at Buffalo, Dept. of Computer Science.

Srihari, S., Cha, S., Arora, H., & Lee, S. (2002). Individuality of Handwriting. Journal of Forensic Sciences, 47(4), 1-17. Retrieved from https://doi.org/10.1520/JFS15447J