Pattern Recognition 45 (2012) 1318–1325

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A novel hybrid CNN–SVM classifier for recognizing handwritten digits

Xiao-Xiao Niu *, Ching Y. Suen

Centre for Pattern Recognition and Machine Intelligence, Concordia University, Suite EV003.403, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8

ARTICLE INFO

Article history:
Received 30 June 2011
Received in revised form
30 August 2011
Accepted 29 September 2011
Available online 19 October 2011

Keywords:
Hybrid model
Convolutional Neural Network
Support Vector Machine
Handwritten digit recognition

ABSTRACT

This paper presents a hybrid model of integrating the synergy of two superior classifiers: Convolutional Neural Network (CNN) and Support Vector Machine (SVM), which have proven results in recognizing different types of patterns. In this model, CNN works as a trainable feature extractor and SVM performs as a recognizer. This hybrid model automatically extracts features from the raw images and generates the predictions. Experiments have been conducted on the well-known MNIST digit database. Comparisons with other studies on the same database indicate that this fusion has achieved better results: a recognition rate of 99.81% without rejection, and a recognition rate of 94.40% with 5.60% rejection. These performances have been analyzed with reference to those by human subjects.

1. Introduction

Handwritten digit recognition is a challenging problem that has been intensely studied for many years in the field of handwriting recognition. Numerous results have been achieved by researchers who have used different algorithms, such as K-Nearest-Neighbors (KNNs), Support Vector Machines (SVMs), Neural Networks (NNs), Convolutional Neural Networks (CNNs), etc. One specific example of handwritten digit recognition research is the work of Ciresan et al. [1] on the MNIST database, where the best recognition rate of 99.65% has been obtained by using a 6-layer Neural Network.

The reason why handwritten digit recognition is still an important area is due to its vast practical applications and financial implications. The industry demands a decent recognition rate with the highest reliability. Higher recognition rate on handwritten digits increases the recognition accuracy for handwritten data, which usually exist in numeral strings. When we consider recognizing a numeral string of a ZIP code, the recognition probability of this string is the multiplication of the recognition probability of each isolated digit (assuming that each digit is correctly separated from the numeral string by the segmentation process). However, it is unrealistic to have a handwritten digit recognition system with 100% recognition accuracy. Hence, the reliability is much more important than the recognition accuracy in real-life systems. A higher reliability can decrease the huge financial loss due to small errors in reading the courtesy amount on cheques for example. However, most of the papers published on the MNIST database have focused mainly on the recognition rate and only a few of them have mentioned the reliability. In this paper, our goals are not only to improve the current recognition performance but also to seek the highest reliability on the applications of handwritten digits.

Feature extraction is one key factor in the success of a recognition system. It requires that features should have the most distinguishable characteristics among different classes while retaining invariant characteristics within the same class. Traditional hand-designed feature extraction is really a tedious and time-consuming task and cannot process raw images, while the automatic extraction methods can retrieve features directly from raw images. Szarvas et al. [2] evaluated the automatically optimized features learned by the Convolutional Neural Network on pedestrian detection, and showed the CNN-features+SVM combination generated the highest accuracy. Mori et al. [3] trained the convolutional spiking neural network with time domain encoding schemes module by module using different fragment images. The outputs of each layer in the model were fed as features to the SVM. 100% face recognition rate was obtained on the 600 images of 20 people. Another example of automatic feature extractors is the trainable feature extraction based on the Convolutional Neural Network described in [4], which showed a high performance for handwritten digit recognition. By using the trainable feature extractor plus elastic distortions or affine distortions, the system obtained the low error rates of 0.56% and 0.54%, respectively. Inspired by this particular work, we propose the hybrid CNN–SVM model. Since both CNN and SVM have already achieved superb performances in digit recognition, we focus our research on their fusion to bring out their best qualities.

In this paper, we have designed a hybrid CNN–SVM model for handwritten digit recognition. This model automatically retrieves features based on the CNN architecture, and recognizes the unknown

* Corresponding author.
E-mail addresses: archernxx@hotmail.com (X.X. Niu), suen@encs.concordia.ca (C.Y. Suen).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.09.021

pattern using the SVM recognizer. High reliabilities of the proposed systems have been achieved by a rejection rule. To verify the feasibility of our methodology, the well-known MNIST digit database is tested.

The rest of this paper is organized as follows: the principles of SVM, CNN and the hybrid CNN–SVM model are described in Section 2. Experimental results and the complexity analysis on the hybrid model are presented and discussed in Section 3. Section 4 compares the differences between machine recognition and human classification on the MNIST database. Conclusions are drawn in Section 5.

2. The new hybrid CNN–SVM system

Our proposed system was designed to integrate the SVM and the CNN classifiers. We will first briefly introduce the SVM theory in Section 2.1, and the CNN structure in Section 2.2. Then, the hybrid CNN–SVM trainable feature extractor model will be presented in Section 2.3, followed by an analysis of its merits at the end of this section.

2.1. SVM classifier

Support Vector Machines [5] with different kernel functions can transform a nonlinear separable problem into a linear separable problem by projecting data into the feature space and then finding the optimal separate hyperplane. This method was initially proposed to solve two-class problems. Later, a few strategies were suggested to extend this technique to multi-class classification problems.

2.1.1. Two-class soft margin Support Vector Machines

Suppose a training set$S=\{(\vec{x}_i,y_i)|(\vec{x}_i,y_i)\in R^m\times R, i=1,2,\ldots,l\}$, where label $y_i\in\{-1,1\},(i=1,2,\ldots,l)$. The soft margin SVM tries to find a maximum margin hyperplane where S can be linearly separated by solving the following primal problem:

$$\min: P(\vec{w},b,\vec{\xi})=\frac{1}{2}\vec{w}^T\cdot\vec{w}+C\sum_{i=1}^{l}\xi_i \tag{1}$$

$$\text{subject to}: \begin{cases} y_i(\vec{w}^T\phi(\vec{x}_i)+b)\geq 1-\xi_i \\ \xi_i\geq 0, \quad i=1,2,\ldots,l \end{cases} \tag{2}$$

where $\vec{w}$ is an m-dimensional vector, b is a scalar. ξ_i is the slack variables. C is the penalty parameter controlling the trade-off between the maximization of the margin and the minimization of the classification errors. Training data $\vec{x}_i$are mapped to a higher-dimensional space by the function $\phi(\cdot)$. To find the solution of the primal optimization problem, it is usually done by solving its dual problem of a Lagrangian formulation:

$$\min: D(\vec{\alpha})=-\sum_{i=1}^{l}\alpha_i+\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}y_iy_j\alpha_i\alpha_jK(\vec{x}_i,\vec{x}_j) \tag{3}$$

$$\text{subject to}: \begin{cases} \sum_{i-1}^{l}y_i\alpha_i=0 \\ 0\leq\alpha_i\leq C, \quad i=1,2,\ldots l \end{cases} \tag{4}$$

where $\alpha_i(i=1,\ldots,l)$ is a positive Lagrange multiplier. Here, the kernel function $K(\vec{x}_i,\vec{x}_j)$ is introduced. It calculates the dot-product for the data in the high-dimensional feature space without doing the explicitly mapping $\phi(\cdot)$. This is called the "kernel trick".

The decision function is defined as

$$g(\vec{x})=sign\left\{\sum_{\vec{x}_i\in SV_s}\alpha_iy_iK(\vec{x}_i,\vec{x})+b\right\} \tag{5}$$

where vector $\vec{x}_i$ is one of the Support Vectors (SVs) when $0<\alpha_i\leq C$.

2.1.2. Multi-class Support Vector Machines

Two approaches are currently used to solve the multiclass problems. One is based on constructing and combining several two-class Support Vector Machines. Three typical strategies applied are: one-against-all, one-against-one, and Directed Acyclic Graph ((DAG)) Support Vector Machines. The other is to consider all the multi-class data in one optimization problem. Hsu et al. [6] compared these two approaches on different datasets, and concluded that the first approach, specifically, one-against-one and DAG methods are more suitable in practice than other methods.

We used LIBSVM [7] to build SVMs in our experiments. LIBSVM is an efficient open source library tool for classification and regression problems. The one-against-one method is implemented for the multi-class SVMs in LIBSVM. Consider a k-class problem, there are$l$ training samples: $\{\vec{x}_1,y_1\},\ldots,\{\vec{x}_l,y_l\}$, where $\vec{x}_i\in R^m, i=1,\ldots,l$ are feature vectors and $y_i\in\{1,\ldots,k\}$ are the corresponding class labels. The one-against-one method constructs $k(k-1)/2$ classifiers, where each classifier uses the training data from two classes chosen out of $k$classes. For training data from the ith and the jth classes, we need to solve the following optimization problem:

$$\min: P(\vec{w}^{ij},b^{ij},\vec{\xi}^{ij})=\frac{1}{2}(\vec{w}^{ij})^T\cdot\vec{w}^{ij}+C\sum_n\xi_n^{ij} \tag{6}$$

$$\text{subject to}: \begin{cases} (\vec{w}^{ij})^T\phi(\vec{x}_n)+b^{ij}\geq 1-\xi_n^{ij}, y_n=i \\ (\vec{w}^{ij})^T\phi(\vec{x}_n)+b^{ij}\leq -1+\xi_n^{ij}, y_n\neq i \\ \xi_n^{ij}\geq 0, \quad n=1,\ldots,k(k-1)/2 \end{cases} \tag{7}$$

In the class decision, LIBSVM applies the "max wins" algorithm. Each classifier gives one vote to its determined class, and the final result is determined by the class that wins the most votes. In case there are more than one class having the same votes, LIBSVM simply chooses the one with the smallest index.

2.1.3. Probability estimates

LIBSVM does not only predict the class label but also provide the probability information for each testing sample. For the problem of k classes, the aim is to estimate each class probability of data $\vec{x}$:

$$p_i=p(y=i|\vec{x}), \quad i=1,\ldots,k \tag{8}$$

For the one-against-one method, p_i is obtained by solving the following optimization problem:

$$\min: W(\vec{p})=\frac{1}{2}\sum_{i=1}^{k}\sum_{j:j\neq i}(r_{ji}p_i-r_{ij}p_j)^2 \tag{9}$$

$$\text{subject to}: \sum_{i=1}^{k}p_i=1, p_i\geq 0, \forall i \tag{10}$$

where r_{ij} is the pairwise class probability defined as

$$r_{ij}\approx p(y=i|y=i,\text{ or } j,\vec{x}) \tag{11}$$

It is estimated during the cross validation in the SVM model selection process. For more implementation details, one can refer to Wu et al. [8]. They have shown that this estimation approach is more stable than voting and the method proposed by Hastie et al. [9].

In our experiments, the SVMs were trained to make the label predictions with probabilities. Judging on the probability values, post-processing could easily be applied in deciding whether to

accept or to reject the candidate, which will be shown in the reliability experiments, in Section 3.2.

2.2. CNN classifier

A Convolutional Neural Network [10] is a multi-layer neural network with a deep supervised learning architecture that can be viewed as the composition of two parts: an automatic feature extractor and a trainable classifier. The feature extractor contains feature map layers and retrieves discriminating features from the raw images via two operations: convolutional filtering and down sampling. The convolutional filtering kernels on feature maps have the size of 5 by 5 pixels, and the down sampling operation after the filtering has a ratio of 2. The classifier and the weights learned in the feature extractor are trained by a back-propagation algorithm.

Instead of using the more complicated LeNet-5 as in [4], we adopted the same simplified CNN structure that was presented in [11]. The architecture of the CNN model is shown in Fig. 1. There are five layers. The input layer is a matrix of the normalized and centralized pattern with size S_1 by S_1. Two feature map layers (N1 and N2) are used to compute the features, with different resolutions. Each neuron on a feature map connects 25 inputs with its previous layers, and they are defined by the 5 by 5 convolutional filtering kernel (known as the "receptive field" in [4]). All the neurons in one feature map share the same kernel and connecting weights (known as the "sharing weights" in [4]). With a kernel size of 5, and a subsampling ratio of 2, each feature map layer reduces the feature size from the previous feature size S to $\lceil (S-4)/2 \rceil$. As shown in Fig. 1, $S_2 = \lceil (S_1-4)/2 \rceil$, and $S_3 = \lceil (S_2-4)/2 \rceil$, where $\lceil x \rceil$ denotes the largest integer not exceeding x. The trainable classifier is the fully connected Multi-Layer Perceptron (MLP), with a hidden layer (N3) and an output layer (N4). In our experiments, the trainable classifier is modified by the SVM classifier, and will be described in detail in the following section.

2.3. Hybrid CNN–SVM model

The architecture of our hybrid CNN–SVM model was designed by replacing the last output layer of the CNN model with an SVM classifier. For output units of the last layer in the CNN network, they are the estimated probabilities for the input sample. Each output probability is calculated by an activation function. The input of the activation function is the linear combination of the outputs from the previous hidden layer with trainable weights, plus a bias term. Looking at the output values of the hidden layer is meaningless, but only makes sense to the CNN network itself; however, these values can be treated as features for any other classifiers.

Fig. 2 shows the structure of the new hybrid CNN–SVM model. Firstly, the normalized and centered input images are sent to the input layer, and the original CNN with the output layer is trained with several epochs until the training process converges. Then, the SVM with a Radial Basis Function (RBF) kernel replaces the output layer. The SVM takes the outputs from the hidden layer as a new feature vector for training. Once the SVM classifier has been well trained, it performs the recognition task and makes new decisions on testing images with such automatically extracted features.

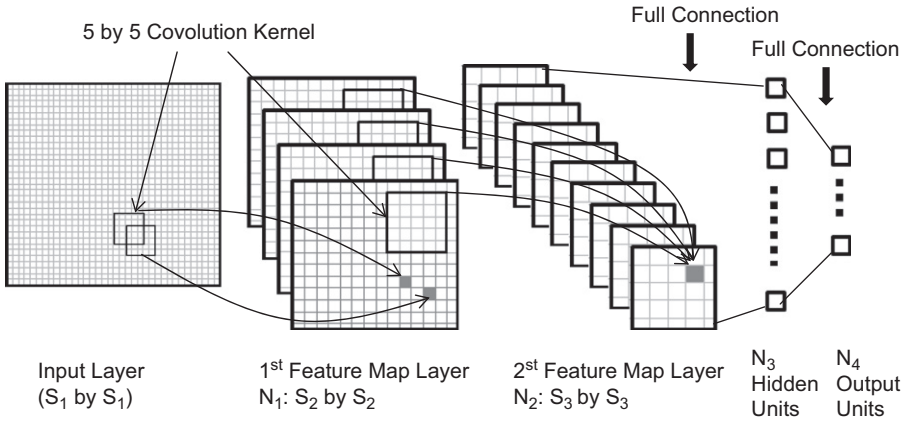

Fig. 1. Structure of the adopted CNN.

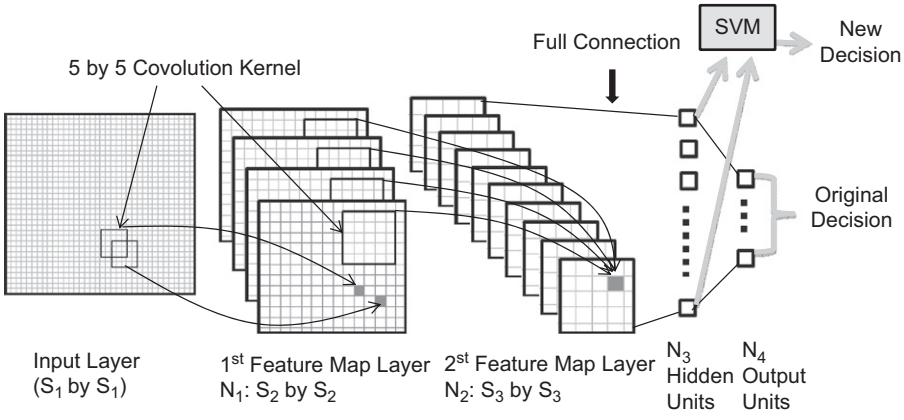

Fig. 2. Structure of the hybrid CNN–SVM model.

The numbers N1, N2, and N3 of different layers have great influence on the generalization ability of the CNN model as discussed in [12,13]. To set up those values, we followed the way similar to [13]. Let N2=50 and N3=100. However, we set N1=25. It is inspired by the sparse and over-complete representation theory in [11,12]. Both researchers initialized the weights of the first feature map layer (N1=50) with 50 sparse and over-complete learned features, and achieved good results on recognizing isolated handwritten numerals. In our CNN model, the weights for the first feature map layer are randomly initialized. But we tried different numbers of the first layer. We found that smaller than 25 (we tried 6) decreased the performance; while greater than 25 (we tried 50) increased the training time a lot with little improvement on the performance. Therefore, we set N1 to be 25 after considering the trade-off between accuracy and time cost.

2.4. Merits of hybrid model

Our expectation that this novel hybrid CNN–SVM model will outperform each individual classifier is based on the fact that the hybrid system compensates the limits of the CNN and SVM classifiers by incorporating the merits of both classifiers. Since the theoretical learning method of CNN is the same as that for the MLP, it is an extension model of the MLP. The learning algorithm of MLP is based on the Empirical Risk Minimization, which attempts to minimize the errors in the training set. When the first separating hyperplane is found by the back-propagation algorithm, no matter whether it is the local or the global minima, the training process stops and the algorithm does not continue to improve the separating hyperplane solution. Therefore, the generalization ability of MLP is lower than that of SVM. On the other hand, the SVM classifier aims to minimize the generalization errors on the unseen data with a fixed distribution for the training set, by using the Structural Risk Minimization principle. The separating hyperplane is a global optimum solution. It is calculated by solving the quadratic programming problem, and the margin area between two classes of training samples reaches its maximum. As a result, the generalization ability of SVM is maximized to enhance the classification accuracy of the hybrid model after replacing the N4 output units in the CNN.

Another limitation of MLP is that it tends to assign a high value (nearly $+1$) to one neuron at the output layer whereas all the remaining neurons have a low value (nearly -1). This causes difficulties in rejecting errors in real applications. But the SVM classifier calculates the estimated probability of each class on one testing data in the classification decision, as we mentioned in Section 2.1.3. This probability information provides a more reliable rank list of label predictions. Beside, using those probability values can help us to design an efficient rejection mechanism.

The advantage of the CNN classifier is that it automatically extracts the salient features of the input image. The features are invariant at a certain degree to the shift and shape distortions of the input characters. This invariance occurs because CNN adopts the weight sharing technique on one feature map. On the contrary, the hand-designed feature extractor needs elaborately designed features or even applies different types of features to achieve the distortion invariance of the individual characters. Furthermore, the topology of handwritten characters is very important because pixels located near each other in space have strong connections. The elementary features like the corners, endpoints, etc. are composed of these nearby pixels. CNN uses the receptive field concept successfully to obtain such local visual features. However, the hand-designed feature extraction methods ignore and lose such topology of the input in most cases. Therefore, the trainable features of CNN can be used instead of the hand-designed features to collect more representative and relevant information, especially for the handwritten digits.

3. Experiments

To evaluate the feasibility of the hybrid model, we conducted experiments on the well-known MNIST handwritten digit dataset. It contains 60,000 training and 10,000 testing samples. It is a subset of the NIST dataset. The images in the MNIST dataset are grayscale numeral bitmaps that have been centered and size normalized to 28×28 pixels. These images were downloaded from [14]. Some samples in this database are illustrated in Fig. 3.

The experiments are conducted in this manner: Section 3.1 presents the results by using the hybrid CNN–SVM model and makes the error analysis. The reliability performance of the model is presented in Section 3.2. In Section 3.3, the complexity of the proposed model is compared with the single CNN classifier on the MNIST testing dataset.

3.1. Experiments on the hybrid CNN–SVM model

Previous researchers [4,12,13] have proven that better generalization can be achieved with an expanded training dataset by using distortion techniques. In our experiments, Simard's elastic distortion [13] was introduced, with additional scaling and rotation transforms applied in the CNN training phase. The training procedure was stopped after 500 epochs as it converged to a fixed value 0.28, as shown in Fig. 4. With this setup, the CNN learning classifier produced an error rate of 0.59% on the testing dataset.

Then, the new hybrid CNN–SVM model was built and trained. The last fully connected layer of CNN was replaced by an SVM classifier to predict labels of the input patterns. One hundred values from the layer N3 of the trained CNN network were used as a new feature vector to represent each input pattern, and were fed to the SVM for learning and testing. To build the SVM in the hybrid model, we used the RBF kernel, and determined the optimal kernel parameter σ and penalty parameter$C$ by applying the 5-fold cross validation method on the training dataset. The process to find the optimal parameters is called the model selection. The grid searching range of each parameter is: $\sigma=[2^3, 2^1, 2^{-1},\ldots, 2^{-15}]$ and $C=[2^{15}, 2^{13},\ldots, 2^{-5}]$. We tried $10\times11=110$ different combinations. The best validation rate was achieved when $C=128$ and $\sigma=2^{-11}$. These parameters were then used to train the hybrid model. A training error rate of 0.11% was obtained, and an error rate of 0.19% on the 10,000 testing data was achieved. Table 1 presents the training and

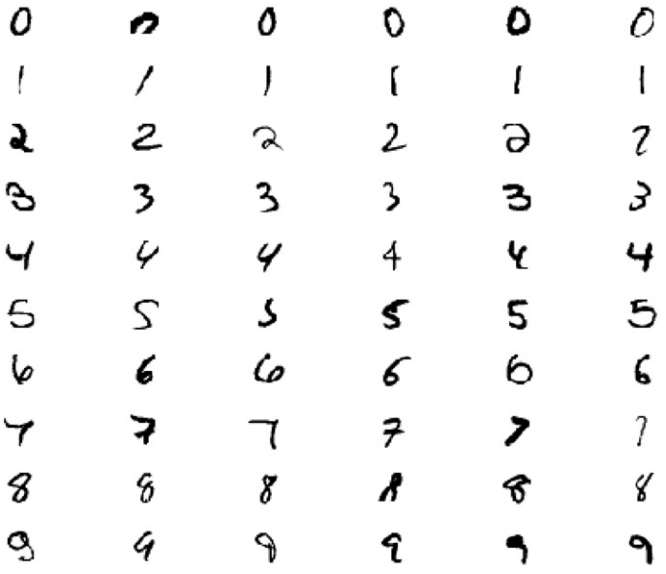

Fig. 3. Sample images in MNIST database.

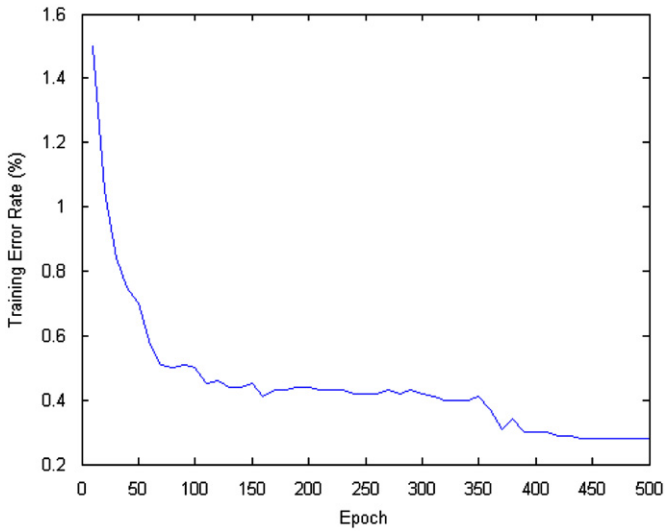

Fig. 4. Trend of training error rates of CNN on the MNIST dataset.

Table 1
Results of different classifiers on MNIST dataset.

Error rate (%)	CNN (distortion)	Hybrid CNN–SVM
Training	0.28	0.11
Testing	0.59	0.19

Table 2
Comparisons of testing results on MNIST dataset.

Reference	Method	Distortion	Error rate (%)
Lauer et al. [4]	TFE-SVM	Affine	0.54
Simard et al. [13]	Convolutional NN	Elastic	0.40
Ranzato et al. [12]	Convolutional NN	Elastic	0.39
LeCun [10]	Boosted LetNet-4	Affine, scaling, squeezing	0.7
Ciresan et al. [1]	6-layer NN	Elastic	0.35
Mizukami et al. [16]	KNN	Displacement computation	0.57
Keysers [17]	KNN	Non-linear deformation	0.52
This paper	Hybrid CNN–SVM	Elastic, scaling, rotation	**0.19**

testing error rates on the MNIST dataset by using the CNN classifier and the hybrid model.

Comparisons with other results of different methods published on the MNIST dataset are listed in Table 2. We chose the best recognition results generated by different learning algorithms with distortions applied on the training data. From the table, the lowest error rate was 0.35% as reported in [1]. However, a significant achievement was made by our proposed hybrid method with the lowest error rate of 0.19%, which boosted the performance by 45.71%$((0.35-0.19/0.35)\times 100\%)$ compared with the best recognition result up to date. Besides, the human recognition error was estimated as 0.2%; even it was not obtained on the whole testing set [15]. Our result is thus comparable to human vision.

Fig. 5 shows all of the 19 misclassified samples, and Table 3 presents the confusion matrix. After analyzing these errors, we found that they can be categorized into two types:

(1) The most frequent confusing pairs are "4–9" and "5–3", which have similar shapes and structures due to people's cursive writing habits. For example, when we closely examine the image "2131.tif" in Fig. 5, even human eyes cannot distinguish whether it is a "4" or a "9" without its label;
(2) The degraded quality of digit images, such as missing strokes ("2655.tiff", "3423.tif", "3559.tif"), broken numerals ("6572.tif"), intruder noises ("3226.tif", "5655.tif") and stroke touching ("948.tif", "9730.tif"). These cases could be caused by people's poor handwritings, or introduced by the scanning procedure, size normalization, and improper segmentations. For the second error category, it is extremely difficult for a machine to make a correct prediction with such ambiguous and degraded inputs.

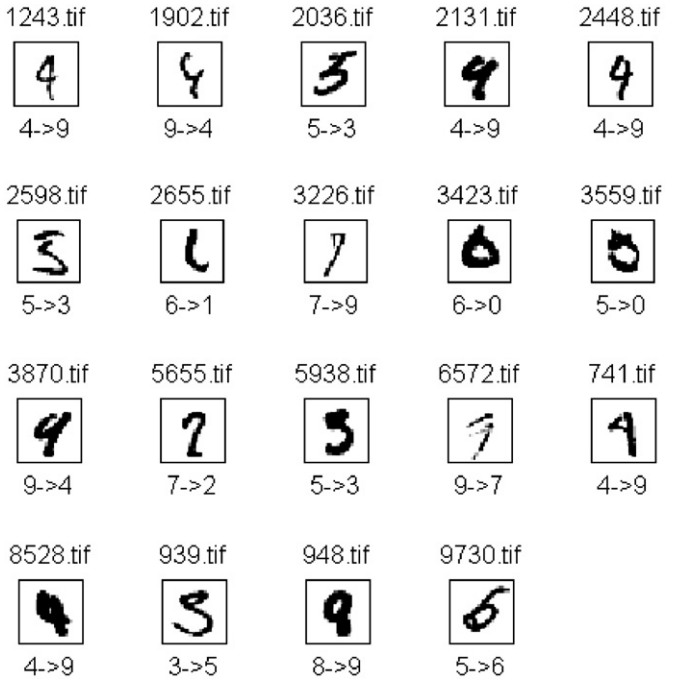

Fig. 5. Nineteen digit images were misclassified by the hybrid CNN–SVM model. The upper label is the ID number of the image in the MNIST testing dataset, and the lower label is the corresponding truth →prediction.

Table 3
Confusion matrix of the hybrid model on the MNIST testing dataset.

Truth	Prediction									
	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3						1				
4										5
5	1			3			1			
6	1	1								
7			1							1
8										1
9					2			1		

3.2. Reliability performance

In order to have a complete evaluation on the proposed model, we investigated the reliability performance on the hybrid model. The reliability performance is achieved through a rejection mechanism in [18]. A test sample is rejected when the difference between the top two probability values in the ranked predictions is smaller than a predefined threshold. Our hybrid model was built to predict the class label with probability information.

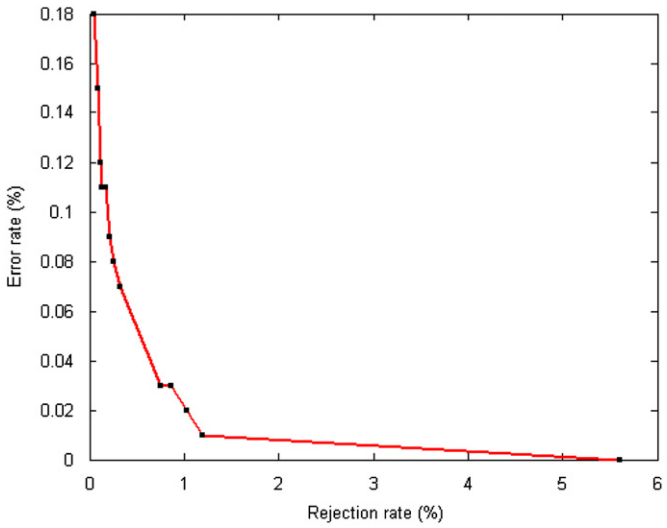

Fig. 6. Error-reject analysis of the hybrid model for the MNIST testing dataset.

Table 4
Comparison of two classifiers in terms of complexity on the 10,000 digits in the MNIST testing dataset.

Comparison factor	CNN	Hybrid CNN–SVM model
Total testing speed (s)	50	56
Memory usage (MB)	1.6	2.9
#SVs	–	1034

Fig. 6 shows the error and rejection rates for the MNIST testing set. When the hybrid model firstly reached zero-level errors, the rejection rate was 5.60%. At this point, the reliability of the system reached 100%, still with a high recognition rate of 94.40%. Its reliability performance exceeds Dong's results in [19], which is 100% reliability with an 8.49% rejection rate by using the HeroSVM library under the same rejection mechanism. Therefore, the hybrid system is really a robust learning model for recognizing handwritten digits.

3.3. Complexity of individual classifiers

The complexity analysis was conducted on the Red hat operating system, which is an open source software based on Linux. It was installed on a PC with Intel Pentium D CPU 3.40 GHZ, and 4.00 GB of RAM. We compared the complexity of our new hybrid model with CNN classifier in the testing process. Three factors were considered: testing speed, memory usage, and the number of SVs. We ignored the analysis of classifiers on the training speed, because the training time of the hybrid model composes of the time of training a CNN classifier and an SVM classifier separately.

Table 4 shows the complexity comparison between two classifier models on the 10,000 MNIST testing samples. The symbol #SVs represents the number of Support Vectors and it plays an important role in analyzing the complexity of an SVM classifier. It reflects the size of the weight model of an SVM, and directly influences the speed of the decision procedure. From Table 4, we noticed that the hybrid model requires 12.00% more time and 81.25% more memory space compared with the CNN model. This figures look quite reasonable due to the introduction of a more sophisticated SVM classifier, which replaced the original simple fully connected output layer of the CNN model. Furthermore, the recognition performance is increased

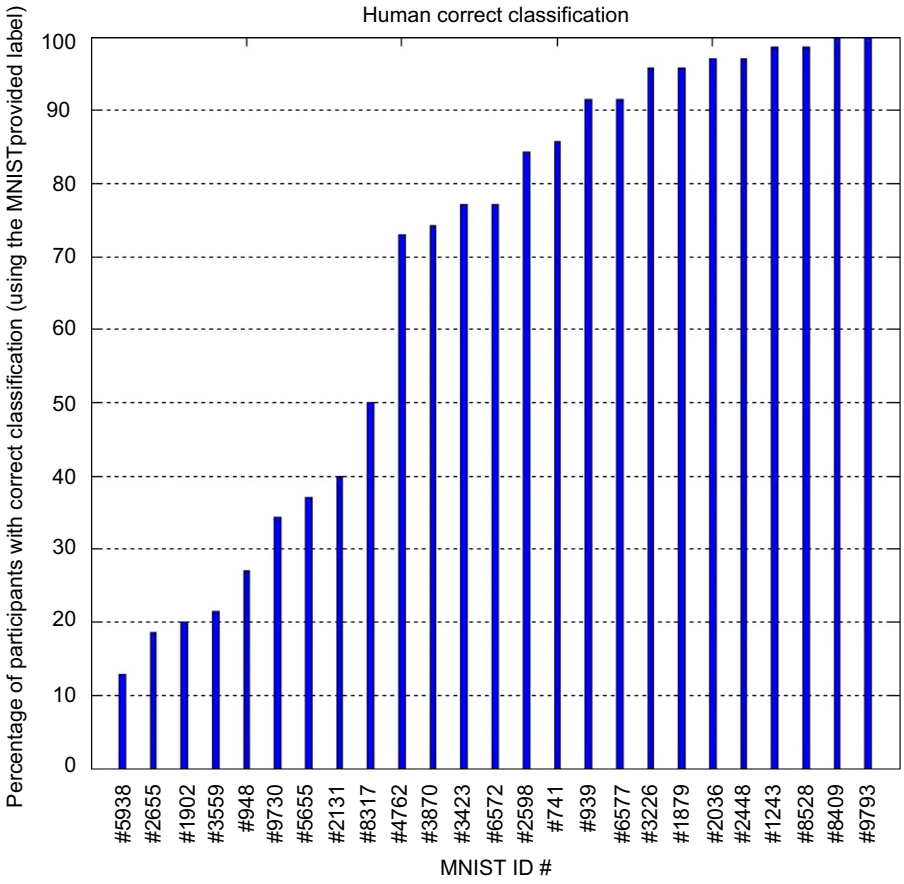

Fig. 7. The percentage of correct classification by humans on 25 testing images from MNIST, using the MNIST provided labels.

from 99.41% by the CNN model to 99.81% by the hybrid model, which is 67.80% (0.19–0.59%) less erroneous than the CNN model. Therefore, it is worth making such sacrifices on the time and memory storage to obtain a higher generalization ability for the hybrid model.

4. Human classification versus machine recognition

In this section, we compare and discuss the differences between humans and machines in the recognition of handwritten digits. Fig. 7 summarizes the percentage of human participants with the correct classifications on the 25 digit images when using the labels provided by MNIST. This survey was conducted at CENPARMI with 70 people participating in 2011. Those 25 digits include all 19 errors produced by the hybrid model. The remaining 6 digits were selected as the most difficult numerals for machines to recognize, after we investigated the errors made by the three classifiers: 6-layer NN [1], VSVM[b] [19,20], and TFE-SVM [4]. The intersection number of misrecognized images is 11 between the hybrid model and 6-layer NN, 10 between the hybrid model and TFE-SVM, and 9 between the hybrid model and VSVM[b]. There are 5 common errors among all the four classifiers. Their MNIST ID numbers are: #1902, #948, #9730, #2131, and #3423.

From Fig. 7, we can see that 2/25 digit images were correctly classified by all the participants; and the rest (23/25 digit images) could be recognized by only a portion of people. For those 2 digits recognized correctly by all the participants, both were correctly classified by the proposed hybrid model without rejections as well.

Table 5
Confusion matrix on the 25 digits by 70 participants.

Truth			Prediction										Percentage of participants with correct recognition (%)
ID#	Shape	Label	0	1	2	3	4	5	6	7	8	9	
#5938		5				61		9					12.9
#2655		6	4	45	6		1		13		1		18.6
#1902		9					54		1	1		14	20.0
#3559		5	29		1		1	15		1	6	17	21.4
#948		8					1				19	50	27.1
#9730		5						24	46				34.3
#5655		7		7	37					26			37.1
#2131		4					28			1		41	40.0
#8317		7			35					35			50.0
#4762		9					19					51	72.9
#3870		9					18					52	74.3
#3423		6	14						54		2		77.1
#6572		9					2			14		54	77.1
#2598		5				11		59					84.3
#741		4		6			60				1	3	85.7
#939		3				64		5				1	91.4
#6577		7		3			1			64		2	91.4
#3226		7		2						67		1	95.7
#1879		8							1	2	67		95.7
#2036		5				2		68					97.1
#2448		4					68					2	97.1
#1243		4					69					1	98.6
#8528		4					69				1		98.6
#8409		8									70		100.0
#9793		4					70						100.0

The 19 numerals misrecognized by the hybrid model belonged to the other 23 digits in the Figure. For the first eight digits in the Figure (from left to right) that could be correctly classified by less than half of participants, all of them were misclassified by the hybrid model. In this case, the handwritten digits that cause difficulty in being recognized by the majority of people can also cause difficulty in being correctly classified by the machine.

More details can be found in Table 5, which shows the confusion matrix on the 25 digits of our survey according to the order of accuracy. For the first digit (#5938) which had the smallest percentage of participants with correct recognition in the Table, there were only 9 subjects (12.9%) correctly recognized it as a digit "5". All the other people regarded it as a digit "3" in the survey. Upon examining this image closely, we see that the digit is so cursively written and there exist no gaps between the top and middle strokes in the upper part of the image. It is so hard for humans to identify what the numeral is without the ground truth, and so it is for the machine. The reason for the misrecognition by the hybrid model might be due to the lack of training samples with such stroke structures. To solve this problem, one way is to import more unseen training samples into the database, and the other way is to use the rejection mechanism to reject it.

5. Conclusion

In this paper, a new hybrid CNN–SVM model has been proposed to solve the handwritten digit recognition problem. This model took the CNN as an automatic feature extractor and it allowed SVM to be the output predictor. The efficiency and feasibility of the proposed model were evaluated in two aspects: the recognition accuracy and the reliability performance. Experimental results on the MNIST digit database showed the great benefit of the proposed hybrid model. It achieved an error rate of 0.19 % with no rejections, and 100% reliability with a rejection rate of 0.56%. Both are the best results up to date compared with other research works.

Our results indicate that the proposed hybrid model is quite a promising classification method in the handwriting recognition field due to three properties: one is that the salient features can be automatically extracted by the hybrid model, while the success of most other traditional classifiers relies largely on the retrieval of good hand-designed features which is a laborious and time-consuming task. The second lies in that the hybrid model combines the advantages of SVM and CNN, as both are the most popular and successful classifiers in the handwritten character recognition field. The third is that the complexity of the hybrid model in the decision process just increases a little bit when compared with the CNN classification model in our experiments, which is desirable when used in practical applications.

Research on the hybrid CNN–SVM learning model is still at an early stage. The performance of the hybrid model can be further improved through the fine tuning of its structure and its parameters. For example, improvements might be made based on the size of the input layer, the number of feature layer maps in layers 2 to 4, the kernel functions used in the model, etc.

Extending the proposed hybrid model to other applications is a task worth investigating. It is very easy to apply our work on the isolated special symbols, such as ",", ".", "?", "!" etc. Without being limited to the handwritten digit recognition, handwritten characters in different languages, such as English, Arabic, French, etc., can also be further studied.

Acknowledgments

This research project was supported by NSERC, the Natural Sciences and Engineering Research Council of Canada, and Concordia University. The authors would also like to thank Dr. Tien D. Bui and Dr. Louisa Lam for their advice, and the anonymous reviewers of this paper for their constructive comments.

References

[1] D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhuber, Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition, CoRR, abs/1003.0358, 2010.
[2] M. Szarvas, A. Yoshizawa, M. Yamamoto, J. Ogata, Pedestrian detection with convolutional neural networks, in: Proceedings of the IEEE on Intelligent Vehicles Symposium, June 2005, pp. 224–229.
[3] K. Mori, M. Matsugu, T. Suzuki, Face recognition using SVM fed with intermediate output of CNN for face detection, in: Proceedings of the IAPR Conference on Machine Vision Applications, Tsukuba Science City, Japan, May 2005, pp. 410–413.
[4] F. Lauer, C.Y. Suen, G. Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recognition 40 (6) (June 2007) 1816–1824.
[5] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[6] C.W. Hsu, C.J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.
[7] C.C. Chang, C.J. Lin, LIBSVM: A Library for Support Vector Machines, Software Available at 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm〉, 2001.
[8] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (2004) 975–1005.
[9] T. Hastie, R. Tibshirani, Classification by pairwise coupling, The Annals of Statistics 26 (1) (1988) 451–471.
[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
[11] W. Pan, T.D. Bui, C.Y. Suen, Isolated handwritten Farsi numerals recognition using sparse and over-complete representations, in: Proceedings of the International Conference on Document Analysis and Recognition, Barcelona, Spain, July 2009, pp. 586–590.
[12] M. Ranzato, C. Poultney, S. Chopra, Y. LeCun, Efficient learning of sparse representations with an energy-based model, in: John Platt Bernhard Schölkopf, Thomas Hofmann (Eds.), Advances in Neural Information Processing Systems, MIT Press, 2006, pp. 1134–1144.
[13] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practice for convolutional neural networks applied to visual document analysis, in: Proceedings of the International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 2, 2003, pp. 958–962.
[14] The MNIST Database of Handwritten Digits, 〈http://yann.lecun.com/exdb/mnist/〉.
[15] P. Simard, Y.L. Cun, J. Denker, Efficient pattern recognition using a new transformation distance, Advances in Neural Information Processing Systems 5 (1993) 50–58.
[16] Y. Mizukami, K. Yamaguchi, J. Warrell, P. Li, S. Prince, CUDA implementation of deformable pattern recognition and its application to MNIST handwritten digit database, in: Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 2000–2004.
[17] D. Keysers, T. Deselaers, C. Gollan, H. Ney, Deformation models for image recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (8) (2007) 1422–1435 August.
[18] H. Cecotti, A. Belaid, Rejection strategy for convolutional neural network for adaptive topology applied to handwritten digits recognition, in: Proceedings of International Conference on Document Analysis and Recognition, Seoul, Korea, 2005, pp. 765–769.
[19] J.X. Dong, HeroSvm 2.1, 〈http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html〉.
[20] C.Y. Suen, J.N. Tan, Analysis of errors on handwritten digits made by a multitude of classifiers, Pattern Recognition Letters 26 (2005) 369–379.

Xiaoxiao Niu received the M.Sc. degree in computer science from Concordia University, Quebec, Canada, in 2011. She is currently working at CENPARMI as a Research Assistant. Her research interests include pattern recognition, handwriting recognition, and image processing.

Ching Y. Suen is the Director of CENPARMI and the Concordia Chair of AI & Pattern Recognition. He received his Ph.D. degree from the University of British Columbia. Currently he is on the editorial boards of several journals related to PR & AI. He has organized numerous international conferences on pattern recognition, handwriting recognition, and document analysis.