Handwriting Recognition on Form Document Using Convolutional Neural Network and Support Vector Machines (CNN-SVM)

Conference Paper · May 2017

DOI: 10.1109/ICoICT.2017.8074699

Handwriting Recognition on Form Document Using Convolutional Neural Network and Support Vector Machines (CNN-SVM)

Darmatasia and Mohamad Ivan Fanany
Machine Learning and Computer Vision Laboratory
Faculty of Computer Science, University of Indonesia
Corresponding Authors: darmatasia@ui.ac.id, ivan@cs.ui.ac.id

***Abstract*—In this paper, we propose a workflow and a machine learning model for recognizing handwritten characters on form document. The learning model is based on Convolutional Neural Network (CNN) as a powerful feature extraction and Support Vector Machines (SVM) as a high-end classifier. The proposed method is more efficient than modifying the CNN with complex architecture. We evaluated some SVM and found that the linear SVM using L1 loss function and L2 regularization giving the best performance both of the accuracy rate and the computation time. Based on the experiment results using data from NIST SD 19 2nd edition both for training and testing, the proposed method which combines CNN and linear SVM using L1 loss function and L2 regularization achieved a recognition rate better than only CNN. The recognition rate achieved by the proposed method are 98.85% on numeral characters, 93.05% on uppercase characters, 86.21% on lowercase characters, and 91.37 on the merger of numeral and uppercase characters. While the original CNN achieves an accuracy rate of 98.30% on numeral characters, 92.33% on uppercase characters, 83.54% on lowercase characters, and 88.32% on the merger of numeral and uppercase characters. The proposed method was also validated by using ten folds cross-validation, and it shows that the proposed method still can improve the accuracy rate. The learning model was used to construct a handwriting recognition system to recognize a more challenging data on form document automatically. The pre-processing, segmentation and character recognition are integrated into one system. The output of the system is converted into an editable text. The system gives an accuracy rate of 83.37% on ten different test form document.**

Keywords—handwriting recognition, form document, CNN, SVM

I. Introduction

Handwriting recognition is a crucial issue in machine learning and computer vision. A variety of techniques and methodologies have been proposed but it still an unresolved issue [1]. However, it is a challenging task especially a handwriting recognition on form document. It comes more complicated than only character recognition. Some noise like a bounding box on form document makes the handwriting recognition systems more complex [2].

Some problems in handwriting recognition are due to the high uncertainty of the input data, as the written characters of each person are different, some characters have a very similar shape, disconnected or distortion characters, the written characters have a different thickness and use of various scanners [3].

Pattern or object recognition is usually done with feature extraction and classification. The feature extraction typically uses a variety of methods to get a representation of the data and then use the classifier to classify the data. The process is conducted manually and separately. Lately, the feature extraction and classification integrated automatically in one process or method. The method used to model high-level abstractions in data [4]. It is often called as deep learning techniques.

Convolutional Neural Network (CNN) is one of the deep learning architecture. It can extract multiple features from low-features to high-features automatically [5]. Currently, CNN is a state of the art of handwriting characters recognition. In [6] reports an accuracy rate of 99.59% on MNIST dataset with modifying the CNN using two training feedback, that is reconstruction feedback and classification feedback.

Some studies modify the CNN with various approaches to improving the accuracy rate and the practice time. Some of them modify the input data to improve the accuracy rate of CNN method. Mega *et al.* [7] have combined GLCM (Gray-Level Co-occurrence Matrix) and CNN for cattle classification. Feature Extraction uses a GLCM method to extract contrast, energy, and homogeneity feature of the image and it is used as input of CNN. Kwolek [8] has modified the architecture of CNN by combining CNN and Gabor filter to detect the facial region. The proposed method gives classification performance better than original CNN.

Modification of CNN related to handwriting recognition mostly conducted with modify the architecture of CNN. Some studies construct a more complex structure to improve the accuracy rate. In [9] propose a sparse CNNs or called DeepCNet for online handwriting character recognition. The pictures are enhanced with signature information, and the sparsity is used to increase the depth of network. It makes a slow speed that allows for the retention of more spatial information. It makes the improvement of generalization ability. Inspired by previous research by [9], Yang [10] also propose a Deep Convolutional Neural Networks (DCNNs). The proposed method uses a Hybrid Serial Parallel (HSP) strategy to ensemble some architecture of DCNN, and it improves the accuracy performance. However, this method is time-consuming. Lately, the dropout strategy becomes a popular technique to prevent the overfitting on the convolutional network. It drops the activations randomly [11]. Other research [12] uses drop-connect strategy to drops the weights randomly. It gives a good performance in accuracy rate but slows down the convergence.

Elleuch [13] has conducted research which slightly different from the approaches previously mentioned. In [13] combines a CNN with another end classifier. The kernel SVM approach using as an end classifier for handwritten Arabic character recognition. The slightly different with the research by [13] are the architecture of CNN and the kind of SVM approach. We used the SVM classifier as an end classifier because it has better generalization ability than neural network on standard CNN. It caused by the SVM classifier working with the Structural Risk Minimization (SRM) principle while the neural network working with the Empirical Risk Minimization (ERM) principle [14].

In this study, we use a linear SVM approach as end classifier to reduce the computation time. The linear SVM is more efficient than kernel SVM both of memory usage and training time [15][16]. It is caused by the linear SVM working in the original dimensions space of the input data while the kernel SVM is mapping the original of input data to a new higher dimensional feature space to identify the class of the input data [16]. In most cases, the use of kernel function on SVM gives accuracy performance better than linear SVM. However, in some cases, the linear SVM outperformed kernel SVM such as for document classification and large dataset [16]. To improve the accuracy performance of linear SVM, we add a regularization to prevent the overfitting during the learning process [17]. In this research, we use L2 regularization that has an analytical solution. The solution of L2 regularization can be calculated computationally efficiently than L1 regularization [18].

The loss function is an important part of learning process. It aims to minimize the objective function. Two popular loss functions are an L1 loss or commonly known as Least Absolute Deviation (LAD) and L2 loss or called Least Square (LS). In this study, we use L1 loss which most robust and resistant to outliers in the data compared with L2 loss [18]. To reduce the training time of linear SVM we use a dual coordinate descent approach [19]. It is faster than ordinary SVM approach where the cost time of training standard SVM is $O(n^3)$, n is training set size [20] while the cost time of [19] is $O(\log(1/\epsilon))$ iterations.

In this research, we use the proposed method to construct a handwriting recognition system on form document automatically. In computer science, a handwriting recognition on a form document also known as Intelligent Character Recognition (ICR). ICR is part of Optical Character Recognition (OCR) used to recognize the handwritten character which written as a single character in the cell [21].

II. Related Work

Some researches have been conducted to develop a variety of methods and algorithms that can be used to recognize a handwritten character.

Hadi *et al.* [22] have conducted research related to handwriting recognition on form document. He uses Freeman Chain Code with the division of the region into nine regions and histogram normalization of chain code as feature extraction. In [22] also proposes four visual feature consisting of top-ratio, right-ratio,wide-ratio, and high-ratio of characters. k-Nearest Neighbour (k-NN) and Artificial Neural Network (ANN) are used to classify the character. Jindal *et al.* conducted research to classify the printed characters and the handwritten characters on form document [2]. It uses Inter-Character Gap (ICG), character height, and baseline features and classification rules to classify the input data.

In paper [23] uses Zernike Moments to extract the feature of numeral and mathematical operators and SVM for classification. In [24] propose Freeman Chain Code to remove a feature from NIST dataset which consisting of uppercase, lowercase, and merger of uppercase and lowercase. Hallale *et al.* [25] propose a directional method for feature extraction on English handwritten characters. The data are classified base on the similarity between the vector feature of data training and the vector feature of data testing.

III. Data and Methodology

In this part, we explain the data and the proposed method used in this study. The method of this research shown in Figure 1.

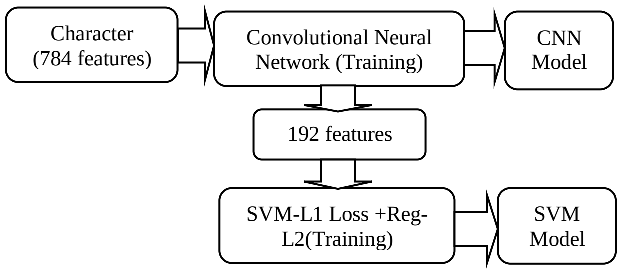

Fig. 1. Research methodology of handwriting recognition

A. Dataset

In this research, we use the NIST SD 19 2nd edition dataset [26] both for training and testing. It consists of numeral, uppercase, lowercase and merger of uppercase and lowercase. The examples of NIST dataset can be seen in Figure 2.

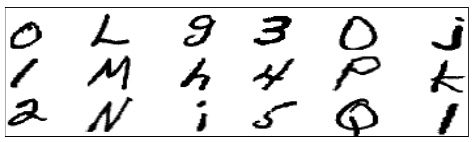

Fig. 2. Examples of the NIST dataset [26]

The original size of the dataset is a 128x128 pixel. Some preprocessing like cropping and image resize conducted on NIST dataset. We crop the character to remove the uninformative background of the image and resize it into the 28x28 pixel.

We also distribute form document to ten different people to test the handwriting recognition system. We use the identity application form for a citizen of Indonesia. In this research, the system only recognizes the characters on Region Of Interest (ROI). It can be seen in Figure 3.

Fig. 3. Application form identity for citizen of Indonesia

B. *Proposed Method*

The proposed method uses a CNN as a powerful feature extraction to extract the feature of input character and a linear SVM for end classifier. CNN was introduced by Lecun and Bengio in the paper [5]. It gives a good performance for local spatial correlation on image [7][27] and has less training [28]. The performance of CNN is explained in the paper [7][27][29].

In general, CNN uses three basic ideas: local receptive fields, shared weight, and pooling. Each local receptive field in the input layer is connected to one neuron in the next layer (hidden layer). Each connection has a weight and single-bias which shared to other local receptive field in the same feature map. Feature map is a map from the input layer to the hidden layer. The weight and bias of each feature map are different thus allowing some of the features can extract from any position [7].

Convolutional Neural Network also contains pooling or sub-sampling layer. The sub-sampling layer is usually used immediately after convolutional layer. Sub-sampling layer simplifies the convolutional layer output information [30]. In the CNN architecture, the last layer is a fully-connected layer. The architecture of fully-connected layer is the same with Multilayer Neural Network. However, a few modifications of backpropagation for training parameters in convolutional and sub sampling-layers are needed.

The proposed architecture of CNN to extract the feature of handwritten character recognition shown in Figure 4. It is a standard architecture of CNN. The network contains five layers. The first four are two sets of convolutional and sub-sampling layers, followed output layer which fully connected layer.

The output of the first convolutional layer has six feature maps with 24x24 pixels and the second convolutional layer has 12 feature maps with 8x8 pixels. The size of kernel or filter which used for convolution is 5x5. The output of the first sub-sampling layer is 12x12 with six feature maps and the second is 4x4 with 12 feature maps. The output of the second sub-sampling layer is transposed into vector feature, and its size is 1x192 features. It is used as input to train the SVM classifier.

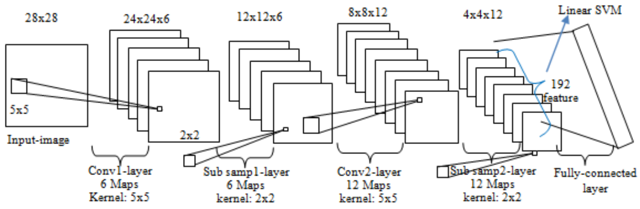

Fig. 4. Architecture of CNN for feature extraction

SVM first introduced by Vapnik [31]. It widely used for classification and regression. The basic idea of SVM is finding the best hyperplane which maximizing the margin of the hyperplane [32]. The hyperplane with maximum margin performs a good generalization [32]. It can be formulated into quadratic programming problem which defined in equation 1 [32].

$$min \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$s.t\ y_i(x_i.w + b) - 1 \geq 0 \quad (1)$$

SVM is a quadratic programming where the class of a new input data can be predicted using equation 2 [32].

$$f(x_d) = \sum_{i=1}^{Ns} \alpha_i\, y_i x_i x_d + b \quad (2)$$

Where x_i is a support vector, Ns is some support vector and x_d is the input data which will be predicted. In the real world, some data are nonlinearly separable. Therefore, the equation two can not be used to predict the input data. It must be modified by adding a slack variable ξ and the maximum margin obtained by the equation 3 [32].

$$min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left(\sum_{i=1}^{N} \xi_i\right)$$
$$s.t\ y_i(x_i.w + b) \geq 1 - \xi_i,\ \xi_i \geq 0 \quad (3)$$

Where C is a penalty parameter due to misclassified data to control the trade-off of the classification error and the margin [32][33]. Kernel trick is another method to solve the nonlinearly separable data [16][32][33]. Kernel approach transforms the input data into a new higher dimensional feature space using a transformation function. The prediction function of kernel trick formulated in equation 4 [32].

$$f(x_d) = \sum_{i=1}^{Ns} \alpha_i\, y_i K(x_i x_d) + b \quad (4)$$

Where K is a kernel function. Some popular kernel functions are polynomial, sigmoid, and Radial Basis Function (RBF). Based on the equation 3, the L1 loss function of SVM can be formulated in equation 5 while the L2 loss function expressed in equation 6 [19].

$$max(1 - y_i w^T x_i, 0) \quad (5)$$
$$max(1 - y_i w^T x_i, 0)^2 \quad (6)$$

L2 regularization on linear SVM added to prevent the overfitting. The primal problem of L2 regularization formulated in equation 7 while the dual problem of L2 regularization expressed in equation 8 [19].

$$\min_{w} = \frac{1}{2} w^T w + C \sum_{i=1}^{l} (max(0, 1 - y_i\, w^T x_i))^2 \quad (7)$$

$$\min_{\alpha} = \frac{1}{2} \alpha^T \bar{Q} \alpha - e^T \alpha$$
$$s.t. 0 \leq \alpha_i \leq U, i = 1, \ldots, l. \quad (8)$$

Where e is the vector of all ones, $\bar{Q} = Q + D$, D is a diagonal matrix and $Q_{ij} = y_i y_j x^T_i x_j$, $U = \infty$ dan $D_{ii} = 1/(2C)$, $\forall_i$.

C. Handwriting Recognition System

In this research, we construct a system to recognize the handwriting characters on form document automatically and convert it into an editable text. The flowchart of the scheme shown in Figure 4.

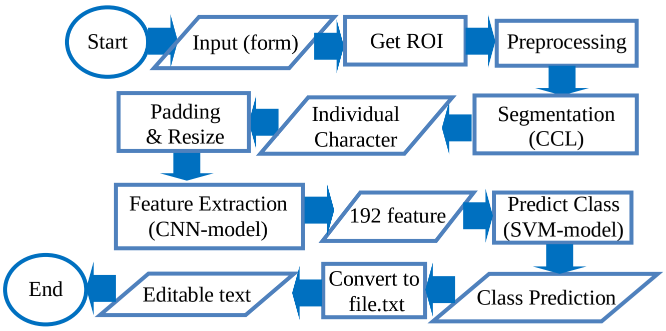

Fig. 5. Flowchart of handwriting character recognition on form document

The system consists of four stages: get ROI (Region of Interest), pre-processing, segmentation and classification. On getting ROI stage, the ROI is cropped according to the specified coordinates. Next, the preprocessing is performed on each ROI.

The preprocessing consist of bounding box removal using the eccentricity criteria, median filter, and bwareaopen. The output image of the preprocessing stage will be segmented using Connected Component Labeling (CCL) method. It aims to get an individual character. CCL is one of popular segmentation process which separates the objects in the image base on the connectivity of image. The advantages of CCL method is not affected by a skewed object [34].

D. Performance Evaluation

In this research, the original CNN and the kind of SVM approach such us kernel SVM, linear SVM using L1 regularization, and linear SVM using L2 loss function both of primal and dual problem approach are evaluated to compare with our proposed method. Each method is tested by using accuracy, precision, and recall. The computation time of each method is also evaluated. We also conducted ten folds cross-validation to validate our proposed method.

IV. Result and Discussion

In this research, we make two scenario experiments. The first experiment, we evaluate the proposed method using NIST dataset both for training and testing. The second experiment, we test the learning model was constructed in the first experiment using a more challenging dataset from the form document. We select 1000 samples randomly from NIST dataset where 80% for training and 20% for testing.

We evaluate the proposed method using Processor Intel(R) Core(Tm) i3-4000M CPU @2.40 GHz, Memory 4.00 GB RAM, Harddisk 500GB, Matlab 2014b 64 bit. The ten folds cross-validation conducted using NVIDIA GTX 1080. The CNN model is constructed by using Rasmus Berg Palm's Deep Learning Matlab Toolbox [35]. A feature of handwritten character according to the CNN model is generated by using Xindong Zhang CNNSVM toolbox [36], linear SVM classifier using Liblinear Matlab Toolbox [37], and kernel SVM using Libsvm Matlab Toolbox [38]. The accuracy rate, precision, and recall of each method are shown in Table I while the computation time of each method is shown in Table II. The accuracy rate, precision, and recall are calculated in percent (%) while the computation time is calculated in seconds (s).

TABLE I. ACCURACY OF HANDWRITING RECOGNITION

Method	**Numeral**			**Uppercase**		
	Acc	Prec	Rec	Acc	Prec	Rec
CNN	98,30	98,30	98,32	92,33	92,33	92,39
CNN+SVM kernel	97,7	97,70	97,71	91,34	91,35	91,48
CNN+SVM L1 Reg-L2 Loss	98,65	98,65	98,66	**93,15**	**93,15**	**93,19**
CNN+SVM L2 Reg-L2 Loss (dual)	98,70	98,60	98,62	93,05	93,06	93,09
CNN+SVM L2 Reg-L2 Loss (primal)	98,75	98,75	98,77	93,05	93,06	93,09
CNN+SVM L2 Reg-L1 Loss (dual)	**98,85**	**98,85**	**98,86**	**93,05**	**93,06**	**93,08**
Method	**Lowercase**			**Numeral+Uppercase**		
	Acc	Prec	Rec	Acc	Prec	Rec
CNN	83,54	83,54	83,68	88,32	88,32	88,84
CNN+SVM kernel	82,21	82,21	82,49	89,12	89,13	89,82
CNN+SVM L1 Reg-L2 Loss	86,07	86,08	85,99	91,02	90,85	91,36
CNN+SVM L2 Reg-L2 Loss (dual)	86,21	86,21	86,13	90,86	90,86	91,35
CNN+SVM L2 Reg-L2 Loss (primal)	86,19	86,19	86,11	90,84	90,94	91,16
CNN+SVM L2 Reg-L1 Loss (dual)	**86,21**	**86,21**	**86,12**	**91,37**	**91,37**	**91,88**

Base on Table I, the proposed method achieves the best accuracy, precision, and recall except for uppercase character. For the uppercase, CNN and SVM with L1 regularization achieve better accuracy rate than proposed method, but however, the method requires more training time than our proposed method. It Shown in Table II.

The kernel SVM uses RBF kernel function. The method only improves the accuracy rate for the merger of numeral and uppercase character. We also identify the most complicated character to be distinguished of each dataset. The most difficult characters to distinguished on numeral characters are 4 and 9, 2 and 3. On uppercase characters are D and O. On lowercase characters are g and q, j and l, and on the merger of numeral and uppercase character are 0 and O, 1 and I, 2 and Z, and 5 and S. It is caused by the similar shape of the characters. The evaluation of the cost time (second) for each method can be seen in Table II. Also, the cost time to generate the feature vector from CNN model on the numeral character is 29,02s, on uppercase is 69,77s, on lowercase is 60,69s, and on the merger of numeral and uppercase is 132,62s.

TABLE II. COMPUTATION TIME OF EACH METHOD

Method	**Numeral**		**Uppercase**	
	Training	Testing	Training	Testing
CNN	4.909,18	8,31	12.387,57	13,74
CNN+SVM kernel	21,07	7,61	164,95	79,54
CNN+SVM L1 Reg-L2 Loss	7,12	0,07	**129,32**	**0,29**
CNN+SVM L2 Reg-L2 Loss (dual)	9,58	0,07	71,93	0,23
CNN+SVM L2 Reg-L2 Loss (primal)	3,46	0,07	26,73	0,26
CNN+SVM L2 Reg-L1 Loss (dual)	**3,56**	**0,07**	**26,16**	**0,23**
Method	**Lowercase**		**Numeral+Uppercase**	
	Training	Testing	Training	Testing
CNN	14.025,51	67,53	14.744,14	21,17
CNN+SVM kernel	204,36	92,27	231,40	156,89
CNN+SVM L1 Reg-L2 Loss	146,08	0,24	191,73	0,38
CNN+SVM L2 Reg-L2 Loss (dual)	91,22	0,25	56,33	0,43
CNN+SVM L2 Reg-L2 Loss (primal)	29,47	0,24	38,43	0,38
CNN+SVM L2 Reg-L1 Loss (dual)	**33,34**	**0,34**	**28,84**	**0,44**

The proposed method is also evaluated using ten folds cross-validation. The mean accuracy of cross-validation on the numeral character is 97.95%, on the uppercase character is 95.45%, on the lowercase character is 89.16%, and on the merger of numeral and uppercase character is 92.59%. While the original CNN achieves a mean accuracy of cross-validation on the numeral character is 97.54%, on the uppercase character is 93.91%, on the lowercase character is 86.20%, and on the merger of numeral and uppercase character is 89.49%. It shows that the proposed method can improve the accuracy rate of handwritten character recognition.

The learning model that has been trained used to construct a system for handwriting character recognition on form document automatically. The system is used to test the proposed method using the forms that have been distributed. The interface of the scheme can be seen in Figure 5. The detailed results of each stage can be seen in Figure 6.

Figure 6 (a-e) shows the result of preprocessing step as described in the previous section. The figure 6(f) is an editable text that contains digital characters according to the class of each input character. After the preprocessing stage, the next step is segmenting the input image into individual character using CCL method. For each particular character applied a 4x4 padding, according to the image background color. It aims to make the input character as closely as possible with the data training to achieve a good prediction. Overall, the system that has been constructed can predict the more challenging handwriting on ten different test form document with an accuracy rate of 83.37%.

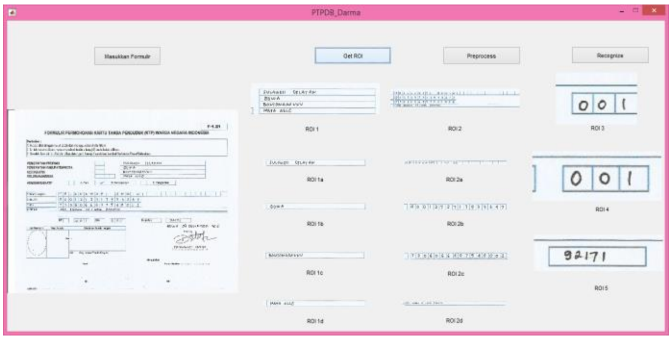

Fig. 5. Interface system of handwriting character recognition on form document

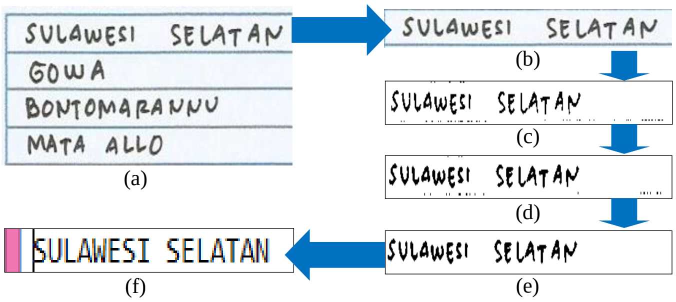

Fig. 6. (a) Example of ROI 1; (b) First Line of ROI 1; (c) Image after applied the eccentricity criteria; (d) Image after applied median filter; (e) Image after applied bwareaopen; (e) Digital character in the editable text.

Some ROI are difficult to be recognized, it happened on ROI which contain the connected character. It 's hard to do the segmentation with the CCL method because the CCL segmentation is based on the connectivity principle. Therefore, CCL performs better on segmenting the characters that are written in the cell such as on ROI 2-5.

The result of this study compared with previous studies which related to handwriting recognition is shown in Table III.

TABLE III. COMPARISON ACCURACY RATE WITH RELATED RESEARCH

Authors	**Method**	**Data-set**	**Number of Class**	**Accuracy (%)**
Hadi, W, & Ramadhani, 2014 [22]	*Freeman Chain Code+* rasio-atas, rasio-kanan, rasio-luas, dan rasio-tinggi, k-NN dan ANN	MNIST	10	k-NN & ANN >88
		MNIST-owner	10	kNN:88,91 ANN:90
		Owner-owner	10	kNN:89,82 ANN:93,6
(Juharwidy-ningsih et al., 2013 [23]	Zernike Moments +SVM	MNIST+ owner	15	97
Nasien et al., 2010 [24]	Freeman Chain Code +SVM	NIST (upper)	26	88,46
		NIST	26	86,007

		(lower)		
		NIST (upper+ lower)	52	73,44
Hallale & Salunke, 2013 [25]	12 directional feature+ similarity	Owner (num+ upper)	36	88,29
Darmatasia, 2016	CNN+SVM L2 regularization L1 Loss Function (dual)	**NIST (num)**	**10**	**98,85**
		NIST (upper)	**26**	**93,05**
		NIST (lower)	**26**	**86,21**
		NIST (num+ upper)	**36**	**91,37**
		NIST-Owner	**36**	**83,37**

As seen in Table III, the proposed method gives the best accuracy using NIST dataset both for training and testing. The proposed method by Hallale [25] achieves better accuracy rate better than our proposed method when using the data of the form document for testing. However, in [25] only classify the individual character that has been segmented and well selected. While in this research, the handwriting recognition conducted on form document which contains a bounding box, noise, and more challenging handwritten characters.

V. CONCLUSION AND FUTURE RESEARCH

In this paper, CNN as a powerful feature extraction method applied to extract the feature of the handwritten characters and linear SVM using L1 loss function and L2 regularization used as end classifier. Based on the experiment results using data from NIST SD 19 2^{nd} edition, both for training and testing, the proposed method achieves an accuracy rate better than only CNN method. The proposed method was also validated using ten folds cross-validation, and it shows that the recognition rate for this proposed method is still able to be improved. The proposed method achieves a better accuracy rate than another previous study.

A system for automatic handwriting recognition on form document has been constructed using the proposed method. Overall, the system can recognize a more challenging handwriting on form document which containing bounding box and some noise.

For the next research, the other segmentation method may be explored to address the connected character problem. Also, some postprocessing like a linguistic post-processing can be considered to achieve a better accuracy.

REFERENCES

[1] A. Shruthi, "Offline Handwritten Word Recognition using Multiple Features with SVM Classifier for," pp. 5989–5995, 2015.

[2] A. Jindal and M. Amir, "Automatic classification of handwritten and printed text in ICR boxes," *Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014*, pp. 1028–1032, 2014.

[3] N. Sharma, T. Patnaik, and B. Kumar, "Recognition for Handwritten English Letters : A Review," vol. 2, no. 7, pp. 318–321, 2013.

[4] "Deep Learning." [Online]. Available: https://www.techopedia.com/definition/30325/deep-learning. [Accessed: 22-Jan-2017].

[5] Lecun Yann and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," *Handb. Brain Theory Neural Netw.*, pp. 255–258., 1998.

[6] L. Chen, S. Wang, W. Fan, J. Sun, and N. Satoshi, "Reconstruction combined training for convolutional neural networks on character recognition," *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. pp. 431–435, 2015.

[7] M. M. Santoni, D. I. Sensuse, A. M. Arymurthy, and M. I. Fanany, "Cattle Race Classification Using Gray Level Co-occurrence Matrix Convolutional Neural Networks," *Procedia Comput. Sci.*, vol. 59, no. October, pp. 493–502, 2015.

[8] B. Kwolek, "Face Detection Using Convolutional Neural Networks and Gabor Filters," *Artif. Neural Networks Biol. Inspirations - ICANN*, pp. 551–556, 2005.

[9] C. V Dec, "Sparse arrays of signatures for online character recognition," pp. 1–10, 2013.

[10] W. Yang, L. Jin, Z. Xie, and Z. Feng, "Improved Deep Convolutional Neural Network For Online Handwritten Chinese Character Recognition using Domain-Specific Knowledge."

[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors arXiv : 1207 . 0580v1 [cs . NE] 3 Jul 2012," pp. 1–18.

[12] M. Zeiler and R. Fergus, "Regularization of Neural Networks using DropConnect," no. 1, 2012.

[13] M. Elleuch, R. Maalej, and M. Kherallah, "A New Design Based - SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition," *Procedia - Procedia Comput. Sci.*, vol. 80, pp. 1712–1723, 2016.

[14] M. Zhang, "SHORT-TERM LOAD FORECASTING BASED ON SUPPORT VECTOR," no. August, pp. 18–21, 2005.

[15] S. Song, Z. Zhan, Z. Long, J. Zhang, and L. Yao, "Comparative Study of SVM Methods Combined with Voxel Selection for Object Category Classification on fMRI Data," *PLoS One*, vol. 6, no. 2, p. e17191, Feb. 2011.

[16] H. Huang and C. Lin, "Linear and Kernel Classification : When to Use Which ?," 2012.

[17] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*. Simpson,Jonathan, 2015.

[18] Anonim, "Differences between L1 and L2 as Loss Function and Regularization," 2013. [Online]. Available: http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/. [Accessed: 05-Jan-2017].

[19] C. Hsieh and C. Lin, "A Dual Coordinate Descent Method for Large-scale Linear SVM," no. 2, 2008.

[20] "Core Vector Machines.pdf." .

[21] "ABBYY FineReader." [Online]. Available: https://www.abbyy.com/.

[22] F. A. Hadi, T. A. B. W, and K. N. Ramadhani, "Freeman Chain Code yang Dimodifikasi Handwritten Numeral Recognition by application of modified Freeman Chain Code Abstrak Pendahuluan Optical Character Recognition (OCR) Freeman Chain Code," 2014.

[23] E. Juharwidyningsih, C. Fatichah, and N. Khotimah, "Pengenalan Karakter Tulisan Tangan Angka dan Operator Matematika Berdasarkan Zernike Moments Menggunakan Support Vector Machine," vol. 2, no. 1, pp. 1–5, 2013.

[24] D. Nasien, H. Haron, and S. S. Yuhaniz, "Support Vector Machine (SVM) for english handwritten character recognition," *2010 2nd Int. Conf. Comput. Eng. Appl. ICCEA 2010*, vol. 1, pp. 249–252, 2010.

[25] S. B. Hallale and G. D. Salunke, "Twelve Directional Feature Extraction for Handwritten English Character Recognition," no. 2, pp. 39–42, 2013.

[26] P. Grother, "NIST Special Database 19 Handprinted Forms and Characters 2 nd Edition NIST Special Database 19 Handprinted Forms and Characters 2 nd Edition."

[27] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," *Proc. Int. Conf. Pattern Recognit. ICPR12*, pp. 10–13, 2012.

[28] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of Training Convolutional."

[29] P. Burkert, F. Trier, M. Z. Afzal, A. Dengel, and M. Liwicki, "DeXpression: Deep Convolutional Neural Network for Expression Recognition," pp. 1–8, 2015.

[30] Michael Nielsen, "Deep Learning," 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap6.html.

[31] C. Cortes and V. Vapnik, "Support Vector Network," vol. 7. 1995.

[32] E. Prasetyo, *Data Mining: Mengolah Data Menjadi Informasi Menggunakan Matlab*, 1st ed. Yogyakarta: ANDI OFFSET, 2014.

[33] K. Sembiring, "Support Vector Machines," no. September, pp. 1–28, 2007.

[34] M. Cheriet, N. Kharma, and C. L. Liu, *Character recognition systems: a guide for students and practioners*. 2007.

[35] R. B. Palm, "DeepLearnToolbox," 2015. [Online]. Available: https://github.com/rasmusbergpalm/DeepLearnToolbox.

[36] X. Zhang, "CNNSVM," 2015. [Online]. Available: https://github.com/xindongzhang/CNNSVM.

[37] Chih-Jen Lin, "LIBLINEAR -- A Library for Large Linear Classification," 2015. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/liblinear/.

[38] C.-C. Chang and C.-J. Lin, "LIBSVM -- A Library for Support Vector Machines," 2016. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/.