Human Action Recognition using Image Processing and Artificial Neural Networks

Chaitra B H
PG Student
Department of CSE, RVCE

Anupama H S
Assistant Professor
Department of CSE, RVCE
Bangalore

Cauvery N K
Professor and Head
Department of ISE, RVCE
Bangalore

ABSTRACT
Human action recognition is an important technique and has drawn the attention of many researchers due to its varying applications such as security systems, medical systems, entertainment. Action recognition is an interesting and a challenging topic of computer vision research due to its prospective use in proactive computing. The developed algorithm for the human action recognition system, which uses the two-dimensional discrete cosine transform(2D-DCT) for image compression and the self organizing map(SOM) neural network for recognition purpose, is simulated in MATLAB. By using 2D-DCT we extract image vectors and these vectors become the input to neural network classifier, which uses self organizing map algorithm to recognize elementary actions from the images (trained). In this paper we have developed and illustrated a recognition system for human actions using a novel self organizing map based retrieval system. SOM has good feature extracting property due to its topological ordering. Using an image database of 30 action images, containing six subjects and each subject having five images with different body postures reflects that the action recognition rate using one of the neural network algorithm SOM is 98.16%.

General Terms
Human Action Recognition (HAR), Artificial Neural Network (ANN).

Keywords
2Dimentional-Discrete Cosine Transform (2D-DCT), Epochs, Minimum Absolute Deviation, Self Organizing Map (SOM).

1. INTRODUCTION
Human action recognition is one of the most successful applications of pattern recognition and image analysis which has recently received significant attention, especially during the past several years. At least two reasons account for this trend: the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. Human Action Recognition has gained momentum and practical vitality in the wake of increased and growing security concerns. A reliable system capable of recognizing various human actions has many important applications. The applications include intelligent surveillance systems [1], health-care systems, and a variety of systems that involve interactions between persons and electronic devices such as human-computer interfaces [5]. The two main aspects of Human communication include verbal (auditory) and non-verbal (visual). Body movements, facial expressions and physiological reactions are some of the basic units of non-verbal communication. Action recognition has proved to be extremely difficult to imitate artificially, although commonalities do exist between each person, they vary considerably in terms of height, weight, shape of the human body and gender. The purpose of action recognition is a computerized analysis of ongoing events from visual data [9].
This paper presents a novel approach to recognize human actions using two dimensional discrete cosine transforms (2D-DCT) and self organize map (SOM) Neural Network as classifier. A block diagram of proposed technique for human action recognition using SOM Neural Network is as shown in the figure 1.1. In the first stage all the 30 action images are compressed for feature processing using two dimensional-discrete cosine transform (2D-DCT). When the 2D-DCT is applied with a mask, high-coefficients in an image are discarded. Then the 2D-IDCT is applied to regenerate the compressed image, which is blurred due to loss of quality and also smaller in size. In next stage uses a self-organizing map (SOM) with an unsupervised learning technique [8] which is trained to classify vectors into groups to recognize if the subject in the input image is "present" or "not present" in the image database. After training all the 30 action images, we now take a single input image, compress it using 2D-DCT and regenerate it using IDCT. All the 30 trained images and untrained input image are simulated. The untrained input image is compared with all trained images, if the action image is classified as present, the best match image is found in the training database using minimum absolute deviation and that image is displayed otherwise if the image is not found then "image is not found in the database" is displayed.

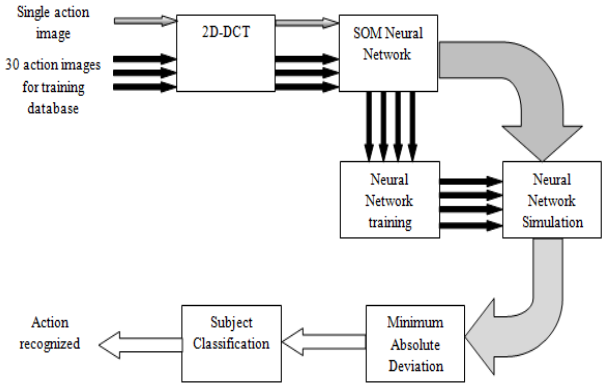

Fig 1.1: Block Diagram of HAR using SOM

2. DISCRETE COSINE TRANSFORM

For feature extraction 2-D Discrete Cosine Transform (2D-DCT) is used. A discrete cosine transform (DCT) expresses a sequence of finite data points in terms of a sum of cosine functions oscillating at different frequencies. The use of cosine rather than sine functions is critical in these applications: for compression [3], it turns out that cosine functions are much more efficient, whereas for differential equations the cosines express particular choice of boundary conditions. The DCT, and in particular the DCT-II, is often used in signal and image processing, especially for lossy data compression, because it has a strong "energy compaction". Most of the signal information tends to be concentrated in a few low-frequency components of the DCT.DCT is real valued and provides a better approximation of a signal with fewer coefficients.

The 2D-DCT of an M ×N matrix A is defined as follows:

$$B_{pq} = \alpha p\, \alpha q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mm} \cos\left(\frac{\pi(2m+1)p}{2m}\right) \cos\left(\frac{\pi(2n+1)q}{2n}\right)$$

$$0 \le p \le M-1$$

$$0 \le q \le N-1$$

The proposed technique uses the DCT transform matrix in the MATLAB Image Processing Toolbox. This technique is efficient for small square inputs such as image blocks of 8 × 8 pixels. The M × M transform matrix T is given by:

$$T_{pq} = \begin{cases} \sqrt{\frac{1}{M}}, & p=0, 0 \le q \le M-1 \\ \sqrt{\frac{2}{M}} \cos\frac{\pi(2q+1)}{2M}, & 1 \le p \le M-1, 0 \le q \le M-1 \end{cases}$$

The DCT transformation matrix is implemented in MATLAB and the feature is extracted. The 2D DCT reduces the size of the data significantly by transforming the image from the spatial representation into the frequency domain. The lower frequencies are characterized by relatively larger magnitude while the higher frequencies have smaller magnitudes. The higher frequency components are ignored as it does not significantly affect the accuracy of work. In short, the 2D DCT coefficient of lower frequency values captures the most relevant information of human actions.

2.1 D-DCT Image Compression

The proposed design technique calculates the 2D-DCT [10] of the image blocks of size 8 × 8 pixels using '8' out of the 64 DCT coefficients for masking. The other 56 remaining coefficients are discarded (set to zero). The image is then reconstructed by computing the 2D-IDCT of each block using the DCT transform matrix computation method. Finally, the output is a set of arrays. Each array is of size 8 × 8 pixels and represents a single image [4]. Empirically, the upper left corner of each 2D-DCT matrix contains the most important values, because they correspond to low-frequency components within the processed image block.

3. SELF ORGANIZING MAPS

Self organizing map also known as Kohonen map [7] is a well known artificial neural network. It is unsupervised learning process in which the learning is based upon the input data which is known as unlabeled data and is independent of the desired output data. Self organizing map can also be termed as a topology preserving map [6]. There is a competition among the neurons to be activated and only one neuron that wins the competition is fired and is called the "winner". Kohonen rule is used to learn the winner neuron and neurons within a certain neighbourhood of the winning neuron. This rule allows the weight of neuron to learn an input vector so this makes it perfect for recognition. Hence in this system SOM is used as classifier. The SOM network used in this system contains N nodes, ordered in two dimensional lattice architecture where each node has 2 or 3 neighbouring nodes. SOM has three phases of life cycle: learning phase, training phase and testing phase.

3.1 Unsupervised Learning

During the learning the neurons having weight closest with the input vector declare as winner [7]. Based on winning neuron weights of all neighbourhood neurons are adjusted by an amount inversely proportional to the Euclidean distance. The learning algorithm is summarized as follows:

1. Initialization: Choose random values for the initial weight vectors $w_{j(0)}$, the weight vector being different for j = 1, 2,...,l where *l* is the total number of neurons:

$$W_i = [W_1, W_2, \ldots\ldots\ldots\ldots\ldots W_l]^{\mathrm{T}} \in R^n \qquad (3.1)$$

2. Sampling: Draw a sample x from the input space with a certain probability:

$$x_i = [x_1, x_2, \ldots\ldots\ldots\ldots\ldots x_l]^{\mathrm{T}} \in R^n \qquad (3.2)$$

3. Similarity Matching: Find the best matching (winning) neuron $i(x)$ at time t, $0 < t \le n$ by using the minimum distance Euclidean criterion:

$$i(x) = arg\ min_j \|x(n) - W_j\|, j = 1,2,\ldots..l \qquad (3.3)$$

4. Updating: Adjust the synaptic weight vector of all neurons by using the update formula:

$$W_j(n+1) = W_j(n) + \eta(n)\, h_{i,j(x)}(n)(x(n) - W_j(n)) \qquad (3.4)$$

Where $\eta(n)$ is learning rate parameter, and $h_{ij}\ x(n)$ is the neighbourhood function centered around the winning neuron. Both $\eta(n)$ and are $h_{ij}\ x(n)$ varied dynamically during learning for best results.

5. Continue with step 2 until no noticeable changes in the feature map are observed.

3.2 Training

During the training phase feature vector are presented to the SOM one at a time. For each node net determines the output unit that is best matches for current input sample. The weight vector for the winner is adjusted with respect to the learning algorithm described in learning phase. At the end of this phase each node has two values: total number of winning times for the subject present in the database, and total number of winning times for the subject not present in the database.

3.3 Testing

During the testing phase firstly every input vector is compared with the all the SOM nodes and then the best matching unit is found based on minimum Euclidean distance as given in equation (3.3). After that final output result is shown.

4. EXPERIMENTAL WORK

4.1 Image Database

An image database is created for the purpose of recognition of actions and is divided into two subsets, for separate training and testing purposes. During SOM training, 30 images are used, containing six subjects (walking, running, jumping. falling, bending, sitting and standing) and each subject having five images with different actions. A preview image of the database for training and testing is as shown in figure 4.1(a) and figure 4.1(b).

Fig 4.1 (a) Image Database for Training

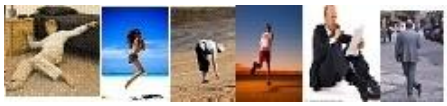

Fig 4.1 (b) Untrained Images for testing

4.2 Validation Technique

All the action images are resized to 8 x 8 pixels and saved; the next step was to compress them by applying the 2D blocked DCT. When the 2D DCT is applied with a mask, high-coefficients in the image are discarded. Then the 2D IDCT is applied to regenerate the compressed image, which is blurred due to loss of quality and also smaller in size. For the image data to be input into the neural network, it should follow the form of only one column, despite the number of rows. Currently, all the resized and DCT compressed action images are in the form of 8 x 8 pixels. Hence the image data needed to be reshaped from an 8 x 8 matrix to a 64 x 1 array for it to be used both for the input and training database of the neural network. SOM's were found to be efficient for image data management and proved to be an accurate closest matching technique of untrained input images with trained database of images. For the design of SOM, a set of 30 action image data, 6 different subjects with 5 different actions for the training database is loaded into MATLAB [10]. A SOM is then created and the Parameters for the SOM network are selected to be a minimum and maximum point for each row on vector; training database. There are 64 minimum and 64 maximum points selected altogether. After the SOM neural network is created, it is trained for 3000 epochs. Figure 4.2 is the SOM layer weight for the 30 action images in the training database and figure 4.3 is the SOM weight vectors is as shown.

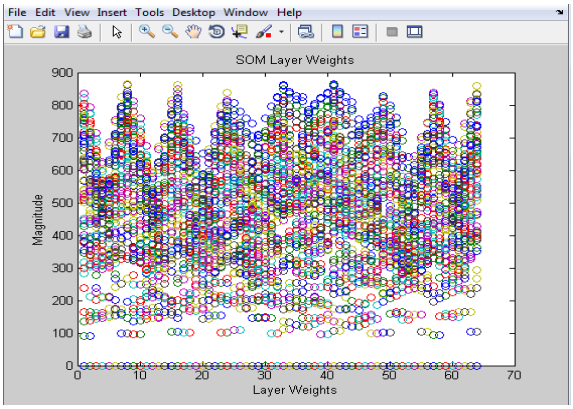

Fig. 4.2 SOM layer weights for the 30 action images in the training database.

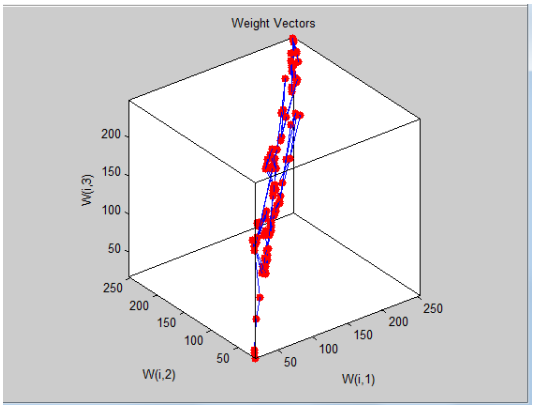

Fig.4.3 SOM weight vectors

After the SOM neural network is trained and simulated for the 30 action images in the training database, the SOM neural network is then simulated for the single input action image of different postures. After the SOM neural network is simulated for the input action image, the image in the training database which is the closest match by the SOM neural network for the input action image is found by finding the minimum absolute deviation. After the closest matched training database images are found, they are then classified. Classification of the subject is the answer of the action recognition system.

4.3 Determining Optimal Number of Epochs for Training

Epochs are neural network training parameters. They are defined as one complete cycle through the neural network for all cases, which present the entire training set to the neural network. Each time a record goes through the net, it is one trial, one sweep of all records is termed as an Epoch. Less number of epochs used for training leads to less training time for the training data set. The goal is to find the optimal number of epochs for training which will produce accurate neural network results and at the same require the least amount of time for program execution. The SOM neural network was tested to determine the optimal number of epochs to be used for neural network training. This test was performed by varying the number of epochs and network training time to find the best possible recognition rate.

Table 4.1 Comparison of number of epochs vs. network training time and Recognition rate

No. of Epochs	Network Training Time (in Seconds)	Recognition Rate (%)
500	113.16	87.12
1000	128.32	90.03
1500	142.64	92.45
2000	155.28	93.89
2500	168.24	95.06
3000	182.39	98.16

Table 4.1 shows the tabulated results generated by varying the number of epochs. The most efficient number of epochs for training and with respect to fastest training time is 3000, with achieves best possible recognition rate as 98.16%.

5. CONCLUSION

This paper presents a novel human action recognition technique that uses features derived from DCT coefficients, along with a self organizing map (SOM)-based classifier. The 2D-DCT and SOM neural network are the heart for the design and implementation of efficient action recognition system. The system was evaluated in MATLAB using an image database of 30 action images, containing six subjects and each subject having 5 images with different actions. After training for approximately 3000 epochs the system achieved a recognition rate of 98.16% for fastest network training time. The system having less computational requirement this make system well suited for low cost, real-time hardware implementation.

6. ACKNOWLEDGMENTS

It is a great pleasure to express sincere and humble gratitude to my guide Ms. Anupama H S for her valuable guidance and encouragement given during the course of work.

7. REFERENCES

[1] L. Weilun, H. Jungong, and P. With, "Flexible human behavior analysis framework for video surveillance applications," *Int. J. Digital Multimedia Broadcast*., vol. 2010, pp. 920121-1–920121-9, Jan. 2010.

[2] G.Strang, "The discrete cosine transform", *SIAM Review*, vol. 41, No.1, pp.135 - 147, 1999.

[3] W. B. Pennebaker and J. L. Mitchell, "JPEG – Still Image Data Compression Standard," *Newyork: International Thomsan Publishing*, 1993.

[4] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd Edition, *Prentice Hall*, 2008, ISBN-13: 9780131687288.

[5] P. Barr, J. Noble, and R. Biddle, "Video game values: Human-computer interaction and games," *Interact. Comput.*, vol. 19, no. 2, pp. 180–195, Mar. 2007.

[6] T. Kohonen, "Self-Organizing Maps," *Springer, Berlin, Heidelberg*, 1995.

[7] Teuvo Kohonen "Self-Organizing Map," Proceedings of the IEEE, vol. 78, No.9, September 1990.

[8] Alexandros Iosifidis, Anastasios Tefas, "View-Invariant Action Recognition Based on Artificial Neural Networks," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 3, March 2012.

[9] Weilong Yang, Yang Wang, and Greg Mori "Recognizing human actions from still Images with Latent Pose," *School of Computing Science*, Simon Fraser University, Burnaby, BC, Canada, September 2012.

[10] Keerti Keshav Kanchi, "Facial Expression Recognition using Image Processing and Neural Network", *International Journal of Computer Science & Engineering Technology* (IJCSET), ISSN: 2229-3345, Vol. 4 No. 05 May 2013.