International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 01 Jan. 2025, pp: 315-319 www.ijaem.net ISSN: 2395-5252

Recognition of Handwritten Numbers Using Machine Learning Algorithm

Thuy NganNgo[1], Quynh Van Nong[2], MinhDuc Ngo[2,*]
[1]Hanoi Metropolitan University, VietNam
[2]Thai Nguyen University of Education, VietNam
[3]Thai Nguyen University of Technology, VietNam
Corresponding Author: Minh Duc Ngo

Date of Submission: 10-01-2025 Date of Acceptance: 20-01-2025

ABSTRACT: In this article, we present the skewed random forest (rODT) machine learning algorithm for handwritten digital character recognition. We propose to use global features (GIST) to represent digital character images in high-dimensional space. Next, we propose a multi-class random skew forest automatic learning algorithm, each member tree uses a hyperplane that effectively divides data at each node of the tree based on linear discriminant analysis (LDA). Building a random skew tree thus enables the algorithm to work well on high-dimensional data generated from the preprocessing step. Experimental results on the real MNIST data set show that our proposed rODT algorithm recognizes very accurately when compared with current identification methods.

KEYWORDS: Machine learning algorithm, handwritten digital characters.

I. INTRODUCTION

Handwritten digit recognition is necessary and widely applied in many fields such as recognizing digits on bank checks, codes on postal service envelopes, or digits on forms. in general. The problem of handwriting recognition in general and handwritten number recognition in particular is a big challenge for researchers. A big problem always lies ahead because the complexity of handwriting recognition depends heavily on the writing style and language expression of the writer. We cannot always write a character exactly the same way. Therefore, building a handwriting recognition system that can reliably recognize any character in all applications is not easy. Recognition systems usually include two steps: extracting features from images and automatically learning from features to be able to recognize characters. The effectiveness of the identification system depends on the methods used in the above two stages. Most current systems (LeCun et al., 1998), (Simard et al., 2003), (Kégl & BusaFekete, 2009) use basic features from character images such as borders, edges, thickness, gray level value, haar-like, with specific processing such as sampling, pixel oscillation, image transformation, adding virtual data. Then the recognition system trains automatic learning models such as k neighbor (kNN), neural network, support vector machine (SVM), boosting.

The system proposed by the author in the article takes two steps: using global features (GIST) to represent digital character images in a high-dimensional space (960 features, dimensions for each image), training Multi-class random skew forest based on linear discriminant analysis (LDA), effectively recognizes numeric characters. Experimental results on the real data set MNIST (LeCun & Cortes, 1989) show that the author's proposed method, training, recognition is fast and accurate when compared with existing methods. The next part of the article is presented as follows: part 2 briefly presents GIST feature extraction from images, part 3 presents our proposed ODT algorithm. Part 4 presents the experimental results followed by conclusions and development directions.

II. CONTENT

2.1 Feature extraction

In the recognition system, the feature extraction step is very important, greatly affecting the effectiveness of training the automatic learning model. The features extracted from the image must achieve the important goal that based on those features, the learning algorithm can best distinguish one numeric character from another. Pioneering studies in the field of recognition (LeCunet al., 1998), (Simard et al., 2003), (Kégl & Busa-Fekete, 2009) all use basic features from the lowest level to

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 01 Jan. 2025, pp: 315-319 www.ijaem.net ISSN: 2395-5252

price. gray level value of each pixel, borders, edges, thickness, haar-like map organization, and other special processing methods such as sampling, pixel oscillation, image transformation. In recent years, the computer vision and image retrieval research community has been particularly interested in two very effective feature types: locally constant features (SIFT) of (Lowe, 2004) and GIST global representation of (Oliva & Torralba, 2001). The SIFT descriptor vectors extracted from images have important properties: they are not changed by scale, translation, or rotation changes, and are not partially changed by affine geometric transformations (changing change of viewing angle) and tolerance to changes in brightness, shading or noise [1].

However, the rotation invariance of the SIFT feature causes disadvantages for numeric character recognition (numbers 9 and 6 can be the same). Furthermore, the SIFT method provides a very poor number of features from the digital character image (less than 10 features). Meanwhile, using GIST global features is not as difficult as SIFT. For that reason, we use the GIST global feature to solve the problem of handwritten number recognition. The GIST method extracts from images a set of important features such as naturalness, extension, roughness, and solidity, allowing the representation of the spatial structure of a scene. To calculate the GIST descriptor feature, the input image is converted into a square, divided into a 4 x 4 grid, and the histograms in the corresponding direction are extracted. The feature extraction principle is based on Gabor transformation in different directions and frequencies.

Each digital character image is extracted GIST features (960-dimensional vector). After this feature extraction step, the image data set is transformed into a table or matrix in which each image is a row with 960 columns (dimensions), each numeric character is labeled (corresponding class is 0, 1, ... , 9).

2.2. Skewed random forest for multi-class classification

The preprocessing step, extracting features from digital character images creates a large-dimensional data set. The next selected classification algorithm must be able to handle large dimensional data well. In a previous study in (Do et al., 2009), we proposed the RF-ODT skew random forest algorithm for effective classification of high-dimensional data. This is an extension of RF-CART proposed by (Breiman, 2001). The effectiveness of a learning algorithm as researched by (Breiman, 1996, 2001) is based on two error components: bias and variance, where the bias error component is the error of the learning model compared to Bayes and Variance is the error due to the variability of the model compared to the randomness of the data samples. In the study, combining multiple classification models into a set of classification models gives higher accuracy than just a single model. Breiman's RF-CART algorithm builds a set of highly efficient and diverse decision trees (with low correlation between member trees). To keep bias low, RF-CART builds trees to maximum depth without cutting branches. To keep correlation between trees low, RF-CART uses bootstrap sampling from the original data set to build a membership tree and randomly selects a subset of attributes to calculate. Best partitioning at the inner nodes of the tree. RF-CART provides high accuracy compared to the best current classification algorithms including Boosting (Freund & Schapire, 1995), SVM (Vapnik, 1995).

Furthermore, it learns quickly and tolerates interference well. However, RF-CART's tree construction only chooses one dimension to partition data at nodes as previously suggested (Breiman et al., 1984), (Quinlan, 1993). Therefore, the accuracy of the tree model is reduced when working with data sets with large dimensions and interdependence. For example, in Figure 1, any single-attribute partition (parallel to a coordinate axis) cannot completely separate the data once into two classes and must be partitioned multiple times. , but multidimensional partitioning (skew, combining two attributes) can be done perfectly in just one pass. Therefore, the single-attribute partitioning used to build a regular tree is not effective in this case [2].

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 01 Jan. 2025, pp: 315-319 www.ijaem.net ISSN: 2395-5252

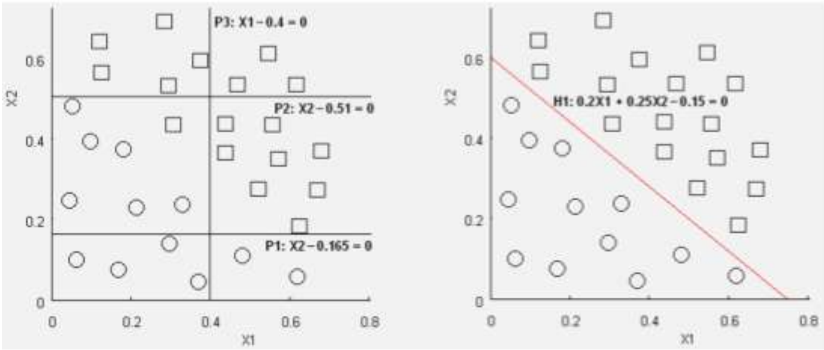

Fig. 1. Single-attribute partition (left), multi-attribute partition (right)

To overcome the above drawback, many decision tree construction algorithms using multi-attribute partitioning (skew) at the nodes are proposed. (Murthy et al., 1993) introduced the OC1 algorithm, a system used to build skew decision trees that uses a hill climbing algorithm to find a good skew partition in the form of a hyperplane. The problem of constructing an optimal skewed decision tree is known to be an NP-hard problem. RF-ODT of (Do et al., 2009) builds random skew trees based on the optimal hyperplane (highly efficient partitioning, good noise tolerance) obtained from SVM training. However, finding the optimal hyperplane by SVM, although effective, is highly complex. To reduce the complexity of the implementation, we propose to replace SVM by linear discriminant analysis LDA, and extend it to the multi-class (greater than 2) classification problem. Considering the problem of binary classification (2 classes), the main idea of LDA (Fisher, 1936) is to find the hyperplane such that when projecting data onto it, the isolation between the data averages of the 2 classes is maximum and The overlap between the two layers is minimal [3].

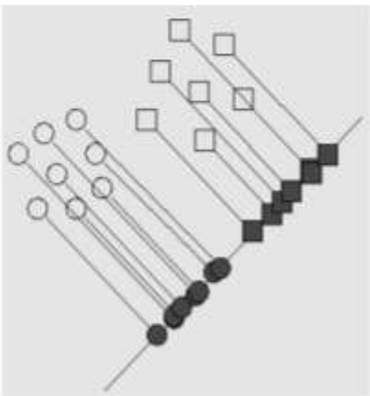

Fig. 2. Illustration of vector (w) used to project 2-dimensional data

Briefly, consider an example of linear binary classification (circle, square) as shown in Figure 2, with m data points xi (i=1,m) in n-dimensional space. The data set is divided into 2 classes R1 (with N1 elements), and R2 (with N2 elements).

LDA finds the optimal hyperplane w for skew partitioning by performing the above linear equations. Note that the hyperplane is partitioned many times by the skewed decision tree up to the leaf node, not just one partition. For that reason, when the linear isolation of the data does not rely on the two centroids m1 and m2 (the case of nonlinear data), the skew tree can still handle this situation. From the beginning of presenting the algorithm until now, we have only focused on the problem of binary classification (2 classes). To expand the algorithm for multi-class classification problems (with more than 2 classes). The main problem is that we have to reduce the problem to a 2-class form so that we can re-implement LDA as described above. To do this, we propose a hierarchical model. Suppose at a node of the skew tree, we have c classes (c > 2). We propose to create two layers (positive layer and negative layer), where each layer contains data from other layers. That is, data from classes close to each other are grouped into one of two classes, positive and negative. At this point, the data at a node has returned to the binary classification problem, we can apply the LDA formula above.

The process continues until the data is completely partitioned. Our skewed random forest algorithm (rODT) proposed for the problem of classifying a data set of m data points xi (i=1, m) in n-dimensional space, is implemented as described in Figure 3. An oblique decision tree (denoted as ODTk) in a random forest consisting of k trees is built as follows: The learning data set Bk is m data elements sampled with return from the original data set. At each node of the tree, randomly select n' dimensions (n'<n) and calculate a skew partition (using LDA as described above) based on these n' dimensions. The tree is built to maximum depth without cutting branches. Oblique random forest

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 01 Jan. 2025, pp: 315-319 www.ijaem.net ISSN: 2395-5252

rODT classifies element x based on the majority vote from the obtained classifications of member trees [4].

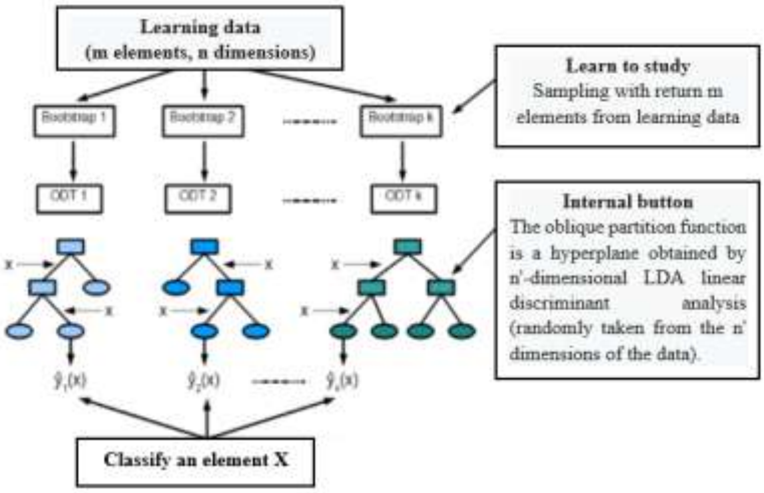

Fig. 3. Skewed random forest algorithm

III. EXPERIMENTAL RESULTS

In the experimental part, we use the data set provided by MNIST (LeCun & Cortes, 1989), which is often used to evaluate the effectiveness of handwritten digital character recognition algorithms. The MNIST data set is derived from the NIST data set provided by the National Institute of Standards and Technology (NIST), then updated by LeCun and divided into 2 separate sets. The learning (training) set includes 60,000 28 x 28 images of handwritten digits used to train an automatic machine learning model. All images in the learning set are aligned and transformed into point data consisting of 60,000 elements (numeric characters) with 784 dimensions which are the gray level values of the points, 10 classes (from 0 to 9). The test set consists of 10,000 images of handwritten digits used for testing, similarly the images in the test set are also transformed. To be able to evaluate the effectiveness of the proposed method (rODT, GIST), we used the program of (Douze et al., 2009) to extract features and at the same time we also installed the rODT algorithm. in C/C++ programming language. We compare the effectiveness of (rODT and GIST) with current algorithms such as AdaBoost.M1 algorithm (Freund & Schapire, 1995), (Witten & Frank, 05), LibSVM (Chang & Lin, 2001), (Vapnik, 1995), CNN convolutional neural network (Simard et al., 2003), (O'Neill, 2006). All results were performed on a personal computer (Intel 3GHz, 2GB RAM) running the Linux operating system. The resulting accuracy is as shown in Table 1. Reference results from the methods of (LeCun et al., 1998), (Kégl & Busa-Fekete, 2009) are also presented in the table.

No	Method	Accuracy (%)
1	1-layer Neural nets (LeCun *et al.*, 1998)	88.00
2	Nearest-neighbor (Euclidean L2) (LeCun *et al.*, 1998)	95.00
3	Convolution net LeNet-1 (LeCun *et al.*, 1998)	98.30
4	Convolution net LeNet-4* (LeCun *et al.*, 1998)	98.90
5	Convolution net LeNet-5* (LeCun *et al.*, 1998)	99.15
6	Convolution Neural Net (CNN)* (Simard *et al.*, 03)	99.10
7	LIBSVM (RBF, $\gamma = 0, 05$, $c = 10^5$)	98.37
8	LIBSVM (Poly, deg = 5, $c = 10^5$)	96.65
9	AdaBoost.M1 (100 trees with C4.5)	95.95
10	Products of boosted stumps (haar)* (Kégl & Busa-Fekete, 2009)	99.12
11	rODT (100 oblique decision trees, GIST)	99.12

Table 1. Results of recognizing the Mnist handwritten character set

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 01 Jan. 2025, pp: 315-319 www.ijaem.net ISSN: 2395-5252

IV. CONCLUSION

We have just presented the skewed random forest machine learning algorithm (rODT) using global features (GIST), allowing accurate recognition of handwritten characters. The preprocessing step extracts global features from the digital character image to produce a data table with large dimensions. We propose a multi-class random skew forest automatic learning algorithm, each member tree uses a hyperplane that effectively divides data at each node of the tree based on linear discriminant analysis (LDA). Experimental results on the real MNIST data set show that our proposed rODT algorithm recognizes very accurately when compared with current identification methods. The proposed method achieves high recognition accuracy but does not require any special processing. Experiments for handwritten character recognition including numeric characters and 26 alphabet characters show that our method is really good. In the near future, we will combine this system with other methods that allow extracting and reading vehicle numbers. The approach can be applied to similar problems in the fields of recognition, classification, and image search.

REFERENCES

[1]. L. Breiman, J.H. Friedman, R.A. Olshen and C. Stone. Classification and Regression Trees. Wadsworth International, 1984.
[2]. L. Breiman. Bagging predictors. Machine Learning 24(2):123–140, 1996.
[3]. L. Breiman. Random forests. Machine Learning 45(1):5–32, 2001.
[4]. C.C. Chang and C.J. Lin. Libsvm – a library for support vector machines. 2001. http://www.csie.ntu.edu.tw/cjlin/libsvm.