Journal of Theoretical and Applied Information Technology
15th September 2022. Vol.100. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CHINESE CHARACTER RECOGNITION USING SUPPORT VECTOR MACHINE

ASLINA BAHARUM[1a*], ROZITA ISMAIL[2], SHALIZA HAYATI A. WAHAB[1b], FARHANA DIANA DERIS[3], NOORSIDI AIZUDDIN MAT NOOR[4], MOHD SHAREDUWAN MOHD KASIHMUDDIN[5]

[1]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Sabah, Malaysia
[2]College of Computing and Informatics, Universiti Tenaga Nasional, Putrajaya Campus, Malaysia
[3]Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, Malaysia
[4]UTM CRES, Faculty of Built Environment and Surveying, Universiti Teknologi Malaysia
[5]School of Mathematical Sciences, Universiti Sains Malaysia, Minden, Penang, Malaysia

E-mail: [1a]aslina@ums.edu.my, [2]irozita@uniten.edu.my, [1b]shaliza@ums.edu.my, [3]diana@utm.my, [4]noorsidi@utm.my, [5]shareduwan@usm.my

ABSTRACT

Optical character recognition is the art of scanning and detecting the word in the images so that the machine can identify and classify the character. Chinese characters are one of the world's most widely used writing systems. It is used by more than one-quarter of the world's population in daily communication. Chinese characters can be considered difficult because they have many categories, complex character structures, similarities between characters, and various fonts or writing styles. There are many known machine learning algorithms for character recognition, but not all can classify Chinese characters with high speed and accuracy. Therefore, this paper proposes recognizing Chinese characters using support vector machines. Support vector machines are a classification of two classes widely used in classification. It produces very accurate results for many classes, making it suitable for recognizing Chinese characters.

Keywords: *Support Vector Machine, Optical Character Recognition (OCR), Character, Classifier, Feature*

1. NTRODUCTION

One of the most difficult fields of image processing and pattern recognition study is optical character recognition [1]. Character recognition aims to identify characters from the image [2]. It stimulates human reading capabilities and scans through the texts so that the computer can understand the word and perform many tasks, like editing or translating the text to other languages. In order to obtain character recognition results more quickly, high-speed pattern matching algorithms are needed to develop real-time character recognition applications. Many machine learning algorithms can be used for character recognition; however, not all machine learning algorithms can detect Chinese characters fast and accurately. Therefore, three machine learning algorithms are discussed in the literature review to determine which algorithms are suitable for this study.

One of the world's most extensively used writing systems is that of Chinese characters [3]. It is used in daily communication by more than a quarter of the world's population, especially in Asia, such as China, Korea, Japan, and Singapore [4]. More than 100,000 Chinese characters are available, with GB2312 being the most widely used official character group and 6763 simplified Chinese characters [5]. Unlike the English character, which contains 52 letters, including uppercase and lowercase, it is less complicated and more accessible to recognize than the Chinese character. Due to several problems, the recognition of Chinese characters was thought to be difficult. First, it has a vast number of categories. All Chinese characters have a specific category to categorize them and identify characters. However, some characters contain other categories' characteristics, making it hard to categorize them. For example, the word 米 should be under the category of 八, but it also seems

to have the characteristic of the category 十 as well. Next, some Chinese characters have a very complicated structure. The most complex Chinese character is, made up of 57 strokes. Third, some of the characters look similar to each other. For examples 已 (yi) and 己 (zi), 土 (tu) and 士 (shi). Although it seems similar to each other, the structure of the character is different. And last but not least, there is a variability of fonts and writing styles in Chinese. In Chinese calligraphy, the five major ones are Zuan, Li, Tsao, Hsin, and Kai. The Chinese character could look slightly or completely different in each style and font style, which might affect the system's identification as another character [6]. Therefore, it requires fast and accurate classification for character recognition as it has thousands of characters to search for.

2. LITERATURE REVIEW

2.1 Support Vector Machine

The Support Vector Machine (SVM) is a recent addition to the artificial intelligence toolkit and is used for classification [7]. SVM has gained momentum in classification because of its accuracy, durability, and indifference to input data types. It produces very accurate results for a large number of classes. Even though the data are implicitly transformed into a high-dimensional feature space (reproducing Kernel Hilbert spaces, or RKHS), they behave as expected [8].

When solving nonlinear problems, kernel functions are used to map data to higher dimensions to find the decision surface so that we can divide it into two classes, as shown in Figure 1. This kernel is the mathematical function used by SVM algorithms to take the data as input and transform it into a higher dimensional form. The linear SVM algorithms use four general kernel functions: polynomial, radial basis function (RBF), and sigmoid.

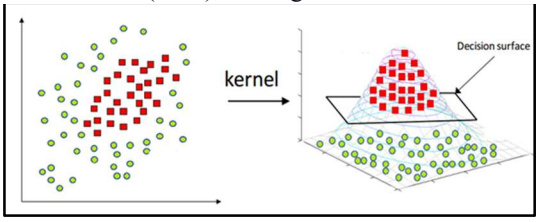

Figure 1: Transformation of the Dataset to Higher Dimensional

The linear kernel is the most straightforward kernel function. The equation of Linear Kernel is k(x,y)=xTy. The polynomial kernel is popular in image processing. Additionally, it is used for issues where all training data has been standardized. The polynomial kernel's equation is k(x,y)=(xTy+1)d, where d is the polynomial degree. The Radial Basis Function (RBF) is the general-purpose kernel. It is the most popular and widely used kernel function when there is no prior knowledge about it. The equation of the RBF is: k(x,y)=exp(-γ||x−y||2). The γ parameter needs to be adjusted carefully since it defines how much influence a single training example has. The sigmoid kernel can be used as the proxy for a neural network. The equation of the sigmoid kernel is k(x,y)=tanh(κxTy+θ).

2.2 K-Nearest Neighbour

K-nearest neighbour (KNN) is a type of non-parametric example-based classification in which all calculations are postponed until after classification, and the function is only addressed locally [9]. KNN is a fundamental categorization method that works best when the user has little to no prior knowledge of how to disseminate data. This is a lazy algorithm, meaning it does not use training data points to make generalizations. All samples are stored in the training data. The algorithm is based on an estimation of the closest neighbourhood. The new cases are classified on their most comparable neighbour class based on a similarity measure, the distance metric [10]. Figure 2 shows the KNN architecture.

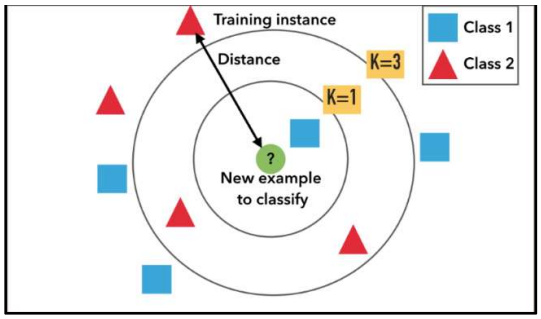

Figure 2: K-Nearest Neighbor Architecture

Strong and noisy training data, effectiveness with massive amounts of training data, no need for a training phase, and ease of learning

complex models are all benefits of KNN. However, on the other hand, the disadvantage of KNN is that we have to specify the parameter k, which is the number of nearest neighbours. It is difficult to apply in high dimensions. High dimensional data causes false intuition, low computing efficiency, data scarcity, a greater need for data and storage, and a higher amount of data to be processed. As a result, the nearest neighbour is no longer close. The distance between data objects becomes less clear. It is not apparent from distance-based learning what kind of distance should be used or which characteristics would lead to the best outcomes. Because we must determine the distance between each query sample and each training sample, the calculation cost is considerable.

2.3 Multilayer Perceptron

A particular kind of feed-forward multilayer artificial neural network is the multilayer perceptron (MLP). It has an output layer, an undetermined number of hidden layers, and input and output layers [11]. Each layer has some nodes known as neurons; the neurons are connected to the neurons of the next layer. These neurons' network connections may be partially or fully connected [6].

The MLP developed a model for the correlation (or dependency) between a set of input-output pairs during training. Each MLP unit conducts several biassed weighted inputs before passing this activation stage through a transfer function to create an output. To reduce error, the model's parameters, or the weights and biases, are adjusted during the training of MLP. The logistic and hyperbolic tangent sigmoid functions are the most often used activation functions in MLP [6]. Figure 3 shows the architecture of a typical MLP.

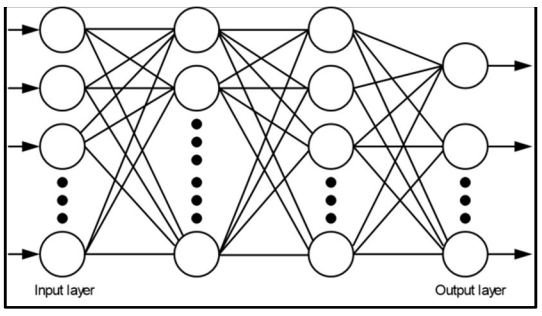

Figure 3: Architecture of a Typical MLP

3. RELATED WORKS

According to Ramanathan et al. [12], the SVM-based English character recognition model performs admirably. Tamil achieved an average accuracy of nearly 99% throughout five iterations when training data for new testing data. Tamil reached an accuracy of roughly 84% in the fourth iteration. The Gabor filter features are extracted during the training phase and entered into the SVM along with their corresponding class values. In the testing phase, the Gabor filter extracts the features, and the SVM evaluates those features and assigns the corresponding class values. In a text file, the values of this class can be instantly transformed to the corresponding characters or ASCII values. In order to train the model, 94 symbols from the data set—14 for English—included letters (lowercase and uppercase), numbers, and symbols in 10 various font sizes.

Additionally, all font styles—bold, italic, regular, and italic—are added so the model can be trained for every circumstance. The SVM model was trained using a data set of 156 symbols, comprising letters, numbers, and 12 different font sizes of 14 sizes, for the Tamil language. Tamil and English SVM models are trained independently. Following the training phase, it is tested by presenting test images using various typefaces. At the end of the research, they achieved an average accuracy of almost 99% over five iterations using the training data and an accuracy of 97.78% in the fifth iteration using the testing data in recognizing the English character.

Besides, Ramanathan et al. [7] also proposed another method of extracting the feature vectors by using the Gabor filters and used SVM in training by using these features. The main objective of this research was to identify various fonts using SVM. They trained the suggested model using six commonly used English fonts: Times New Roman, Arial, Comic Sans MS, Courier New, Algerian, and Tahoma. The four standard font styles were bold, italic, and bold-italic. Initially, 216 samples were used in the model training phase, and then 432, 648, and 864 samples were added throughout time. At the end of the research, they achieved good accuracy, with an average of around 95%, even with 216 samples using the SVM model to recognize the English character of various font styles. Due to

excessive learning, there is little variation in accuracy for larger event outcomes.

Furthermore, Naik and Desai [13] presented an online handwritten recognition system for the Gujarati language. They used SVM, K-Nearest Neighbor, and multilayer perceptron to classify the strokes using a hybrid feature set. Three kernel functions are used for the SVM: linear, polynomial, and RBF. In the KNN classifier, they tested with three different values, which are k=5, k=7, and k=9. They use different classifications using different parameter values to compare results in terms of accuracy and execution time. The training sample used a data set of 3000 samples collected from different authors from various groups and gender levels to train the system and then tested by 100 different authors after the system was trained and tuned. Before classifying the Gujarati character, the characters undergo a pre-processing process and feature extraction. The method used for pre-processing is normalization, smoothing, and resampling. To eliminate size variance across all strokes, the bilinear interpolation approach is utilized to equalize the height and width of a stroke. After that, smoothing takes the noise out of a stroke.

Last but not least, resampling is employed to eliminate the impact of writing stroke speed change to resample the recorded stroke coordinates. After obtaining the pre-processed coordinates, they extract a feature—a special piece of data specific to each stroke—using hybrid feature extraction techniques. Once the extracted feature got, it will be used to classify the characters. Utilizing the SVM with RBF kernel, they outperformed all other classifiers in accuracy, achieving a 91.63% accuracy with an execution time per stroke of 0.063 seconds. The accuracy rate for employing the MLP is the lowest, at 86.72%, and the processing time was as long as 1.062 seconds on average.

4. RESULT AND DISCUSSION

The proposed methodology for training the SVM model is in figure 4. Once the training phases are completed, the SVM model will be tested with a new set of testing data.

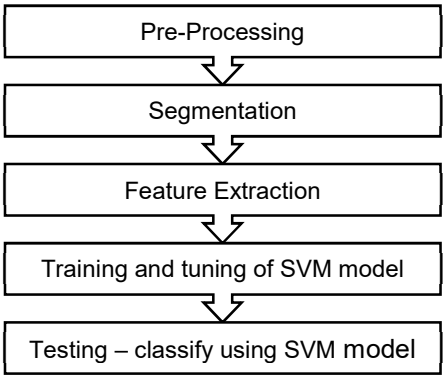

Figure 4: Flow Chart of Training and Testing SVM Model

4.1 Pre-processing

Since the dataset is obtained from the handwritten Chinese character database, no scanning process is required. The images from the database will be normalized, and skewness will be corrected to reduce skewness. After that, the images will then convert to binary images.

4.2 Segmentation

Once the image is processed, segmentation will be taken to extract all the characters from the image. The following phases are performed in the process of segmentation, line segmentation and character segmentation. Inline segmentation, each row in the document is divided using a horizontal profile. In character segmentation, eight connected component analyzes are used to divide each character into a row.

4.3 Feature Extraction

Every character contains some unique information known as features. We can use different methods to extract these features to classify the characters. In this research, the Gabor filter will be used to do feature extraction. After the image is segmented, the image will be resized to a 64x64 binary image. Then the images are passed through a bank of 24 Gabor filters to obtain the characteristic vectors of each image.

4.4 Training and Turning

After obtaining the feature vector, the feature vector is added with its class value. These vectors are assigned to the SVM to draw the optimum yield limit with the maximum distance to the nearest point, also known as the support vector, in the data set. Data sets are converted into higher dimensional spaces using a

radial base function kernel (RBF) to improve accuracy in classifying characters. In addition, the RBF kernel function parameters must be adjusted to obtain the highest accuracy. Any algorithm optimization can also be used to find the best value for a parameter.

4.5 Testing

Once the training and tuning process is complete, the SVM model is ready to classify the new inputs of Chinese characters. The new test image will undergo a process of pre-processing, segmentation, and feature extraction. But this time, after the feature vector is obtained, it goes through the improved SVM model to get the model results.

5. CONCLUSION

Chinese characters are complex as they are not written in alphabets that most people know. People who cannot read and understand Chinese characters, especially foreign tourists who travel to places that mainly use the Chinese language, need translation software. Fast and accurate translation software is ideal for helping break the language barrier to a certain extent.

The proposed paper identifies the Chinese character using the support vector machine (SVM) model. The performance and accuracy of the Chinese character recognition system will be evaluated after the SVM model's output is obtained. In future works, the dataset from the HCL2000 database [14] will be collected to develop and train the SVM model. Once the SVM model is finely trained and tuned, it will then collect a new dataset to test the SVM model. In addition, other feature extraction techniques can be applied to extract the character's features for modelling training and comparing the classification accuracy of various feature extraction techniques. The finely tuned SVM model will then be useful to be implemented in modern communication technologies.

REFERENCES:

[1] Majed Valad Beigi, "Handwritten Character Recognition Using BP NN, LAMSTAR NN and SVM," Northwestern University (Evanston, Illinois), 2015. http://users.eecs.northwestern.edu/~mvb541/EECS349/Report.pdf.

[2] Savitha Attigeri, "Neural Network based Handwritten Character Recognition system," *International Journal Of Engineering And Computer Science*, Vol. 7, No. 03, 2018, pp. 23761–23768, doi: 10.18535/ijecs/v7i3.18.

[3] Xu-Yao Zhang, Fei Yin, Yan-Ming Zhang, Cheng-Lin Liu, and Yishua Bengio, "Drawing and Recognizing Chinese Characters with Recurrent Neural Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, 2018, pp. 849–862, doi: 10.1109/TPAMI.2017.2695539.

[4] Ruwei Dai, Chenglin Liu, and Baihua Xiao, "Chinese character recognition: history, status and prospects," *Frontiers of Computer Science in China*, Vol. 1, No. 2, 2007, pp. 126–136, doi: 10.1007/s11704-007-0012-5.

[5] Zhouhui Lian and Jianguo Xiao, "Automatic shape morphing for Chinese characters," *SIGGRAPH Asia 2012 Technical Briefs on - SA '12*, 2012, doi: 10.1145/2407746.2407748.

[6] Gurpreet Singh and Manoj Sachan, "Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition," *IEEE Xplore*, Dec. 01, 2014. https://ieeexplore.ieee.org/document/7238334.

[7] R. Ramanathan, K. P. Soman, L. Thaneshwaran, V. Viknesh, T. Arunkumar, and P. Yuvaraj, "A Novel Technique for English Font Recognition Using Support Vector Machines," *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, 2009, doi: 10.1109/artcom.2009.89.

[8] Andreas Ch. Braun, Uwe Weidner, and Stefan Hinz, "Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — A comparison," *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2011, doi: 10.1109/whispers.2011.6080861.

[9] M. Manjusha and R. Harikumar, "Performance analysis of KNN classifier and K-means clustering for robust classification of epilepsy from EEG signals," *IEEE Xplore*, 2016, pp. 2412–2416, doi: 10.1109/WiSPNET.2016.7566575.

[10] T. Ananda Babu and P. Rajesh Kumar, "Characterization and classification of uterine magnetomyography signals using KNN classifier," *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, 2018, doi: 10.1109/spaces.2018.8316337.

[11] O. Batsamhan and Y. P. Singh, "Mongolian character recognition using multilayer perceptron (MLP)," *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, Vol. 2, 2002, doi: 10.1109/iconip.2002.1198132.

[12] R. Ramanathan, S. Ponmathavan, N. Valliappan, L. Thaneshwaran, A. S. Nair, and K. P. Soman, "Optical Character Recognition for English and Tamil Using Support Vector Machines," *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, 2009, doi: 10.1109/act.2009.155.

[13] Vishal A. Naik and Apurva A. Desai, "Online handwritten Gujarati character recognition using SVM, MLP, and K-NN," *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2017, doi: 10.1109/icccnt.2017.8203926.

[14] H. Zhang, J. Guo, G. Chen and C. Li, "HCL2000 - A Large-scale Handwritten Chinese Character Database for Handwritten Character Recognition," 2009 10th International Conference on Document Analysis and Recognition, 2009, pp. 286-290, doi: 10.1109/ICDAR.2009.15.