ISSN:0975-9646

G. Venkata Narasimhulu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (5) , 2012,5025 - 5030

# Back Propagation Neural Network based Gait Recognition

G. Venkata Narasimhulu
*Dept of ECE*
*Tirumala Engineering College*
*Hyderabad,India*

Dr. S. A. K. Jilani
*Dept of ECE*
*Madanapalle Institute of Tech.and Science*
*Chithoor,India*

***Abstract*** **– We describe a new method for recognizing humans by their gait using back propagation neural network(BPNN), BPNN algorithm is used to recognize humans by their gait patterns. Automatic gait recognition using Fourier descriptors and independent component analysis (ICA) for the purpose of human identification at a distance. Firstly, a simple background generation algorithm is introduced to subtract the moving figures accurately and to obtain binary human silhouettes. Secondly, these silhouettes are described with Fourier descriptors and converted into associated one-dimension signals. Then ICA is applied to get the independent components of the signals. For reducing the computational cost, a fast and robust fixed-point algorithm for calculating ICs is adopted and a criterion how to select ICs is put forward. Lastly, the nearest neighbour (NN), support vector machine (SVM) and back propagation neural network (BPNN) classifiers are chosen for recognition and this method is tested on the small UMD gait database and the NLPR gait database. Gait recognition aims essentially to address this problem by identifying people based on the way they walk [1]. Gait recognition has 3 steps. The first step is pre-processing, the second step is feature extraction and the third one is classification. This paper focuses on the classification step that is essential to increase the CCR (Correct Classification Rate). Multilayer Perceptron (MLP) is used in this work. In this paper we apply MLP NN for 11 views in our database and compare the CCR values for these views. In experiments, higher gait recognition performances have been achieved.**

***Keywords*** **- Gait Recognition, Gait Biometrics, ICA, BPNN, MLP Neural Network, NN Classifier, SVM Classifier, BPNN classifier.**

## I. INTRODUCTION

The study of human gait has been increased extensive interests in various fields such as clinical analysis, computer animation, athletic performance analysis, visual surveillance, robotics and biometrics.

Gait Biometrics is a new powerful tool for reliable human identification and it makes use of human physiology characteristics such as face, iris, finger prints and hand geometry for identification.

Gait recognition can be broadly classified into two types: model-based and model-free approach [2].Model based approaches purpose to explicitly model human body or motion and performs model matching in each frame of a walking sequence[3,4,5]. Model based methods aim to model human body analysis of the parts of body such as hand, torso, thigh, legs and foot and perform model matching in each frame of a walking sequence to measure these parameters. This paper proposes a method of automatic gait recognition using Fourier descriptors and independent component analysis (ICA), which achieves high recognition accuracy results. The method proposed in the paper can be mainly divided into three procedures including human motion detection, feature representation and gait recognition. The main advantages of our approach in this paper are 1) based on Fourier descriptors and ICA 2) One dimensional signals are applied to represent the changing of moving silhouettes which can decrease computational cost effectively. 3) three classifiers namely nearest neighbour (NN), support vector machine (SVM) and back propagation neural network (BPNN) are applied for recognition and experimental results are compared with the three different classifiers. 4) It is easy to implement and has better recognition accuracy.

The neural networks can construct nonlinear decision boundaries without prior assumptions about the statistics of input data.

In next section, Feature extraction will be discussed briefly. Training and projection is discussed in section 3, the systems design is described in section 4, the experimental results are discussed in section 5, conclusion is given in section 6.

## II. FEATURE EXTRACTION

Before training and recognition, each image sequence including a walking figure is converted into an associated temporal sequence of distance signals at the pre-processing stage .An important cue in determining underlying motion of a walking figure is temporal changes of the walker's silhouette. To make the proposed method insensitive to changes of color and texture of clothes, we use only the binary silhouette. Additionally, for the sake of computational efficiency, we convert these 2D silhouette changes into an associated sequence of 1D signals to approximate temporal pattern of gait. This process is illustrated in Fig. 1. After the moving silhouette of a walking figure has been tracked, its outer contour can be easily obtained using a border following algorithm.

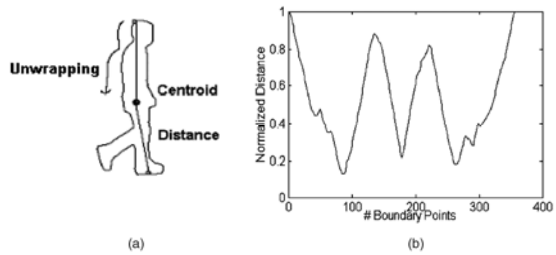


Fig. 1. Silhouette representation: (a) illustration of boundary extraction and counter clockwise unwrapping and (b) the normalized distance signal consisting of all distances between the centroid and the pixels on the boundary

Then, we may compute its shape centroid $(x_c,y_c)$.By choosing the centroid as a reference origin, we unwrap the outer contour counter clockwise to turn it into a distance signal $S=\{d_1,d_2,\ldots,d_{i,\ldots,}d_{Nb}\}$ that is composed of all distances $d_i$ between each boundary pixel $(x_i, y_i)$ and the centroid

$$d_i=[(x_i - x_c)^2 + (y_i - y_c)^2]^{1/2} \quad (1)$$

This signal indirectly represents the original 2D silhouette shape in the 1D space. To eliminate the influence of spatial scale and signal length, we normalize these distance signals with respect to magnitude and size. First, we normalize its signal magnitude through L1-norm. Then, equally spaced resampling is used to normalize its size into a fixed length (360 in our experiments). By converting such a sequence of silhouette images into an associated sequence of 1D signal patterns, we will no longer need to cope with those likely noisy silhouette data[1].

## III. TRAINING AND PROJECTION

### A. PCA Training

The purpose of PCA training is to obtain several principal components to represent the original gait features from a high-dimensional measurement space to a low-dimensional eigenspace. The training process similar to [11] is illustrated as follows:

Given s classes for training, and each class represents a sequence of distance signals of one subject's gait. Multiple sequences of each person can be freely added for training.

Let $D_{i,j}$ be the $j^{th}$ distance signal in class i and $N_i$ the number of such distance signals in the $i^{th}$ class. The total number of training samples is

$N_t = N_1+ N_2 + . . . + N_s$, and the whole training set can be represented by $[D_{1,1},D_{1,2},\ldots D_1,N_1, D_{2,1},\ldots,D_s,N_s]$ We can easily obtain the mean $m_d$ and the global covariance matrix $\sum$ of such a data set by

$$\text{Mean}(m_d) = 1/N_t \sum_{i=1}^{s} \sum_{j=1}^{N_i} D_{i,j} \quad (2)$$

$$\sum = 1/N_t \sum_{i=1}^{s} \sum_{j=1}^{N_i} (D_{i,j} - m_d)(D_{i,j} - m_d)^T \quad (3)$$

If the rank of the matrix Σ is N, then we can compute N nonzero eigenvalues $\lambda_1,\lambda_2,..,\lambda_N$ and the associated eigenvectors $e_1, e_2, \ldots,e_N$ based on SVD (Singular Value Decomposition).

.

Generally speaking, the first few eigenvectors correspond to large changes in training patterns. Therefore, for the sake of memory efficiency in practical applications, we may ignore those small eigenvalues and their corresponding eigenvectors

using a threshold value Ts

$$W_k = \sum_{i=1}^{k} \lambda_i \;/\; \sum_{i=1}^{N} \lambda_i > Ts \quad (4)$$

where $W_k$ is the accumulated variance of the first k largest eigenvalues with respect to all eigenvalues. In our experiments, $T_S$ is chosen as 0.95 for obtaining steady results[1].

### B. Projection

Taking only the K<N largest eigenvalues and their associated eigenvectors, the transform matrix E= $[e_1, e_2,\ldots,e_N]$ can be constructed to project an original distance signal into a point $P_{i,j}$ in the k-dimensional eigenspace.

$$P_{i,j}=[ e_1, e_2,\ldots,e_N]^T D_{i,j} \quad (5)$$

Accordingly, a sequential movement of gait can be mapped into a manifold trajectory in such a parametric eigenspace. It is well-known that k is usually much smaller than the original data dimension N. That is to say, eigenspace analysis can drastically reduce the dimensionality of input samples. For each training sequence, the projection centroid $C_i$ in the eigenspace is accordingly given by averaging all single projections corresponding to each frame in the sequence [1].

$$C_i= 1/Ni \sum_{j=1}^{N_i} P_{i,j} \quad (6)$$

## IV. SYSTEMS DESIGN

### A. Independent Component Analysis (ICA)

Gait recognition using independent component analysis (ICA) for the purpose of human identification at a distance. Firstly, a simple background generation algorithm is introduced to subtract the moving figures accurately and to obtain binary human silhouettes. Secondly, these silhouettes are described with Fourier descriptors and converted into associated one-dimension signals. Then ICA is applied to get the independent components of the signals. For reducing the computational cost, a fast and robust fixed-point algorithm for calculating ICs is adopted and a criterion how to select ICs is put forward.

### B. Fourier Descriptors

Gait recognition using Fourier descriptors. Gait recognition refers to automatic identification of an individual based on his/her style of walking; it is a new biometrics recognition technology. Gait contour by using Fourier descriptors, to make a periodical analysis on the height and width ratio of the gait image, and to solve the problems resulting from an image sequence of different gait cycles by using dynamic time warping. By utilizing the presented algorithm, experiments made on the CMU data bank have achieved a comparatively high correct identification ratio.

### C. Neural Network (NN)

Neural neworks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in the graphic below Fig.2.

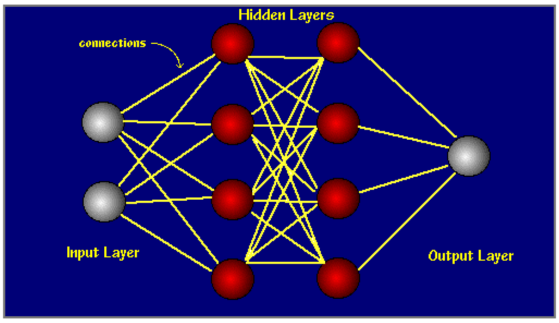


Fig 2. Neural Network

The neural network can be defined as an interconnection of neurons. Neural networks makes use of the fact that the recall of information can be effected in two ways. The recall can be performed in the feed forward mode. The feed forward have no memory. Feed forward network's behaviour does not depend on what happened in the past but rather what happens now. The network responds only to its present input. The area of Neural Networks probably belongs to the borderline between the Artificial Intelligence and Approximation Algorythms.The NNs are used in (to name few) universal approximation (mapping input to the output), tools capable of learning from their environment, tools for finding non-evident dependencies between data and so on. The Neural Networking algorithms (at least some of them) are modelled after the brain (not necessarily - human brain) and how it processes the information.

### *D. Back Propagation Neural Network (BPNN)*

Back propagation networks are necessarily multilayer perceptrons (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network. Non-linear activation functions that are commonly used include the logistic function, the softmax function, and the Gaussian function. The back propagation algorithm for calculating a gradient has been rediscovered a number of times, and is a special case of a more general technique called automatic differentiation in the reverse accumulation mode.

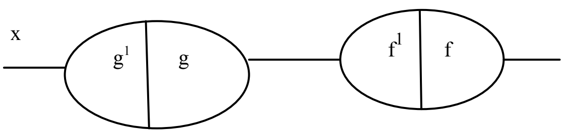


Fig. 3. Network for the composition of two functions.

The B-diagram of Figure 3 contains only two nodes. In the feed-forward step, incoming information into a unit is used as the argument for the evaluation of the node's primitive function and its derivative. In this step the network computes the composition of the functions f and g.

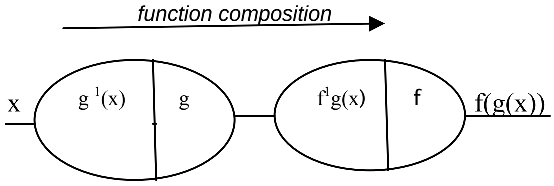


Fig. 4. Result of the feed-forward step

Figure 4 shows the state of the network after the feed-forward step. The correct result of the function composition has been produced at the output unit and each unit has stored some information on its left side. In the back propagation step the input from the right of the network is the constant 1. Incoming information to a node is multiplied by the value stored in its left side. The result of the multiplication is transmitted to the next unit to the left. We call the result at each node the traversing value at this node.

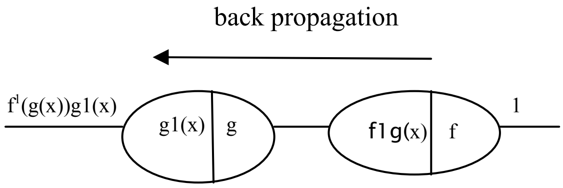


Fig. 5. Result of the back propagation step

Figure 5 shows the final result of the back propagation step, which is f0(g(x))g0(x), i.e., the derivative of the function composition f(g(x)) implemented by this network. The back propagation step provides an implementation of the chain rule. Any sequence of function compositions can be evaluated in this way and its derivative can be obtained in the back propagation step. We can think of the network as being used backwards with the input 1, whereby at each node the product with the value stored in the left side is computed.

### *E. Back Propagation Algorithm (BPA)*

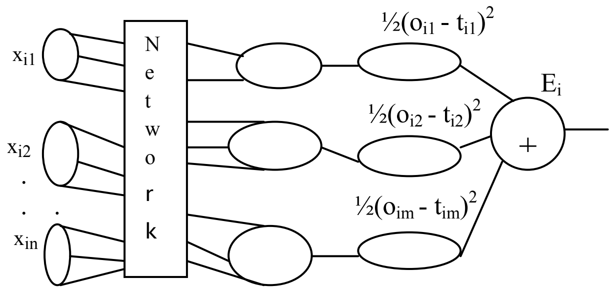


Fig.6. Extended network for the computation of the error function

The back propagation algorithm is used to find a local minimum of the error function. The network is initialized with randomly chosen weights. The gradient of the error function is computed and used to correct the initial weights. Our task is to compute this gradient recursively

The first step of the minimization process consists of extending the network, so that it computes the error function automatically. Figure 6 shows how this is done. Every one of the j output units of the network is connected to a node which evaluates the function $1/2\ (o_{ij} - t_{ij})^2$, where $o_{ij}$ and $t_{ij}$ denote the $j^{th}$ component of the output vector $o_i$ and of the target $t_i$. The outputs of the additional m nodes are collected at a node which adds them up and gives the sum $E_i$ as its output. The same network extension has to be built for each pattern ti. A computing unit collects all quadratic errors and outputs their sum $E_1 + \cdot \cdot \cdot + E_p$. The output of this extended network is the error function E.

We now have a network capable of calculating the total error for a given training set. The weights in the network are the only parameters that can be modified to make the quadratic error E as low as possible. Because E is calculated by the extended network exclusively through composition of the node functions, it is a continuous and differentiable function of the ` weights w1,w2, . . . ,w` in the network. We can thus minimize E by using an iterative process of gradient descent, for which we need to calculate the gradient

$$\nabla E = [\partial E / \partial w_1, \partial E / \partial w_2, \ldots\ldots\ldots\ldots \partial E / \partial w_l] \text{------}(7)$$

Each weight is updated using the increment

$$\blacktriangle w_i = -\gamma\ \partial E / \partial w_i \text{ for } i=1,2.........l$$

where γ represents a learning constant, i.e., a proportionality parameter which defines the step length of each iteration in the negative gradient direction.

With this extension of the original network the whole learning problem now reduces to the question of calculating the gradient of a network function with respect to its weights. Once we have a method to compute this gradient, we can adjust the network weights iteratively. In this way we expect to find a minimum of the error function, where $\nabla E = 0$.

We can now formulate the complete back propagation algorithm and prove by induction that it works in arbitrary feed-forward networks with differentiable activation functions at the nodes. We assume that we are dealing with a network with a single input and a single output unit.

Consider a network with a single real input x and network function F. The derivative F0(x) is computed in two phases:

1. Feed-forward: the input x is fed into the network. The primitive functions at the nodes and their derivatives are evaluated at each node. The derivatives are stored.

2.Back propagation: the constant 1 is fed into the output unit and the network is run backwards. Incoming information to a node is added and the result is multiplied by the value stored in the left part of the unit. The result is transmitted to the left of the unit. The result collected at the input unit is the derivative of the network function with respect to x.

### *F. Support Vector Machines (SVM)*

Support vector machines can perform binary classification and regression estimation tasks. Given a set of two-class labelled data $(x_i, y_i)$, $i = 1, 2, .., n$ and $y_i = \pm 1$, an SVM learns a separating hyper-plane $\langle w, x \rangle + b = 0$, where $x_i \in R^n$, $w \in R^n$, and $b \in R$. In the linear hyper-plane, the SVM looks for a discriminating plane that maximises the margin by minimising $||w||^2/2$, subject to $y_i (\langle w, x_i \rangle + b) \geq 1$ for all *i*. In the linear non-separable case, the optimal separating hyper-plane can be computed by introducing slack variables $\xi_i = 1, 2, ..,..n$ and an adjustable parameter $C$ and then minimising

$$\| \mathbf{w} \|^2/2 + \mathbf{C} \sum_i \xi_i, \text{ subject to } y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i$$

and $\xi_i \geq 0$ for all i .............................................. (8)

Lagrange multiplier $\alpha_i$ is used for solving the non-separable case by introducing the dual optimisation. The separating hyper-plane of linear function is not adjustable in many practical cases and takes the kernel function $K(\bullet)$ such that $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. This $\alpha_i$ can be computed by solving the quadratic optimisation problem as

$$\text{Min } W(\alpha) = -\sum_{i=1}^{n} \alpha_i + \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j\, y_i y_j\, K(x_i, x_j)$$

$$\text{s.t } \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C \text{ for all i } \text{------------ (9)}$$

Support vectors are the training examples with $\alpha_i > 0$. Specifically unbounded support $x_{usv}$ vectors [6] are with $0 < \alpha_i < C$ and bound support vectors with $\alpha_i = C$. The parameters of the separating hyper-plane are

$$W = \sum_{i=1}^{n} \alpha_i y_i x_i \text{ and } b = y_{usv} - \langle W, X_{usv} \rangle \text{ .................(10)}$$

In the present study, the implementation of the SVM is based on the working set selection strategy of SVM$^{light}$ and there kernels of linear ($\langle x_i, x_j \rangle$), polynomial $[(\langle x_i, x_j \rangle + 1)^p]$ and radial basis function $(\exp(-||x_i - x_j||^2 / p2))$ are chosen.

Automated gender classification system in human gait using support vector machine. Support vector machines (SVMs) and neural network are employed as classifiers for this gender classification task in a way of 10-fold cross validation (CV).

### *G. Multilayer Perceptron Neural Network (MLPNN)*

**Fig. 7. The walking sequences captured from 11 different views**

**Table 1. Compared Results for Lateral View**

| Method | Classifier | CCR |
|---|---|---|
| **NED** ( L. Wang, T. Tan) | **NN** (Nearest Neighbour) | **65** |
| **NED** ( L. Wang, T. Tan) | **ENN** (Euclidean Nearest Neighbour) | **75** |
| **NED (In this paper)** | **NN** (Neural Network) | **82** |

**Table 2. Results for all views based on the proposed method**

| S.No | View (In degrees) | Method | Classifier | CCR |
|---|---|---|---|---|
| 1 | 0 | NED | NN(Neural Network) | 50 |
| 2 | 18 | NED | NN(Neural Network) | 57 |
| 3 | 36 | NED | NN(Neural Network) | 70 |
| 4 | 54 | NED | NN(Neural Network) | 75 |
| 5 | 72 | NED | NN(Neural Network) | 75 |
| 6 | 90 | NED | NN(Neural Network) | 82 |
| 7 | 108 | NED | NN(Neural Network) | 62 |
| 8 | 126 | NED | NN(Neural Network) | 60 |
| 9 | 144 | NED | NN(Neural Network) | 54 |
| 10 | 162 | NED | NN(Neural Network) | 50 |
| 11 | 180 | NED | NN(Neural Network) | 54 |

## V. EXPERIMENTAL RESULTS

The classification process is carried out through three different methods, namely the nearest neighbour (NN), support vector machine (SVM) and classifier derived from the ICs. NN classifier is a very simple classifier and we use Euclidian distance to evaluate the discriminatory of two gait sequences. SVM classifiers have high generalization capabilities in many tasks especially in the object recognition problem. Neural network classifier [7],[8] and [9] is a very useful classifier which is widely used in multiple class classification. Generally, BPNN has multiple layers, we can simple it into three layers i.e. input layer, hidden layer and out layer.

Two famous public gait databases, namely University of Maryland database (UMD) and Chinese National Laboratory of Pattern Recognition (NLPR) database. NLPR database includes 20 subjects and four sequences for each view angle and have three angles namely laterally ($0^0$), obliquely ($45^0$) and frontally ($90^0$). Tables (3,.4& 5) gives the experimental results separately using different classifiers on the two databases (UMD and NLPR).

The average accuracy of polynomial kernel (p=6) was the best with around 100% for training, 95% for CV and 96% for testing in the 19 features. The experimental results shows that (Table.6), Polynomial kernel is better than linear kernel or RBF kernel [10] in this gender classification task.

**Table 3. Recognition results using NN classifier**

| Ranks | UMD database (6 persons, 1 view) | | NLPR database (20 persons, 3 views) | |
|---|---|---|---|---|
| | *90 ICs selected* | *Using all ICs* | *300 ICs selected* | *Using all ICs* |
| Rank 1 | 100% | 100% | 75% | 75% |
| Rank 2 | 100% | 100% | 85% | 85% |
| Rank 10 | 100% | 100% | 90% | 90% |

**Table 4. Recognition results using SVM classifier**

| Ranks | UMD database (6 persons, 1 view) | | NLPR database (20 persons, 3 views) | |
|---|---|---|---|---|
| | *90 ICs selected* | *Using all ICs* | *300 ICs selected* | *Using all ICs* |
| Rank 1 | 100% | 100% | 82.5% | 81.9% |
| Rank 2 | 100% | 100% | 87.6% | 86.1% |
| Rank 10 | 100% | 100% | 92.1% | 89.6% |

**Table 5. Recognition results using BPNN classifier**

| Ranks | UMD database (6 persons, 1 view) | | NLPR database (20 persons, 3 views) | |
|---|---|---|---|---|
| | *90 ICs selected* | *Using all ICs* | *300 ICs selected* | *Using all ICs* |
| *Rank 1* | 100% | 100% | 84.6% | 84.1% |
| Rank 2 | 100% | 100% | 89.4% | 88.6% |
| Rank 10 | 100% | 100% | 95.1% | 92.3% |

From the three tables (Table.3,Table.4& Table. 5), we can find that BPNN classifier is the best classifier used in gait recognition compared to the another two classifiers (NN and SVM). NN classifier is worst. NN classifier as it is very simple and it can save a large of computational time for our recognition which is very important for gait recognition system. SVM is a new classifier as it has strong generalization and it is very suit for two classifications. SVM classifier is the first choice other wise BPNN classifier is our best choice and it has the best recognition accuracy.

**Table 6. Experimental Results of 10-fold Cross Validation (CV) test**

| Kernel | *P* | Fts | SVs | Classification Rate (%) | | | FLOP |
|---|---|---|---|---|---|---|---|
| | | | | Training | CV | Testing | |
| Linear | | 26 | 121.1 ± 4.2 | 94.4±0.5 | 92.4±3.2 | 93.9±1.7 | 54.2±11.8 |
| | | 19 | 140.9 ± 6.2 | 93.4±0.4 | 91.24±3.2 | 94.6±1.2 | 67.2±15.2 |
| **Poly-nomial.** | 2 | 26 | 64.1 ±2.3 | 100±0.0 | 95.8±1.8 | 94.4±0.9 | 1.6±0.5 |
| | | 19 | 57.4 ± 3.4 | 100±0.0 | 95.4±3.3 | 96.1±0.7 | 1.9±1.3 |
| | **6** | 26 | 68.3 ± 4.3 | 100±0.0 | 94.0±4.8 | 93.9±0.7 | 0.9±0.4 |
| | | **19** | **63.8 ± 3.2** | **100±0.0** | **95.6±2.7** | **95.7±0.7** | **1.0±0.7** |
| RBF | 1.5 | 26 | 137.6 ± 4.6 | 96.8±0.3 | 94.4±2.8 | 95.7±0.9 | 4.1±0.4 |
| | | 19 | 133.7 ± 5.1 | 96.8±0.5 | 93.6±2.8 | 96.5±0.6 | 4.7±1.1 |
| | 2 | 26 | 145.6 ± 6.6 | 96.2±0.4 | 93.8±3.2 | 95.4±0.9 | 3.5±0.7 |
| | | 19 | 143.3 ± 4.4 | 95.8±0.4 | 94.2±2.2 | 96.7±0.3 | 4.0±0.9 |

## VI. CONCLUSION

This paper has proposed gait recognition method based on human silhouettes using Fourier descriptors (FD) and independent component analysis (ICA). The median background extraction method is better than the mean method and has less computational cost than the least mean square method. The results of SVM with polynomial kernel have produced very good classification rates which were 96% for 100 subjects on average. MLP neural network algorithm has higher recognition accuracy. Based on the experimental results, fourth, fifth and sixth views (equal to 72, 90 and 108 degrees) have better CCR results and the best one is obtained for a 90 degree lateral view. The results are shown in the tables 1 and 2. Table 1 shows that the obtained results are better than [1] (7% increase in CCR) and the table 2 shows the results for all views based on the proposed method and represents 5% to 25% increase in fourth and fifth views and 12% to32% increase in sixth view compared to the previous methods.

## REFERENCES


[1] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette Analysis-Based Gait recognition for Human Identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, pp. 1505-1518, 2003

[2] M. Ekinci, Gait recognition using multiple projections, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2006, pp. 517–522.

[3]A.Bazin,M.Nixon,Gaitverificationusingprobabilisticmethods,in:IEEE Workshop on Applications of Computer Vision,2005,pp.60–65.

[4] C.BenAbdelkader,R.Culter,H.Nanda,L.Davis, Eigen Gait: motion based recognition of people using image self-similarity,in:Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication,2001,pp.284–294.

[5] A.Bobick,A.Johnson,Gait recognition using static activity-specific parameters,in:Proceedings of the IEEE Computer Vision and Pattern Recognition,2001,pp.423–430

[6] Joachims, T.: Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer (2002)

[7] Hagan,M.T., Demuth, H.B.Beale, M.: Neural network Design.China machine Press and CITIC Publishing House. Beijing(2002)

[8] Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey, 2nd ed.(1999)

[9] Rajapakse, J.C., Wang, L.P.(Eds.): Neural Information Processing: Research and Development. Springer, Berlin (2004)

[10] Shin, M., and Park, C.: A Radial Basis Function Approach to Pattern Recognition and Its Applications. ETRI Journal, **22**(2) (2000) 1-10

[11] P. Huang, C. Harris, and M. Nixon, "Human Gait Recognition in Canonical Space Using Temporal Templates," IEE Proc. Vision Image and Signal Processing Conf., vol. 146, no. 2, pp. 93-100, 1999


## AUTHOR'S BIOGRAPHY

**Mr G.Venkata Narasimhulu**[1] is an Associate professor in ECE Dept at Tirumala Engineering College, ECE Dept (Affiliated to JNT University, Hyderabad), Ranga Reddy-Dist, AP, India. He received his Masters degree from Sri Venkateswara University, Tirupati, A.P., India. He has published many articles in National and International Journals.. He has been attended in the organization of a number of workshops His main research interest includes image processing, Digital Signal Processing, Neural Networks,

and Bioinformatics. He has memberships in technical organization like IE(MIE), IETE(MIETE) and ISTE(MISTE). He has fellow ship of ISECE (FISECE).

**Dr. S. A. K. Jilani**[2]

Professor, Dept. of ECE, Madanapalle Institute of Tech. and Science, Madanapalle, Chithoor -Dist, A.P., India.