Multi Prototype Fuzzy Pattern Matching for Traffic Signs Recognition

Dr. Dhabe P. S.
Department of Information Technology and MCA
Vishwakarma Institute of Technology
Pune, Maharashtra 411037, India

priyadarshan.dhabe@vit.edu

Geetesh Lokhande, Dhruv Sonone & Surbhi Audichya
Department of Information Technology and MCA
Vishwakarma Institute of Technology
Pune, Maharashtra 411037, India
{geetesh.lokhande13 & dhruv.sonone14 & surbhi.audichya14}@vit.edu

Abstract: - This paper presents a novel method of multi prototype fuzzy matching. The ability of classification and recognition is tested with a realistic database of Indian traffic signs. It calculated multiple prototypes of a single class using a fuzzy parameter. Its performance is found to be far greater than the traditional method of prototype matching and recognition where only a single class prototype is used. The proposed method also eliminated the serious barriers of a single prototype method where we cannot increase the recognition using the same training data set used for the calculation of prototype after its calculation.

Key-Words: - prototype matching, classification, recognition, traffic sign, traffic sign recognition.

1. INTRODUCTION

BACKGROUND

The traffic and the road signs considered in this paper are those that use a symbolic or a visual language about the road(s) ahead that can be interpreted by drivers. The terms are interchangeably in this paper and elsewhere might also appear in combination, as ‘Indian traffic signs’. A type of sign that is not considered in this paper is the direction sign, in which the upcoming directions for getting to numbered routes or named towns are essentially by text and not symbolically.

An automatic means of recognising and detecting the traffic signs can make a signification amount of contribution to this goal by providing a quick method of detecting, classifying and logging signs. This method will help to develop the inventory consistently and accurately. After this is done, the detection of obscured and disfigured signs becomes easier for human operator.

The Road and traffic sign recognition is a field of study that can be used to help the development of an inventory system (where a real-time recognition is not required) or aid the development of an in-car advisory system (where real-time recognition is necessary).

COMPLEXITY OF THE RECOGNITION TASK

A normal road in the most of the cities in the world like one shown in the figure 1.0 below presents a complex scene.

Figure 1: A traffic scene in the middle of a street

It may include vehicles, people different coloured vehicles, numerous shops and their signs and a number of traffic signs to control the traffic on the road. Fundamentally, if a person is asked to point the traffic sign in the image, they can do it very easily. But, from the point of view of a computer vision, this image contains some difficulties which are:

1. The existence of a number of similar objects.
2. Presence of obstacles that may partially or totally occlude the sign.
3. The amount of information in the scene is vast and some time is needed to analyse the scene and extract the required information.

AIM & OBJECTIVE

While Road and traffic sign recognition is still a hot research topic, matching the class prototype is a traditional method used for this task. In this method, a single prototype for a class is generally taken as average of all the patterns used to calculate class prototype. After calculating the Euclidean distance between prototype and applied pattern, a minimum distance classifier is used. This method is well suited for the pattern recognition tasks where the patterns within a class has less deviation from one another. Since in case of traffic signs, there is quite an amount of variation in the signs of a pattern. Road signs are acquired using a digital camera for the purpose of analysis. Following facts can be detected immediately by observation:

1. Each sign has a considerable variation in their shape.
2. The angle of orientation is different as the still image is captured using a moving camera.
3. Different perspectives may produce images from different angles.
4. Signs may suffer from motion blur.
5. Weather condition may change the color of the signs.

Figure 2: Variations in shape of a traffic sign

So, if we were to calculate a single prototype of a class of the signs depicted in figure 2, then that prototype will not be able to capture all the peculiarity of all the patterns within that class.

The central idea of this paper is to calculated more than one prototype for a single class to capture such peculiarities of all the possible shapes. From figure 2, can be observed that we can create separate prototypes from the first, second and third sign and from fourth, fifth, sixth and seventh sign. Then we can expect to get a better recognition with prototype matching. From this idea we worked in this direction and out work leads to a fuzzy system that can be used for recognition and classification of traffic signs.

The remaining part of this paper is organised as follows. Section 2 discussed the image database of 864 variations of different traffic sings used for the experimentation purpose along with image normalisation with respect to scale and translation. Section 3 explores the method used for calculation of multiple prototypes. Section 4 elaborated on a 2-D example explaining the behaviour of the proposed method for better understanding. The Experimental results are given the section 5 along with performance comparisons. Conclusions are withdrawn in section 6 based on the empirical evidences and Reference are cited at the end of this paper.

2. IMAGE DATABASE

The image dataset used in this works consists of the images of the Indian road and traffic symbols. These images were obtained from reliable internet source. The images were classified into Mandatory signs, Cautionary signs and Informatory signs. For the sake of simplicity of this project, we only used the Mandatory signs. These sings were translated and scaled into size of 200x200px and stored as .bmp images. The images were then also transformed into grayscale mode, in order to make it easier to calculate the number of white and black pixels.

There were a total of 36 images in the Mandatory sign category. For each of the image (class) we created about 24 variations (features) so the images vary in terms of rotation, distortion, wrap, perception and color.

Figure 3: Classes of Mandatory signs

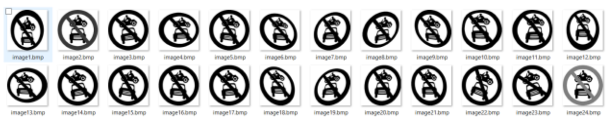

Figure 4: Variations of one of the classes

3. MULTIPLE PROTOTYPE CALCULATION

In order to calculated the multiple prototypes used by the proposed method, we have used a ‘fuzzy membership function. This function calculated membership of input pattern R_h in a class prototype C_i as shown in (1):

$$f(C_i) = \begin{cases} 0, & \gamma.d \geq 1 \\ 1-\gamma.d, & \gamma.d < 1 \end{cases}$$

Equation 1

Where, we calculate distance d as

$$d = \left(\sum_{j=0}^{m}\left(C_{ij} - R_{hj}\right)^2\right)^{1/2}$$

Equation 2

Here, γ is the sensitivity parameter that regulates how quickly the membership value decreases as the distance between cluster prototype and the input pattern increases, where $\gamma > 0$. The membership function designed in Equation (1) holds all the properties of fuzzy sets like normality and convexity. The plot membership functions for a 2-D class prototype (0.5, 0.5), with $\gamma = 1$ & $\gamma = 4$ are shown in figure 5.

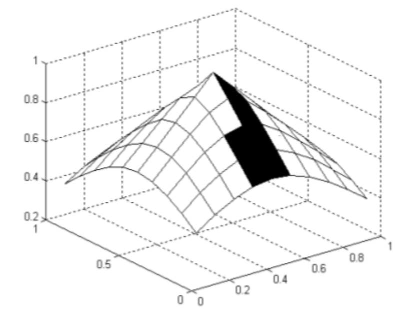

Figure 5a: Membership function plot for γ=1

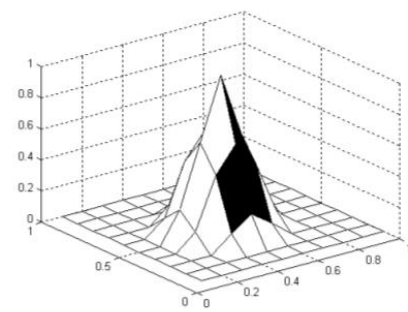

Figure 5b: Membership function plot for γ=4

Prototype calculation requires to find out natural grouping of patterns within a class and then the largest group within the class is considered for calculation of a single prototype for this selected group. In order to find a pattern group, we have used a parameter called the grouping factor $0 < \alpha \leq 1$ which defines the size of pattern group and thus controls number of prototypes created within a class. More value of α leads to the creation of more prototypes within the given class.

Let that the pattern set R=$\{R_h | h = 1,2 \ldots k\}$, contains n dimensional k patterns of m class. To calculate pattern grouping of i^{th} class patterns, we have to get a set S of all patterns belonging to i^{th} class, where $S \subset R$. To determine the pattern group of this class, all the patterns are applied to each of the pattern assuming them as a class prototype and fuzzy membership of each pattern is calculated in all other patterns using (1). The patterns that give fuzzy membership value larger than α are counted for all the patterns. Let $R_j \subset S$ is the pattern with the maximum count and S^1 be the set of these p patterns falling close around R_j with fuzzy membership $\geq \alpha$.

Then the class prototype is computed as

$$C_{ij} = \frac{1}{p}\left(\sum_{j=1}^{p} s_i^1\right) \text{ for } i = 1, 2, 3 \text{ …. n}$$

Equation 3

Now the patterns which are already grouped and present in S^1 are removed from S and the patterns which are not grouped are considered for the calculation of next prototype within the same class. This process repeats until all the patterns of a class are grouped and S will become empty. The same steps are repeated for class C =1, 2, 3… *m*. Thus for each class we get multiple prototypes.

4. A 2-D EXAMPLE

To explain the calculation if multiple prototypes an example in 2-D pattern space would be easier. The patterns are selected to capture all the behavioural possibilities of the algorithm. The selected patterns are listed in the table below and its scatter plot is shown in figure 6. All the patterns are belonging to the same class. It can be confirmed from the scatter plot too that we cannot capture all the peculiarities of all the patterns using a single prototype. It is expected that at least three prototypes are needed for better recognition. The order of data presentation to the system is same as that shown in the table below.

P1=[0.5,0.7]	P2=[0.6,0.7]	P3=[0.55,0.65]
P4=[0.5,0.6]	P5=[0.6,0.6]	P6=[0.1,0.2]
P7=[0.2,0.2]	P8=[0.15,0.15]	P9=[0.1,0.1]
P10=[0.2,0.1]	P11=[0.7,0.3]	P12=[0.8,0.3]
P13=[0.75,0.25]	P14=[0.7,0.2]	P15=[0.8,0.2]

By selecting α = 0.85 we have given patterns listed within the table above. The first largest group selected for calculation of prototype contains (P1, P2, P3, P4, P5) and the calculated prototype is (0.55, 0.65). Now these patterns are removed from the dataset and the remaining ten patterns from P6..P15 are considered for pattern grouping and prototype calculation. When the remaining ten patterns are considered, it has selected largest group containing patterns (P6, P7, P8, P9, P10) and calculated second prototype as (0.15, 0.15).

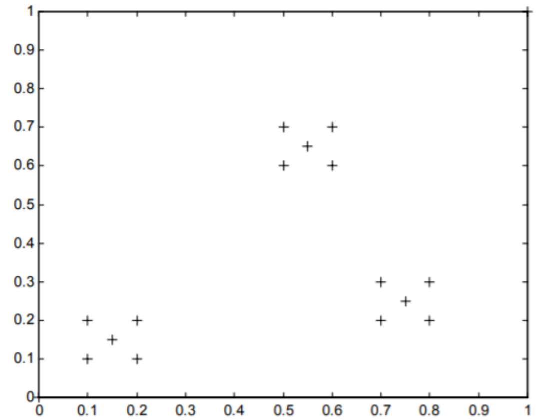

Figure 6: Scatter plot of 2-D example

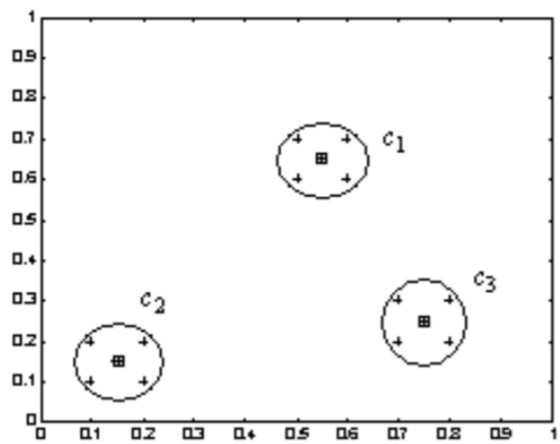

Figure 7: Pattern groups and multiple prototypes

After removing these patterns only five patterns are remained in the dataset and the algorithm has selected largest group containing (P11, P12, P13, P14, P15) and calculated prototype as (0.75, 0.25). The patterns falling in the largest group when removed from the dataset and when dataset becomes empty, the algorithm is halted by successfully giving us multiple prototypes.

Thus algorithm has created three prototypes for the patterns of same class. From this example it is clear that algorithm has created three prototypes for the patterns belonging to the same class. The pattern groups are circled and given the name as c_1, c_2 and c_3 as shown in the figure 7. The calculated prototypes are indicated by square dots. From observation of Figure 7, it is clear that algorithm did the job we expected to get natural pattern grouping within a single class.

The number of prototypes created can be increased by increasing the value of α. The less the value of α, the algorithm calculates less number of prototypes.

5. EXPERIMENTAL RESULTS

We have simulated the proposed method in Python. For this experimentation purpose, we have used the database of Indian road and traffic sign images as discussed in Section 2 (Image Database). We decolorized the original image into 8 equal parts. So the dimensionality of feature vector for a single image is (1,8) and for the complete database of 864 i.e. (36*24) images is (864,8).

After getting the feature matrix R of size (864,8), the values in it are found arbitrary integers to bring all the values within the interval [0,1], we have used

$$R = \frac{R}{\max(R)}$$

where, max(R) is the maximum value in the matrix R. These bring all the values in the interval [0, 1]. The feature matrix contains patterns of 36 classes i.e. 24 patterns of each class.

We have used first 10 pattern features from R for calculation of prototypes i.e. 144 pattern features per class and remaining 180 are used for testing recognition. The classification is found to be 76 percent. Now the remaining pattern features which were not considered for calculation of class prototype are used for testing recognition. It has given 70 percent recognition rate. This is shown in the table 1.

Table 1: When number of features selected are 10

	Percentage
Classification	76.38
Recognition	70.55

Time required for creation of prototype since the loading of the image is tabulated in table 2.

Table 2: Time from loading images to prototype calculation

Alpha	Gamma	Time for Prototype Creation (sec)
0.85	0.14	54.15
0.70	0.14	51.10
0.70	0.70	52.18

When α=0.14 and $\gamma = 0.85$, it has created a total of 36 prototypes and while α=0.85 and $\gamma = 4$, it has created a total of 123 prototypes. The number of prototypes created for different values of alpha and gamma is tabulated in table 3.

Table 3: No. of prototypes created for different values of alpha and gamma

Alpha	Gamma	Number of Prototypes created
0.14	0.85	36
0.85	4	123
0.85	1	60

6. CONCLUSION

The task of Traffic and Road sign recognition, especially in the domain of machine learning is still in the developing stage and a little far from complete. This is because we have not used the entire dataset for the classification and recognition. We have used only the mandatory signs. It still has to be tested with the Informatory and the Cautionary signs.

But, whatever dataset we used, we found that the proposed method of multi prototype is found to be better than the traditional single method of prototype per class. It also adds flexibility that makes it possible to increase classification and recognition rate by increasing total number of prototype created per class, which could not be possible for single class prototype which is one of the challenges of that method. By increasing the value of α we can increase the total number of prototypes created per class.

7. REFERENCES

[1]. Milind E. Rane, Dhabe P. S. and J. B. Patil, "Multi prototype fuzzy pattern matching for handwritten character recognition".

[2]. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 2nd Ed., Pearson education, 2003.

[3]. Hung-Pin Chiu and Din-Chang Tseng, "Invariant handwritten Chinese character recognition using fuzzy min-max neural networks", Pattern recognition letters, Vol. 18, pp. 481-491, 1997.

[4]. S.J. Parantonis and P.J.G. Lisboa, "Translation, rotation and scale invariant pattern recognition by high-order neural network and moment classifiers", IEEE Trans. Neural networks, vol.3, No.2, pp.241-251, 1992.

[5]. P.M Patil. P.S. Dhabe, U.V. Kulkarni and T. R. Sontakke, "Recognition of handwritten characters using modified fuzzy hyper line segment neural network", in proc. IEEE int. conference on fuzzy systems FUZZ_IEEE 03, May 2003.

[6]. P. K. Simpson, "Fuzzy min-max neural networks – Part 1: Classification," IEEE Trans. on Neural Networks, Vol. 3, No. 5, pp. 776-786, 1992.