Abstract—In this paper, we present a fast learning neural network classifier for human action recognition. The proposed classifier is a fully complex-valued neural network with a single hidden layer. The neurons in the hidden layer employ the fully complex-valued hyperbolic secant as an activation function. The parameters of the hidden layer are chosen randomly and the output weights are estimated analytically as a minimum norm least square solution to a set of linear equations. The fast leaning fully complex-valued neural classifier is used for recognizing human actions accurately. Optical flow-based features extracted from the video sequences are utilized to recognize 10 different human actions. The feature vectors are computationally simple first order statistics of the optical flow vectors, obtained from coarse to fine rectangular patches centered around the object. The results indicate the superior performance of the complex-valued neural classifier for action recognition. The superior performance of the complex neural network for action recognition stems from the fact that motion, by nature, consists of two components, one along each of the axes.

I. INTRODUCTION


Recently complex-valued classifiers available in the literature are used to solve real valued classification tasks such as Multi Layer Multi Valued Network (MLMVN) [4], the single layer complex-valued neural network with phase encoded input features [5] referred to as ‘Phase Encoded Complex Valued Neural Network (PE-CVNN)’ and Phase Encoded Complex-valued Extreme Learning Machine (PE-CELM) [6]. A multi-valued neuron used in the MLMVN [4] uses multiple-valued threshold logic to map the complex-valued input to C discrete outputs using a piecewise continuous activation function, where C is the total number of classes. The transformation used in the MLMVN is not unique, leading to misclassification. The misclassification is further enhanced by the increase in number of sectors inside the unit circle in multi-category classification problems with more number of classes (C). Moreover, the MLMVN uses a derivative-free global error correcting rule for network parameter update that requires significant computational effort.

In the PE-CVNN and PE-CELM presented in [5], [6], the real-valued input feature is phase encoded in $[0, \pi]$ to obtain the complex-valued input feature. This transformation retains the relational property and spatial relationship among the real-valued input features [5]. However, the activation functions used in [5] are similar to the split complex-valued activation functions and do not preserve the phase information of the error signal during the backward computation. This might result in inaccurate estimation of the decision function while performing classification problems. Moreover, the gradient descent based learning, presented in [5], also requires significant time to train the classifier.

In this paper, we propose a complex-valued classifiers that require considerably insignificant computational effort compared to the other classifiers. The classifiers have a non-linear hidden layer and a linear output layer. The neurons in the hidden layer of these networks use the fully complex-valued activation function of the type of a hyperbolic secant function [7]. Similar to the C-ELM (Complex Extreme Learning Machines) [8], the parameters of the hidden neurons are chosen randomly and the output parameters of the networks are estimated analytically.

Action recognition is one of the crucial computer vision tasks in applications like video surveillance and monitoring. Developing algorithms for human action recognition is hard due to the complexity of the scene. Typical action recognition approaches [9], [10], [11], [12], [13] assume that each video clip has been pre-segmented to have a single action. Motion information between consecutive image frames provides vital cues about the object dynamics. These motion information over an image plane is a two-dimensional vector, corresponding to horizontal and vertical displacement. This motion information is basically complex in nature and hence it should be treated as complex valued feature rather than real-valued feature for effective action modeling. In our proposed approach we have extracted accumulated motion information (optical-flow) over a small time window for extracting complex-valued motion features for each frame to represent the corresponding action. We have used the Weizmann human action dataset for evaluating the performance [12].

The paper is organized as follows. Section 2 describes the fast learning fully complex-valued neural classifier. Section 3 explains action recognition using Complex-valued Neural Classifier and the performance evaluation. Section 4 provides concluding remarks and future directions.

II. DESCRIPTION OF THE CLASSIFIER

In this section, we present the detailed description of fast learning fully complex-valued neural classifier.
A. Classification problem definition

Let us assume \( N \) random observations \( \{(x_1, c_1), \ldots, (x_t, c_t), \ldots, (x_N, c_N)\} \), where \( x_t \in \mathbb{C}^m \) are the \( m \)-dimensional complex-valued input features of \( t \)-th observation and \( c_t \in \{1, 2, \ldots, C\} \) is its class label. The coded class label in the complex domain \( y^t \) are obtained using:

\[
y^t_l = \begin{cases} 
  1 + i, & \text{if } c_t = l, \\
  -1 - i, & \text{otherwise},
\end{cases} \quad l = 1, 2, \ldots, C \tag{1}
\]

Now, the classification problem in the complex domain can be viewed as finding the decision function \( F \) that maps the complex-valued input features to the complex-valued coded class labels, i.e., \( F : \mathbb{C}^m \rightarrow \mathbb{C}^C \), and then predicting the class labels of new, unseen samples with certain accuracy.

B. Fast learning complex-valued classifiers

The fully complex-valued classifier is a single hidden layer network, with a non-linear hidden layer and a linear output layer as shown in Fig. 1.

The neurons in the hidden layer of the neural classifier employ the fully complex-valued activation function of the type of a hyperbolic secant function [7], and their responses are given by

\[
y^t_{hj} = \text{sech} \left( u^T_j (x_t - v_j) \right); \quad j = 1 \cdots K \tag{2}
\]

where \( u_j \) is the complex-valued scaling factor and \( v_j \) is the center of the \( j \)-th neuron, and \( \text{sech}(x) = 2/(e^x + e^{-x}) \).

The neurons in the output layer employ linear activation function and the output of the classifier is given by:

\[
\hat{y}_l = \sum_{j=1}^{K} w_{lj} y^t_{hj}, \quad l = 1, \ldots, C \tag{3}
\]

where \( w_{lj} \) are the complex-valued weight connecting the \( l \)-th output neuron and the \( j \)-th hidden neuron.

The class labels can be estimated from the outputs using:

\[
\hat{c} = \max_{l=1,2,\ldots,C} \text{real} \left( \hat{y}_l \right) \tag{4}
\]

Eq. (3) can be written in a matrix form as

\[
\hat{Y} = WH \tag{5}
\]

where \( W \) is the matrix of all output weights connecting the hidden layer, and \( H \) is the \( K \times N \) matrix of the response of the hidden neurons for the samples in the training data set given by

\[
H (V, B, Z) = \begin{bmatrix}
\text{sech}(u_1 \| z_1 - v_1 \|) & \cdots & \text{sech}(u_1 \| z_N - v_1 \|) \\
\vdots & \ddots & \vdots \\
\text{sech}(u_K \| z_1 - v_K \|) & \cdots & \text{sech}(u_K \| z_N - v_K \|)
\end{bmatrix} \tag{6}
\]

Similar to the C-ELM [8], the parameters of the hidden neurons \( (u_j, v_j) \) are chosen randomly and the output weights \( W \) are estimated by the least squares method according to:

\[
W = Y H^\dagger \tag{7}
\]

where \( H^\dagger \) is the Moore-Penrose inverse of the hidden layer output matrix, and \( Y \) is the complex-valued coded class label.

The proposed fully complex-valued neural classifier can be summarized as:

- For a given training set \( (X, Y) \), select the appropriate number of hidden neurons \( K \).
- Choose the scaling factor \( U \) and the neuron centers \( V \) randomly.
- Calculate the output weights \( W \) analytically: \( W = Y H^\dagger \).

In this paper, selection of an appropriate number of hidden neurons is done using the addition/deletion of neurons to obtain an optimal performance, as discussed in [14] for real-valued networks.

III. HUMAN ACTION RECOGNITION USING COMPLEX-VALEUDED NEURAL CLASSIFIER

Human action recognition is one of the crucial tasks in applications such as video surveillance. It is a challenging problem due to complexity of the real-life problem such as background clutter, occlusion illumination and scale variations. There have been many proposals for modeling and representing actions starting from Hidden Markov Models (HMM) [15], [10] through interest point models [11], [9], to silhouette tunnel shape models [12], [13]. All these
approaches utilize real valued spatio-temporal features to represent various actions. The motion information between two frames is a vital source of information to model human actions. The motion information is basically complex in nature and hence it should be treated as complex valued feature rather than real-valued feature for effective action modeling.

A. Action Representation

The motion flow between consecutive frames are obtained using Lukas and Kanade’s Optical Flow technique [16]. We represent the action as a motion flow-history image over a short period of time window \((w)\) (say, 5 frames) as given in equation (8).

\[
H_k(i, j) = \sum_n F_n(i, j), \quad n \in \{(k-w), \ldots k\} \tag{8}
\]

Here, \(H_k(i, j)\) indicates the flow history of \(k\)th frame at \((i, j)\)th location and \(F_n(i, j)\) represents the optical flow between \(n\)th and \((n-1)\)th frame at \((i, j)\)th location. For 'Wave2' action represented in Fig. 2(a), the corresponding flow-history image is shown in Fig. 2(c). The background subtracted image shown in Fig. 2(b), yields the object center about which a rectangle patch is chosen as shown in Fig. 3(a) for feature extraction. This representation is very useful in recognizing actions instantly as video frames arrive. The action recognition for a wider time window could be achieved from the recognition results at frame level.

B. Feature Extraction

First, the optical flow for each frame of the video sequences are obtained. The object center for each frame is obtained from the mask generated by subtracting the given background video frame from the current frame. Now, the flow vectors for the current frame are accumulated from the past few frames. In our experiment, motion information from 6 previous frames are accumulated for feature extraction.

In the experiment, 54 rectangular patches surrounding the object, of hierarchically increasing sizes are utilized. The mean value of non-zero motion vectors in each block is computed as the representative motion vector as given in Eq. (9).

\[
F_k^b = \frac{\sum_{(i,j) \in R_b} H_k(i, j)}{\sum_{(i,j) \in R_b} I_k(i, j)}, \tag{9}
\]

where,

\[
I_k(i, j) = \begin{cases} 1, & \text{if } |H_k(i, j)| > 0, \\ 0, & \text{otherwise} \end{cases} \tag{10}
\]

Here, \(F_k^b\) indicates the representative motion vector corresponding to \(b\)th rectangular block in \(k\)th frame, \(R_b\) indicates the pixels that belong to \(b\)th rectangle block and \(I(i, j)\) is a binary matrix indicating the locations of non-zero motion vectors.

Figure 3 illustrates the process of feature extraction at frame level. The final feature vectors are obtained by finding the mean optical flow within each of the rectangular patches surrounding the object center, in a hierarchical fashion as illustrated in Fig. 3(a-f). Initially the mean flow vectors are obtained at finer resolution using 36 windows each of size 6 \(\times\) 4, symmetrically laid about the image center. The remaining features are obtained by merging the windows sequentially to sizes of: 12 \(\times\) 8 (Fig. 3(b)), 18 \(\times\) 12 (Fig. 3(c)), 36 \(\times\) 12 (Fig. 3(d)), 18 \(\times\) 24 (Fig. 3(e)) and 36 \(\times\) 24 (Fig. 3(f)) as illustrated.

The representative motion vectors of all the 54 patches are collected as the feature vector. Since each motion vector has \(x\) and \(y\) component, they are combined to form a complex number \((Mv_x + i.Mv_y)\). The final feature vectors are, 54-dimensional complex numbers. These feature vectors are scaled appropriately to keep the real/imaginary values within -1 to 1 range.

C. Performance Evaluation

In this section, we present performance of fully complex-valued neural classifier on human action recognition. For this purpose, we use Weizmann Human Action Dataset. This dataset\(^1\) contains 90 low resolution video sequences (180 \(\times\) 144), where 10 actions were performed by 9 subjects. The background videos were also provided for all action sequences. The 10 actions include: bend, jack, jump, pjump, run, side, skip, walk, wave1 and wave2. Totally 5036 frame level feature vectors were obtained from 10 action classes. For training, randomly 3000 frame level features were used and tested against the remaining 2036 frames. The performance of the classifier is evaluated using average and overall accuracies.

The average \((\eta_a)\) (Eq. (11)) and over-all \((\eta_o)\) (Eq. (12)) classification efficiencies of the classifiers derived from their confusion matrices are used as the performance measures for comparison in this study.

\[
\eta_a = \frac{1}{C} \sum_{i=1}^{C} \frac{q_{ii}}{N_i} \times 100\% \tag{11}
\]

\[
\eta_o = \frac{\sum_{i=1}^{C} q_{ii}}{\sum_{i=1}^{C} N_i} \times 100\% \tag{12}
\]

where \(q_{ii}\) is the total number of correctly classified samples in the class \(c_i\) and \(N_i\) is the total number of samples belonging to a class \(c_i\) in the data set.

In our simulation study, we used a fully complex-valued neural network with 400 hidden neurons. The number of hidden neurons are chosen based on incremental-decremental strategy given in [14]. The overall/average training and test accuracy is around 93%. The performance for individual action is given in table I and the confusion matrix is given in table III for test. Some actions like bend and wave2 are having similar motion flow as Wave1, hence the performance of these actions are below average. This could possibly overcome by using additional information such as silhouette and global features such as object trajectory. In order to compare the results with regular real-valued classifiers, same dataset

\(^1\)http://wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html
Fig. 2. (a) Original image (b) corresponding background subtracted mask (c) accumulated flow.

TABLE III
ACTION RECOGNITION RESULTS - CONFUSION MATRIX (TEST)

<table>
<thead>
<tr>
<th>Action</th>
<th>Bend</th>
<th>Jack</th>
<th>Jump</th>
<th>pJump</th>
<th>Run</th>
<th>Side</th>
<th>Skip</th>
<th>Walk</th>
<th>Wave1</th>
<th>Wave2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bend</td>
<td>80.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19.6</td>
<td>0</td>
</tr>
<tr>
<td>Jack</td>
<td>0</td>
<td>94.8</td>
<td>0</td>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>Jump</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pJump</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>91.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Run</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>93.4</td>
<td>0</td>
<td>4.4</td>
<td>2.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Side</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>93.4</td>
<td>2</td>
<td>2.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Skip</td>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0.55</td>
<td>1.1</td>
<td>0</td>
<td>94.5</td>
<td>2.19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Walk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wave1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wave2</td>
<td>0.42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14.6</td>
<td>85</td>
<td></td>
</tr>
</tbody>
</table>

TABLE I
ACTION RECOGNITION RESULTS USING COMPLEX-VALUED ELM CLASSIFIER

<table>
<thead>
<tr>
<th>Action</th>
<th>Accuracy ηi (%)</th>
<th>Accuracy ηi (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test</td>
<td>Training</td>
</tr>
<tr>
<td>Bend</td>
<td>80.4</td>
<td>77.1</td>
</tr>
<tr>
<td>Jack</td>
<td>94.8</td>
<td>96.1</td>
</tr>
<tr>
<td>Jump</td>
<td>99.5</td>
<td>98.7</td>
</tr>
<tr>
<td>pJump</td>
<td>94.9</td>
<td>99.8</td>
</tr>
<tr>
<td>Run</td>
<td>93.4</td>
<td>97.5</td>
</tr>
<tr>
<td>Side</td>
<td>93.4</td>
<td>98.5</td>
</tr>
<tr>
<td>Skip</td>
<td>94.5</td>
<td>98.7</td>
</tr>
<tr>
<td>Walk</td>
<td>100</td>
<td>99</td>
</tr>
<tr>
<td>Wave1</td>
<td>100</td>
<td>94.8</td>
</tr>
<tr>
<td>Wave2</td>
<td>85</td>
<td>82.6</td>
</tr>
</tbody>
</table>

TABLE II
ACTION RECOGNITION RESULTS USING SVM CLASSIFIER

<table>
<thead>
<tr>
<th>Action</th>
<th>Accuracy ηi (%)</th>
<th>Accuracy ηi (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test</td>
<td>Training</td>
</tr>
<tr>
<td>Bend</td>
<td>24.1</td>
<td>33.1</td>
</tr>
<tr>
<td>Jack</td>
<td>93.3</td>
<td>91.7</td>
</tr>
<tr>
<td>Jump</td>
<td>65.9</td>
<td>78.1</td>
</tr>
<tr>
<td>pJump</td>
<td>39.1</td>
<td>48</td>
</tr>
<tr>
<td>Run</td>
<td>80</td>
<td>87.8</td>
</tr>
<tr>
<td>Side</td>
<td>74.2</td>
<td>76.5</td>
</tr>
<tr>
<td>Skip</td>
<td>76.9</td>
<td>81.2</td>
</tr>
<tr>
<td>Walk</td>
<td>90.6</td>
<td>92.8</td>
</tr>
<tr>
<td>Wave1</td>
<td>76.6</td>
<td>73.8</td>
</tr>
<tr>
<td>Wave2</td>
<td>40.1</td>
<td>38.4</td>
</tr>
</tbody>
</table>
was trained with various classifiers including multi class SVM, using liblinear package\(^2\). The prediction results, for the test data, for all the classifiers were in the range of 65% to 67%, which is 25% less than the proposed complex-valued neural classifier. The performance for individual action using SVM is given in table II. This result indicates the importance of using complex valued data for motion information, which by nature consists of components in two directions.

Recognizing actions at sequence level could be achieved by utilizing the frame level detection results and global features such as trajectory information. Future research direction will be towards increasing the performance with global features such as object trajectory information with local frame-level complex feature. Since there is no frame level action recognition results available in literature for the dataset, we will be converting the frame level result to sequence level result and compare the performance.

### IV. Conclusion

The exceptional universal classification ability of the complex-valued neural network is attributed to their inherent orthogonal decision boundaries. In this paper, we present an efficient, fast learning complex-valued classifier for human action recognition. Here, the input weights are randomly selected and the output weights are calculated analytically. The performances of the classifier is evaluated using benchmark action recognition sequences. The results indicate superior performance of the proposed complex-valued neural classifier over the traditional real-valued classifiers.

\(^2\)http://www.csie.ntu.edu.tw/~cjlin/liblinear

### REFERENCES

