Handwritten Signature Verification System using machine Learning Approach

Pooja Gaikwad[1], Kashaf Pathan[2], Purva Patil[3], Laxmi Pagare[4], R.P. Dahake[5]

[1,2,3,4] Student, Department of Information Technology, MET's Institute of Engineering, Nashik, INDIA

[5] Professor, Department of Information Technology, MET's Institute of Engineering, Nashik, INDIA

ABSTRACT

In the field of biometric, offline handwritten signature verification is most referenced procedure for authentication of a person during financial transaction. A signature is the "seal of approval" for verifying the approval of a person and remains the most preferred means of verification. This verification system mainly aims at verifying the discriminating the forged signature from the genuine signatures. In this work, Convolutional Neural Networks (CNN) have been used to learn features from the pre-processed genuine signatures and forged signatures dataset. The CNN used is inspired by Inception V1 architecture (GoogleNet). The architecture uses the concept of having different filters on same level so that the network would be wider instead of deeper. In this paper, the proposed model is tested on few publicly available datasets on kaggle.com. has been successful in verifying handwritten signature images provided with an extensive precision level.

Keywords— Offline handwritten signature, classification, algorithms, artificial intelligence, CNN

I. INTRODUCTION

Signature has been a distinguishing element for individual identification through years. This area incorporates a background marked by ages which begins from the Roman Empire. The legalization of the handwritten signature has been declared by the British government in the 19th century. Signatures have been used for automatic clearing of cheques in the banking industry as well as confirming the legality of any document regarding properties, real estate and agreements. Despite an expanding number of electronic choices to paper cheques and the other materials which are generally authorized by the handwritten signature, fraud detection has been a continuous issue in each area where the handwritten signatures are used. Since banks pay less attention to verifying signatures on cheques and Debit and Shopping cards, a system capable of verifying whether a signature is a forgery or not will prove beneficial. As signature is the primary mechanism for authentication and authorization in legal transactions and documents, the need for an efficient automated solution for signature verification has increased. Therefore, developing a secure and robust platform that automatically authenticates the signatures in few seconds based on the sample dataset of original signatures of the owner is the objective of this system.

II. LITERATURE SURVEY

In "Handwritten signature recognition method based on fuzzy logic" [1], this paper author suggest a new method for handwritten signature recognition based on fuzzy features of the curvature. In the work authors suggested a method for recognition of fuzzy 2D primitives via a technology of soft computing.

The authors, [2] observed that segmentation accuracy of Devanagari text characters is completely depends on accurately segmented lines and words from handwritten documents. Author identified many issues and challenges for segmenting lines and words from these handwritten documents. Global threshold and Otsu's optimum threshold methods were experimented by author and found 85.12% segmentation accuracy.

The authors [3] designed and implemented a segmentation technique for segmentation of multiconnected handwritten Devnagri compound characters to extract the features and recognize the hand-written Devnagri compound characters.

There are different classifiers that can be used for this application and the followings are the alternatives that was used in this study.

A. **Multinomial Naive Bayes Classifier** - MultinomialNB implements the naive Bayes algorithm for multinomial distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. [5]

B. **Bernoulli Naive Bayes Classifier - BernoulliNB** implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binaryvalued (Bernoulli, Boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter).

C. **Logistic Regression Classifier** - In the multiclass case, the training algorithm uses the onevs-rest (OvR) scheme if the 'multi_class' option is set to 'ovr' and uses the cross-entropy loss, if the 'multi_class' option is set to 'multinomial'. (Currently the 'multinomial' option is supported only by the 'lbfgs' and 'newtoncg' solvers) [6]. This class implements regularized logistic regression using the liblinear library, newton-cg and lbfgs solvers. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

D. **Random Forests** - Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. [7]

III. PROPOSED ARCHITECTURE

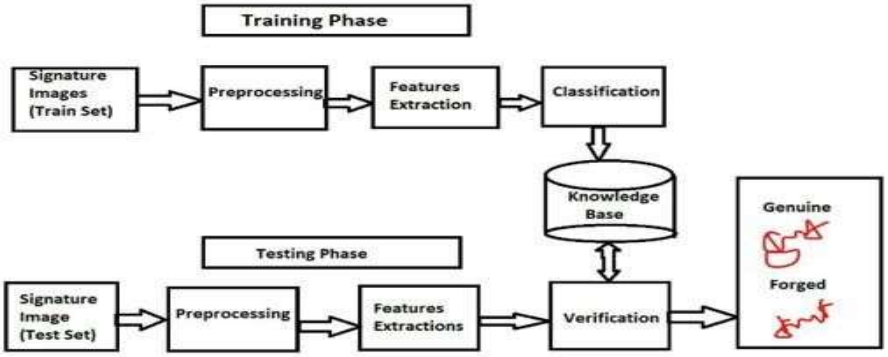

Figure 1 System Architecture

Fig. 1 represents the architecture of proposed handwritten verification system. proposed handwritten signature identification system which identify the authenticity of given signature of a person. The design of a handwriting verification system is divided into two phases: Training phase and Testing phase. A training phase consist of four major steps 1) extraction of a signature image from a dataset 2) pre-processing of image 3) Feature extraction 4) Neural network training. A testing phase consists of five important steps 1) Retrieval of a signature to be tested from a dataset 2) Image pre-processing- Signature preprocessing is a necessary step to improve the accuracy of Feature extraction and Classification and to reduce their computational needs. The purpose of preprocessing phase is to make signatures standard and ready for feature extraction. 3) Feature extraction – The success of a signature verification system greatly depends on Feature extraction. An ideal feature extraction technique extracts a minimal feature set that maximizes interpersonal distance between signature examples of various persons while minimizing intrapersonal distance for those belonging to the same person 4) Application of extracted features to a trained neural network 5) Testing output generated from a neural network.

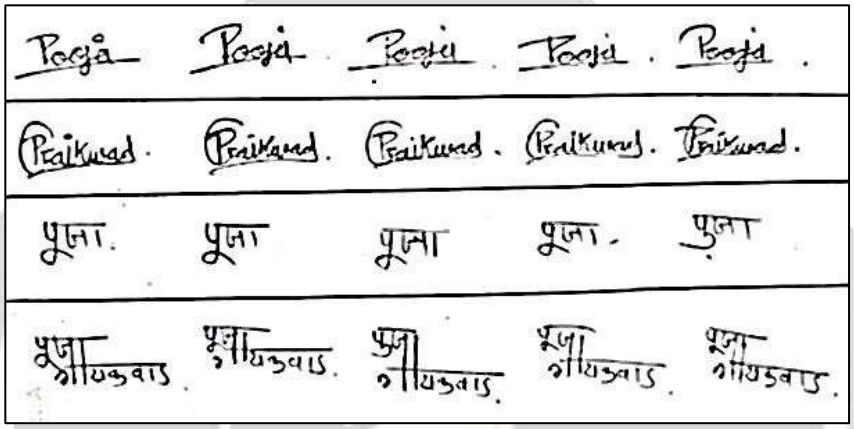

Figure2 Samples in Latin and Devanagari

IV. TECHNOLOGIES

A. Python Programming

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

B. Streamlit

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps. so let's get started! Make sure that you have Python 3.6 - Python 3.8 installed.

C. Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano

V. RESULT

In our work, we binarize the images and store them appropriately. We then do the required file handling and management operations to split the batches of images based on a required split ratio. After the models are built, plots of accuracy and loss are made, we create models for different splits of data and plot the training and validation accuracies to get an idea of the presence of any overfitting or underfitting. On splitting the dataset in the different ration on trial basis, we obtained a maximum accuracy of 97% percent on the validation set as shown in table. There is very little overfitting as the training and testing accuracies are almost equal.

Split Ratio	**Training Accuracy range**	**Validation Accuracy range**
1:2	96 - 99	85 - 92
2:3	95 - 98	89 - 95
7:3	97 – 99.50	95 - 99

Table 1 - Accuracy Table

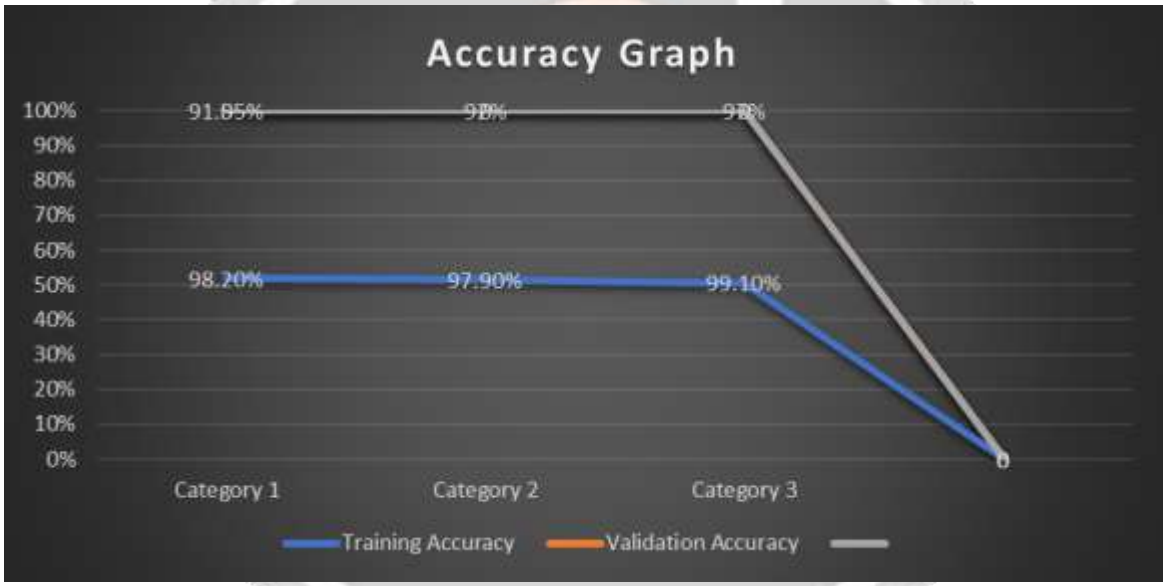

Figure 2 - Accuracy Graph

Figure 3 - Result

VI. CONCLUSION

With the highest accuracy of more than 92.50% and minimal training time of about an hour, it is evident that this offline signature verification with an acceptable level of accuracy has been realized without the need for use of image preprocessing and extra feature selection techniques. The reason why the performance of roughly 80% is ok and acceptable for signature verification is the fact that in banks, for example, a customer writes his signature more than once in the hope that one of the signatures is very close to the one enrolled into the system and so that if the system doesn't correctly identify one, it correctly identifies the other. So, we don't need a system that has to be 100% accurate.

VII. REFERENCES

[1] Handwritten signature recognition method based on fuzzy logic Igor V. Anikin;Ellina S. Anisimova 2016 Dynamics of Systems, Mechanisms and Machines (Dynamics) Year: 2016 | Publisher: IEEE

[2] Vijay More, Madan Kharat, Shyamrao Gumaste, "Segmentation of Devanagari Handwritten Text Using Thresholding Approach", International Journal of Scientific & Technology Research (IJSTR), Volume 9, Issue 3, 2277-8616, March 2020, pp.6594-6605.

[3] Prashant Yawalkar, Dr. M. U. Kharat, Dr. S. V. Gumaste, "Segmentation of Multiple Touching Handwritten Devnagari Compound Characters: Image Segmentation

[4] for Feature Extraction", Feature Dimension Reduction for Content-Based Image Identification, IGI Global, A volume in the Advances in Multimedia Kumar and K. Bhatia, "A robust offline handwritten signature verification system using writer independent approach," 2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall), 2017.

[5] Ertam and G. Aydin, "Data classification with deep learning using Tensorflow," 2017 International Conference on Computer Science and Engineering (UBMK), 2017.

[6] S. Choi and K. Lee, "A CUDA-based implementation of convolutional neural network," 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), 2017.

[7] L. N. Harfiya, A. W. Widodo, and R. C. Wihandika, "Offline signature verification based on pyramid histogram of

oriented gradient features," 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), 2017.

[8] M. S. Chowdhury and M. S. Rahman, "Towards optimal shallow ANN for recognizing isolated handwritten Bengali numerals," 2016 9th International Conference on Electrical and Computer Engineering (ICECE), 2016