Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 7, No. 1, February 2016), Pages: 49-66
www.jacr.iausari.ac.ir

A New Hierarchical Architecture Based on SVM for Persian License Plate Character Recognition

Amir Ebrahimi Ghahnavieh✉, Abolghasem A. Raie

Mobile Robots Research Lab., Faculty of Electrical Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran ,Iran

amir.ebrahimi66@aut.ac.ir; raie@aut.ac.ir

Received: 2015/03/20; Accepted: 2015/07/25

Abstract

Each license plate recognition system is composed of three main parts, namely, license plate detection, character segmentation and character recognition. In this paper, we focus on the improvement and innovation of the character recognition step. For this purpose, a new hierarchical architecture based on Support Vector Machines (SVMs) is suggested for Persian license plate characters recognition. Clustering characters in the hierarchical architecture is proposed based on a new criterion using a confusion matrix. The criterion and the confusion matrix have not been used for clustering in this way. The hierarchical architecture precision is 97/4% and recognizes the entire characters of a license plate in approximately 60ms. All evaluations and comparisons among the performances of previous methods and the proposed method in this paper have been done in the same hardware and software test bed. The dataset obtained from images in real conditions such as, day, night, different distances and angles.

Keywords: *License Plate Recognition, Persian Character Recognition, Hierarchical Architecture, Support Vector Machine.*

1. Introduction

With the development of automobile industry and increasing transportation, the traffic monitoring and extracting traffic parameters by human become more difficult and nearly impossible. Thus, in developed countries the concept of intelligent transport system emerges. The intelligent transportation system is aimed at designing a comprehensive and applied model to control and monitor traffics a part of which is the automatic license plate recognition system. License plate recognition system prepares conditions in which the existing number in automobile license plate could be automatically extracted by computer from still images or video frames with image processing methods to be used as specific characters. Automatic toll payment, identification and issuing fines for offending automobiles automatically are among the examples of using a license plate recognition system [1].

The license plate recognition system contains software and hardware parts. This paper focuses on the software part. The software parts of each system are composed of three stages: license plate detection, character segmentation, and character recognition. In this paper, the main purpose is to improve system in character recognition step. This part is the last but not the least main stage among these three phases for license plate recognition. Various studies have been conducted in both fields of automotive license

plate recognition systems, foreigner and Persian. Two comprehensive overviews on many current techniques could be found in [1,2]. Automobile license plate recognition is still a challenging field for Persian license plate [3-9,10].

A simple way to design a classifier for the license plate characters recognition is the template matching technique. The first stage in template matching is to give system templates as training samples. Then, each character is compared to the templates. The template which has the nearest distance from the character is introduced as the output. Given to the standardized plate format and similar character's font, this method represents desired accuracy but, it is not recommended because of its low speed for real-time license plate recognition systems. In [11], the excellence of probabilistic artificial neural networks in comparison to the template matching method for recognizing the automobile license plate characters is highlighted.

The neural networks are the most popular type of classifiers in character recognition [3-5,12]. These networks are able to segment the data space into several classes simultaneously and can segment classes non-linearly. If a neural network with enough hidden layers and number of neurons is properly designed, it could reach high accuracy. Any types of Multilayer Neural Networks (MLPs), RBF, Kohonen, ART, etc. could be used to recognize the license plate characters. Among these, the MLPs are used according to their user friendly and easy training. The MLP networks have two major drawbacks [13,14]. First, there is not a theoretical relationship between the structure of the network (number of neurons in each layer and the number of layers) and the classifier output. Second, the separator hyper plane does not have maximum margin. This is because as soon as the back propagation error algorithm reaches a separator plane (the termination condition is satisfied), it will stop the training process and does not follow the optimal plane. In other words, the result may be trapped in a local minimum. To remove these defects, SVMs are proposed.

2. Support Vector Machines

The SVM is a binary optimal classifier, which separates the classes with the most appropriate margin. Gaining the best margin increases the SVM generalization capability [14,15]. This issue motivates us to take advantage of this classifier in our proposed method. In SVMs a linear decision function for binary classification and multidimensional data is used. This function will only have two outputs of -1 or +1 corresponding to each class. In the case that the data can be segmented linearly, the separator function is a hyper plane that segments the two class points such as Fig. 1. Reference 10 involves flexibility by using slack variables in the constraints to allow the possibility of misclassification of a limited number of training examples. For simplicity and given to existing images quality in dataset, linear SVM is used.

Differences between SVMs and neural networks are in finding a separator plane in a way that SVM classifier finds the best separator plane, while the neural networks terminate the training process by finding the first plane which satisfies the separation constraints. Another advantage of the SVM compared to neural networks is the ability to reject data which are closer to founded hyper plan in the SVM. These data are likely to be irrelevant and non-characters [13]. In addition, the data space in SVMs could be extended to a higher dimension that gives opportunity to solve problems that are more complex. SVMs, unlike neural networks that work with entire training samples, operate on data which are closer to the separating boundary. In other words, they are not

sensitive to a large dataset. These training samples which are closer to the boundary are known as support vectors.

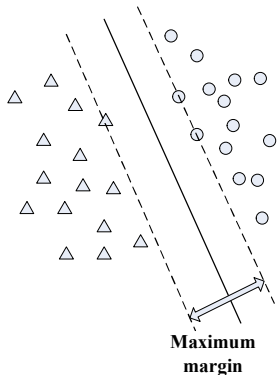

Figure 1. Finding the maximum margin by SVM

The advantage of using hierarchical architectures which includes speed of the train and test are fully expressed in [16,17]. In [18], the SVM classifier is compared to RBF and MLP classifiers in a hierarchical architecture and gives the best results. Moreover, due to the fact that SVM is an optimal binary classifier, it seems to be suitable for a hierarchical architecture.

A typical hierarchical architecture is as Fig. 2 in which some classes are separated in each node. Let to explain some points about key words in this part. There are some circles in each hierarchical architecture. Each of these circles is called a “node” and is considered as a classifier that has some branches getting out of it. It should be noted that the upper branch of a node goes into it and the lower branches go out of it. These branches classify the classes to some macro classes. Then, each macro class is divided into other macro classes by its branches. This continues when macro classes are divided into individual classes. Where macro classes have more classes, it is called high level that starts from zero level, where all classes are there. Low level is where the macro classes broke into classes and the tree is terminated. Some limitations are placed on the number of branches entering or leaving the nodes in the following.

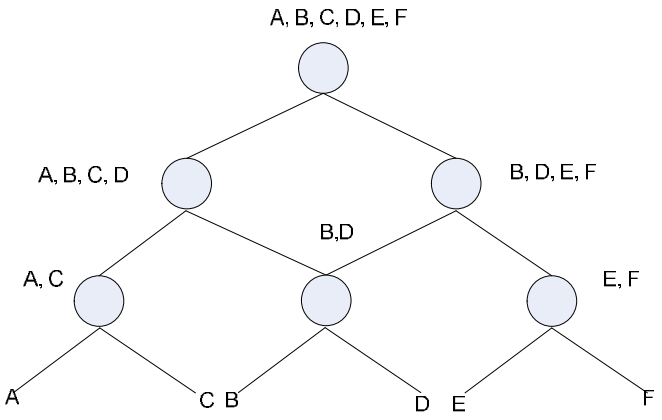

Figure 2. A typical hierarchical architecture

In this article we will try to use the benefits of SVM in a proposed hierarchical architecture. Therefore, first, a hierarchical architecture is designed based on the Persian license plate character recognition problem and then, a novel clustering algorithm is presented. In clustering, each node should divide the classes into two macro classes in order to make classes of each macro class closer to each other and two macro classes far from each one. Hence, our clustering algorithm is based on a criterion for nearness and distance of two classes in each node. This criterion is depending on a confusion matrix and separates the classes into different macro classes so that classes with high similarity are classified in final levels of the hierarchical architecture. This matter results in more accurate classifiers for similar-shaped characters which is the main advantage of our proposed algorithm. Details of our algorithm and the results are brought in Sec 4.1 and Sec 4.2, respectively. The designed hierarchical architecture and the proposed clustering method are completely new in this paper and have not been used elsewhere.

This paper is organized as follows. First, a review on multiclass SVM methods and common methods for clustering in hierarchical architectures is mentioned. Next, the proposed method which is an SVM-based hierarchical architecture is explained in addition to our new clustering method and its results. Finally, the results of the previous methods and the proposed method on a dataset of Persian characters are presented and analyzed.

3. Related Works

3.1 Multiclass SVM

As we have already mentioned, the SVM classifier is a binary one. To use it for automotive license plate character recognition, this classifier should be generalized to multiclass. The main disadvantage of the SVM classifier is its binary mode that could face multiclass problems with difficulty. In this section, the overall approaches to make SVM multiclass and the proposed method is discussed. Some of these methods have been used in handwritten character recognition and some of them have been used for license plate character recognition. Also, some methods have not been employed for character recognition yet.

There are two main approaches to generalize the SVM to multiclass; the first approach is to combine all optimization problems and all the constraints together to construct a multiclass SVM which is not recommended due to the computational cost and problem complexity. The second approach is to combine binary SVMs with one another. This method is the simplest and based on combination of the SVMs, the accuracy and speed of processing is different. Some of these combinations are presented as follows.

The One-Against-All method is the simplest method which is called OAA through this document. In this method, there is one SVM for each class which separates it from the other ones. Regarding to N classes, N SVM classifiers are also designed and verified for a new test sample. After designing the SVMs, the outputs of all the SVMs for each test data are obtained and the separator with positive output is selected as the winner. A disadvantage of this system is the imbalance of data in training an SVM for one class in a way that the number of the training samples of the relevant class is much more fewer than the samples of other classes. The output of the SVM can also be

ambiguous because instead of one SVM, several SVMs can give positive output or all the SVMs could give negative output and a blind area is appeared.

Given that, in [19] for handwritten character recognition, an SVM with less distance from hyper plane would be the winner which is unfair, irrational and necessarily would not give the appropriate result. For this reason, several strategies for mapping outputs of the SVMs are introduced based on their output histogram for license plate character recognition [20]. Having mapped output, it is possible to compare them in terms of magnitude (distance from hyper plane). Among the proposed strategies, using MLP to map the output of SVMs has the best results. In this paper, this method is referred to as MLP-strategy.

Another method of character recognition using SVM is to apply SVM for each pair of classes which is used in [21] to recognize handwritten characters. In this method, the number $N(N-1)/2$ SVMs are trained for N classes. Each input image is sent to all the SVMs and the output of each one is obtained. It is expected that the looked-for class wins more than other classes. This method is called One-Against-One and we call it OAO in the rest of this article. Authors of [21] claimed that this proposed method has more accuracy than other method used in [13]. However, forcing all the SVMs to give a wrong answer may affect the final decision. Also, it creates ambiguity when two classes have the same number of winning. Moreover, the size of the classifier increases sharply with increasing the number of classes and so this method would be ineffective for large problems [14].

Regarding to DAG method that is shown for 4 classes in Fig. 3, a test image is given to each one of the SVMs (circles) separately and SVM outputs determine the subsequent path in a tree root [14]. In this method, first, a list of all classes is defined. Each node is responsible to make decision between the first and the last node from the list by using a pre-trained SVM. Having found the desired class, the other class is removed from the list and the list is updated. This process continues to the next nodes with updated list and proceeds until only one class remains in the list. In this method, there are $N(N-1)/2$ SVMs for N classes but just $N-1$ of them are utilized.

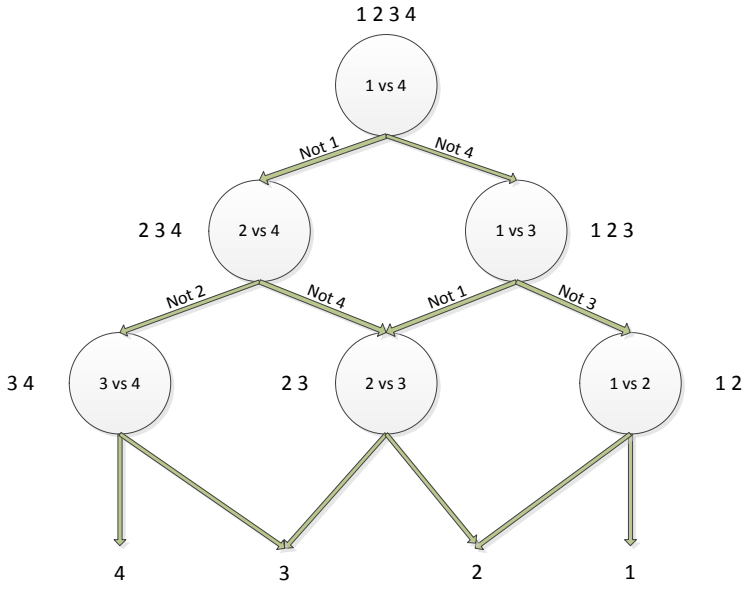

Figure 3. DAG method proposed in [14]

The way we pass to obtain the output is known as evaluation path. By merging the classes below the root, this method improves the generalization capability of the recognition system. In this method, the way to select root source has a great effect on the final result.

NSVM method is used for handwritten character recognition in [22]. Given to OAO, it is noted that there are $N(N-1)/2$ possible SVMs for N classes. In NSVM method, $N/2$ of them are selected randomly. Outputs of them are considered as input of a neural network and the neural network is trained. Finally, the output of the neural network is known as the system output.

Another solution is to combine a neural network and an SVM that is proposed in [13] for handwritten character recognition which is called MLP-SVM in here. The main idea of this method is that the two larger outputs of a neural network are used as two most probable characters and these characters are distinguished from each other by an SVM. Given that the neural network output is not probabilistic, there is no guarantee to raise accuracy by SVM. Thus, in [7,9] the probabilistic classifiers such as a probabilistic neural network and maximum likelihood model are used to reach most probable characters in license plate character recognition and are shown that these probabilistic classifiers give higher accuracy compared to neural networks. The two latter methods are called PNN-SVM and ML-SVM in this paper.

In [23], intuitive clustering is used along with SVM. Intuitive clustering means the characters that are similar in shape can be laid in one cluster. In [24], Kmeans clustering method is used along with SVM and OAO method is used in each cluster for classification. In [25,26], the error correcting code is used to assign SVM classifiers to each class. This method gives high accuracy if assigned codes have high hamming distances with each other. In [26], a probabilistic approach is used for the output of each SVM.

Various methods, advantages, and disadvantages of multiclass SVM have been already described. A number of methods were used for the handwritten character recognition and some of them were used for automobile license plate character recognition. In Sec. 4, a new method based on hierarchical architecture is described.

3.2 Clustering

In this article we will try to use the SVM in a proposed hierarchical architecture. But, first of all a kind of clustering is needed to design a hierarchical architecture. From one perspective, the clustering is conducted by two approaches: bottom-up and top-down. In bottom-up approach, classes are merged in lower level layers and to reach top levels, macro classes are merged together to form larger macro classes [27]. Top-down approach performs unlikely so that the larger macro classes in higher levels are broken down to reach separate classes in the lowest level. In [28], it is discussed that the first approach cannot guarantee to have suitable classifier. In this article, the top-down method is used for clustering classes and macro classes.

The simplest way for clustering is to test all different combinations together in each level and select two groups which have the maximum margins. These two groups should be selected in a way that the distances between the two groups goes maximum and the standard deviation in each group goes minimum. This method needs complex calculations and for instance, for breaking one node with N classes all $2^N/2-1$ cases should be reviewed. If we put the equality limitation in number of classes in two macro

classes in each level, the $1/2\times C_{N/2}^{N}$ cases should be considered. The disadvantage of this method is that different cases need altered training of the classifiers and resolving the problem repeatedly. This study is expensive for large values of M.Therefore, another solution for clustering the classes is needed.

Different criteria are considered for clustering the classes in a top-down manner. In [29], a spherical shell is used. This method uses the average of the classes for segmentation and chooses a radius in the data space so that a macro class is considered inside the sphere and the other macro class out of sphere. In [28], K-means clustering method is applied which uses the mean of the data. According to using mean value, these two methods are not appropriate when we face complex data distribution. In [27], both the mean and the distribution of each class are employed using the bottom-up approach. The lower levels, which are similar to each other, regarding to their proposed criteria, merged together and form a new macro class. This will continue until they reach the higher levels. In [18,30] a confusion matrix is used to cluster classes and similar classes like {o, O, Q, 0} are separated in the lower levels of the classifier. In design of our hierarchical architecture, we try to separate the similar characters in lower levels to divide the generalization capability between them equally.

From another perspective, the clustering is carried out by two approaches: one-against-others and some-against-others [27]. In one-against-others approach, each class is separated from other classes in each node. In some-against-others method, in each node, classes are broken down into two smaller macro classes with the same number of classes. The first approach gives the initial clustered classes this opportunity to have more generalization capability and some kind of priority that have weighty effect on final result. The second approach is recommended so that all classes could have the same generalization capability. Until now, some properties have been introduced to lead hierarchical architecture clustering. These properties are summarized as follows:

a) Only one branch should be entered into each node. Experimental results have led us to this conclusion.
b) SVM is binary classifier and therefore, to get benefit of it, two branches should get out of each node.
c) Each node breaks its own classes into two macro classes with the same number of classes. Equality of number of classes, gives us this assurance to have the same generalization capability for all classes and macro classes.
d) The available 24 classes are broken into 3 macro classes with 8 classes in each one. Due to the exponent 2 of the value 8, all design limitations could be satisfied.
e) Each node should divide the classes into two macro classes in order to make classes of each macro class closer to each other and two macro classes far from each one.

Therefore, it is required to introduce a criterion for nearness and distance of two macro classes.

4. The Proposed Method

Using hierarchical architecture and considering that SVM is a binary classifier, requires the existing classes number to be exponent two but it does not meet such a requirement. The number of the characters of automobile license plate in the dataset is 24 – containing 15 letters and 9 numbers – which is not usable in a hierarchical

architecture. To solve this problem, a multi-class classifier is used on the top of the hierarchical architecture. The multi-mode structure classifier is a probabilistic neural network here. First, by using a probabilistic neural network, 24 classes are broken into 3 macro classes, and then these 8 macro classes are broken down into the given classes in different levels by 21 SVM classifiers. The overall shape of the hierarchical architecture is as Fig. 4. In what follows, we will study how to cluster the classes into macro classes to reach the best results.

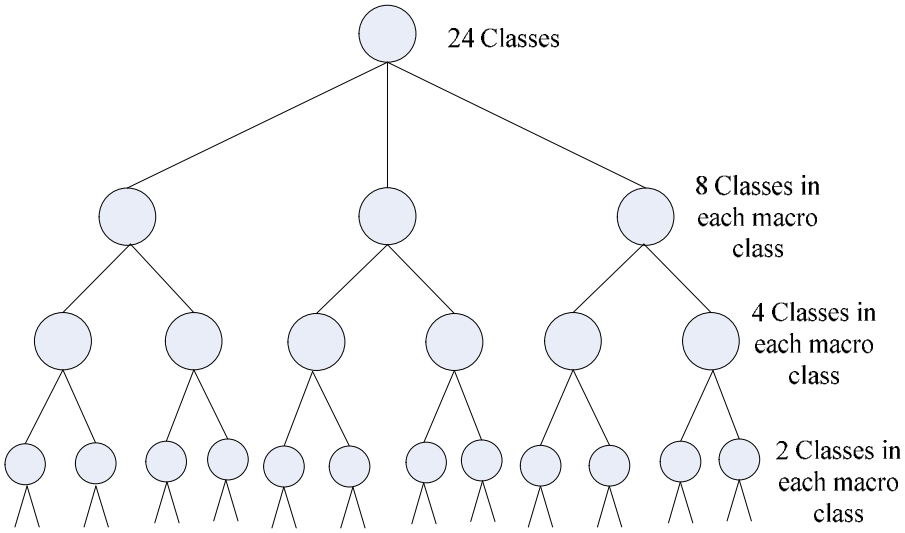

Figure 4. The used hierarchical architecture

4.1 The proposed clustering method

In proposed method, a confusion matrix is used for clustering. In [7] a matrix is introduced as the confusion matrix that presents the misclassification of the characters. The matrix from all three methods MLP-SVM, PNN-SVM and ML-SVM are added together for half of the test samples and employed as the confusion matrix for hierarchical architecture clustering.

The presented matrix in [19] is called here as S matrix so that $S(i, j)$ is the similarity between classes i and j and D_i is considered as the error vector for class i as:

$$D_i(j) = S(i, j) + S(j, i), \quad j = 1,2,\ldots 24 . \tag{1}$$

And also a value is considered as the error value for each class as:

$$E_i = \sum_j D_i(j). \tag{2}$$

The clustering algorithm:

1. All the available classes for breaking are listed which is called the *base list* in here.
2. Two classes from this list are searched to find two classes with the lowest error value. After finding the classes, they are set into two different macro classes and used as a baseline for breaking other classes.
3. For each founded classes, a list is formed; *list1* and *list2*. In each list just one class exists so far.
4. The following vector is formed:

$$F1_i = \sum_{m \in list1} D_m - \sum_{n \in list2} D_n \ . \tag{3}$$

Then $\arg \max_{i \in base\ list} F1_i$ is founded and the corresponding class is removed from the base list and added to the list1.

1. Afterwards, the $F2_i$ vector is formed by updated lists as below:

$$F2_i = \sum_{n \in list2} D_n - \sum_{m \in list1} D_m \ . \tag{4}$$

This time $\arg \max_{i \in base\ list} F2_i$ is founded and the corresponding class is removed from the base list and added to the list2.

2. If the number of remained classes is equal to four, the algorithm is terminated. Else, the algorithm is continued from step 4.

Algorithm continues until 4 classes are remained in each macro class being broken down into two macro classes with two classes in each one. In this situation, 2 classes i and j in the base list with 4 classes i, j, m, n are examined to maximize $D_i(j) + D_m(n)$. Consider that $D_i(j) = D_j(i)$ and $D_m(n) = D_n(m)$. After finding i, j, m, n, classes i and j are laid in one macro class and classes n and m in another macro class. The flowchart of this algorithm is shown in Fig 5.

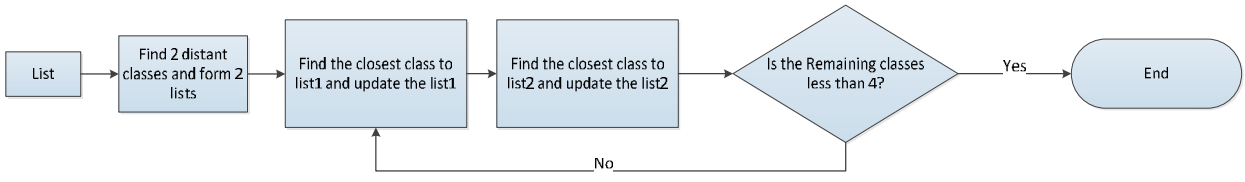

Figure 5. The flowchart of clustering algorithm

4.2 Clustering results

24 available classes are presented in Table 1. According to the proposed algorithm, the classes are categorized in each level.

Table 1. 24 available classes

۱-۲-۳-۴-۵-۶-۷-۸-۹-الف-ب-س-د-ع-ق-هـ-ح-ل-م-ن-ص-ط-و-ی

Level 0:

At this level, total classes are divided to 3 macro classes with 8 classes in each one. 8 classes which have 87% of misclassifications in [7] are laid in macro class 2 and for the rest of the classes, the above algorithm is used. In this level, a probabilistic neural network is used for classification.

Table 2. Macro classes of level 0

Macro class 1	**Macro class 2**	**Macro class 3**
ی-د-ح-ع-م-ط-ص-س	۲-۳-۴-۵-۶-۷-هـ-و	الف-ل-ن-ق-ب-۱-۹-۸

Level 1:

Table 3. Macro classes of level 1

Macro class 4	**Macro class 5**	**Macro class 6**	**Macro class 7**	**Macro class 8**	**Macro class 9**
ی-د-ح-ع	م-ط-ص-س	الف-۱-۹-۸	ل-ن-ق-ب	۵-۶-هـ-و	۷-۴-۳-۲

Level 2:

Table 4. Macro classes of level 2

Macro class 10	**Macro class 11**	**Macro class 12**	**Macro class 13**	**Macro class 14**	**Macro class 15**
ح-ع	ی-د	ص-س	م-ط	۹-۱	۸-الف
Macro class 16	**Macro class 17**	**Macro class 18**	**Macro class 19**	**Macro class 20**	**Macro class 21**
ن-ق	ل-ب	هـ-۵	و-۶	۷-۲	۴-۳

In what follows, for each macro class pair, a classifier is designed. 21 SVM classifiers are trained using our dataset. For testing each character, it passes a path to reach the corresponding class according to the outputs of the SVM classifiers. In hierarchical method, if the number of the classifiers is equal to $N-1, \log_2 N$ of them are used for each test character. For our hierarchical method, 3 SVM classifiers and 1 PNN classifier in baseline level are employed.

5. Results and Discussion

5.1 Test Bed

The license plate character recognition system of this paper is implemented by a PC with Core2 Quad CPU 2.67GHz and 3.5 GB RAM. The software which is used for simulation is MATLAB 2010. The dataset contains about 20145 characters that are obtained from images in real situations i.e. day, night, different distances and angles and provided by Bani Nick Pardazesh Company [31]. Available images often have angles, the distance between camera and license plates is variable and there are differences between camera and plate in height. The dataset images are resized to 20×12 low-resolution images where some images can be seen with noise, distortion and elimination. Detailed explanations regarding the dataset and all types of images are presented in [32].

5.2 Test Results

Character recognition contains two phases of features extraction and classification. Zoning method is used to extract 15 features. The zoning features are obtained from determining the average of light pixels in each zone. In this section, we tend to compare the mentioned methods in Sec. 3.1 and the proposed method. For training, 1200 characters (50 samples for each class) are used randomly and the test data is divided into two parts. The first part consists of 9479 characters and is used for achieving confusion matrix by testing three methods MLP-SVM, ML-SVM and PNN-SVM. This confusion matrix is used for clustering the characters in proposed hierarchical architecture.

The second part of the test data contains 9466 characters, which is used to compare different classification methods. For all the classifiers, the same training data (1200 characters) and parameters are used. For MLP-based methods a 15×20×24 structure is used. It should be noted that the neural networks have extensive parameters for training. The used structure and training parameters presented suitable results on the dataset but they are not optimized for the neural networks. Table 5 shows the accuracy and speed of implemented methods on the second part of the test data. The second column is accuracy and the third column is the run time for the whole test samples. Since each Persian license plate has 8 characters, it takes 8 times for a system to recognize the whole of a license plate, instead of just a single character. The average time to recognize a license plate is shown in the last column of Table 5.

Table 5. Accuracies and recognition speeds obtained for several methods

Method	Accuracy (%)	Testing time (s)	Recognition time (ms)
The Proposed Method	97.41	70.92	59.93
PNN-SVM [7]	96.70	67.89	57.37
ML-SVM [7]	95.87	118.04	99.76
MLP-SVM [13]	95.86	109.93	92.90
PNN [11]	96.23	60.41	51.05
DAG [14]	94.89	127.96	108.14
MLP strategy [20]	78.94	143.82	121.55
NSVM [22]	91.94	120.37	101.73

Among simulated methods, the OAA method with about 30% accuracy and OAO method with 2 seconds processing time on each license plate are not appropriate. However, the accuracy of OAO is about 95% and the processing time of OAA for a license plate is about 65ms. The difference between the three classifiers of MLP-SVM, ML-SVM and PNN-SVM is in the way of separating the most probable characters before applying the SVM. Among these three methods, the PNN-SVM is shown to be the fastest processing time and also provides a somewhat better output accuracy than the other methods. In all the above three methods, the SVM uses two probable characters.

Figure 6 shows the percentage of the presence of the correct character when choosing more than two probable characters for all the three mentioned approaches. As can be observed, the PNN-SVM and ML-SVM methods have had a better performance in recognizing the probable character, compared with the MLP-SVM method. Also, the PNN has performed better when one character was to be chosen as the output; however, with the increase in the number of more probable characters, the ML has been more successful although the two methods differ slightly and ultimately exhibit the same performance.

As it is observable from Table 5, the fastest method is PNN due to its parallel structure [11]. Also, adding SVM classifiers to the PNN method improved its accuracy slightly. In remaining methods, NSVM and MLP-strategy are not reliable for an ALPR system with 91.94% and 78.94% accuracy, respectively. However, DAG and our proposed method give more desirable accuracy. The proposed clustering algorithm clusters the similar characters in final stages of the hierarchical architecture and gives the same generalization capability between them. Therefore, similar characters will be classified by more accurate classifiers. This is the main advantage of our proposed method.

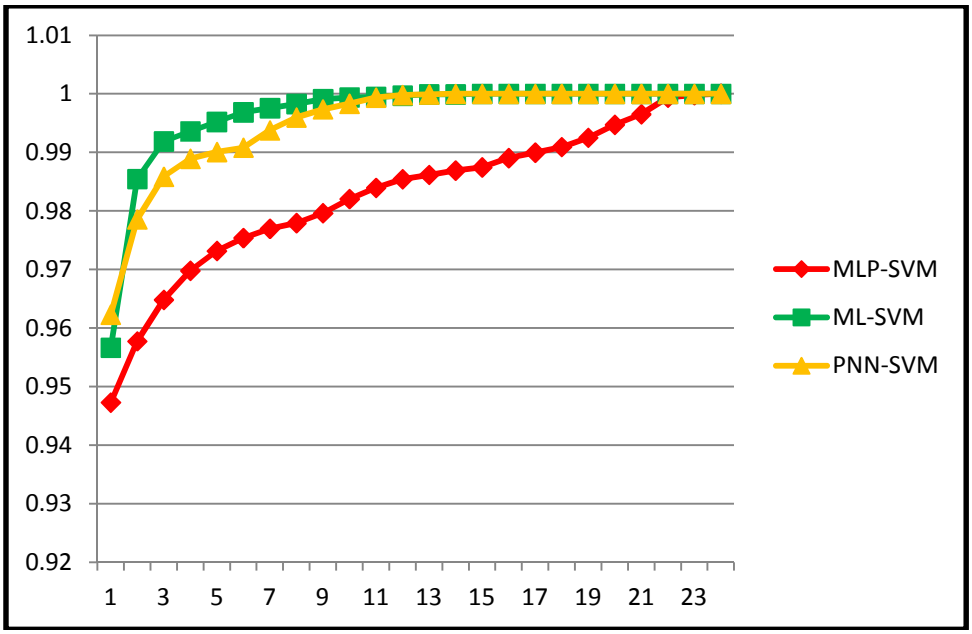

Figure 6. The probability of the presence of the correct character when choosing more than two probable characters

5.3 Misclassifications Analysis

Some samples of misclassifications in our proposed method are shown in Table 6. As it is obvious, some of the characters are difficult to be recognized even by the human eye. In Table 7, the joint misclassification percentage matrix of the more accurate studied methods is given. The goal of this table is to discuss the possibility of improving our proposed method by combining with other methods. In other words, we would like to investigate if there are misclassified characters of the proposed method being difficult to be recognized by other methods or not. For instance, in Table 7, the row corresponding to PNN and the column corresponding to DAG show that 77.87% of the misclassified characters of PNN are also misclassified in DAG.

Table 6. Some samples of misclassifications in our proposed method

Character image	Correct class	Misclassified with:
	٢	٣
	٣	٢
	٤	٢
	٤	٦
	٧	٢

Most of the misclassifications of the DAG are misclassified by other methods, regarding the large values of the column corresponding to DAG. The main disadvantage of DAG is the effect of selecting its root. In this method, the first and the last classes – class#1 and class#24 in here – are selected as the root and therefore, they exit from the competition sooner than the other classes. Suppose that the first and the last 6 classes are in category#1 and 12 middle classes are in category#2. About 98% of the DAG misclassifications occur in the category#1 compared with 81% for other studied methods of Table 5 in average. In DAG, about 29% of the samples are misclassified with category#2 compared with 18% for other studied method of Table 5 in average. This means that DAG tends to misclassify the classes in the first category with the classes in the second one.

By analyzing the row corresponding to the proposed method, we can say that combining MLP and SVM could give a more accurate classifier due to the joint misclassification percentage value between the MLP-SVM and the proposed method. It is noticeable to mention that 3%, 6%, 27% and 64% of misclassifications happened in PNN classifier, level0 SVM classifiers, level1 SVM classifiers and level2 SVM classifiers, respectively. It shows that making decisions on characters in lower levels of the hierarchical architecture is more difficult.

Table 6. The joint misclassification percentage matrix

	The Proposed Method	**PNN-SVM [7]**	**ML-SVM [7]**	**MLP-SVM [13]**	**PNN [11]**	**DAG [14]**
The Proposed Method	-	51.02	22.04	21.63	49.39	52.65
PNN-SVM [7]	40.06	-	56.41	35.26	68.59	95.51
ML-SVM [7]	13.81	45.01	-	50.13	38.62	89.26
MLP-SVM [13]	13.52	28.06	50	-	48.72	56.38
PNN [11]	33.89	59.94	42.30	53.5	-	77.87
DAG [14]	26.65	61.57	72.11	45.66	57.44	-

5.4 Low Resolution Challenges

The resolution of the characters is a significant challenge in character recognition systems. Table 8 shows a comparison between the related works in terms of accuracy and resolution. As it is observable, many researches are very accurate in high resolution images of handwritten characters [19,23] and license plates characters [11,33]. Nevertheless, classification methods are not satisfying for low resolution images of handwritten characters [22] and license plates characters [12]. The effectiveness of our proposed method on low-resolution characters is obvious in Table 8. The goal of bringing this table is to show how low resolution characters lead to least accurate results. Since the dataset is different in all of the below methods, comparing the results

of this table is pointless. Nevertheless, analysis of the results in Sec 5.2 and Table 5 which includes most of the methods in Table 8 are on the same dataset and valid.

Table 7. A comparison between the related works in terms of accuracy and resolution

Ref.	Publication Year	Classification Method	Resolution	Character Recognition Application	Accuracy (%)
The Proposed Method	-	SVM-based Hierarchical Structure	20×12	License Plates (Persian)	97
[5]	2012	Neural Networks	30×60	License Plates (Persian)	91
[34]	2012	SVM	32×18	License Plates	96
[7]	2012	ML-SVM	20×12	License Plates (Persian)	96
[7]	2012	PNN-SVM	20×12	License Plates (Persian)	97
[22]	2012	Neural Networks + SVM (NSVM)	16×16	Handwritten	85
[4]	2011	Neural Networks	Not Rep.	License Plates (Persian)	92
[23]	2011	SVM	32×32	Handwritten	98
[33]	2011	SVM	Not Rep.	License Plates	96
[24]	2010	SVM (OAO)	Not Rep.	Handwritten (Persian)	95
[3]	2010	Neural Networks	Not Rep.	License Plates (Persian)	96
[20]	2009	Neural Networks + SVM* (MLP strategy)	16×16	Handwritten	95
[18]	2009	SVM-based Hierarchical Structure	32×32	Handwritten	85
[17]	2008	SVM-based Hierarchical Structure	Not Rep.	License Plates	82
[27]	2008	SVM-based Hierarchical Structure	Not Rep.	Handwritten	98
[12]	2006	Neural Networks	12×9	License Plates	90
[11]	2005	PNN	30×20	License Plates	99
[13]	2003	Neural Networks + SVM (MLP-SVM)	Not Rep.	Handwritten	98
[19]	2000	SVM (OAA)	64×64	Handwritten (Persian)	94

* Separate classifiers for digits and letters.

6. Conclusion and Suggestions

In this paper, a hierarchical architecture based on the SVM is proposed for Persian license plate character recognition. The proposed hierarchical architecture is designed based on a new clustering method and a confusion matrix which has 97/41% accuracy in 59.93ms. The aim of the proposed clustering algorithm was to cluster the similar characters in final stages of the hierarchical architecture and give the same generalization capability between them. The used confusing matrix is obtained from the misclassification matrix of MLP-SVM, PNN-SVM and ML-SVM and the clustering results show that how similar characters are laid in the same macro classes.

In the result section, the accuracy and the speed of described methods are presented and compared with each other based on the dataset. The OAA and OAO were rejected due to low accuracy and speed, respectively. Among other methods, the hierarchical architecture and PNN was the best method in terms of accuracy and speed, respectively. The main disadvantage of a hierarchical architecture is that if an error happens in top levels, it spreads to lower levels. This problem could be solved by using probabilistic classifiers. In a way that instead of a strict decision, a probabilistic value is used in each level.

References

[1] Christos-Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos and Eleftherios Kayafas, "License plate recognition from still images and video sequences: A Survey," IEEE Transactions on Intelligent Transportation Systems, Vol. 9, No. 3, pp. 371–391 (2008).

[2] Shan Du, Mahmoud Ibrahim, Mohamed Shehata and Wael Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 23, No. 2, pp. 311–325 (2013).

[3] Mojdeh Akhtari and Karim Faez, "The application of a CICA neural network on Farsi license plates recognition," 10th International Conference on Hybrid Intelligent Systems, pp. 205–208 (2010).

[4] Sedigheh Ghofrani and Mehran Rasooli, "Farsi license plate detection and recognition based on characters features," Majlesi Journal of Electrical Engineering, Vol. 5, No. 2 pp. 44–51 (2011).

[5] Mohammad Kadkhod and Ebrahim Zarei Zefreh, "License plate recognition system for Persian vehicles," Journal of Academic and Applied Studies, Vol. 2(4), pp. 1–11 (2012).

[6] Mohammad Mahdi Dehshibi and Rahele Allahverdi, "Persian vehicle license plate recognition using multiclass AdaBoost," 3rd International Conference on Signal Acquisition and Processing (ICSAP), Vol. 1, pp. 265–268 (2011).

[7] Ebrahimi A and Raie A., "License plate character recognition using multiclass SVM," Journal of American Science, Vol. 8(1s), pp. 38–42 (2012).

[8] A. A. Shahraki, A. E. Ghahnavieh and S. A. Mirmahdavi, "License plate extraction from still images," In Proceedings of IEEE, 4th International Conference on Intelligent Systems, Modelling and Simulation (ISMS'2013), Bangkok, Thailand, pp. 45-48 (2013).

[9] A. E. Ghahnavieh, A. Amirkhani-Shahraki and A. A. Raie, "Enhancing the license plates character recognition methods by means of SVM," In Proceedings of IEEE, 22nd Iranian Conference on Electrical Engineering (ICEE 2014), Tehran, Iran, (2014).

[10] Amir Ebrahimi, Abdollah Amirkhani, Abolghasem A. Raie and Mohammad R. Mosavi, "Car license plate recognition using color features of Persian license plates," Journal of Advances in Computer Research, Vol. 6, No. 4, 2015.

[11] Yafeng Hu, Feng Zhu, and Xianda Zhang, "A novel approach for license plate recognition using subspace projection and probabilistic neural network," Advances in Neural Networks, Lecture Notes in Computer Science, Vol. 3497, pp. 216–221 (2005).

[12] C. Anagnostopoulos, I. Anagnostopoulos, E. Kayafas, and V. Loumos, "A license plate recognition system for intelligent transportation system applications," IEEE Transactions on Intelligent Transportation Systems, Vol. 7, No. 3, pp. 377– 392 (2006).

[13] A. Bellili, M. Gilloux and P. Gallinari, "An MLP-SVM combination architecture for offline handwritten digit recognition," International Journal on Document Analysis and Recognition, Vol. 5, No. 4, pp. 244–252 (2003).

[14] John C. Platt, Nello Cristianini and John Shawe-Taylor, "Large margin DAGs for multiclass classification", MIT Press, pp. 547–553 (2000).

[15] C. Cortes and V. Vapnik, "Support-vector network," Machine Learning, Vol. 20, pp. 273–297 (1995).

[16] Platt, J., "Fast training of support vector machines using sequential minimal optimization," Advance in Neural Information Processing Systems, pp. 336–342 (1999).

[17] Lihong Zheng, Xiangjian He, Qiang Wu, Wenjing Jia, Bijan Samali and Marimuthu Palaniswami, "A hierarchically combined classifier for license plate recognition," 8th IEEE International Conference on Computer and Information Technology (CIT), pp. 372–377 (2008).

[18] Tapan Kumar Bhowmik, Pradip Ghanty, Anandarup Roy and Swapan Kumar Parui, "SVM-based hierarchical architectures for handwritten Bangla character recognition," International Journal of Document Analysis and Recognition, Vol. 12, pp. 97–108 (2009).

[19] Javad Sadri, Ching Y. Suen and Tien D. Bui, "Application of support vector machines for recognition of handwritten Arabic/Persian digits," 2nd Iranian Conference on Machine Vision and Image Processing (MVIP), Vol. 1, pp. 273–297 (2003).

[20] Tiago C. Mota and Antonio Carlos G. Thome, "One-against-all-based multiclass SVM strategies applied to vehicle plate character recognition," In Proceedings of International Joint Conference on Neural Networks, pp. 2153–2159 (2009).

[21] Renata F. P. Neves, Alberto N. G. Lopes Filho, Carlos A.B.Mello and Cleber Zanchettin, "A SVM based off-line handwritten digit recognizer," IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 510–515 (2011).

[22] Hassiba Nemmour and Youcef Chibani, "N-SVM combination and tangent vectors for handwritten alphanumeric character recognition," 5th International Symposium on I/V Communications and Mobile Network (ISVC), pp. 1–4 (2012).

[23] Lucian-Ovidiu Fedorovici and Florin Dragan, "A comparison between a neural network and a SVM and Zernike moments based blob recognition modules," 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, pp. 253–258 (2011).

[24] Mehdi Salehpour, and Alireza Behrad, "Cluster based weighted SVM for the recognition of Farsi handwritten digits," 10th Symposium on Neural Network Applications in Electrical Engineering, Faculty of Electrical Engineering, University of Belgrade, Serbia, pp. 219–223 (2010).

[25] Zelong Wang, Fengxia Yan, Feng He and Jubo Zho, "A new SVM multi-class classification method based on error-correcting code," International Conference on Computational Intelligence and Security, pp. 21–24 (2008).

[26] Zhanyi Wang, Weiran Xu, Jiani Hu and Jun Guo, "A multiclass SVM method via probabilistic error-correcting output codes," International Conference on Internet Technology and Applications, pp. 1–4 (2010).

[27] Lili Cheng, Jianpei Zhang, Jing Yang and Jun Ma, "An improved hierarchical multi-class support vector machine with binary tree architecture," International Conference on Internet Computing in Science and Engineering, pp. 106–109 (2008).

[28] Yu-Chiang Frank Wang and David Casasent, "New support vector-based design method for binary hierarchical classifiers for multi-class classification problems," Elsevier: Neural Networks, Vol. 21, pp. 502–510 (2008).

[29] Vural, V. and Dy, J.G., "A hierarchical method for multi-class support vector machines," In Proceedings of the International Conference on Machine Learning, pp. 105–113 (2004).

[30] Friedhelm Schwenker, "Hierarchical support vector machines for multi-class pattern recognition," 4th International Conference on knowledge-Based Intelligent Engineering Systems & Allied Technologies, Brighton, UK, Vol. 2, pp. 561–565 (2000).

[31] http://www.baninick.com

[32] Amir Ebrahimi Ghahnavieh, Mahmoud Enayati and Abolghasem A. Raie, "Introducing a large dataset of Persian license plate characters," Journal of Electronic Imaging, Vol. 23, No. 2, 023015 (2014). doi:10.1117/1.JEI.23.2.023015.

[33] Ying Wen, Yue Lu, Jingqi Yan, Zhenyu Zhou, Karen M. von Deneen, and Pengfei Shi, "An algorithm for license plate recognition applied to intelligent transportation system," IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 3, pp. 830–845 (2011).

[34] Liuy Ongchun and Yang Jing, "Research of license plate character features extraction and recognition," 2nd International Conference on Computer Science and Network Technology, pp. 2154–2157 (2012).