A COMPARISON OF POSTURE RECOGNITION USING SUPERVISED AND UNSUPERVISED LEARNING ALGORITHMS

Maleeha Kiran, Chee Seng Chan, Weng Kin Lai
Centre for Multimodal Signal Processing
Mimos Berhad,Technology Park Malaysia
57000 Kuala Lumpur, Malaysia
Emails: {malehaa.kiran;cs.chan;lai}@mimos.my

Kyaw Kyaw Hitke Ali, Othman Khalifa
Department of Electrical & Computer Engineering
International Islamic University Malaysia
53100 Gombak, Selangor, Malaysia
Email: khalifa@iiu.edu.my

KEYWORDS

Posture Recognition, Machine Learning, Artificial Intelligence, Image Processing

ABSTRACT

Recognition of human posture is one step in the process of analyzing human behaviour. However, it is an ill-defined problem due to the high degree of freedom exhibited by the human body. In this paper, we study both supervised and unsupervised learning algorithms to recognise human posture in image sequences. In particular, we are interested in a specific set of postures which are representative of typical applications found in video analytics. The algorithms chosen for this paper are K-means, artificial neural network, self organizing maps and particle swarm optimization. Experimental results have shown that the supervised learning algorithms outperform the unsupervised learning algorithms in terms of the number of correctly classified postures. Our future work will focus on detecting abnormal behaviour based on these recognised static postures.

INTRODUCTION

At present the majority of automated surveillance systems act passively meaning they are usually employed to analyze events after an incident has occurred. However to deter such incidents from occurring, it is imperative that the security system pre-emptively alert security personnel whenever any suspicious behaviour or event is detected. In order to do that, the system should have the capability to analyze human behaviour. One of the key aspects of analyzing human behaviour is correctly recognizing different types of human posture from static images. This is needed as we are interested in recognizing acts which can later turn into suspicious activities. For example if the system can detect the act of climbing or jumping from a static image then appropriate alert can be raised before it becomes an intrusion detection activity. In addition the detection of such an act from a static image can be done in a smaller time period thereby making the system more efficient and responsive. This is in contrast to the detection of suspicious activity such as intrusion detection which comprises of a combination of acts and would require the system to monitor a human for a longer period of time.

Ideally the recognition of human postures should be performed by using only one static camera and in real time as the majority of applications for video surveillance applications make use of one static camera to observe the scene and any analysis is done in real time (Boulay et al., 2003). This would also imply that the computational cost of the system is a crucial aspect of such systems. However the downside is that there is usually a tradeoff between computational complexity and classification precision.

In this paper, our aim is to study and compare the performance of both supervised and unsupervised learning algorithms to recognize a few challenging human posture. In particular, we are interested in a specific set of postures which are representative of typical applications which are found in the area of video analytics. The algorithms chosen for this paper are Self-Organizing Maps (SOM), K-means, Artificial Neural Network (ANN), and Particle Swarm Optimization (PSO). Experimental results have shown that the supervised learning algorithms outperformed the unsupervised learning algorithms in terms of the number of correctly classified postures.

The rest of the paper is structured as follows. Section II discusses the related work in this area. Section III discusses various supervised learning algorithms. Section IV presents the unsupervised learning algorithms. Section V shows the experimental results in terms of posture recognition. Section VI concludes the paper with discussion and future work.

RELATED WORK

One of the earlier human motion analysis systems applied the method of template matching. Bobick and Davis (Bobick and Davis, 1996) proposed a view-based approach to the representation and recognition of action using temporal templates. They made use of the binary Motion Energy Image (MEI) and Motion History Image (MHI) to interpret human movement in an image sequence. First, motion images in a sequence were extracted by differencing, and these motion images were accumulated in time to form MEI. Then, the MEI was enhanced into MHI which was a scalar-valued image. Taken together, the MEI and MHI were con-

Proceedings 24th European Conference on Modelling and Simulation ©ECMS Andrzej Bargiela, Sayed Azam Ali David Crowley, Eugène J.H. Kerckhoffs (Editors)
ISBN: 978-0-9564944-0-5 / ISBN: 978-0-9564944-1-2 (CD)

sidered as a two-component version of a temporal template, a vector-valued image, in which each component of each pixel was some function of the motion at that pixel position. Finally, by representing the templates by its seven Hu-moments, a Mahalanobis distance was employed to classify the action of the subject by comparing it to the Hu-moments of pre-recorded actions. Bradski and Davis (Bradski and Davis, 2002) pick up the idea of MHI and develop timed MHI (tMHI) for motion segmentation. tMHI allow determination of the normal optical flow. Motion is segmented relative to object boundaries and the motion orientation. Hu-moments are applied to the binary silhouette to recognise the pose. In Bobick and Davis (Bobick and Davis, 1996), an action is represented by several feature images. Principle Component Analysis (PCA) is applied for dimensionality reduction. Finally, each action is represented by a manifold in the PCA space. Motion history images can also be used to detect and interpret actions in compressed video data.

Yi et. al. (Yi et al., 2005) present the idea of a Pixel Change Ratio Map (PCRM) which is conceptually similar to the MHI. However, further processing is based on motion histograms which are computed from the PCRM. Weinland et. al. (Weinland et al., 2006) suggest replacing the motion history image by a 4D motion history volume. For this, they first compute the visual hull from multiple cameras. Then, they consider the variations around the central vertical axes and use cylindrical coordinates to compute alignments and comparisons. One of the main problem of template matching approaches are the recognition rate of objects based on 2D image features is low, because of the nonlinear distortion during perspective projection and the image variations with the viewpoint's movement. These algorithms are generally unable to recover 3D pose of objects. The stability of dealing effectively with occlusion, overlapping and interference of unrelated structures is generally poor.

Spagnolo et. al. (Spagnolo et al., 2003) proposed a fast and reliable approach to estimate body postures in outdoor visual surveillance. The sequences of images coming from a static camera is trained and tested for recognition. The system uses a clustering algorithm and therefore manually labelling of the clusters is required after the training stage. The features extracted are the horizontal and vertical histograms of binary shapes associated with humans. After training, the Manhattan distance is used for building clusters and for recognition. The main strengths of their method are high classification performance and relatively low computational time which allows the system to perform well in real time.

Buccolieri et. al. (Buccolieri et al., 2005) used active contours and Artificial Neural Networks (ANN) for their posture recognition system. With regards to the feature extraction, localization of moving objects in the image and human posture estimation are performed. The classification is performed by the radial basis functions neural network. Their approach has some advantages such as low sensitivity to noise, fast processing speed, and the ability to handle some degree of occlusion. However, the system is limited to recognizing only three postures, namely standing, bending and squatting postures.

The literature work surveyed focuses on developing a solution required for posture or action recognition tailored to the requirements of specific problems being faced by the authors. There is little effort devoted to doing a comparative analysis of techniques to help decide which technique is suited for the purpose of posture recognition. This paper seeks to address this gap in the current literature work.

SUPERVISED LEARNING

Artificial Neural Network

ANN is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example.

The most basic building block in the neural network is the perceptron. It is characterised by several input lines with weights associated to them. The input is collected and summed and an output is given according to the output function *f*. The output is formally defined as follows:

$$o = f(\sum_{i=1}^{n} w_i \vec{x}_i) \tag{1}$$

where o is the output function and w_i the weight associated with input line *i*. Assume the feature values of the samples are real numbers, and their labels are either 0 or 1 denoting two different classes. The delta rule adapts the weights w_i according to the following formula:

$$w_i(t+1) = w_i(t) + \eta(y_i(t) - o(t))\vec{x}_i(t) \tag{2}$$

where $y_i(t)$ is the desired response, i.e. the real class label, and η an adjustable parameter called the learning rate which usually has a value between 0 and 1. The *t* is used to indicate the time step. Note that there is no change in the weights if the perceptron produces a correct answer, y(t)-o(t) = 0, in other cases the weights are adapted. This makes it unsuited for solving the XOR problem, which need a nonlinear solution (Minsky and Papert, 1969).

The invention of the back-propagation rule caused a major breakthrough in neural network research. In principle the rule is very simple: calculate the error made by the network and propagate it back through the network layers. This back-propagated error is used to update the weights. The sample $\vec{x}$ is fed to the network and produces an output on the right side $\vec{o}$. The input pattern $\vec{x}$ is propagated through the network in the following way:

$$o_j^{(1)} = f\left(\sum_{k=1}^{N} w_{jk}^{(1)} \vec{x}_k\right) \quad (3)$$

$$o_i^{(2)} = f\left(\sum_{j=1}^{M} w_{ij}^{(2)} o_j^{(1)}\right) \quad (4)$$

where $o_j^{(1)}$ and $o_i^{(2)}$ denotes the output of a hidden unit and an output unit respectively. The variables *N* and *M* denote the number of input units and the number of hidden units. A weight from a unit to another unit is denoted by $w_{ij}^{(1)}$ where *j* is the source of the connection, *i* is the target and *l* the layer. The final output of the network can be written as:

$$o_i^{(2)} = f\left(\sum_{j=1}^{M} w_{ij}^{(2)} f(\sum_{k=1}^{N} w_{jk}^{(1)} \vec{x}_k\right) \quad (5)$$

where $o_j^{(1)}$ has been replaced by Eq. 3 . The output of the network has to be judged using some error criterion. The criterion determines the size of the error to be back propagated. In general, the Mean Squared Error (MSE) criterion is employed:

$$E = \frac{1}{2}\frac{1}{|L|}\sum_{i=1}^{|L|}\sum_{\vec{x}\in L}[y(\vec{x}) - o(\vec{x})]^2 \quad (6)$$

where $y(\vec{x})$ is the desired network output value for the sample $\vec{x}$ under investigation and —L— is the size (cardinality) of the learning set. The objective during the training is to minimise the error function (Eq. 6) by choosing the appropriate weights.

Self-organizing maps

SOM (T.Kohonen, 1995) is a type of ANN that is trained using unsupervised learning to produce a two dimensional, discretised representation of the input space of the training samples, called a map. The map preserves the topological properties of the input space. This makes SOM useful for visualizing low-dimensional views of high-dimensional data, similar to multidimensional scaling. Same to most of the ANN, SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector. Associated with each node in this neural network is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid.

The SOM describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest weight vector to the vector taken from data space and to assign the map coordinates of this node to the input vector. The goal of learning in the SOM is to cause different parts of the network to respond similarly to certain input patterns. This is partly motivated by how visual, auditory or other sensory information is handled in separate parts of the cerebral cortex in the human brain. The weights of the neurons are initialized either to small random values or sampled evenly from the subspace spanned by the two largest principal component eigenvectors. With the latter alternative, learning is much faster because the initial weights already give good approximation of SOM weights.

The network must be fed a large number of example vectors that represent, as close as possible, the kinds of vectors expected during mapping. The examples are usually administered several times. The training utilizes competitive learning. When a training example is fed to the network, its Euclidean distance to all weight vectors is computed.

The neuron with weight vector most similar to the input is called the best matching unit (BMU). The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance from the BMU. The update formula for a neuron with weight vector $W_v(t)$ is:

$$W_v(t+1) = W_v(t) + \Theta(v,t)\alpha(t)(D(t) - W_v(t)) \quad (7)$$

where $\alpha(t)$ is a monotonically decreasing learning coefficient and *D(t)* is the input vector Duda et al. (2000); Alpaydin (2004). The goal of learning in self-organizing maps is to cause different parts of the network to respond similarly to certain input patterns. The network must be fed a large number of example vectors that represents, as close as possible, the kinds of vectors expected during mapping. The examples are usually administered several times and the training utilizes competitive learning.

UNSUPERVISED LEARNING

K-means Algorithm

As similar to other partitional clustering algorithms, K-means algorithm is generally an iterative algorithm that converge to local optimal (Costa and Cesar, 2001). Employing the general form of iterative clustering, the steps of K-means algorithm are:

1. Using the output from PSO as initial K cluster centroids

2. **Repeat**
   (a) **For** each pattern, z_p in the dataset **do**
   Compute its membership $u(M_k|z_p)$ to each centroid m_k and its weight $w(z_p)$
   end loop
   (b) Recalculate the K cluster centroids, using

$$m_k = \frac{\sum\limits_{\forall z_p} u(m_k|z_p)w(z_p)z_p}{\sum\limits_{\forall z_p} u(m_k|z_p)w(z_p)} \tag{8}$$

until a stopping criterion is satisfied

where $u(m_k|z_p)$ is the membership function which quantifies the membership of pattern z_p to cluster *k*. For K-means algorithm in this paper, the membership and weight function are defined as

$$u(m_k|z_p) = \begin{cases} 1 & \text{if} \quad d^2(z_p, m_k) = \arg\min_k\{d^2(z_p, m_k)\} \\ 0 & \quad \text{otherwise} \end{cases} \tag{9}$$

$$w(z_p) = 1 \tag{10}$$

Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is a computational paradigm based on the phenomenon of collective intelligence exhibited by the social behaviour of bird flocking and fish schooling (Kennedy and Eberhart, 1995). In a real number space, each individual possible solution can be modelled as a particle that moves through the problem hyperspace. The position of each particle is determined by the vector $x_i \in \Re_n$ and its movement by the velocity of the particle $v_i \in \Re_n$ (Kennedy, 1997), as shown in Eq 11.

$$\overrightarrow{x_i}(t) = \overrightarrow{x_i}(t-1) + \overrightarrow{v_i}(t) \tag{11}$$

The information available for each individual is based on its own experience (the decisions that it has made so far and the success of each decision) and the knowledge of the performance of other individuals in its neighbourhood. Since the relative importance of these two factors can vary from one decision to another, it is reasonable to apply random weights to each part, and therefore the velocity will be determined by

$$\overrightarrow{v_i}(t) = \alpha\overrightarrow{v_i}(t) + \phi_1 r(\cdot)(p_i - \overrightarrow{x_i}(t-1)) + \phi_2 r(\cdot)(p_g - \overrightarrow{x_i}(t-1)) \tag{12}$$

where α called inertia is a parameter within the range [0, 1] and is often decreased over time; ϕ_1 and ϕ_2 are two constants, often chosen so that $\phi_1 + \phi_2 = 4$, which control the degree to which the particle "'follow the herd"' thus stressing exploitation (higher values of ϕ_1); $r(\cdot)$ is a uniformly random number generator function that returns values within the interval (0,1); and g is the particle in i's neighbourhood with the current neighbourhood-best candidate solution.

According to the formulation above, the following procedure can be used for implementing the PSO algorithm (Shi, 2004).

1. Initialize the swarm by assigning a random position in the problem hyperspace to each particle.

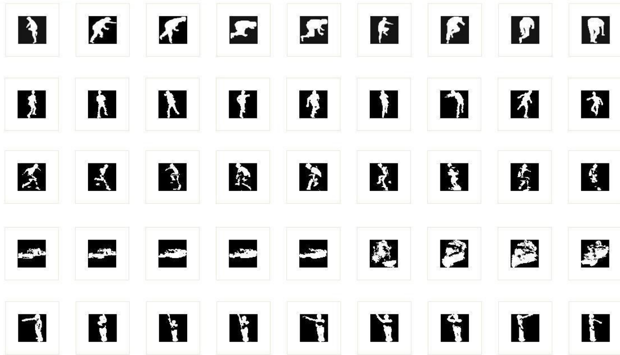

Figure 1: Example posture silhoutte from dataset. Top to bottom: Climbing, Fighting, Jumping, Lying Down and Pointing

2. Evaluate the fitness function for each particle.

3. For each individual particle, compare the particle's fitness value with its p_{best}. If the current value is better than the p_{best} value, then set this value as the p_{best} and the current particle's position, x_i, as p_i.

4. Identify the particle that has the best fitness value. The value of its fitness function is identified as g_{best} and its position as p_g.

5. Update the velocities and positions of all the particles using Eq. 11 and Eq. 12.

6. Repeat steps (2) to (5) until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

EXPERIMENTAL RESULTS

In this section, the comparison between supervised and unsupervised training to posture recognition is presented. Our dataset consist of five challenging postures silhouette, e.g. lying, jumping, fighting, climbing and pointing and a sample dataset is illustrated in Fig. 1. The dataset is collected in such a way that no occlusion between intra body-parts occurs and the camera distance to subject is constant for each posture. Similar constraints have been imposed by other researchers as many techniques and algorithms are at an experimental stage performing very well in tightly controlled laboratory settings as to (Spagnolo et al., 2003; Korde and Jondhale, 2008).

For each of the supervised or unsupervised methods we first perform the pre-processing and feature extraction as (Herrero-Jaraba et al., 2004) in order to remove noise and illumination problems occurring during the data capture phase. The feature extracted from each binary image comprises of a silhouette of the human figure. Secondly the input data (silhouette) is cropped to a 50x50 pixels to compensate the processing time needed. Finally, according to Section III and IV, each supervised and unsupervised algorithms are employed to learn and classify five different postures. All the algorithms are implemented in MATLAB and the learning and testing data is chosen empirically. The parameters settings for the algorithms can be summarized as below:

- The ANN consists of 2500 input neurons as each image is represented by a vector of 50x50 pixels in size with 100 dataset for each of the 5 posture classes. There are 60 hidden neurons and 5 output neurons, representing one for each of the posture class. The network was trained with the back-propagation using the Levenberg-Marquardt algorithm (Hagan and Menhaj, 1994). This algorithm appears to be the fastest method for training moderate-sized feed-forward neural networks (up to several hundred weights). Meanwhile, the activation function used for the ANN is a log sigmoid function as neural network learn much faster when the activation function is represented by a hyperbolic tangent (Negnevitsky, 2005).

- In K-means, the system is trained with 20 empirically chosen samples per posture and then increased to 50 samples per posture. However, we noticed that increasing the number of samples did not improve the recognition rate significantly therefore we revert back to using 20 samples are selected.

- The 2D lattice in SOM algorithm consists of 40 rows and 40 columns and the epoch is set to 200.

- PSO algorithm was initialised as to (Teng et al., 2010). In each separate experiment, different combinations of static or dynamic acceleration and static or dynamic inertia were used (thus leading to 4 different combinations). In this paper, the number of particle is set to 10 and cluster number is 5 for each experiments.

The system is trained with a dataset of 100 images per posture class. Testing is done a different set of test images (20 per posture class). Each image is classified as belonging to a particular posture class and its recognition accuracy is calculated. The overall effectiveness of these algorithms is determined using this recognition accuracy. This overall accuracy measurement determines how well the algorithms are able to correctly classify postures in a given dataset. The number of correctly classified posture is referred to as the True Positives (TP). Any postures that are not correctly classified are considered False Positives (FP). The overall accuracy is thus calculates as Eq. 13:

$$\text{Overall Accuracy} = \frac{\sum \text{TP for all cluster}}{\text{Total Number of Data}} \quad (13)$$

The recognition results for the dataset using different algorithms are shown in Table 1. From the analysis of the results, the following hypotheses can be made:

Table 1: Average Recognition Accuracy for the Posture Classes

Algorithm	Pointing	Jumping	Lying Down	Fighting	Climbing
SOM	45%	85%	65%	100%	100%
Neural Network	100%	100%	100%	100%	100%
K-means	30%	25%	20%	15%	65%
PSO	33%	5%	23%	4%	39%

- For all postures, the supervised algorithms outperform the unsupervised algorithms in terms of the percentage of correct classification. The main reason is that supervised learning algorithms learn a mapping from x to y, given a training set made of pairs (x_i, y_i), while the unsupervised learning algorithms find similarity in the training data and the resulting cluster should match the intuitive classification of the data. As the posture classes show large intra-class variation, the clusters discovered are not very distinctive for each posture class.

- On average, the supervised approaches were able to identify the more difficult poses of climbing and fighting postures very well as indicated in Table 1. Nevertheless, the recognition of the lying down posture by SOM was not as good as that obtained by ANN even though this particular posture is significantly different from the others.

- ANN was able to achieve 100% recognition accuracy for all the posture categories. The most probable reason for this is because the dataset of images used for training and testing are free from excessive noise and the posture silhouette of the human is easily distinguished in the image. The dataset of images used in experimental and testing work was collected in controlled environment and it was possible to keep it free from noise or any occlusion from other objects in the scene.

- PSO algorithm achieved the worst recognition rate in the experiment. One of the main reason as indicated by, Merwe and Engelbrecht (van der Merwe and Engelbrecht, 2003) lies in utilizing the PSO algorithm's optimal ability which, if given enough time, could generate a more compact clustering results from the low dimensional dataset than the traditional K-means clustering algorithm. However, when clustering large datasets, the slow shift from the global searching stage to the local refining stage causes the PSO clustering algorithm to require many more iterations to converge to an optimum solution than the K-means.

As Table 1 shows, the PSO and K-means algorithms achieved a low recognition rate for posture classes. Hence for these two approaches we decided to revise the experiment and extract a feature set for representing the human posture instead of using only the human silhouette image. For our study, we have selected six

Table 2: Recognition rate for K-means and PSO algorithms after feature extraction

Algorithm	Pointing	Jumping	Lying Down	Fighting	Climbing
K-means	50%	40%	80%	20%	60%
PSO	60%	40%	90%	30%	60%

features from the human body to be used in representing a single sample of silhouette image. These are the location of the head (H), left arm (A^L), right arm (A^R), left leg (L^L), right leg (L^R) and torso (T). The points of the features, $S = ([H_x, H_y], [A^L_x, A^L_y], [A^R_x, A^R_y], [L^L_x, L^L_y], [L^R_x, L^R_y], [T_x, T_y])$ selected represents the feature points as shown in Figure 2 where *x* and *y* are the pixels coordinate.

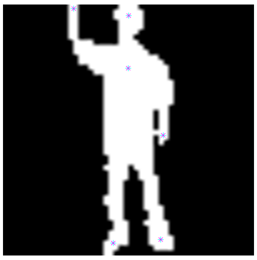

Figure 2: Feature points of the six body parts

The points obtained from the samples are used in the training as feature set and then for testing with the K-Means and PSO algorithms. Generally, K-Means is able to converge to a stable solution within a limited number of iteration number compared to PSO which will take a longer time to generate an optimal solution. Hence, for effective comparison, we fixed the number of iterations to be 300 for both the algorithms. The number of samples used in the training and testing phase is 10 samples per posture respectively. Table 2 gives the new results of revised experiment.

It can be surmised from the Table 2 that:

- K-means is able to converge to a solution quickly within a small time range. However, it tends to produces lower accuracies compared to PSO as can be observed in the case of fighting and pointing postures.

- Generally, the results of 'Fighting' and 'Jumping' postures are the lowest for both the algorithms. This could be due to the large similarities of the individual posture as compared to other postures. Including feature extraction stage with PSO has significantly improved the recognition rate for all the posture classes especially for the case of 'Fighting' and' Jumping' which saw the recognition rate jump to 30% and 40% respectively. Even K-means saw an improvement in the recognition rate using a feature set.

- The 'Lying Down' posture produces the highest accuracy for both the algorithms (as shown in Table 3) which was only 20% and 23% before the feature set was included. In the 'Lying Down' posture the orientation of the body plays a vital part in its recognition rate as the orientation is horizontal whereas for others it is vertical

Hence having an appropriate feature set plays a vital role to improve the recognition rate of the posture classes especially for K-means and PSO. The feature set can be made more effective if we extract the angle information for the feature set which includes the angle of the head and limbs with respect to the base line of the sample. Another modification could be to compute the distance of the nearest point of the subject to the base line of the image to determine the orientation of the subject.

CONCLUDING REMARKS

Automated surveillance systems are increasingly being used to analyze human behaviour to detect any suspicious activity. Being able to correctly recognize human posture contributes towards the detection of such an activity. We investigated several supervised and unsupervised algorithm to recognize human postures. The system was first trained to recognize the various postures and then tested against them. The results showed that the unsupervised algorithms tend to give low recognition rates as compared to the supervised algorithms. However the recognition rate from these algorithms showed promising improvement once we extracted a feature set from the posture dataset and tested it with them.

Future work will investigate how the recognition accuracy of the pair of lying down and jumping postures, even though quite distinctive, can be further improved. This improvement is needed especially in the case for unsupervised algorithms. The research work reported in this paper is in its preliminary stages but we intend to obtain accuracy results of the learning algorithms using publicly available databases such as CAVIAR. Finally we also plan to investigate a better selection of feature set to represent the postures and the effect this will have on improving the recognition accuracy of the system.

REFERENCES

Alpaydin, E. (2004). *Introduction to Machine Learning*. The MIT Press.

Bobick, A. and Davis, J. (1996). Real-time recognition of activity using temporal templates. In *Proceedings of IEEE CS Workshop on Applications of Computer Vision*, pages 39–42.

Boulay, B., Bremond, F., and Thonnat, M. (2003). Human posture recognition in video sequence. In *IEEE International Workshop on Visual Surveillance, Performance Evaluation of Tracking & Surveillance*, pages 23–29.

Bradski, G. and Davis, J. (2002). Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13(3):174–184.

Buccolieri, F., Distante, F., and Leone, A. (2005). Human posture recognition using active contours and radial basis function neural network. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 213–218.

Costa, L. and Cesar, R. M. (2001). *Shape Analysis and Classification: Theory and Practice*. CRC Press.

Duda, R. O., Hart, P., and Stork, D. G. (2000). *Pattern Classification*. Wiley - Interscience.

Hagan, M. and Menhaj, M. (1994). Training feed-forward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993.

Herrero-Jaraba, E., Orrite-Urunuela, C., Monzon, F., and Buldain, D. (2004). Video-based human posture recognition. In *Proceedings of the 2004 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, pages 19–22.

Kennedy, J. (1997). The particle swarm: social adaptation of knowledge. In *IEEE International Conference on Evolutionary Computation*, pages 303–308.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.

Korde, S. and Jondhale, K. C. (2008). Hand gesture recognition system using standard fuzzy c-means algorithm for recognizing hand gesture with angle variations for unsupervised users. In *Proceedings of the 2008 First international Conference on Emerging Trends in Engineering and Technology*, pages 681–685.

Minsky, M. L. and Papert, S. A. (1969). *Perceptrons*. Cambridge: MIT Press.

Negnevitsky, M. (2005). *Artificial Intelligence: A Guide to Intelligent Systems*. Pearson Education Ltd.

Shi, Y. (2004). Particle swarm optimisation. *IEEE Neural Network Society*, Feature Article:8–13.

Spagnolo, P., Leo, M., Leone, A., Attolico, G., and Distante, A. (2003). Posture estimation in visual surveillance of archaeological sites. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 277–283.

Teng, S. L., Chan, C. S., Lim, M. K., and Lai, W. K. (2010). Hybrid particle swarm optimization for data clustering. In *Proceedings of SPIE on the 2^{nd} International Conference of Digital Image Processing, Singapore*, volume 7546, pages 75460E1–75460E6.

T.Kohonen (1995). *Self-Organizing Maps*. Springer Series in Information Sciences.

van der Merwe, D. and Engelbrecht, A. (2003). Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation*, volume 1, pages 215–220.

Weinland, D., Ronfard, R., and Boyer, E. (2006). Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257.

Yi, H., Rajan, D., and Chia, L.-T. (2005). A new motion histogram to index motion content in video segments. *Pattern Recognition Letters*, 26(9):1221–1231.

AUTHOR BIOGRAPHIES

MALEEHA KIRAN received her Master degree in Computer and Information Engineering from International Islamic University, Malaysia in 2009. She is currently working as a Researcher in the Centre for Multimodal Signal Processing, MIMOS Berhad. Her research interests include image processing and machine learning.

CHEE SENG CHAN received his Ph.D. (Artificial Intelligence) from the University of Portsmouth, U.K. in 2008. Currently, he is a Senior Researcher in Centre of Multimodal Signal Processing, MIMOS Berhad, Malaysia. He previously held research appointment at the University of Portsmouth, U.K. His research interests include computer vision, machine learning, fuzzy qualitative reasoning, cognitive science; esp.: human motion recognition in images and video, biomedical image analysis, compositionality, graphical models and perceptual organization. He is a member for both IET and IEEE.

WENG KIN LAI holds Ph.D. (Engineering) from the University of Auckland, New Zealand. He has been with MIMOS Berhad for over 16 years and is currently the Principal Researcher of the research team developing solutions for video analytics. Weng Kin is a senior member of IEEE , a Fellow of IET and Member of the Governing Board of APNNA. He has served three full terms as the Hon Secretary of IET Malaysia local network, and is currently their Hon Treasurer. His current research interests are in machine intelligence and pattern recognition.

KYAW KYAW HITKE ALI holds B.Eng (Hons) in Electronics-Computer and Information Engineering from International Islamic University Malaysia (ranked 1^{st} out of 620 engineering students). Apart from co-authoring one journal article and a number of conference papers, he has also won several awards in research exhibitions at both national and university levels.

OTHMAN KHALIFA holds a Ph.D. (Digital Image Processing) from Newcastle University, U.K. He has worked in industry for eight years and is currently a Professor and Head of the department of Electrical and Computer Engineering, International Islamic University Malaysia. His areas of research interest are communication systems, information theory and coding, digital image/video processing, coding and compression, wavelets, fractal and pattern recognition. He is a senior member of IEEE.