The design results obtained by the MMFA algorithm are summarised with those by the SSA algorithm by Haines [6] in Table 1. It should be noted that they have the same quality of solutions. In this experiment, $T_s = 5.0$, $\lambda = 0.98$, $T_f = 5 \times 10^4$, and $N_f = 100$ were used. Fig. 1 shows the learning curve of the objective function in $n$ and the changes of the average states for all the nodes for the case when $n = 6$.

Table 2: Execution times of MMFA and SSA for various $n$ ($k = 5$, CPU time(s))

<table>
<thead>
<tr>
<th>$n$</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSA</td>
<td>2.04</td>
<td>2.16</td>
<td>2.31</td>
<td>2.41</td>
<td>2.52</td>
<td>2.59</td>
</tr>
<tr>
<td>MMFA</td>
<td>15.10</td>
<td>15.72</td>
<td>16.03</td>
<td>16.57</td>
<td>17.12</td>
<td>17.86</td>
</tr>
</tbody>
</table>

Table 3: Execution time for various $N_f$ ($k = 5$, $n = 6$, CPU time(s))

<table>
<thead>
<tr>
<th>$N_f$</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution time</td>
<td>2.70</td>
<td>3.05</td>
<td>2.78</td>
<td>2.04</td>
<td>2.24</td>
</tr>
</tbody>
</table>

Neural-vision based approach for real-time road traffic applications

M.Y. Siyal, M. Fathy and F. Dorry

Indexing terms: Image processing, Edge detection, Neural networks

The authors describe a novel neural network and edge based image processing approach for road traffic analysis. An edge detection technique to detect vehicles is used, while a back propagation neural network is used to track and count vehicles. The neural network is trained for various road traffic conditions and is able to analyse complex traffic conditions better than the heuristic approach. The results show that this approach provides better results than the traditional image processing techniques.

Introduction: Traffic data collection and analysis by image processing techniques has been investigated by various researchers for a number of years [1-4]. Many techniques have been proposed to speed up operations and some intelligent approaches have been developed to compensate for the effects of lighting, shadows, occlusions etc. However, these techniques have not yielded good results due to various problems such as inefficiency of background updating and thresholding techniques and false detection of vehicles.

The first image processing algorithm required in traffic application is filtering and segmentation. Background differencing is the simplest technique used to segment traffic images. However, this technique has problems of accurately updating the background frame and automatically selecting a suitable threshold value. Edge detection based segmentation of traffic has the advantage of being less sensitive to the variation of ambient lighting and shadows. However, combined background differencing and edge detection has the advantage of eliminating stationary vehicles, shadows and the road markings and is less sensitive to variations of lighting [2].

The second image processing step is to analyse the results and to measure the traffic parameters. So far, research efforts in this area have been concentrated on investigating traditional approaches, which have not provided satisfactory results [1, 3, 5]. In this Letter a novel neural network approach is introduced which can be trained for different kind of roads and can measure various traffic parameters. We have conducted extensive tests and experiments and have compared results with the heuristic image processing methods. The results show that the neural network approach provides better results.

Vehicle detection: The algorithm used for vehicle detection is based on applying lowpass filtering and edge detection operations on windows located across the road. Following the application of edge detection operation, the number of pixels having values greater than the threshold is used to recognise a vehicle [6]. The threshold value is automatically selected by differencing the edges of the current picture from the background picture and analysing the histogram for a number of frames. After applying the edge detection operation, a vector with values 0 or 1 is obtained for each window. A zero value determines no vehicles and 1 shows a vehicle on that frame.

Measuring traffic parameters by neural network: To analyse the status vector for each road window, a neural network with inputs, seven outputs and a hidden layer was used for pattern recognition and traffic analysis. In this method, the pattern vector of 10 consecutive frames is applied to a back propagation network and the number of passed vehicles is computed by neural networks. Each input of the neural network is the percentage of white points for that frame and each output represents the number of passed vehicles for 10 frames.

The training of a neural network is carried out for each road separately and a sample which consists of 10 consecutive frames should be little different from other samples. Otherwise, a large number of training samples will be required for training. In our approach, 200 samples are used for training and the network has produced a designed response after 100K training. Following the training of the network, the computed weights (W) are extracted and are used for a vehicle detection program to compute the outputs as follows:

Table 1: Design results by MMFA and SSA ($k = 5$)

<table>
<thead>
<tr>
<th>$n$</th>
<th>Algorithm</th>
<th>Design result</th>
<th>$E(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>SSA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>SSA</td>
<td>MMFA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>7</td>
<td>MMA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>SSA</td>
<td>MMFA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>8</td>
<td>SSA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>SSA</td>
<td>MMFA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>9</td>
<td>SSA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>SSA</td>
<td>MMFA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>10</td>
<td>SSA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>SSA</td>
<td>MMFA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>11</td>
<td>SSA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
<tr>
<td>SSA</td>
<td>MMFA</td>
<td>$\pm 1.0, 0.765, 0.285$</td>
<td>241.54</td>
</tr>
</tbody>
</table>

References

\[ S_j = \sum_i W_{ij} \cdot X_i \quad i = 0 \ldots \text{input} \quad j = 1 \ldots \text{hidden} \quad (1) \]

Where, \text{input} and \text{hidden} are the numbers of inputs and hidden layers in our system. These values are constant and equal to 10. The output of each hidden layer is computed using a signed function as follows:

\[ h_j = \left(1 + e^{-x_j}\right)^{-1} \quad j = 1 \ldots \text{hidden} \quad (2) \]

Inserting the value of \( S_j \) from eqn. 1, the main outputs of the neural network is as follows:

\[ S_j = \sum_i W_{ij} \cdot h_i \quad i = 0 \ldots \text{hidden} \quad j = 1 \ldots \text{output} \quad (3) \]

When \( h_j \) is computed from eqn. 2 and the output is a constant value equal to 6. The output of the network is:

\[ O_j = \left(1 + e^{-x_j}\right)^{-1} \quad j = 1 \ldots \text{output} \quad (4) \]

Results: The algorithm has been implemented in a user friendly package in an MS Windows environment. The results of counting vehicles by a heuristic method is shown in Fig. 1, while Fig. 2 represents the results of vehicle counting by a neural network approach. As can be seen, this approach provides better results. To further improve the accuracy of the vehicle counting operation, the windows are again divided into two sections and a neural network with 20 inputs is used for analysis. For this training network, 150 training samples were selected and the network produced a designated response after 100K training. With this approach the accuracy achieved was &gt; 97%.

**Fig. 1** Results of counting vehicles (heuristic approach)

**Fig. 2** Results of counting vehicles by neural network approach and dividing window into two halves

Conclusions: In this Letter, a novel neural network and image processing approach was described to analyse and measure road traffic parameters. To further increase the accuracy in the case of vehicle not moving on their lanes, half size windows were used. Using neural networks is more accurate than using the heuristic approach as considering all the states of the window pattern is difficult in a real-world traffic scene because the vehicles move with different speeds and directions. In this case, an intelligent approach such as the neural network, which is trained for many traffic scenes and situations, can compute the parameters more accurately.

References


Practical stability criteria for cellular neural networks

P.P. Civalleri and M. Gilli

**Indexing terms:** Cellular neural nets, Stability criteria

New sufficient conditions for the existence of stable equilibrium points in cellular neural networks (CNNs), described by space-invariant templates, are presented. Extensive simulations have shown that such conditions also guarantee the complete stability of the network.

Cellular neural networks (CNNs) are arrays of dynamical cells [1, 2] that are suitable for the formulation and solution of many complex computational problems. Most applications (e.g. image processing) require the network to be completely stable, i.e. that every trajectory converges to an equilibrium point. The study of the stability of such networks (that are large-scale dynamical systems) is a cumbersome task and for this reason only a few results are available [3–8]. Two considerations are needed: (i) the simulations show that such conditions also guarantee the complete stability of the class of stable CNNs is much larger than the subclass, for which a rigorous proof of stability is already available; (ii) CNN design requires simple stability criteria that should be directly expressed in terms of the template elements.

In this Letter we present some sufficient conditions for the existence of stable equilibrium points in both 1D and 2D template CNNs. Such conditions are different from those reported in [7, 8] and in several cases give rise to less strong constraints. Moreover, they can be checked by simply looking at the template elements and therefore are suitable for design.

We have verified, by extensive simulations, that our sufficient conditions seem to also guarantee the stability of the network.

We consider an autonomous CNN described by a space-invariant template and assume that the cells are arranged on a rectangular matrix, composed by \( N \) rows and \( M \) columns. We also assume that the input terms are null. (The results, however can be easily extended to constant inputs.)

The dynamics of the network is governed by the following set of normalised equations:

\[
\dot{x}_{i,j}(t) = -x_{i,j}(t) + \sum_{k=k}^{k_{x}} \sum_{h=h}^{h_{y}} A_{k,h} y_{k,h}(t) \quad (1)
\]

\[
k_{x}(i, r) = \max(1 - i, -r) \quad k_{y}(i, r) = \min(N - i, r)
\]

\[
l_{x}(i, r) = \max(1 - j, -r) \quad l_{y}(i, r) = \min(M - j, r)
\]

© IEE 1997

Electronics Letters 22nd May 1997 Vol. 33 No. 11