RECOGNITION OF TRAFFIC SIGNS BY ARTIFICIAL NEURAL NETWORK

Dan Ghica, Si Wei Lu and Xiaobu Yuan
Department of Computer Science Memorial University of Newfoundland
St. John's, Newfoundland, Canada A1C 5S7 swlu@cs.mun.ca

ABSTRACT

An artificial neural network system for traffic sign recognition is proposed in the paper. The input image is first processed for extraction of color and geometric information. A morphological filter is applied to increase the saliency by eliminating smaller objects and by linking together objects broken in disjoint parts due to noise. The coordinates of the resulting objects are determined, and the objects are isolated from the original image according to these coordinates. After this, the objects are normalized and sent to the neural network which performs the recognition. The neural network consists of classification subnetwork, winner-takes-all subnetwork (hopfield network), and validation subnetwork. By introducing the new concept of a validation sub-network, the network enhance the capabilty to correctly classify the different traffic signs and avoid to misclassify the non-traffic signs into a traffic sign. The system is tested by simulation as a whole and in part on a large amount of data acquired by a video camera attached to a vehicle frame by frame. The performance is encouraging. It produced excellent results except for the images under very poor illumination such that the color threshold (preprocessing) fails to extract the color information.

I. INTRODUCTION

The system processes images acquired by a video camera attached to a vehicle frame by frame. The task of the system is to identify the presence of traffic signs. A neural process can be used to recognize the traffic signs. The neural network determines a mapping between an input feature space and a finite output set of classes. There are several existing networks such as BP net [1] and ART net [2] which can be used for classification. But, BP net is not suitable for our purpose. Since, BP net divides the input feature space exhaustively among classes, while our input data contain inputs which do not belong to any category. There is a number of inputs which are not traffic signs but either noise or "false targets" (cars, walls etc). BP net always attends to classify an object into a traffic sign class which is closest to the object. Therefore, it is impossible for the network to decide whether an object is a sign or not.

The ART network works on a different principle than the BP. ART has a clustering principle. For each class, the network generates a prototype and then assigns an input to that class if it is "close enough" (in a Hamming distance sense) to that pattern. If no prototype is close enough to the input, a new class and a new prototype are generated. But, it is virtually impossible to come up with a representative set of input pattern (images of objects) which are not traffic signs. The ART will generate unlimited number of classes for the objects which are not traffic signs. These objects cannot be grouped as a stand-alone class (not_traffic_sign class).

The proposed neural network structure is built in a hierarchical manner. At the first level, the classification subnetwork (BP net) receives the input image. The output of the BP subnetwork is forward to MAXNET subnetwork [3] located at the second level. The heteroassociative memory [4] at the third level is used to recall the prototype which is mapped by the input image. The validation subnetwork measures the distance between the prototype and the input image by a generalize metric.

The system was trained to recognized stop, yield, no entry and various warning signs, and it was tested on over 100 images gathered with a video camera on the road. The recognition system had a 100% success rate in correctly recognizing the images, provided that they were detected correctly by the previous stages of the system (feature extraction).

II. Object Detection by Color Threshold

Human drivers use two kind of visual information about the signs: color information and geometric information [5]. Only a limited set of standard colors are used for traffic signs: red (stop, prohibition), green (directional guidance), blue (school and service signs), yellow (warning), white (regulation), orange (construction) and brown (tourist information). The shapes are: rectangle, triangle, diamond, octagon, pentagon. The color information is quite easy to process. The amount of information conveyed by color is not as large as the amount conveyed by shape. A number of other objects having the same colors as the traffic signs may be part of the image. However, for its simplicity we use color to determinate and isolate objects in the picture which can be potential traffic signs.

Thresholding is one of the earliest techniques developed for segmenting digital images [6]. The purpose of a thresholding operation is to classify all the pixels of an image into object pixels and background pixels. The image is transformed from a greylevel (or color) image into a binary image. The criteria we use in our color thresholding is that a pixel is an object pixel if its color is "close enough" to a reference color and a background pixel otherwise. To asses the closeness between a pixel and a reference color we shall compute the distance in the color space between the pixel color and the reference color. The color space is the RGB color space. The basis vectors in this space are Red, Green, Blue, therefore any color c is a linear combination of these three colors:

$c = c1 * Red + c2 * Green + c3 * Blue, \quad 0 \leq c1, c2, c3 \leq 1$

Given a reference color r=(r1, r2, r3), the distance between an arbitrary color c and r is:

$d = \sqrt{(r1-c1)^2 + (r2-c2)^2 + (r3-c3)^2}$

The thresholding function F therefore is given by:

$$F(c) = \begin{cases} (0,0,0) & \text{if } \|r-c\| \leq t \\ c & \text{if } \|r-c\| > t \end{cases}$$

where the t is a suitable threshold. An image with a no_entry sign is used to illustrate the system (see Figure 1). The up-left part of the Figure 1 is the original image. The result of the color thresholding operations is a binary image which is shown at the down-left part of the Figure 1. The no-entry sign, the red car, and other red objects are detected as well as noisy spots throughout the picture.

III. Noise Reduction by Morphological Filter

Because of the nature of the color thresholding, patterns are loaded with a considerable amount of noise. Hence, separating the objects from the background and recognizing them becomes difficult and the results tend to be less accurate. To eliminate noise and small gaps, a morphological filtering [7] has been applied based on the following set of rules:

1. If a pixel belongs to an object, it should be activated.

2. If a pixel does not belong to an object, it should be deactivated.

3. If a pixel is surrounded by "enough" activated pixels, then it belongs to an object.

4. If a pixel is surrounded by "enough" deactivated pixels, then it does not belong to an object.

5. If all the pixels obey rules 1-4, then no more filtering is necessary.

6. The closer two pixels are, the greater their mutual influence is.

This filter is implemented by a modified Hopfield Network [8]. For pixel (i,j) in images, there is a neuron (i,j) in the Hopfield Network. To understand the following implementation of these rules, we should always bare in mind the pixel-neuron correspondence.

Let $T_{ij,kl}$ be the weight between neurons corresponding to pixels (i,j) and (k,l). Consider the rule 6, $T_{ij,kl} = 1/d^2$, where d is the Euclidean distance between the pixels, $d = (i-k)^2 + (j-l)^2$. Here those weights from the neurons far away to neuron at (i,j) are very small. For simplicity, the neurons not in the neighborhood n(i,j) (for example a 7×7 window) are not connected to the neuron at (i,j). The input to the neuron (i,j) is:

$$net_{ij} = \sum_{(k,l)\in n(i,j)} T_{ij,kl} \quad y_{kl}(t)$$

where $y_{kl}(t)$ is the output from neuron (k,l).

All the excitation functions are :

$$y_{ij}(t+1) = \begin{cases} +1 & net_{ij} > t_1 \\ y_{ij}(t) & t_2 < net_{ij} < t_1 \\ -1 & net_{ij} < t_2 \end{cases}$$

with t_1, t_2 the thresholds of the neurons. The set of thresholds for the neuron activation functions is pre-assigned such that the neuron can determine if it should be activated or deactivated.

The filtering will stop when the filtered image no longer changes.

The network has been tested on various noisy images. The results are very good. Most of the noise are eliminated. We should also note that the number of iterations was very small (usually less than ten). Since the network is a parallel computational device, the whole process is extremely fast.

IV. Object Recognition Using an Neural Network

After the filtering is performed, the objects are clipped from the background and normalized to a specified size, and send to recognition.

A neural process, as any process, can be defined as a mapping θ between an input feature space I and an output space O.

$$\theta : I \rightarrow O \tag{1}$$

A *validation subnetwork* is defined as a mapping ϑ between the output space O and the input feature space I, given in Equation 2. Its purpose is to re-map the output to the input in an attempt to "reconstruct" the input after it has been classified. The validation subnetwork generates what we call a *prototype* of the class to which the input has been assigned to:

$$\vartheta : O \rightarrow I \tag{2}$$

Once the input has been recreated, a measure of the closeness between the actual input and the prototype (validated replica) of the input needs to be computed. The measure is to be chosen according to the particular features of the actual problem, and it can vary from a simple Hamming distance to a complex weighted distance, as we will see in the examples that follow. An *evaluation subnetwork* is constructed to measure the similarity Ξ between two objects in the input feature space I, and it can be a distance measurement, as shown in Equation 3.

$$\begin{gathered} \Xi(i, i') = m, \\ m \in R^+ \text{ and } i, i' \in I \end{gathered} \tag{3}$$

Given an input object i, the output o produced by the network will therefore be:

$$o = \theta(i)$$

and the *validated replica* (the prototype) of the input i' produced by the validation subnetwork is:

$$i' = \vartheta(o) = \vartheta(\theta(i)) = (\vartheta \circ \theta)(i) \tag{4}$$

A measure m needs to be computed by comparing the original input i with its validated replica i', as in Equation 5.

$$m = \Xi(i, i') = \Xi(i, \vartheta(o)) = \Xi(i, \vartheta(\theta(i))) \tag{5}$$

Finally, a decision function Ψ maps the measure of the distance m to a decision space D, associating the appropriate decisions to the various distances between the input data and the validated replica.

$$\Psi : R^+ \rightarrow D$$

In a typical case, Ψ will be a thresholding function and D a binary set of decisions.

An input i will be called *validable* if the network has the possibility to produce a valid class c as output:

$$\Xi(i, (\vartheta \circ \theta)(i)) < \rho \tag{6}$$

Figure 2 represents a general diagram of how a validation subnetwork is used to validate.

Considering that the feature space has m dimensions, we want to train the network with n input patterns which are classified into n different classes. Given a matrix storing the stimulus S and a matrix storing the responses R, the set of weights W of the classification subnetwork needs to be determined such as

$$R = W \times S$$

Assuming that S has full column rank, it has been shown [?] that

$$W = R \times S^\dagger \tag{7}$$

which is equivalent to

$$W = R \times (S^T \times S)^{-1} \times S^T \tag{8}$$

Without lack of generalization, from a classification point of view, since the collection of classes is finite, we can choose a response matrix of the form

$$R = (r_1 \, r_2 \, \ldots \, r_n) \tag{9}$$

where

$$r_j = (r_{j,k}); r_{jk \; j \neq k} = 0; r_{jj} = 1; 0 \leq j \leq n$$

which is

$$R = I_{nn} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix} \tag{10}$$

For any input i the output p will therefore be

$$p = W \cdot i; \qquad p = (p_1 \, p_2 \, \ldots \, p_n)^T \tag{11}$$

To decide to which class the input i is mapped, the function θ is introduced:

$$o = \theta(p) = (o_1 \; o_2 \; \dots \; o_n)^T$$

such as

$$o_j = \begin{cases} 1 & \text{if } (\forall k, 0 \leq k \leq n)(p_j \geq p_k) \\ 0 & \text{otherwise} \end{cases}$$

The validation subnetwork will measure the distance between the prototype (Vector $\bar{p}$) and the input image (Vector $\bar{y}$) by the generalize metric

$$(\bar{p} - \bar{y})' U(\bar{y} - \bar{y})$$

where U is a weight matrix.

In the roadsign recognition system, the noise usually happens along the edges of the traffic signs. This means that the weight of the mismatches which occur along the edges should be smaller than of those who occur in the inside or on the outside of the object. We can train the network to obtain the weights.

V. Conclusions

This paper has presented the artificial neural network with a validation subnetwork improving the performance of neural architectures.

The backpropagation network (BP) and the ART1 network both failed to perform correctly when faced with inputs which represented invalid patterns. The ART1 showed a better behavior, but the addition of new classes after almost every invalid pattern presented was a problem very difficult to solve. After the adoption of the validation sub-network, the network was trained and the weights determined, we can build the actual network, with the set of weights fixed. The speed of the system is fast enough to allow real time processing of images captured by a camera mounted on a vehicle. The most computationally intensive parts of the system (filtering, image recognition) are handled by parallel architectures which are able to process information at a rate even higher than the rate the camera can provide it.

The system is tested by simulation, as a whole and in part, and we conclude that the results are encouraging. In Figure 1 , the down-right four images show the recognized sign and the prototyps of the sign. a prototype for the traffic sign which is recalled by the associative memory, and the difference measurement between the prototype and input images. A vigilance parameter ρ is used to measure the difference. The traffic sign is recognized because the difference between the images is smaller than parameter ρ.

The classification subnetwork assigned the car image to the closest class, which is the no-entry sign class. Hence the associative memory also recalled the prototype for no-entry sign. But the difference measurement between the two images is much larger than the parameter ρ. Hence, the car image is classified as no_traffic_sign class.

In terms of reliability, the system was tested on a large amount of data gathered with a video camera and, in conditions of good illuminations, it produced excellent results. However, in poorly illuminated images (too dark) the method of color thresholding proved inadequate and the road signs could not be identified.

References

[1] Gorman and Sejnowski "Analysis of hidden units in a layered network", *Neural Networks*, vol 1, no 1, pp 76-90, 1988

[2] Carpenter and Grossberg "Invariant Pattern Recognition and Recall by an Attentive Self-Organizing ART architecture in a Nonstationary World", *Proceedings of the IEEE First International Conference on Neural Networks*, pp II-737-II-735, 1987

[3] Yoh-Han Pao *Adaptive Pattern Recognition and Neural Networks.* Addison-Wesley 1989.

[4] T. Kohonen *Self-Organization and Associative Memory.* pp. 128-133, Springer-Verlag, Berlin (1984).

[5] Guibert, Keryer, Attia "Optical Roadsign Recognition to Improve Active Safety Features", *Proceedings of SPIE - The International Society for Optical Engineering*, pp 230-234, 1993

[6] Narashiman "Some Further Experiments in Parallel Processing of Pictures", *IEEE Transactions on Electronic Computers*, vol EC-12, pp 748-750, 1963

[7] Gonzales and Woods *Morphology, Digital Image Processing*, pp 518-548, 1993

[8] Sreeram, Aiyer, Niranjan and Fallside "A Theoretical Investigation into the Performance of the Hopfield Model", *IEEE Transactions on Neural Networks*, vol 1, no 2, pp 204-215, 1990

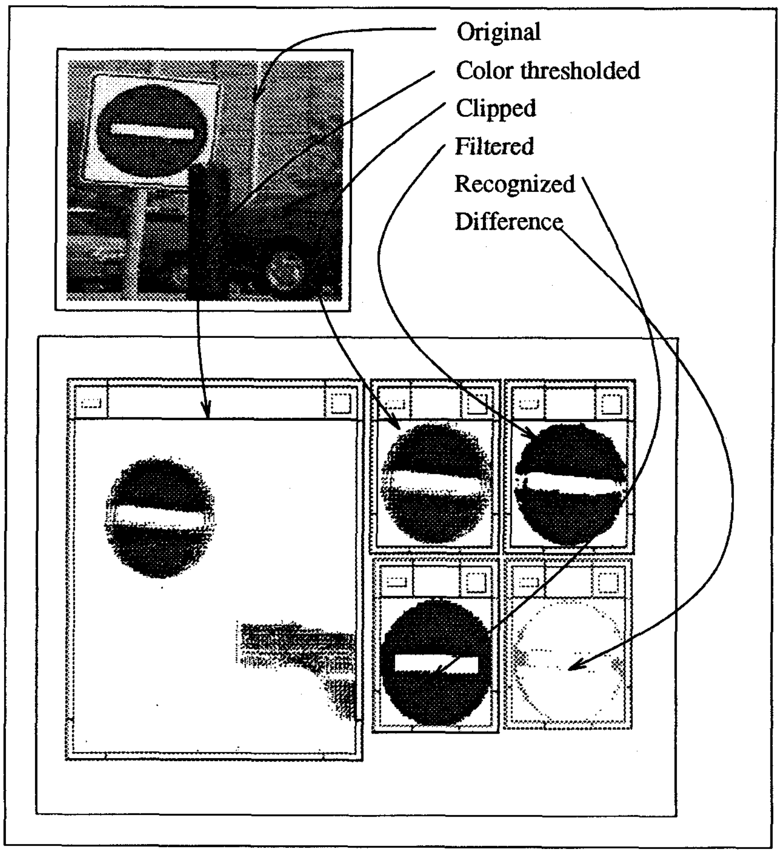

Figure 1: The evolution of the input image at various stages in the road sign recognition system.

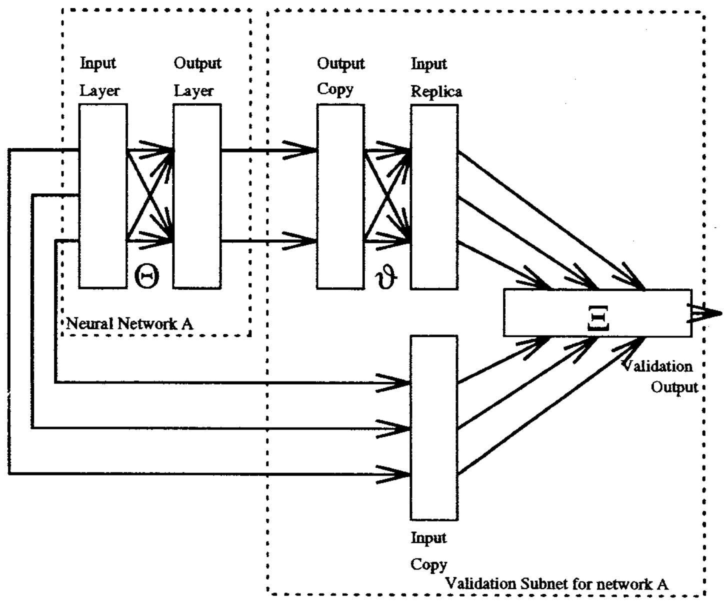

Figure 2: The general architecture of a validation subnetwork.