

## Motion clustering on video sequences using a competitive learning network

Salih GÖRGÜNOĞLU<sup>1,\*</sup>, Şafak ALTAY<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Karabük University, Karabük, Turkey

<sup>2</sup>Department of Electronics and Computer Education, Karabük University, Karabük, Turkey

Received: 09.03.2012 • Accepted: 22.08.2012 • Published Online: 17.01.2014 • Printed: 14.02.2014

**Abstract:** It is necessary to track human movements in crowded places and environments such as stations, subways, metros, and schoolyards, where security is of great importance. As a result, undesired injuries, accidents, and unusual movements can be determined and various precautionary measures can be taken against them. In this study, real-time or existing video sequences are used within the system. These video sequences are obtained from objects such as humans or vehicles, moving actively in various environments. At first, some preprocesses are made respectively, such as converting gray scale, finding the edges of the objects existing in the images, and thresholding the images. Next, motion vectors are generated by utilizing a full search algorithm. Afterwards, k-means, nearest neighbor, image subdivision, and a competitive learning network are used as clustering methods to determine dense active regions on the video sequence using these motion vectors, and then their performances are compared. It is seen that the competitive learning network significantly determines the classification of dense active regions, including motion. Moreover, the algorithms are analyzed in terms of their time performances.

**Key words:** Motion estimation, competitive learning network, video processing, clustering

### 1. Introduction

Motion analysis is the evaluation of movement occurring in an environment. This evaluation process includes the determination of the direction, speed, size, and density of movement. Moving objects on video sequences obtained by a video camera can either be a human being or other living thing, or could be a moving vehicle.

Motion analysis is used in various fields, such as agriculture, security, medicine, traffic, and sports. Observing the growth of a plant, tracking the movement of a vehicle within traffic, or observing the movements of a runner's hands or legs depend on the processes of movement analysis. Motion analysis can also be used in observing the movements of crowds.

Many academic studies have been implemented on motion analysis. Some examined only the motion of 1 person. In [1], the motion of 1 person was analyzed in an indoor environment and the human action was segmented utilizing 2 different techniques based on color intensity and motion. Some studies searched groups of people and evaluated the motion of people in groups. The authors in [2] tracked groups of people and the tracking process was realized in both indoor and outdoor environments.

In addition to that, there are studies that deal with the estimation of traffic density by watching the motion of vehicles in traffic. They find regions where the traffic density is higher. In a study using adjusted images, the vehicle's speed and traffic intensity were estimated [3]. Some studies deal with the determination

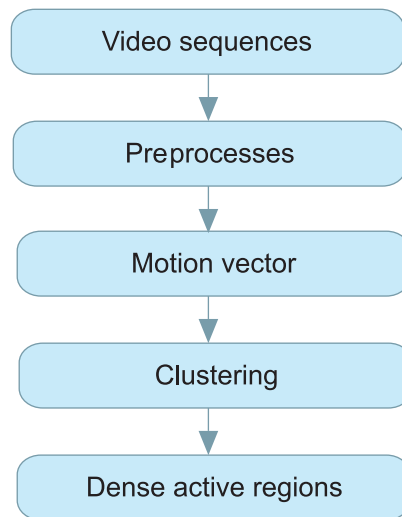
\*Correspondence: [sgorgunoglu@karabuk.edu.tr](mailto:sgorgunoglu@karabuk.edu.tr)

of the number of vehicles in real-time traffic flux. Traffic intensity is estimated due to the presence of a vehicle in the sequence [4,5].

Many academic studies were made on artificial neural networks. In [6], a special node-splitting criterion was presented based on the competitive learning network, which is a self-creating model, and this model gives better clustering or quantization results. Furthermore, the competitive learning network was used in various studies. These studies were about network intrusion detection [7], adaptive feedback controller [8], and feature maps integrations [9]. In another study, multilayer perceptrons were used to determine different team tactics in volleyball. The team tactic patterns were used to train multilayer perceptrons [10]. Another study dealt with a neural network of motion perception and speed discrimination [11]. There were also studies that used different types of cameras, as catadioptric [12] and multicamera systems [13,14]. A technique based on m-medoids was used in [15] to classify motion and anomaly detection. In this study, object motion trajectories were modeled as a time series and used in the algorithm. Another study was about the estimation of dominant motion direction on video sequences, and the k-means algorithm was used to classify motion vectors [16].

## 2. Motion analysis

The motion analysis process mainly consists of 3 basic stages: preprocesses, the determination of motion vectors using the full search algorithm, and clustering utilizing the competitive learning network. At first, video images are subjected to specific preprocessing operations in the motion analysis process. Motion vectors occurring in the moving regions in video images are determined after these processes. Finally, the motion vectors are grouped and the motion analysis process is implemented in this study, which is shown in Figure 1.



**Figure 1.** Motion analysis process.

### 2.1. Preprocessing operations

Video sequences consist of frames, and consecutive frames are compared for motion estimation. Two frames are selected for comparison by leaving 5 frames apart from each other, since there must be a definite interval between the frames for formation of the movement. Various preprocessing operations are applied to the frames that will be compared after determination. The first preprocessing operation is finding the RGB values of the frames' pixels. R denotes the level of red brightness, G denotes the level of green brightness, and B denotes

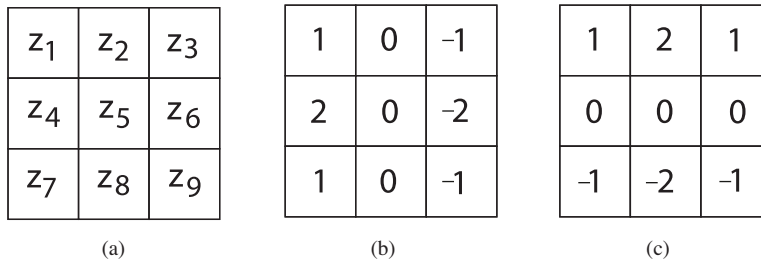
the level of blue brightness. The frames are converted into gray scale using the RGB values. The process of converting to gray scale is done by summation after multiplication of RGB values singly with certain coefficients. The process of converting gray scale is done according to Eq. (1), where  $g(x, y)$  represents the pixel value.

$$g(x, y) = 0.299R + 0.587G + 0.114B \tag{1}$$

Moreover, the frames are separated into 4 regions within this process and the average values of the pixels converted into gray scale in each region are calculated as given in Eq. (2), where  $M$  and  $N$  represent the sizes of each region.

$$\text{Average value} = \frac{1}{MN} \sum_{u=0}^N \sum_{v=0}^M g(x, y) \tag{2}$$

Later, the edges of the objects are found using the Sobel filter. The neighborhoods of the pixels and operators of the Sobel filter are given in Figure 2. Sobel operators are also used to determine the direction of a pixel or the local direction of a block in an image [17]. The pixel's gradient values of the x and y directions are calculated as given in Eqs. (3) and (4). The pixel value to be assigned is calculated as given in Eq. (5).



**Figure 2.** a) Neighborhoods of the pixels, b) horizontal operator, c) vertical operator of the Sobel filter.

$$G_x = Z_1 + 2Z_4 + Z_7 - Z_3 - 2Z_6 - Z_9 \tag{3}$$

$$G_y = Z_1 + 2Z_2 + Z_3 - Z_7 - 2Z_8 - Z_9 \tag{4}$$

$$G = \sqrt{G_x^2 + G_y^2} \tag{5}$$

Frames are converted into black-and-white form by utilizing the average value obtained from 4 separated regions. The thresholding process is performed according to Eq. (6).

$$g(x, y) = \begin{cases} 255, & \text{if } (g(x, y) \geq \text{average value}) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

After these processes, 2 frames are compared with each other in order to find the active region. A new image is formed; in this form, the regions are white with a white-colored first frame and a black-colored second frame, and the rest of them are black. Regions of the white pixel in the new image show the active region [18]. The process for finding the active region is carried out according to Eq. (7). Finally, noises are cleaned on the image by a filter.

$$g(x, y) = \begin{cases} 255, & \text{if } (g_1(x, y) = 255 \text{ and } g_2(x, y) = 0) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

## 2.2. Full search algorithm

The full search algorithm is one of the block-matching algorithms. Block-matching algorithms are generally used in motion estimation and motion analysis. Frames are divided into macroblocks in block-matching algorithms. Motion vectors are formed via comparing macroblocks situated in the first frame with macroblocks in the search region located in the second frame. Macroblocks are compared for each probability in the search region in the full search algorithm, which is the most accurate one among the block-matching algorithms [19,20].

Motion vectors are found using the full search algorithm after the end of the preprocessing operations. Initially, the frames are separated into  $W \times W$  pixel macroblocks for the full search algorithm. The macroblocks of the first frame's active region are matched on a  $3 \times 3$  macroblock region of the second frame by scanning. The sum of absolute differences (SAD) is computed at all points in a  $3 \times 3$  macroblock region [21]. The  $W \times W$  pixel regions that have the lowest SAD are matched. The SAD is computed as follows:

$$SAD = \sum_{u=0}^w \sum_{v=0}^w |MB_1(x+u, y+v) - MB_2(x'+u, y'+v)|. \quad (8)$$

Here,  $MB_1$  denotes the macroblock that will be compared in the first frame and  $MB_2$  denotes the macroblock in the search region in the second frame of the equation.  $(x, y)$  represent the coordinates of the macroblock pixels. In many academic studies, the solutions of the problems or the algorithms are given as pseudocodes [22,23]. The pseudocode of the full search algorithm that we follow in this study is shown in Figures 3 and 4.

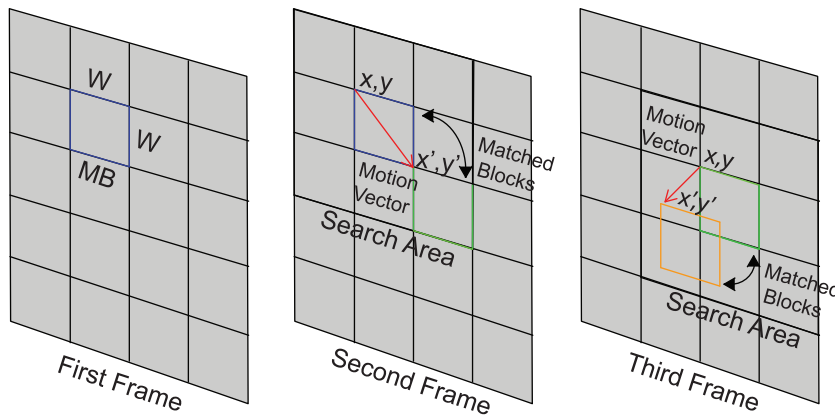


Figure 3. Sequence of the full search algorithm.

## 2.3. Clustering methods

### 2.3.1. k-Means algorithm

Video sequences are divided into  $K$  regions using the motion vector. This method provides satisfactory results in the video sequences, in which one person (or more) is moving. In this method, it is necessary to determine the cluster number ( $K$ ) initially. Many academic studies have been conducted on the k-means algorithm for different purposes [24–26].

```

Step 1 : Start
Step 2 : Divide frames that will be compared into macroblocks having  $W \times W$  dimension
Step 3 : Define search region as  $3 \times 3$  macroblock
Step 4 : For each  $MB_1(x,y)$  in the first frame
{
  Step 4.1 : For each  $i \in \{x - W, x - (W - 1), \dots, x + W\}$ 
  {
    For each  $j \in \{y - W, y - (W - 1), \dots, y + W\}$ 
    {
      Calculate SAD value between  $MB_2(i,j)$  in the second frame and  $MB_1(x,y)$ 
      If  $i = x - W$  and  $j = y - W$  The lowest value = SAD
      If  $SAD < \text{The lowest value}$  { The lowest value = SAD,  $x' = i, y' = j$  }
    }
  }
  Step 4.2 : Form motion vector between  $(x,y)$  and  $(x', y')$  points
}
Step 5 : End

```

**Figure 4.** The pseudocode of the full search algorithm [16].

### 2.3.2. Nearest neighbor algorithm

In this method, the regions including the active regions are divided into 2 clusters. Clustering is based on the Euclidian distance and the nearest neighbor values of 2 clusters are taken into consideration in calculating the distance between the clusters. The nearest neighbor algorithm has been used in various academic studies [27–30].

### 2.3.3. Image subdivision algorithm

In this method, video sequences are partitioned or divided into 25 regions as  $64 \times 48$  pixels at first. Next, the numbers of the motion vectors of the macroblocks in each region are determined. The region that includes the most motion vectors is considered as the dense active region. It is seen that there are many studies on the image subdivision algorithm [31–34].

### 2.3.4. Competitive learning network

Artificial neural networks are developed by taking the human nervous system into consideration. Artificial neural cells, similar to human brain neurons, exist within its structure. Many artificial neurons compose the artificial neural network by conjoining.

Artificial neural networks are quite popular because of their learning and generalization ability, fast real-time operation, and ease of implementation features. The efficiency of artificial neural networks originates from the weight connections in a network. It is stated that each artificial neuron or processing element has weighted inputs, summation function, activation function, and an output. The architecture, learning rule, and activation function determine the behavior of artificial neural networks [35,36].

Each artificial neural network has a different number of inputs and outputs, which varies according to the type of the problem. There are supervised and unsupervised artificial neural networks according to the learning

method of the network. Artificial neural networks are widely used in diagnosis of diseases [37–39]; optical design and computation [40–44]; recognition of biometrical features such as voice, fingerprint, signature, and retina; and the solving of classification problems [45–60].

The competitive learning network is one of the unsupervised neural networks, since target values are not given. It is generally used in classification problems [61]. All of the outputs are connected to each other and all of the inputs as well. Only an output becomes active and its weights change [62]. The model of a 2-input and 3-output competitive learning network is shown in Figure 5.

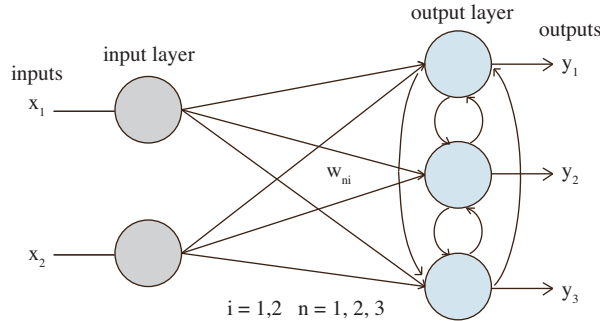


Figure 5. Competitive learning network.

In this study, the competitive learning network is used to cluster motion vectors. The number of samples is equal to the number of the motion vectors occurring in the network. The coordinate values of the motion vectors are given to the network as input values; therefore, the number of inputs is 2. A maximum of 3 different moving regions (maximum number of outputs can be 3) can be determined with a 3-output network in this structure. The clusters are formed as the number of output weights that are updated; that is, if the weights of 2 outputs are updated, then the number of the cluster is 2.

The Euclidean distance between each sample and output is calculated, as given in Eq. (9) below.

$$d_n = \sqrt{\sum_{i=1}^m (x_i - w_{ni})^2}, \quad m = 2 \tag{9}$$

In this equation,  $d_n$  is the sum of the Euclidean distance between the  $i$ th sample of  $x$  and the weight of the  $n$ th output, and  $m$  is the number of inputs. The output that has the smallest distance wins and if the  $k$ th output is equal to the smallest distance  $d_k$ , then the  $k$ th output  $y_k$  gets a value of 1 or 0, as shown in Eq. (10).

$$y_k = \begin{cases} 1, & \text{if } (d_k = \min(d_1, d_2, \dots, d_n)), 1 \leq k \leq n \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Finally, the weights of the outputs are updated according to Eq. (11). Here,  $w_{ni}$  is the current weight between the  $n$ th output and the  $i$ th input, and  $w_{ni+1}$  denotes the new weight value between the  $n$ th output and the  $i$ th input.

$$w_{ni+1} = w_{ni} + \eta y_n (x_i - w_{ni}) \tag{11}$$

Here,  $\eta$  denotes the learning rate. The centers of the clusters  $(x_n, y_n)$  are determined as given in Eq. (12).

$$(x_n, y_n) = (w_{n1}, w_{n2}), \quad n = 1, 2, 3 \tag{12}$$

Motion clustering using the competitive learning network algorithm that is utilized in this study is shown in Figure 6.

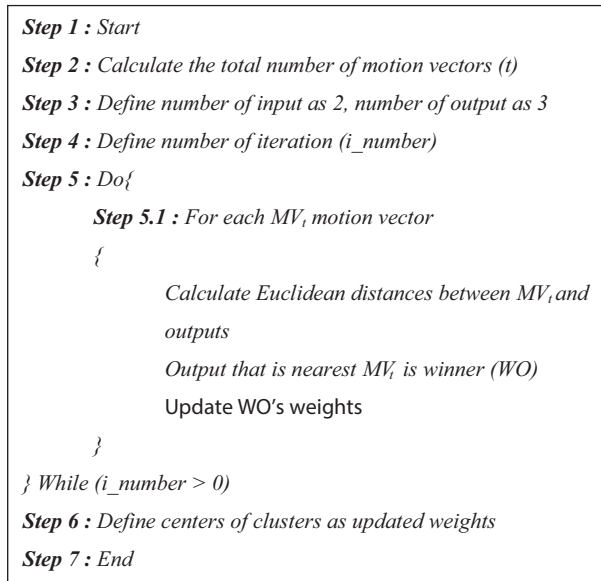


Figure 6. Motion clustering using the competitive learning network algorithm.

### 3. Experimental results

Software is developed in order to perform motion analysis using the C# visual programming language. The algorithms mentioned above can be applied to any video sequence selected by the user with this software. The

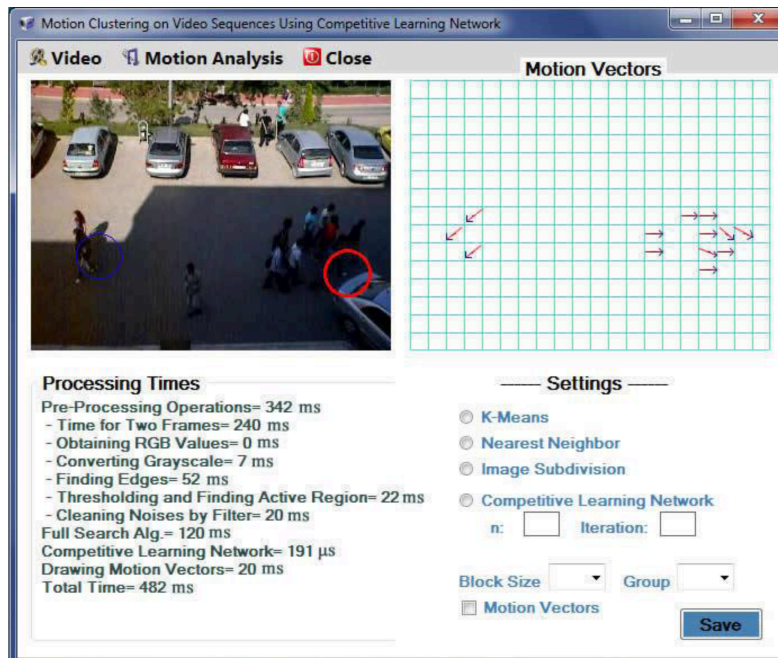


Figure 7. Software interface.

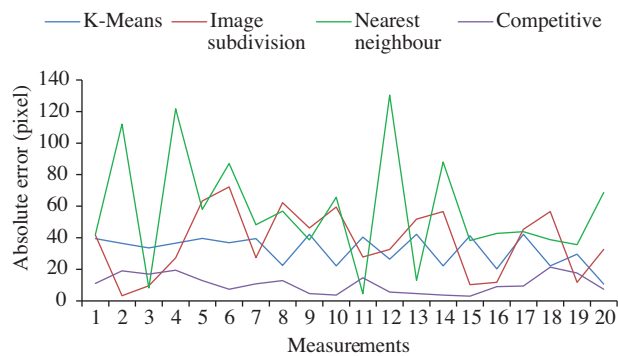
motion vectors and centers of the clusters can be seen in the screen by the users and the user can determine the clustering method, number of iterations, learning rate of the competitive learning network, block size, and number of groups. Furthermore, the user can observe the execution time of the algorithms utilized.

A screenshot of the software developed is given in Figure 7. In the video scene that is analyzed, the motion vectors formed and the processing times of the algorithms can be seen on the screen.

Screenshots, and the detection and classification of the direction of the motion in these screenshots obtained in the experimental studies, are displayed in Figures 8a and 8b, respectively. The competitive learning network is used as a clustering method. The video sequences utilized are at pixel dimensions of  $320 \times 240$ . The blue and red circles displayed by the program indicate the centers of the clusters, which denote the moving regions. The red circle displays the center of the most intensive motion. The absolute errors obtained as a result of 20 different measurements of the clustering methods are given in Figure 9.



**Figure 8.** Screenshots displaying the center of the motion direction obtained and their clustering: a) motion in video sequence is clustered as bidirectional; b) motion in video sequence is clustered as 3-directional.



**Figure 9.** Absolute errors of the algorithms.

The average durations and mean absolute errors of the algorithms used in the study are given in the Table. The average durations are calculated by taking the average value of 50 different measures, taken from 5 different video sequences, into consideration. The sizes of the macroblocks utilized within the study are set to  $10 \times 10$ ,  $16 \times 16$ , and  $20 \times 20$  pixels. The number of moving macroblocks is limited to 10. The iteration count is taken as 30 and the learning rate is taken as 0.5 for the competitive learning network. Since preprocessing algorithms



are carried out in terms of pixels, a change in the size of the macroblock does not affect the processing period. Moreover, as the macroblock size in the full search algorithm increases, the search region and process time increase, as well. In order to determine the success of the algorithms, the absolute error is obtained according to 20 different measurements, which are calculated with the help of Eq. (13) below.

$$\text{error} = |x - x_{approx}| + |y - y_{approx}| \quad (13)$$

In this equation,  $x$  and  $y$  denote the true values, and  $x_{approx}$  and  $y_{approx}$  denote the approximate values.

It is seen that the competitive learning network has the smallest error value out of the 4 algorithms discussed above. On the other hand, the processing time of the competitive learning network lasts longer compared to the other algorithms.

The average durations and absolute errors of the algorithms are obtained from a notebook computer consisting of the following hardware: Intel(R) Pentium(R) Dual T3400 2.16 GHz microprocessor, 3 GB RAM, and Windows 7 operating system. In addition, the camera frame rate used within the system is 25.

**Table.** Average duration and absolute errors of the algorithms.

Process		Time			Error
		10 × 10 pixel block	16 × 16 pixel block	20 × 20 pixel block	
Preprocessing operations	Time for 2 frames	240 ms	240 ms	240 ms	-
	Obtaining RGB values	1.75 ms	1.75 ms	1.75 ms	-
	Converting gray scale	7.36 ms	7.36 ms	7.36 ms	-
	Finding edges	55.66 ms	55.66 ms	55.66 ms	-
	Thresholding and finding active region	19.3 ms	19.3 ms	19.3 ms	-
	Cleaning noises	21.66 ms	21.66 ms	21.66 ms	-
Full search algorithm		16.88 ms	101.83 ms	238.74 ms	-
k-Means algorithm		16.07 $\mu$ s	9.86 $\mu$ s	7.61 $\mu$ s	32.74
Nearest neighbor algorithm		58.62 $\mu$ s	44.45 $\mu$ s	37.47 $\mu$ s	57.09
Image subdivision algorithm		33.19 $\mu$ s	31.80 $\mu$ s	29.95 $\mu$ s	37.47
Competitive learning network		319.23 $\mu$ s	170.18 $\mu$ s	131.13 $\mu$ s	10.72

#### 4. Conclusions

In this study, software that clusters moving regions according to motion vectors on video sequences based on clustering methods is successfully developed. The k-means, nearest neighbor, image subdivision, and competitive learning network are used as clustering methods. It is seen that the competitive learning network provides more accurate results and the number of clusters ( $K$ ) is obtained automatically according to the motion vectors in this algorithm. More motion centers can be determined by increasing the competitive learning network's output number in environments that have scattered motion. On the other hand, the total duration of the algorithms is considerably important when the moving objects' speeds are very high. Therefore, the duration performances of the algorithms are also compared.

## References

- [1] L. Shao, L. Ji, Y. Liu, J. Zhang, “Human action segmentation and recognition via motion and shape analysis”, *Pattern Recognition Letters*, Vol. 33, pp. 438–445, 2012.
- [2] S.J. McKenna, S. Jabri, Z. Duric, H. Wechsler, A. Rosenfeld, “Tracking groups of people”, *Computer Vision and Image Understanding*, Vol. 80, pp. 42–56, 2000.
- [3] C. Maduro, K. Batista, P. Peixoto, J. Batista, “Estimation of vehicle velocity and traffic intensity using rectified images”, 15th IEEE International Conference on Image Processing, pp. 777–780, 2008.
- [4] H.S. Mohana, M. Ashwathakumar, G. Shivakumar, “Vehicle detection and counting by using real time traffic flux through differential technique and performance evaluation”, *International Conference on Advanced Computer Control*, pp. 791–795, 2009.
- [5] C. Ozkurt, F. Camci, “Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks”, *Mathematical and Computational Applications*, Vol. 14, pp. 187–196, 2009.
- [6] H. Xiong, M.N.S. Swamy, M.O. Ahmad, I. King, “Branching competitive learning network: a novel self-creating model”, *IEEE Transactions on Neural Network*, Vol. 15, pp. 417–429, 2004.
- [7] J.Z. Lei, A. Ghorbani, “Network intrusion detection using an improved competitive learning neural network”, *Communication Networks and Services Research*, pp. 190–197, 2004.
- [8] H. Gu, “Adaptive synchronization for competitive neural networks with different time scales and stochastic perturbation”, *Neurocomputing*, Vol. 73, pp. 350–356, 2009.
- [9] J. Steffen, M. Pardowitz, J.J. Steil, H. Ritter, “Integrating feature maps and competitive layer architectures for motion segmentation”, *Neurocomputing*, Vol. 74, pp. 1372–1381, 2011.
- [10] J.M. Jäger, W.I. Schöllhorn, “Identifying individuality and variability in team tactics by means of statistical shape analysis and multilayer perceptrons”, *Human Movement Science*, Vol. 31, pp. 303–317, 2012.
- [11] J. Chey, S. Grossberg, E. Mingolla, “Neural dynamics of motion processing and speed discrimination”, *Vision Research*, Vol. 38, pp. 2769–2786, 1998.
- [12] J.C. Bazin, C. Démonceaux, P. Vasseur, I.S. Kweon, “Motion estimation by decoupling rotation and translation in catadioptric vision”, *Computer Vision and Image Understanding*, Vol. 114, pp. 254–273, 2010.
- [13] J.H. Kim, M.J. Chung, B.T. Choi, “Recursive estimation of motion and a scene model with a two camera system of divergent view”, *Pattern Recognition*, Vol. 43, pp. 2265–2280, 2010.
- [14] J. Kim, M. Hwangbo, T. Kanade, “Spherical approximation for multiple cameras in motion estimation: its applicability and advantages”, *Computer Vision and Image Understanding*, Vol. 114, pp. 1068–1083, 2010.
- [15] S. Khalid, “Activity classification and anomaly detection using m-medoids based modelling of motion patterns”, *Pattern Recognition*, Vol. 43, pp. 3636–3647, 2010.
- [16] S. Görgünoğlu, Ş. Altay, B. Şen, “Estimation of dominant motion direction on video sequences”, 2nd International Symposium on Computing in Science & Engineering, pp. 9–15, 2011.
- [17] A. Çavuşoğlu, S. Görgünoğlu, “A fast fingerprint image enhancement algorithm using a parabolic mask”, *Computers & Electrical Engineering*, Vol. 34, pp. 250–256, 2008.
- [18] Z. Telatar, “Video dizilerinde ayrit sezmeğe dayalı hareket kestirimi”, *Gazi Üniversitesi Mühendislik Mimarlık Dergisi*, Vol. 24, pp. 245–255, 2009 (article in Turkish).
- [19] R.T.N. Rajaram, V. Vasudevan, “Optimisation of the one-dimensional full search algorithm and implementation using an EPLD”, 13th International Conference on VLSI Design, pp. 336–341, 2000.
- [20] Z. Jing, J. Liangbao, C. Xuehong, “Implementation of parallel full search algorithm for motion estimation on multi-core processors”, 2nd International Conference on Next Generation Information Technology, pp. 31–35, 2011.

- [21] A.V. Paramkusam, V.S.K. Reddy, “An optimal fast full search motion estimation algorithm in video coding”, *International Conference on Signal Processing, Communication, Computing and Networking Technologies*, pp. 598–603, 2011.
- [22] E. Arcaklıoğlu, A. Çavuşoğlu, A. Erişen, “Thermodynamic analysis of refrigerant mixtures for possible replacements for CFCs by an algorithm compiling property data”, *Applied Thermal Engineering*, Vol. 26, pp. 430–439, 2006.
- [23] E. Arcaklıoğlu, A. Çavuşoğlu, A. Erişen, “An algorithmic approach towards finding better refrigerant substitutes of CFCs in terms of the second law of thermodynamics”, *Energy Conversion and Management*, Vol. 46, pp. 1595–1611, 2005.
- [24] M. Laszlo, S. Mukherjee, “A genetic algorithm that exchanges neighboring centers for k-means clustering”, *Pattern Recognition Letters*, Vol. 28, pp. 2359–2366, 2007.
- [25] K.L. Chung, J.S. Lin, “Faster and more robust point symmetry-based k-means algorithm”, *Pattern Recognition*, Vol. 40, pp. 410–422, 2007.
- [26] M.C. Chiang, C.W. Tsai, C.S. Yang, “A time-efficient pattern reduction algorithm for k-means clustering”, *Information Sciences*, Vol. 181, pp. 716–731, 2011.
- [27] Y. Zeng, Y. Yang, L. Zhao, “Pseudo nearest neighbor rule for pattern classification”, *Expert Systems with Applications*, Vol. 36, pp. 3587–3595, 2009.
- [28] S.H. Cha, S.N. Srihari, “A fast nearest neighbor search algorithm by filtration”, *Pattern Recognition*, Vol. 35, pp. 515–525, 2002.
- [29] G. Beliakov, G. Li, “Improving the speed and stability of the k-nearest neighbors method”, *Pattern Recognition Letters*, Vol. 33, pp. 1296–1301, 2012.
- [30] N. García-Pedrajas, D. Ortiz-Boyer, “Boosting k-nearest neighbor classifier by means of input space projection”, *Expert Systems with Applications*, Vol. 36, pp. 10570–10582, 2009.
- [31] K. Sasazaki, S. Saga, J. Maeda, Y. Suzuki, “Vector quantization of images with variable block size”, *Applied Soft Computing*, Vol. 8, pp. 634–645, 2008.
- [32] K.L. Chung, P.C. Chen, “An efficient algorithm for computing moments on a block representation of a grey-scale image”, *Pattern Recognition*, Vol. 38, pp. 2578–2586, 2005.
- [33] E. Bernardis, S.X. Yu, “Pop out many small structures from a very large microscopic image”, *Medical Image Analysis*, Vol. 15, pp. 690–707, 2011.
- [34] L. Nanni, A. Lumini, “Descriptors for image-based fingerprint matchers”, *Expert Systems with Applications*, Vol. 36, pp. 12414–12422, 2009.
- [35] S. Yiğit, R. Eryiğit, F.V. Çelebi, “Optical gain model proposed with the use of artificial neural networks optimised by artificial bee colony algorithm”, *Optoelectronics and Advanced Materials – Rapid Communications*, Vol. 5, pp. 1026–1029, 2011.
- [36] S. Tankiz, F.V. Celebi, R. Yildirim, “Computer aided design model for a quantum-cascade laser”, *IET Circuits, Devices & Systems*, Vol. 5, pp. 143–147, 2011.
- [37] N. Süt, Y. Çelik, “Prediction of mortality in stroke patients using multilayer perceptron neural networks”, *Turkish Journal of Medical Sciences*, Vol. 42, pp. 886–893, 2012.
- [38] B. Taşdelen, S. Helvacı, H. Kaleağası, A. Özge, “Artificial neural network analysis for prediction of headache prognosis in elderly patients”, *Turkish Journal of Medical Sciences*, Vol. 39, pp. 5–12, 2009.
- [39] O. Er, N. Yumusak, F. Temurtas, “Chest diseases diagnosis using artificial neural networks”, *Expert Systems with Applications*, Vol. 37, pp. 7648–7655, 2010.
- [40] S. Yigit, B. Tugrul, F.V. Celebi, “A complete CAD model for type-I quantum cascade lasers with the use of artificial bee colony algorithm”, *Journal of Artificial Intelligence*, Vol. 5, pp. 76–84, 2012.
- [41] F.V. Celebi, T. Altindag, “An accurate optical gain model using adaptive neuro-fuzzy inference system”, *Optoelectronics and Advanced Materials Rapid Communications*, Vol. 3, pp. 975–977, 2009.

- [42] F.V. Celebi, “A different approach to gain computation in laser diodes with respect to different number of quantum-wells”, *Optik*, Vol. 116, pp. 375–378, 2005.
- [43] F.V. Celebi, K. Danisman, “Neural estimator to determine alpha parameter in terms of quantum-well number”, *Optics & Laser Technology*, Vol. 37, pp. 281–285, 2005.
- [44] F.V. Celebi, K. Danisman, “A different approach for the computation of refractive index change in quantum-well diode lasers for different injection levels”, *Proceedings of the SPIE – The International Society for Optical Engineering*, Vol. 61, pp. 384–388, 2004.
- [45] S. Partal, İ. Şenol, A.F. Bakan, K.N. Bekiroğlu, “Online speed control of a brushless AC servomotor based on artificial neural networks”, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 19, pp. 373–383, 2011.
- [46] O. Özgönel, T. Yalçın, “A complete motor protection algorithm based on PCA and ANN: a real time study”, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 19, pp. 317–334, 2011.
- [47] M. Aktaş, H.İ. Okumuş, “Stator resistance estimation using ANN in DTC IM drives”, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 18, pp. 197–210, 2010.
- [48] İ. Dalkıran, K. Danışman, “Artificial neural network based chaotic generator for cryptology”, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 18, pp. 225–240, 2010.
- [49] G. Vachkov, Y. Kiyota, K. Komatsu, S. Fujii, “Real-time classification algorithm for recognition of machine operating modes by use of self-organizing maps”, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 12, pp. 27–42, 2004.
- [50] K. Radhakrishnan, H. Larijani, “Evaluating perceived voice quality on packet networks using different random neural network architectures”, *Performance Evaluation*, Vol. 68, pp. 347–360, 2011.
- [51] J.D. Wu, S.H. Ye, “Driver identification based on voice signal using continuous wavelet transform and artificial neural network techniques”, *Expert Systems with Applications*, Vol. 36, pp. 1061–1069, 2009.
- [52] F.V. Çelebi, “A proposed CAD model based on amplified spontaneous emission spectroscopy”, *Journal of Optoelectronics and Advanced Materials*, Vol. 7, pp. 1573–1579, 2005.
- [53] R. Kumar, B.R. Deva Vikram, “Fingerprint matching using multi-dimensional ANN”, *Engineering Applications of Artificial Intelligence*, Vol. 23, pp. 222–228, 2010.
- [54] J.D. Wu, C.T. Liu, “Finger-vein pattern identification using SVM and neural network technique”, *Expert Systems with Applications*, Vol. 38, pp. 14284–14289, 2011.
- [55] M.M.M. Fahmy, “Online handwritten signature verification system based on DWT features extraction and neural network classification”, *Ain Shams Engineering Journal*, Vol. 1, pp. 59–70, 2010.
- [56] A. Fallah, M. Jamaati, A. Soleamani, “A new online signature verification system based on combining Mellin transform, MFCC and neural network”, *Digital Signal Processing*, Vol. 21, pp. 404–416, 2011.
- [57] L. Chen, W. Xue, N. Tokuda, “Classification of 2-dimensional array patterns: assembling many small neural networks is better than using a large one”, *Neural Networks*, Vol. 23, pp. 770–781, 2010.
- [58] Anbazhagan, N. Kumarappan, “A neural network approach to day-ahead deregulated electricity market prices classification”, *Electric Power Systems Research*, Vol. 86, pp. 140–150, 2012.
- [59] S. Kulluk, L. Ozbakir, A. Baykasoglu, “Training neural networks with harmony search algorithms for classification problems”, *Engineering Applications of Artificial Intelligence*, Vol. 25, pp. 11–19, 2012.
- [60] M. Ghiassi, C. Burnley, “Measuring effectiveness of a dynamic artificial neural network algorithm for classification problems”, *Expert Systems with Applications*, Vol. 37, pp. 3118–3128, 2010.
- [61] Y. Yang, K. Chen, “An ensemble of competitive learning networks with different representations for temporal data clustering”, *International Joint Conference on Neural Networks*, pp. 3120–3127, 2006.
- [62] J. Liu, M. Gao, “Unsupervised classification algorithm for intrusion detection based on competitive learning network”, *International Symposium on Information Science and Engineering*, Vol. 1, pp. 519–523, 2008.