Neurocomputing ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Human action recognition using a fast learning fully complex-valued classifier ☆

R. Venkatesh Babu [a,*], S. Suresh [b], R. Savitha [b]

[a] Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, India
[b] School of Computer Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 11 August 2011
Received in revised form
29 February 2012
Accepted 2 March 2012
Communicated by L. Shao

Keywords:
Action recognition
Complex-valued neural networks
Optical flow
Human Action
Orthogonal decision boundaries

ABSTRACT

In this paper, we use optical flow based complex-valued features extracted from video sequences to recognize human actions. The optical flow features between two image planes can be appropriately represented in the Complex plane. Therefore, we argue that motion information that is used to model the human actions should be represented as complex-valued features and propose a fast learning fully complex-valued neural classifier to solve the action recognition task. The classifier, termed as, "fast learning fully complex-valued neural (FLFCN) classifier" is a single hidden layer fully complex-valued neural network. The neurons in the hidden layer employ the fully complex-valued activation function of the type of a hyperbolic secant function. The parameters of the hidden layer are chosen randomly and the output weights are estimated as the minimum norm least square solution to a set of linear equations. The results indicate the superior performance of FLFCN classifier in recognizing the actions compared to real-valued support vector machines and other existing results in the literature. Complex valued representation of 2D motion and orthogonal decision boundaries boost the classification performance of FLFCN classifier.

1. Introduction

Action recognition is one of the crucial computer vision tasks in applications such as video archive indexing, human computer interaction, video surveillance and monitoring. Developing algorithms for human action recognition is hard owing to the complexity of the scene due to occlusion, clutter, illumination change, appearance or viewpoint change and intra-class action variations. Typical action recognition approaches [2–6] assume that each video clip has been pre-segmented to have a single action performed by single subject.

Image based action representation is one of the earlier approaches, where the region of interest (ROI) is obtained by background subtraction. Feature extraction is performed within the bounding box of ROI. In one of the initial works by Bobick and Davis [7], the extracted silhouettes are used to construct binary motion energy image (MEI), which indicates where motion occurs. A motion history image (MHI) is also constructed, where pixel intensities are a recency function of the silhouette motion. These two templates are compared using *Hu* moments [8] for recognizing actions. Babu et al. [9] extracted motion history from compressed video by utilizing the readily available motion vector information. Yamato et al. [10] used grid-based silhouette mesh features to form a compact codebook of observations for representing actions using hidden Markov model. Extracting silhouettes in real-life videos is challenging and prone to noise. The noisy silhouettes are handled by phase correlation [11], by constructing space–time volume over silhouette images [5,12] or by using shape context descriptors [13,14]. Gorelick et al. [5], stack silhouettes over a given sequence to form a space–time volume. Then, the solution of the Poisson equation is used to derive local space–time saliency and orientation features. Global features for a given temporal range are obtained by calculating weighted moments over these local features. These features are then used to recognize actions using nearest neighbor classification technique. Guo et al. [15] extracted the feature vectors from segments of silhouette tunnels. The classification is performed by approximating the log-covariance of query segment by sparse linear combination of action dictionaries. In order to improve the robustness of the global image-based approach against partial occlusion, viewpoint variation and noise, the ROI is divided into a fixed spatial or temporal grid. Each cell in the grid describes the image observation locally, and the matching function is changed accordingly from global to local. Recently, Shao et al. [16] represented actions using correlogram of body poses obtained from the silhouette sequences. Kellokumpu et al. [17] calculate local binary patterns along the temporal dimension and store a

☆ An earlier brief version of this paper has appeared in the Proceedings of the International Joint Conference on Neural Networks, 2011 (IJCNN 2011) [1].

* Corresponding author.
E-mail address: venky@serc.iisc.ernet.in (R. Venkatesh Babu).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.03.003

histogram of non-background responses in a spatial grid. Thurau and Hlavoc [18] use Histograms of Oriented Gradients (HOG, [19]) and focus on foreground edges by applying non-negative matrix factorization. Lu and Little [20] use principal component analysis to reduce the dimension of the HOG descriptor.

Efros et al. [21] calculate optical flow in person-centered images in order to model relative motions among different locations of object. Here, the objects are players in football game occupying a very small region in the image plane. The motion descriptor is constructed from the positive and negative components of horizontal and vertical flow vectors leading to four distinct bins. The descriptors are then matched to a database of pre-classified actions using the *k*-nearest-neighbor framework. Danafar and Gheissari [22] extended this work by dividing the region of interest (ROI) into horizontal slices that approximately contain head, body and legs. Ali and Shah [23] derive 11 kinematic features from the optical flow. These include divergence, vorticity, symmetry and gradient tensor features. Principal component analysis is applied to determine the dominant kinematic modes. The actions are classified using multiple instance learning, in which each action video is represented by a bag of kinematic modes.

Local representations of actions provide better robustness to viewpoint variance, noisy background subtraction and partial occlusion. Laptev [24] extended the Harris corner detector [25] to 3-dimension (3D). Space–time interest points are those points where the local neighborhood has a significant variation in both the spatial and the temporal domain. The scale of the neighborhood is automatically selected for space and time individually. The descriptors of these interest points are often used to generate a codebook by clustering. A local descriptor is described as a codeword contribution. A frame or sequence can be represented as a bag-of-words, or a histogram of codeword frequencies to represent the corresponding action [4,2]. Mattivi et al. [26] described space–time interest points using local binary pattern on three orthogonal planes for recognizing human actions. Shao and Mattivi [27] evaluated various features for part based human action recognition. Poppe [28] and Weinland et al. [29] presented a detailed survey on human action recognition. Most of the action recognition approaches rely on computationally expensive features, extracted from space–time volumes or silhouette tunnels information which may not be suitable for real-time applications.

Motion information (optical flow) between consecutive image frames provide vital cues about the object dynamics. Most of the reported works either utilize local features such as space–time interest points or utilize global features such as shape/silhoutte. We need the dynamics of the object both at local and global levels in order to effectively model the actions. Optical flow vectors are the most suitable candidate for extracting simple dynamic features hierarchically from local pixel level to global object level. Motion field of an image plane is a two-dimensional vector (corresponding to horizontal and vertical displacements) and shall suitably be represented in the Complex plane. We advocate the notion that representing motion information as complex values, rather than as two decoupled real values is more effective for modeling actions. Even, simple motion-based feature can be very effective in recognizing actions if represented in Complex domain.

Therefore, in this paper, the motion features that represent each action are represented in the Complex domain by extracting the accumulated motion information (optical-flow) over a small time window. Then, we employ a complex-valued neural classifier to perform the action recognition task on the *Weizmann* human action dataset [5] and *KTH* human motion dataset [30]. The action recognition task is considered as a classification task, where the objective of the classifier is to approximate the decision function that maps the complex-valued input features to the corresponding action class labels. As the input features are inherently complex-valued, a fully complex-valued neural classifier can learn the decision function more efficiently than real-valued neural classifiers.

The development of a fully complex-valued neural network is limited by the Complex domain, as Liouville's theorem [31] states that an entire and bounded function is a constant function in the Complex domain. On the contrary, a constant function is not acceptable as an activation function. These conflicting requirements led to reducing the essential properties of a fully complex-valued activation function to be analytic and bounded *almost everywhere* [32]. A set of elementary transcendental functions with finite singularities are suggested as activation functions for a fully complex-valued multi-layer perceptron network in [32]. In addition, a fully complex-valued radial basis function network with a fully complex-valued activation function that satisfies the reduced essential properties of a fully complex-valued activation function has been developed in [33]. However, these networks use a gradient descent based batch learning algorithm that are computationally intensive. Moreover, the singularities of the activation functions affect the convergence of these networks [34]. Therefore, in this paper, we propose a fast learning fully complex-valued neural (FLFCN) classifier that is computationally less intensive and does not suffer from issues due to convergence.

FLFCN classifier is a single hidden layer fully complex-valued neural network with the hidden layer employing a fully complex-valued activation function of the type of hyperbolic secant [33]. The parameters of the hidden neurons of the network are chosen randomly and the output parameters are computed analytically, and hence, learning using FLFCN is fast. It is shown that the decision boundaries formed by the real and imaginary parts of the output neurons are orthogonal to each other. These orthogonal decision boundaries of a fully complex-valued neural network help to perform classification tasks efficiently. The action recognition performance of FLFCN classifier is studied in comparison to the support vector machine (SVM) classifier and other results available in the literature for *Weizmann* and *KTH* datasets. The results show the superior classification ability of FLFCN.

The paper is organized as follows. Section 2 explains action representation. Section 3 presents a survey on complex-valued neural networks and a detailed description of the fast learning fully complex-valued neural classifier. In Section 4, the performance of the proposed classifier is studied in detail. Section 5 presents the concluding remarks and future directions.

2. Human action representation

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 scenarios such as background clutter, occlusion, illumination and scale variations. There have been many proposals for modeling and representing actions starting from hidden Markov models [35,3] through interest point models [4,2], to silhouette tunnel shaped models [5,6]. 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 inherently complex-valued and hence, a fully complex-valued neural classifier can learn the decision function more accurately than real-valued neural classifiers.

2.1. Action representation

The motion flow between consecutive frames is obtained using Lukas and Kanade's optical flow technique [36]. We represent the

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

action as accumulated motion flow over a short period of time window (w) (say, 4–6 frames) as given below

$$G_k(i,j) = \sum_n F_n(i,j), \quad n \in \{(k-w), \ldots, k\}. \tag{1}$$

Here, $G_k(i,j)$ indicates the accumulated flow for the kth frame at (i,j)th location and $F_n(i,j)$ represents the optical flow between nth and $(n-1)$th frame at (i,j)th location. For 'Wave2' action represented in Fig. 1(a), the corresponding accumulated flow image is shown in Fig. 1(c). The background subtracted image, shown in Fig. 1(b), yields the object center about which a rectangle patch is chosen as shown in Fig. 1(d) 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.

2.2. Feature extraction

First, the optical flow for each frame of the video sequences is 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 six 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 patch is computed as the representative motion vector as given below

$$F_k^b = \frac{\sum_{(i,j)\in R_b} G_k(i,j)}{\sum_{(i,j)\in R_b} I_k(i,j)}, \tag{2}$$

where

$$I_k(i,j) = \begin{cases} 1 & \text{if } |G_k(i,j)| > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Here, F_k^b indicates the representative motion vector corresponding to bth rectangular block in kth frame, R_b indicates the pixels that belong to bth rectangle block and $I(i,j)$ is a binary matrix indicating the locations of non-zero motion vectors. The final frame-level features are extracted by cascading the representative motion vector of each block.

$$\mathbf{z}_k^f = [F_k^1 F_k^2 \ldots F_k^{54}], \tag{4}$$

where $\mathbf{z}_k^f$ is the frame-level feature of kth frame. Fig. 2 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. 2(a)–(f). Initially, the mean flow vectors are obtained at finer resolution using 36 windows each of size 6×4, symmetrically laid about the image center. The remaining features are obtained by merging the windows sequentially to sizes of: 12×8 (Fig. 2(b)), 18×12 (Fig. 2(c)), 36×12 (Fig. 2(d)), 18×24 (Fig. 2(e)) and 36×24 (Fig. 2(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 components, they are combined to form a complex number $(z_x + i.z_y)$. The final feature vectors have 54 dimensions, each of which is a complex number (i.e., $\mathbf{z}_t^f \in \mathbb{C}^{54}$). These feature vectors are normalized such that the real/imaginary parts of the features are in the range $[-1,1]$. Similarly, the class label for each action is encoded in the Complex domain $\mathbf{y}_k$ as

$$y_k^l = \begin{cases} 1+i1 & \text{if } c_k = l, \\ -1-i1 & \text{otherwise,} \end{cases} \quad l = 1,2,\ldots,C. \tag{5}$$

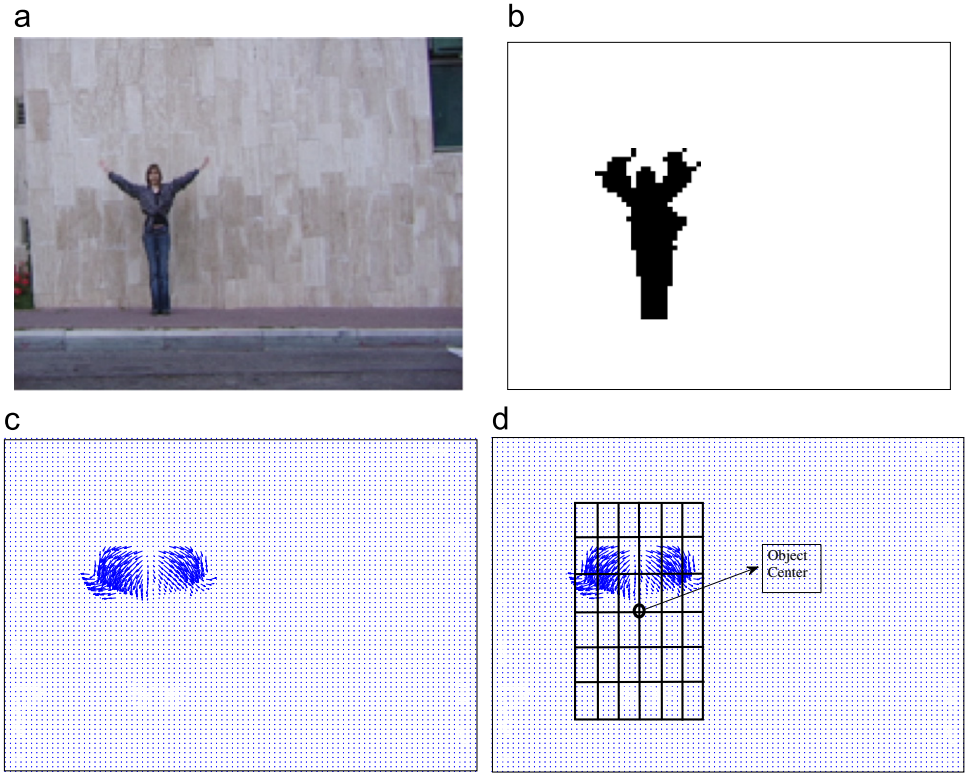

Fig. 1. (a) Original image; (b) corresponding background subtracted mask; (c) accumulated flow; and (d) feature extraction from patches surrounding object center.

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

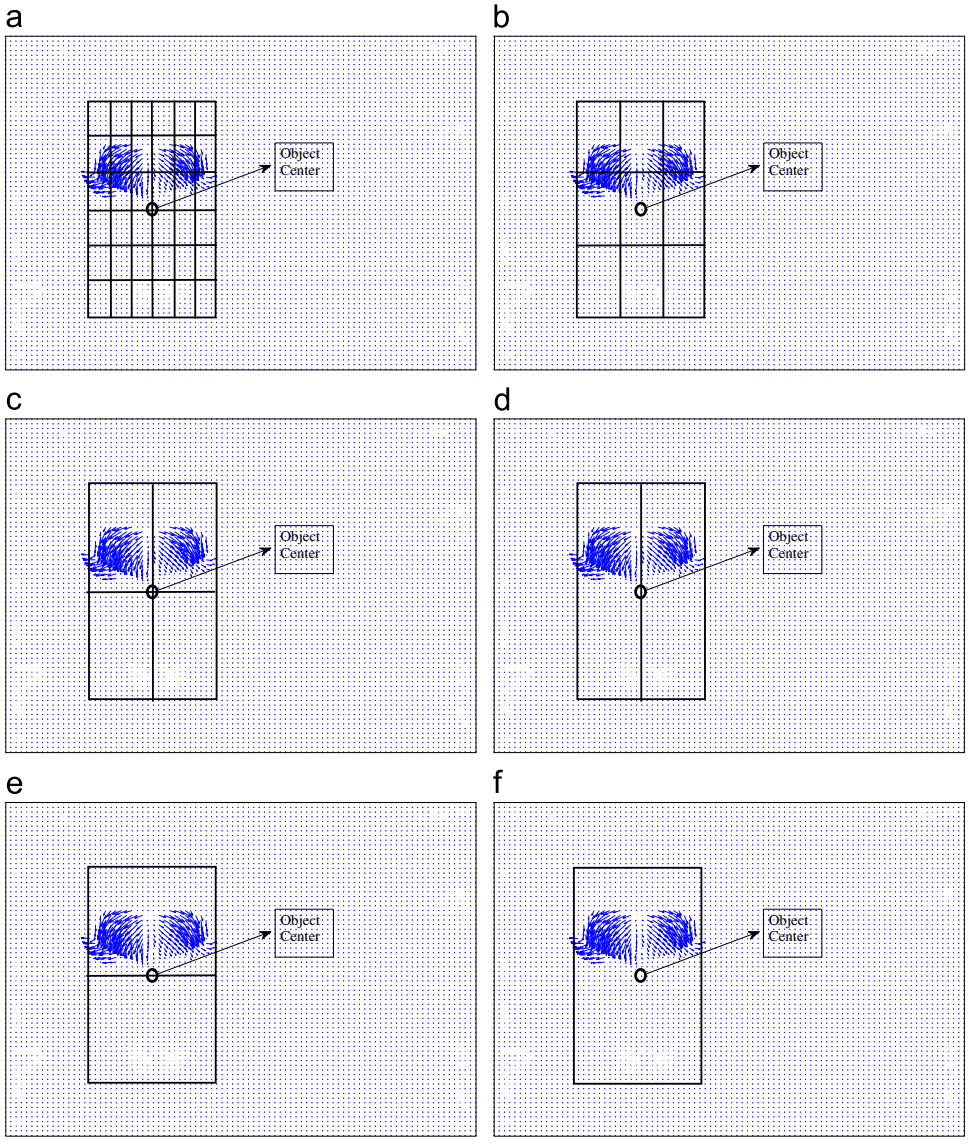

Fig. 2. Feature extraction: Fine to coarse strategy (a) first level fine image patches (No. of patches=36); (b) second level patches (No. of patches=9); (c) third level coarse patches (No. of patches=4); and (d)–(f) other coarse resolution patches (No. of patches=5).

Therefore, the action recognition task is defined as estimating the decision function that maps the complex-valued feature vector ($\mathbf{z}_k^f$) to their corresponding action class labels ($\mathbf{y}_k$), i.e., estimate $F : \mathbb{C}^m \rightarrow \mathbb{C}^C$, and then predict the actions of new, unseen actions with certain accuracy.

Since the input features and the coded class labels are both in the Complex domain, we present a computationally efficient fully complex-valued learning algorithm for recognizing human actions. In the next section, we briefly introduce complex-valued neural networks, followed by the detailed description of the proposed fast learning fully complex-valued neural classifier (FLFCN).

3. Complex-valued neural networks

Complex-valued neural networks have better computational power than the real-valued networks [37]. T. Nitta showed that a complex-valued neural networks with split-type of activation function have orthogonal decision boundaries, enabling them to perform classification effectively [38].

The main challenge in developing a fully complex-valued neural networks is the choice of an appropriate activation function. This is due to Liouville's theorem [31], which states that an entire and bounded function is a constant function in the Complex domain. Since activation function in a neural network has to be both differentiable and bounded, Kim and Adali [32] reduced the above requirement for a complex-valued activation function to be analytic and bounded *almost everywhere*. They also suggested a set of elementary transcendental functions as possible choice of an activation function for a fully complex-valued multilayer perceptron (FC-MLP) network. Later, Huang et al. [39] developed a complex-valued extreme learning machine using these activation functions to solve approximation problems. It was observed in [34] that the singularities of these activation functions affect the convergence and the learning capability of the FC-MLP network.

Since the radial basis function networks are well-known for their localization properties and decision making abilities, they can be used to solve classification tasks efficiently. A fully complex-valued radial basis function (FC-RBF) network with a fully complex-valued Gaussian-like sech ($\mathrm{sech}(z) = 2/(e^z + e^{-z})$)

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

activation function has been developed in [33]. The sech function satisfies all the desired properties of a fully complex-valued activation function and has a better approximation ability than other complex-valued radial basis function networks [33]. It has singularities at $(2k+1)/2\pi i$, where $k \in \mathbb{C}$ and is analytic and bounded almost everywhere. The presence of orthogonal decision boundaries in FC-RBF and its classification abilities has been shown in [40]. Recently, a meta-cognitive learning algorithm for the gradient descent based FC-RBF network was presented in [41] and it was also shown that the meta-cognitive component improved the classification ability of FC-RBF classifier. However, the gradient-descent based learning algorithm derived in [33,41] is computationally intensive, which affects the speed of learning. Recently, a sequential learning algorithm with a self-regulatory learning mechanism that chooses appropriate samples to participate in the training process has been developed in [42]. A computationally efficient and accurate fast learning fully complex-valued relaxation network has been developed in [43]. In addition to the above algorithms, several computationally efficient fast learning complex-valued classifiers have been developed to solve real-valued classification problems in [44–46]. As these classifiers have been originally developed to solve real-valued classification problems, the neurons in the input layer of these classifiers employ a non-linear transformation function to map the real-valued input features to the Complex domain.

In the next section, we present a fast learning fully complex-valued neural (FLFCN) classifier for human action recognition. The basic building block of FLFCN classifier is FC-RBF network, which is a single hidden layer network. It employs the fully complex-valued sech activation function in the hidden layer and a linear activation function in the input/output layer. It must be noted that as we are using complex-valued input features to represent the action recognition task, the neurons in the input layer of the FLFCN are linear, unlike those in [44–46]. During training, the hidden layer parameters of FLFCN classifier are chosen randomly and the output weights are estimated analytically, as the solution to a set of linear equations. Therefore, training FLFCN classifier takes significantly lesser computational time and it exploits the advantages of the fully complex-valued sech activation function. The following section presents a detailed description of the classifier.

3.1. A fast learning fully complex-valued neural (FLFCN) classifier

Let the frame level features of the various subjects and their corresponding coded action labels be represented by the training dataset

$$\{(\mathbf{z}_1^f,\mathbf{y}_1),\ldots,(\mathbf{z}_k^f,\mathbf{y}_t),\ldots,(\mathbf{z}_N^f,\mathbf{y}_N)\},$$

where $\mathbf{z}_k^f$[1] $\in \mathbb{C}^m$ are the m-dimensional complex-valued input features of the kth observation (note that for the frame-level features, $m=54$) and $\mathbf{y}_t \in \mathbb{C}^C$ are its encoded action labels. Now, we present a fast learning fully complex-valued neural classifier that performs the mapping of the complex-valued feature vectors to their corresponding action class labels.

The basic building block of the fast learning fully complex-valued neural classifier is the FC-RBF network with the complex-valued sech activation function as shown in Fig. 3. FLFCN classifier is a single hidden layer network with a non-linear hidden layer and a linear input/output layer.

The neurons in the hidden layer of FLFCN classifier employ the fully complex-valued activation function of the type of a hyperbolic secant function [33], and their responses are given by

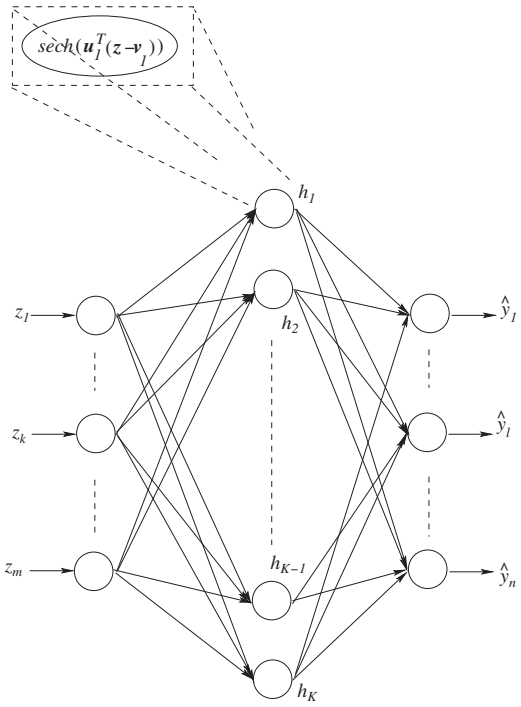

Fig. 3. The architecture of the fast learning fully complex-valued neural classifier.

$$h_j = \mathrm{sech}(\mathbf{u}_j^T(\mathbf{z}_k-\mathbf{v}_j)), \quad j=1,\ldots,K, \ k=1,\ldots,N, \tag{6}$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are the complex-valued scaling factor and center of the jth hidden neuron, respectively, $\mathrm{sech}(z)=2/(e^z+e^{-z})$.

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

$$\hat{y}_l = \sum_{j=1}^{K} w_{lj} y_h^j, \quad l=1,\ldots,C, \tag{7}$$

where w_{lj} is the complex-valued weight connecting the jth hidden neuron and the lth output neuron.

Eq. (7) can be written in a matrix form as

$$\hat{Y} = WH, \tag{8}$$

where W is the matrix of all output weights connecting the hidden layer to the output layer, and H is a $K \times N$ matrix of the response of the hidden neurons for all the samples in the training dataset given by

$$H(V,U,Z) = \begin{bmatrix} \mathrm{sech}(\mathbf{u}_1^T\|\mathbf{z}_1-\mathbf{v}_1\|) & \cdots & \mathrm{sech}(\mathbf{u}_1^T\|\mathbf{z}_N-\mathbf{v}_1\|) \\ \vdots & \vdots & \vdots \\ \mathrm{sech}(\mathbf{u}_j^T\|\mathbf{z}_1-\mathbf{v}_j\|) & \cdots & \mathrm{sech}(\mathbf{u}_j^T\|\mathbf{z}_N-\mathbf{v}_j\|) \\ \vdots & \vdots & \vdots \\ \mathrm{sech}(\mathbf{u}_K^T\|\mathbf{z}_1-\mathbf{v}_K\|) & \cdots & \mathrm{sech}(\mathbf{u}_K^T\|\mathbf{z}_N-\mathbf{v}_K\|) \end{bmatrix}.$$

During training, the parameters of the hidden neurons ($\mathbf{u}_j$, $\mathbf{v}_j$) are chosen randomly and the output weights W are estimated by the least squares method according to

$$W = YH^{\dagger}, \tag{9}$$

where $H^{\dagger}$ is the Moore–Penrose pseudo-inverse of the hidden layer output matrix, and Y is the complex-valued coded class label. Thus, the output weights can be estimated analytically, and the learning using FLFCN classifier is fast.

[1] The superscript f is dropped in future discussions for notational convenience.

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

After computation of the predicted outputs ($\hat{\mathbf{y}}$), the class labels can be estimated from the outputs using:

$$\hat{c} = \max_{l=1,2,\ldots,C} \text{real} \quad (\hat{y}_l). \tag{10}$$

The learning algorithm of FLFCN classifier can be summarized as:

- For a given training set (Z, 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: $W = YH^{\dagger}$.

The classification performance of FLFCN is affected by the number of hidden neurons. In this paper, selection of an appropriate number of hidden neurons is done by adding neurons until an optimal performance is achieved [47], as shown below:

Step 1. Select a network with a minimum configuration ($K = m + C$).
Step 2. Select the input weights randomly and compute the output weights analytically.
Step 3. Use leave-one cross-validation to determine training/validation accuracy from the training data.
Step 4. Increase K until the validation accuracy improves and return to Step 2.
Step 5. If the validation accuracy decreases as K increases, then stop.

In Appendix A, we show that the decision boundaries formed by the real and imaginary parts of the neurons in the output layer of FLFCN classifier are orthogonal to each other. The two decision boundaries that divide the Complex plane into four regions help FLFCN classifier to perform better. Based on this proof, we state the following lemma:

Lemma 3.1. *The decision boundaries formed by the real and imaginary parts of an output neuron of FLFCN classifier are orthogonal to each other.*

In the next section, we use FLFCN classifier to perform human action recognition tasks and highlight the advantages of the orthogonal decision boundaries of FLFCN classifier.

4. Performance evaluation

In this section, we present the results of the human action recognition problem using FLFCN classifier. For this purpose, we use *Weizmann*[2] and *KTH*[3] datasets. The *Weizmann* dataset contains 90 low resolution video sequences (180×144), 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. The *KTH* dataset contains six actions: *walking*, *jogging*, *running*, *boxing*, *hand waving and hand clapping*, performed several times by 25 subjects. In *KTH* dataset, many sequences are recorded with a slightly shaking camera, and moreover this dataset does not provide background models and extracted silhouettes.

The performance of FLFCN classifier is compared with the real-valued support vector machine classifier and the existing results in the literature for these datasets. The average classification efficiency (η_a) of the classifier derived from their confusion matrices is used as the performance measure for comparison in this study

$$\eta_a = \frac{1}{C}\sum_{l=1}^{C}\frac{q_{ll}}{N_l} \times 100\%, \tag{11}$$

where q_{ll} is the total number of correctly classified samples in the class c_l and N_l is the total number of samples belonging to a class c_l in the dataset.

Table 1
Action recognition results for *Weizmann* dataset using FLFCN classifier at frame level.

Action	Accuracy η_a(%) test	Accuracy η_a(%) training
Bend	80.4	77.1
Jack	94.8	96.1
Jump	100	98.7
pJump	91.5	89.5
Run	93.4	97.5
Side	93.4	97.8
Skip	94.5	98.7
Walk	100	99
Wave1	100	94.8
Wave2	85	82.6

4.1. Recognition at frame level

For *Weizmann* dataset, totally 5036 frame level samples (feature vectors) were obtained from 10 action classes. For training, 3000 randomly chosen frame level samples are chosen and the performance of FLFCN classifier is tested using the remaining 2036 frames. In our simulation study, we use a FLFCN classifier with 400 hidden neurons. The number of hidden neurons are chosen based on incremental–decremental strategy given in [47]. The average training and test accuracy is around 93%. The performance for individual action is given in Table 1 and the confusion matrix of the samples in the testing dataset is given in Table 2. Some actions like *bend* and *wave2* have similar motion flow as *wave1*, hence the performance of these actions are below average. In the following subsection, we show that the performance of these classes greatly improve, by considering the neighboring frames. In order to compare the results with regular real-valued classifiers, the same dataset is used for training multi-class SVM, using the lib-linear package.[4]

The prediction results for the test data, for all the classifiers are in the range of 65–67%, which is 25% less than the proposed FLFCN classifier. The performance for individual actions using SVM is given in Table 3.

For each action in *KTH* dataset, the sequences without scale change are considered. Totally, 10,398 frame level samples were obtained from six action classes. Randomly chosen 5000 samples were used for training and tested using the remaining samples. FLFCN classifier is trained with 200 hidden neurons. The average training and test accuracy is around 86%. The performance for individual action is given in Table 4 for both FLFCN and SVM classifiers. The prediction results for the test data using SVM classifier is around 56%, which is 25% less than the proposed FLFCN classifier.

These results indicate the importance of using complex-valued features for motion information, which by nature consists of components in two directions.

[2] http://wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html
[3] http://www.nada.kth.se/cvap/actions/
[4] http://www.csie.ntu.edu.tw/~cjlin/liblinear

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

Table 2
Action recognition results for *Weizmann* dataset at frame level—confusion matrix (test).

Action	Bend	Jack	Jump	pJump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	80.4	0	0	0	0	0	0	0	19.6	0
Jack	0	94.8	0	1.6	0	0	0	0.4	1.6	1.6
Jump	0	0	100	0	0	0	0	0	0	0
pJump	0.5	0	0	91.5	0	0	0	0	8	0
Run	0	0	0	0	93.4	0	4.4	2.2	0	0
Side	0	0	2	0	0	93.4	2	2.6	0	0
Skip	1.6	0	0	0.55	1.1	0	94.5	2.19	0	0
Walk	0	0	0	0	0	0	0	100	0	0
Wave1	0	0	0	0	0	0	0	0	100	0
Wave2	0.42	0	0	0	0	0	0	0	14.6	85

Table 3
Action recognition results for *Weizmann* dataset using SVM classifier at frame level.

Action	Accuracy η_a (%) test	Accuracy η_a (%) training
Bend	24.1	33.1
Jack	93.3	91.7
Jump	63.9	78.1
pJump	39.1	48
Run	80	87.8
Side	74.2	76.5
Skip	76.9	81.2
Walk	90.6	92.8
Wave1	76.6	73.8
Wave2	40.1	38.4

Table 4
Action recognition test results for *KTH* dataset at frame level.

Action	Accuracy η_a (%) FLFCN	Accuracy η_a (%) SVM
Boxing	80.7	64.2
Hand clapping	86.3	69.2
Hand waving	88.1	83.3
Jogging	75.0	47.9
Running	87.8	21.9
Walking	96.4	52.2

4.2. Recognition at sub-sequence Level

Recognizing actions at sub-sequence level is achieved by segmenting the video into many overlapping sub-shots. In our experiments with *Weizmann* dataset, each sub-shot contains eight consecutive frames. The feature vector for each sub-shot is obtained by combining individual frame features

$$\mathbf{z}_k^s = [\mathbf{z}_{k-7}^f \cdots \mathbf{z}_k^f], \tag{12}$$

where $\mathbf{z}_k^s$ is the feature vector of kth sub-sequence and $\mathbf{z}_k^f$ is the feature vector corresponding to kth frame (see Eq. (4)). Thus, each sub-sequence is represented using a 54×8-dimensional complex-valued feature vector.

Totally, there are 4385 sub-shots from 10 action classes. For training, 3000 randomly chosen sub-shot features are used and the performance of FLFCN classifier is tested using the remaining 1385 sub-shots.

In this experiment, we chose around 800 hidden neurons based on incremental–decremental strategy. The average training and test accuracy is around 98%. The performance for individual action is given in Table 5 and the confusion matrix for the samples in the testing dataset is given in Table 6. Here, the performance of *bend* and *wave2* actions have improved approximately by 10% compared to the frame-level results. The confusion of the above actions with *wave1* action has greatly reduced due to the temporal support from neighboring frames. The other actions (*jack*, *pJump*, *run*, *side* and *skip*) also show good performance improvement compared to the frame-level recognition. As the hidden layer parameters of the classifier are chosen randomly, we present the best, worst and mean test performance of FLFCN classifier over 20 trials in Table 7. From the table, it can be observed that random selection of the hidden layer parameters does not affect the performance of the classifier significantly. The number of hidden neurons in the trials was kept in the range of 800–850. To highlight the advantages of using a complex-valued classifier in performing the action recognition task, the performance of FLFCN classifier is also compared against that of the real-valued multi-class SVM with real-valued features using the lib-linear package. The training accuracy for the sub-sequence level classification using SVM classifier is almost 100%, whereas the test performance is around 87%. The performance for

Table 5
Action recognition results at sub-sequence level for *Weizmann* dataset using FLFCN classifier.

Action	Accuracy η_a (%) test	Accuracy η_a (%) training
Bend	90.4	85.1
Jack	99.5	100
Jump	100	100
pJump	100	100
Run	100	100
Side	100	100
Skip	99.1	100
Walk	100	100
Wave1	100	99.7
Wave2	95.8	96.3

Table 6
Action recognition results at sub-sequence level for *Weizmann* dataset—confusion matrix (test).

Action	Bend	Jack	Jump	pJump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	90.4	0	0	0	0	0	0	0	9.6	0
Jack	0	99.5	0	0	0	0	0	0	0	0.5
Jump	0	0	100	0	0	0	0	0	0	0
pJump	0	0	0	100	0	0	0	0	8	0
Run	0	0	0	0	100	0	0	0	0	0
Side	0	0	0	0	0	100	0	0	0	0
Skip	0	0	0	0	0	0	99.1	0	0.9	0
Walk	0	0	0	0	0	0	0	100	0	0
Wave1	0	0	0	0	0	0	0	0	100	0
Wave2	0	0	0	0	0	0	0	0	4.2	95.8

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

Table 7
Test performance of the proposed approach (for *Weizmann* dataset): best, worst and mean.

Best	Worst	Mean
98.5%	97.4%	97.9%

Table 8
Action recognition results for *Weizmann* dataset using SVM classifier at sub-sequence level.

Action	Accuracy η_i (%) test	Accuracy η_i (%) training
Bend	83.1	100
Jack	88.2	100
Jump	78.9	100
pJump	72.8	100
Run	94.7	100
Side	83.7	100
Skip	87.3	99.6
Walk	95.1	100
Wave1	96.6	100
Wave2	93.0	100

Table 9
Action recognition test results for *KTH* dataset at sequence level.

Action	Accuracy η_a (%) FLFCN	Accuracy η_a (%) SVM
Boxing	89.8	69.8
Hand clapping	95.6	67.5
Hand waving	95.3	86.8
Jogging	81.7	60.7
Running	90.6	31.9
Walking	96.6	66.2

individual actions using SVM is given in Table 8. This result indicates the poor generalization performance of the traditional classifiers, which use real-valued features.

In the experiment with *KTH* dataset, each sub-shot contains two consecutive frames. Totally, there are 10,084 sub-shots from six action classes. For training, 5000 randomly chosen sub-shot features are used and the performance of FLFCN classifier is tested using the remaining sub-shots. In this experiment, we chose around 600 hidden neurons based on incremental–decremental strategy. The average test accuracy is 91.6%. The performance for individual action is given in Table 9 for both FLFCN and SVM classifiers. The average test performance of SVM classifier is 63.8%. It can be observed from Table 9 that the proposed classifier performs much better than the traditional SVM classifier. The performance of the proposed approach is similar to [49]. Since the proposed approach uses simple optical flow based features, it achieves real-time performance of 25–30 frames/s. Whereas 3D-Harris based approach runs only at 1.6 frames/s [50] due to the huge computational cost involved in detecting 'space–time interest points'.

Tables 10 and 11 compare the action recognition results of the proposed approach with some of the state-of-the-art methods for *Weizmann* and *KTH* datasets, respectively.

In this work, we use simple optical flow features, whereas Gorelick et al. [5] and Guo et al. [15] use silhouette information that is prone to noise and difficult to extract in real-life scenarios is used. Whereas Wang and Mori [48] use bag-of-words model for recognizing actions. Here, the motion descriptors are obtained after tracking and stabilizing the persons in the video. Hence, the performance of the algorithm [48] largely depends on tracking algorithm used. From Tables 10 and 11, it can be observed that the performance of the proposed approach is on par with the aforementioned state-of-the-art methods. Moreover, the computational effort has been significantly reduced during both the feature extraction and the action recognition phase.

Table 10
Comparison of proposed method with state-of-the-art methods for *Weizmann* dataset.

Method	Proposed	Guo et al. [15]	Gorelick et al. [5]	Ali et al. [23]
Performance	97.93%	96.74%	97.83%	95.75%

Table 11
Comparison of proposed method with state-of-the-art methods for *KTH* dataset.

Method	Proposed	Yang et al. [48]	Schuldt et al. [30]	Ali et al. [23]	Laptev et al. [49]
Performance	91.6%	91.2%	71.7%	87.7%	91.8%

5. Conclusion

In this paper, we have proposed an efficient, fast learning fully complex-valued neural classifier (FLFCN) for human action recognition. As the optical flow between two image planes is effectively represented in the Complex domain, a set of simple complex-valued optical flow features are used for action representation. A complex-valued neural classifier has been developed to recognize various human actions. FLFCN is a fully complex-valued neural network with a sech activation function at the hidden layer. The input weights of FLFCN classifier are selected randomly and the output weights are computed analytically. The orthogonality of decision boundary of FLFCN is proved and the performance of the classifier is evaluated using *Weizmann* and *KTH* human action datasets. The action recognition performance of FLFCN is compared against real-valued SVM classifier and state-of-the-art results available in the literature. The results indicate the superior performance of FLFCN classifier over the other real-valued classifiers.

Acknowledgments

The authors wish to express grateful thanks to the referees for their useful comments and suggestions to improve the presentation of this paper. The second and third authors also wish to thank the Ministry of Education (MoE), Singapore for the financial support through Tier-I (No. M58020020) funding to conduct this study.

Appendix A. Orthogonality of decision boundaries of FLFCN classifier

Let the m-dimensional input features[5] be $\mathbf{z}=\mathbf{z}^R+i\mathbf{z}^I \in \mathbb{C}^m$, hidden layer responses be $\mathbf{h} \in \mathbb{C}^K$, given by $\mathbf{h}=\mathbf{h}^R+i\mathbf{h}^I$, the input to each hidden neuron be $O_k=O_k^R+iO_k^I=\mathbf{u}_k^T(\mathbf{z}-\mathbf{v}_k); k=1,\ldots,K$, the output weights be $w_{kj}=w_{kj}^R+iw_{kj}^I$ and the predicted output of the network be $\widehat{\mathbf{y}}=\widehat{\mathbf{y}}^R+i\widehat{\mathbf{y}}^I \in \mathbb{C}^C$. In this section, we will consider a single output neuron of FLFCN classifier and show that

[5] In this discussion, we drop the subscript k for convenience of representation.

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

the decision boundaries formed by the output neuron are orthogonal. The proof can then be easily extended to all the neurons in the output layer of FLFCN classifier.

The predicted output of the lth output neuron of the network ($\widehat{y}_l$) is given by

$$\widehat{y}_l = \sum_{k=1}^{K} w_{lk} h_k, \quad l = 1, \ldots, n, \tag{A.1}$$

where

$$h_k = h_k^R + i h_k^I = \mathrm{sech}(O_k) = \mathrm{sech}(O_k^R + i O_k^I), \tag{A.2}$$

$$h_k = \frac{2(\cos(O_k^I)\cosh(O_k^R) - i\sin(O_k^I)\sinh(O_k^R))}{\cos(2O_k^I) + \cosh(2O_k^R)}. \tag{A.3}$$

Using the laws of differentiation and the trigonometric and hyperbolic product of sum identities in Eq. (A.3), the following can be derived for the kth hidden neuron:

$$\frac{\partial h_k^R}{\partial O_k^R} = \frac{\sinh(O_k^R)\cos(3O_k^I) - \sinh(3O_k^R)\cos(O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}, \tag{A.4}$$

$$\frac{\partial h_k^I}{\partial O_k^I} = \frac{\sinh(O_k^R)\cos(3O_k^I) - \sinh(3O_k^R)\cos(O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}, \tag{A.5}$$

$$\frac{\partial h_k^R}{\partial O_k^I} = \frac{\cosh(O_k^R)\sin(3O_k^I) - \cosh(3O_k^R)\sin(O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}, \tag{A.6}$$

$$\frac{\partial h_k^I}{\partial O_k^R} = \frac{\cosh(3O_k^R)\sin(O_k^I) - \cosh(O_k^R)\sin(3O_k^I)}{(\cos(2O_k^I) + \cosh(2O_k^R))^2}. \tag{A.7}$$

From Eqs. (A.4)–(A.7), it can be observed that

$$\frac{\partial h_k^R}{\partial O_k^R} = \frac{\partial h_k^I}{\partial O_k^I} \tag{A.8}$$

and

$$\frac{\partial h_k^R}{\partial O_k^I} = -\frac{\partial h_k^I}{\partial O_k^R}. \tag{A.9}$$

Hence, the sech activation function satisfies the Cauchy Riemann equations.

Eq. (A.1) can be written as

$$\widehat{y}_l = \widehat{y}_l^R + i\widehat{y}_l^I = \sum_{k=1}^{K} ((w_{lk}^R \cdot h_k^R - w_{lk}^I \cdot h_k^I) + i(w_{lk}^R \cdot h_k^I + w_{lk}^I \cdot h_k^R)). \tag{A.10}$$

It can be observed from Eq. (A.10) that

$$\widehat{y}_l^R = \sum_{k=1}^{K} (w_{lk}^R \cdot h_k^R - w_{lk}^I \cdot h_k^I) \rightarrow \mathbf{C}^R \tag{A.11}$$

and

$$\widehat{y}_l^I = \sum_{k=1}^{K} (w_{lk}^R \cdot h_k^I + w_{lk}^I \cdot h_k^R) \rightarrow \mathbf{C}^I \tag{A.12}$$

are the decision boundaries for the real and imaginary parts of the output, respectively, and are constants for a given set of inputs and network parameters. Here, '→' means 'which is'.

The normal vector $Q^R(\mathbf{z}^R, \mathbf{z}^I)$ to the real-part of the decision boundary ($\mathbf{C}^R$) and the normal vector $Q^I(\mathbf{z}^R, \mathbf{z}^I)$ to the imaginary-part of the decision boundary ($\mathbf{C}^I$) are

$$Q^R(\mathbf{z}^R, \mathbf{z}^I) = \left(\frac{\partial \widehat{y}_l^R}{\partial z_1^R} \cdots \frac{\partial \widehat{y}_l^R}{\partial z_j^R} \cdots \frac{\partial \widehat{y}_l^R}{\partial z_m^R} \frac{\partial \widehat{y}_l^R}{\partial z_1^I} \cdots \frac{\partial \widehat{y}_l^R}{\partial z_j^I} \cdots \frac{\partial \widehat{y}_l^R}{\partial z_m^I} \right) \tag{A.13}$$

and

$$Q^I(\mathbf{z}^R, \mathbf{z}^I) = \left(\frac{\partial \widehat{y}_l^I}{\partial z_1^R} \cdots \frac{\partial \widehat{y}_l^I}{\partial z_j^R} \cdots \frac{\partial \widehat{y}_l^I}{\partial z_m^R} \frac{\partial \widehat{y}_l^I}{\partial z_1^I} \cdots \frac{\partial \widehat{y}_l^I}{\partial z_j^I} \cdots \frac{\partial \widehat{y}_l^I}{\partial z_m^I} \right). \tag{A.14}$$

The normal vectors of the decision boundaries are orthogonal to each other *iff* their dot product is zero. The dot product is

$$Q^R(\mathbf{z}^R, \mathbf{z}^I) \cdot Q^I(\mathbf{z}^R, \mathbf{z}^I) = \sum_{j=1}^{m} \left(\frac{\partial \widehat{y}_l^R}{\partial z_j^R} \cdot \frac{\partial \widehat{y}_l^I}{\partial z_j^R} + \frac{\partial \widehat{y}_l^R}{\partial z_j^I} \cdot \frac{\partial \widehat{y}_l^I}{\partial z_j^I} \right). \tag{A.15}$$

From Eqs. (A.11) and (A.12), it can be seen that

$$\frac{\partial \widehat{y}_l^R}{\partial h_k^R} = w_{lk}^R, \quad \frac{\partial \widehat{y}_l^I}{\partial h_k^R} = w_{lk}^I, \quad \frac{\partial \widehat{y}_l^R}{\partial h_k^I} = -w_{lk}^I \quad \text{and} \quad \frac{\partial \widehat{y}_l^I}{\partial h_k^I} = w_{lk}^R. \tag{A.16}$$

Moreover, the input to the neurons in the hidden layer is given by

$$\begin{aligned} O_k &= O_k^R + i O_k^I = (\mathbf{u}_k^R + i\mathbf{u}_k^I)^T ((\mathbf{z}^R - \mathbf{v}_k^R) + i(\mathbf{z}^I - \mathbf{v}_k^I)) \\ &= \sum_{j=1}^{m} (u_{kj}^R + i u_{kj}^I)((z_j^R - v_{kj}^R) + i(z_j^I - v_{kj}^I)) \\ &= \sum_{j=1}^{m} [u_{kj}^R (z_j^R - v_{kj}^R) - u_{kj}^I (z_j^I - v_{kj}^I) + i(u_{kj}^R (z_j^I - v_{kj}^I) + u_{kj}^I (z_j^R - v_k^R))]. \end{aligned} \tag{A.17}$$

From Eq. (A.17), the following can be deduced:

$$\frac{\partial O_k^R}{\partial z_j^R} = u_{kj}^R, \quad \frac{\partial O_k^R}{\partial z_j^I} = -u_{kj}^I, \quad \frac{\partial O_k^I}{\partial z_j^R} = u_{kj}^I, \quad \text{and} \quad \frac{\partial O_k^I}{\partial z_j^I} = u_{kj}^R. \tag{A.18}$$

Using the expressions in Eqs. (A.16) and (A.18) and the Cauchy Riemann equations shown in Eqs. (A.8) and (A.9), the following can be derived:

$$\begin{aligned} \frac{\partial \widehat{y}_l^R}{\partial z_j^R} &= \frac{\partial \widehat{y}_l^R}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^R} + \frac{\partial \widehat{y}_l^R}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^R} + \frac{\partial \widehat{y}_l^R}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^R} + \frac{\partial \widehat{y}_l^R}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^R} \\ &= w_{lk}^R \left(\frac{\partial h_k^R}{\partial O_k^R} u_{kj}^R + \frac{\partial h_k^R}{\partial O_k^I} u_{kj}^I \right) + w_{lk}^I \left(\frac{\partial h_k^I}{\partial O_k^R} u_{kj}^R + \frac{\partial h_k^I}{\partial O_k^I} u_{kj}^I \right) \\ &= w_{lk}^R \left(\frac{\partial h_k^R}{\partial O_k^R} u_{kj}^R + \frac{\partial h_k^R}{\partial O_k^I} u_{kj}^I \right) + w_{lk}^I \left(-\frac{\partial h_k^R}{\partial O_k^I} u_{kj}^R + \frac{\partial h_k^I}{\partial O_k^I} u_{kj}^I \right), \end{aligned} \tag{A.19}$$

$$\begin{aligned} \frac{\partial \widehat{y}_l^R}{\partial z_j^I} &= \frac{\partial \widehat{y}_l^R}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^I} + \frac{\partial \widehat{y}_l^R}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^I} + \frac{\partial \widehat{y}_l^R}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^I} + \frac{\partial \widehat{y}_l^R}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^I} \\ &= w_{lk}^R \left(\frac{\partial h_k^R}{\partial O_k^R}(-u_{kj}^I) + \frac{\partial h_k^R}{\partial O_k^I} u_{kj}^R \right) - w_{lk}^I \left(\frac{\partial h_k^R}{\partial O_k^I} u_{kj}^I + \frac{\partial h_k^R}{\partial O_k^R} u_{kj}^R \right), \end{aligned} \tag{A.20}$$

$$\begin{aligned} \frac{\partial \widehat{y}_l^I}{\partial z_j^R} &= \frac{\partial \widehat{y}_l^I}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^R} + \frac{\partial \widehat{y}_l^I}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^R} + \frac{\partial \widehat{y}_l^I}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^R} + \frac{\partial \widehat{y}_l^I}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^R} \\ &= w_{lk}^R \left(-\frac{\partial h_k^R}{\partial O_k^I}(u_{kj}^R) + \frac{\partial h_k^R}{\partial O_k^R} u_{kj}^I \right) + w_{lk}^I \left(\frac{\partial h_k^R}{\partial O_k^R} u_{kj}^R + \frac{\partial h_k^R}{\partial O_k^I} u_{kj}^I \right), \end{aligned} \tag{A.21}$$

$$\begin{aligned} \frac{\partial \widehat{y}_l^I}{\partial z_j^I} &= \frac{\partial \widehat{y}_l^I}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^I} + \frac{\partial \widehat{y}_l^I}{\partial h_k^R}\frac{\partial h_k^R}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^I} + \frac{\partial \widehat{y}_l^I}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^R}\frac{\partial O_k^R}{\partial z_j^I} + \frac{\partial \widehat{y}_l^I}{\partial h_k^I}\frac{\partial h_k^I}{\partial O_k^I}\frac{\partial O_k^I}{\partial z_j^I} \\ &= w_{lk}^R \left(\frac{\partial h_k^R}{\partial O_k^I}(u_{kj}^I) + \frac{\partial h_k^R}{\partial O_k^R} u_{kj}^R \right) + w_{lk}^I \left(-\frac{\partial h_k^R}{\partial O_k^R} u_{kj}^I + \frac{\partial h_k^R}{\partial O_k^I} u_{kj}^R \right). \end{aligned} \tag{A.22}$$

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

Substituting Eqs. (A.19)–(A.22) into Eqs. (A.15), we get

$$\sum_{j=1}^{m}\left(\begin{array}{l}
w_{lk}^{R^2}\left[-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^R}u_{kj}^{R^2}+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^{I^2}+\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^Ru_{kj}^I-\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^Ru_{kj}^I\right]\\
+w_{lk}^Rw_{lk}^I\left[\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^{R^2}-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I+\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^{I^2}\right]\\
+w_{lk}^Rw_{lk}^I\left[\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^{R^2}+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I+\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^{I^2}\right]\\
+w_{lk}^{I^2}\left[-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^{R^2}-\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^Ru_{kj}^I+\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^Ru_{kj}^I+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^{I^2}\right]\\
+w_{lk}^{R^2}\left[-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^{I^2}+\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^Ru_{kj}^I-\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^Ru_{kj}^I+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^{R^2}\right]\\
+w_{lk}^Rw_{lk}^I\left[-\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^{I^2}+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I+\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I-\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^{R^2}\right]\\
+w_{lk}^Rw_{lk}^I\left[-\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^{I^2}-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I-\frac{\partial h_k^R}{\partial O_k^R}\frac{\partial h_k^R}{\partial O_k^I}u_{kj}^Ru_{kj}^I-\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^{R^2}\right]\\
+w_{lk}^{I^2}\left[-\frac{\partial h_k^R}{\partial O_k^I}\frac{\partial h_k^R}{\partial O_k^R}u_{kj}^{I^2}-\left(\frac{\partial h_k^R}{\partial O_k^R}\right)^2u_{kj}^Ru_{kj}^I+\left(\frac{\partial h_k^R}{\partial O_k^I}\right)^2u_{kj}^Ru_{kj}^I+\frac{\partial h_k^R}{\partial O_k^I}\frac{\partial h_k^R}{\partial O_k^R}u_{kj}^{R^2}\right]
\end{array}\right)=0.$$

Hence

$$Q^R(\mathbf{z}^R,\mathbf{z}^I)\cdot Q^I(\mathbf{z}^R,\mathbf{z}^I)=0. \tag{A.23}$$

Based on the above result, we can state that the decision boundaries formed by the real and imaginary parts of an output neuron in FLFCN network with any fully complex-valued activation function that satisfies the Cauchy Riemann conditions are orthogonal to each other.

Appendix B. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org.10.1016/j.neucom.2012.03.003.

References

[1] R.V. Babu, S. Suresh, Fully complex-valued ELM classifiers for human action recognition, in: International Joint Conference on Neural Networks, 2011, pp. 2803–2808.
[2] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: Proceedings of International Conference on Pattern Recognition, 2004, pp. 32–36.
[3] R.V. Babu, B. Anantharaman, K.R. Ramakrishnan, S.H. Srinivasan, Compressed domain action classification using HMM, Pattern Recognition Lett. 23 (10) (2002) 1203–1213.
[4] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial–temporal words, Int. J. Comput. Vision 79 (3) (2008) 299–318.
[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space–time shapes, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2247–2253.
[6] K. Guo, P. Ishwar, J. Konrad, Action recognition from video by covariance matching of silhouette tunnels, in: Proceedings of Brazilian Symposium on Computer Graphics and Image Processing, 2009, pp. 299–306.
[7] A.F. Bobick, J.W. Davis, The recognition of human movement using temporal templates, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 257–267.
[8] M.-K. Hu, Visual pattern recognition by moment invariants, IEEE Trans. Inf. Theory 8 (2) (1962) 179–187.
[9] R.V. Babu, K.R. Ramakrishnan, Recognition of human actions using motion history information extracted from the compressed video, Image Vision Comput. 22 (8) (2004) 597–607.
[10] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time sequential images using hidden Markov model, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 1992, pp. 379–385.
[11] A.S. Ogale, A. Karapurkar, Y. Aloimonos, View-invariant modeling and recognition of human actions using grammars, in: Revised Papers of the Workshops on Dynamical Vision (WDV05 and WDV06), Lecture Notes in Computer Science, 2007, pp. 115–126.
[12] A. Yilmaz, M. Shah, A differential geometric approach to representing the human actions, Comput. Vision Image Understanding 119 (3) (2008) 335–351.
[13] F. Lv, R. Nevatia, Single view human action recognition using key pose matching and Viterbi path searching, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[14] Z. Zhang, Y. Hu, S. Chan, L.-T. Chia, Motion context: a new representation for human action recognition, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 817–829.
[15] K. Guo, P. Ishwar, J. Konrad, Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels, in: Proceedings of International Conference on Pattern Recognition (Semantic Description of Human Activities Contest), 2010.
[16] L. Shao, D. Wu, X. Chen, Action recognition using correlogram of body poses and spectral regression, in: IEEE International Conference on Image Processing (ICIP), 2011.
[17] V. Kellokumpu, G. Zhao, M. Pietikinen, Human activity recognition using a dynamic texture based method, in: Proceedings of the British Machine Vision Conference, 2008, pp. 885–894.
[18] C. Thurau, V. Hlavc, Pose primitive based human action recognition in videos or still images, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–6.
[19] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[20] W.-L. Lu, J.J. Little, Simultaneous tracking and action recognition using the PCAHOG descriptor, in: Proceedings of the Canadian Conference on Computer and Robot Vision, 2006, pp. 1–6.
[21] A.A. Efros, A.C. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: Proceedings of the International Conference on Computer Vision, vol. 2, 2003, pp. 726–733.
[22] S. Danafar, N. Gheissari, Action recognition for surveillance applications using optic flow and SVM, in: Proceedings of the Asian Conference on Computer Vision, 2007, pp. 457–466.
[23] S. Ali, M. Shah, Human action recognition in videos using kinematic features and multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2) (2010) 288–303.
[24] I. Laptev, On space–time interest points, Int. J. Comput. Vision 64 (2/3) (2005) 107–123.
[25] C. Harris, M. Stephens, A combined corner and edge detector, in: Proceedings of the Alvey Vision Conference, 1988, pp. 147–151.
[26] R. Mattivi, L. Shao, Human action recognition using LBP-TOP as sparse spatio-temporal feature descriptor, in: International Conference on Computer Analysis of Images and Patterns, vol. 5702, 2009, pp. 740–747.
[27] L. Shao, R. Mattivi, Feature detector and descriptor evaluation in human action recognition, in: ACM International Conference on Image and Video Retrieval (CIVR), 2010.
[28] R. Poppe, A survey on vision-based human action recognition, Int. J. Comput. Vision 28 (2/3) (2010) 976–990.
[29] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding (2011) 224–241.
[30] C. Schuldt, L. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: IEEE International Conference on Pattern Recognition, Vol. 3, 2004, pp. 32–36.
[31] R. Remmert, Theory of Complex Functions, Springer Verlag, New York, USA, 1991.
[32] T. Kim, T. Adali, Fully complex multi-layer perceptron network for nonlinear signal processing, J. VLSI Signal Process. 32 (1/2) (2002) 29–43.
[33] R. Savitha, S. Suresh, N. Sundararajan, A fully complex-valued radial basis function network and its learning algorithm, International Journal of Neural Systems 19 (4) (2009) 253–267.
[34] R. Savitha, S. Suresh, N. Sundararajan, P. Saratchandran, A new learning algorithm with logarithmic performance index for complex-valued neural networks, Neurocomputing 72 (16–18) (2009) 3771–3781.
[35] T. Starner, A. Pentland, Visual recognition of american sign language using hidden Markov models, in: International Conference on Automatic Face and Gesture Recognition, 1995, pp. 189–194.
[36] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proceedings of DARPA Image Understanding Workshop, 1981, pp. 121–130.
[37] T. Nitta, The computational power of complex-valued neuron, in: Artificial Neural Networks and Neural Information Processing ICANN/ICONIP, Istanbul, Turkey, Lecture Notes in Computer Science, 2003, pp. 993–1000.
[38] T. Nitta, Orthogonality of decision boundaries of complex-valued neural networks, Neural Comput. 16 (1) (2004) 73–97.
[39] M.B. Li, G.B. Huang, P. Saratchandran, N. Sundararajan, Fully complex extreme learning machines, Neurocomputing 68 (1–4) (2005) 306–314.
[40] R. Savitha, S. Suresh, N. Sundararajan, H.J. Kim, A fully complex-valued radial basis function classifier for real-valued classification problems, Neurocomputing 78 (1) (2012) 104–110.
[41] R. Savith, S. Suresh, N. Sundararajan, A meta-cognitive learning algorithm for a fully complex-valued radial basis function network, Neural Comput. 24 (5) (2012) 1297–1328.
[42] S. Suresh, R. Savitha, N. Sundararajan, A sequential learning algorithm for complex-valued self-regulating resource allocation network – CSRAN, IEEE Trans. Neural Networks 22 (7) (2011) 1061–1072.

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003

[43] S. Suresh, R. Savitha, N. Sundararajan, A fast learning fully complex-valued relaxation network (FCRN), in: IEEE International Joint Conference on Neural Networks, 2011 (IJCNN 2011), 2011, pp. 1372–1377.
[44] R. Savitha, S. Suresh, N. Sundararajan, Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems, Inf. Sci. 187 (1) (2012) 277–290.
[45] R. Savitha, S. Suresh, N. Sundararajan, A fast learning complex-valued neural classifier for real-valued classification problems, in: International Joint Conference on Neural Networks (IJCNN 2011), 2011, pp. 2243–2249.
[46] R. Savitha, S. Suresh, N. Sundararajan, H.J. Kim, Fast learning fully complex-valued classifiers for real-valued classification problems, in: D. Liu, et al. (Ed.), ISNN 2011, Part I, Lecture Notes in Computer Science (LNCS), vol. 6675, 2011, pp. 602–609.
[47] S. Suresh, S.N. Omkar, V. Mani, T.N.G. Prakash, Lift coefficient prediction at high angle of attack using recurrent neural network, Aerosp. Sci. Technol. 7 (8) (2003) 595–602.
[48] Y. Wang, G. Mori, Human action recognition by semilatent topic models, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1762–1774.
[49] I. Laptev, M. Marszaek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[50] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: British Machine Vision Conference, 2009, p. 127.

R. Venkatesh Babu completed his Ph.D. in 2003, from the Department of Electrical Engineering, Indian Institute of Science, Bangalore, India. He held post-doctoral positions at NTNU, Norway and IRISA/INRIA, Rennes, France, through ERCIM fellowship. Subsequently, he worked as a research fellow at NTU, Singapore. He spent couple of years working in industry. Currently, he is working as an Assistant Professor at Supercomputer Education and Research Centre (SERC), Indian Institute of Science, Bangalore, India. His research interests include video analytics, human–computer interaction, computer vision, compressed domain video processing.

S. Suresh received the B.E. degree in Electrical and Electronics Engineering from Bharathiyar University in 1999, and M.E. (2001) and Ph.D. (2005) degrees in Aerospace Engineering from Indian Institute of Science, India. He was a post-doctoral researcher in School of Electrical Engineering, Nanyang Technological University from 2005 to 2007. From 2007 to 2008, he was in INRIA Sophia Antipolis, France as an ERCIM research fellow. He was in Korea University for a short period as a visiting faculty in Industrial Engineering. From January 2009 to December 2009, he was in Indian Institute of Technology – Delhi as an Assistant Professor in Department of Electrical Engineering. Currently, he is working as an Assistant Professor in School of Computer Engineering, Nanyang Technological University, Singapore since 2010. His research interest includes flight control, unmanned aerial vehicle design, machine learning, optimization and computer vision.

R. Savitha received the B.E. degree in Electrical and Electronics Engineering from Manonmaniam Sundaranar University in 2000, and M.E. degree in Control and Instrumentation Engineering from College of Engineering Guindy, Anna University, India in 2002. She received the Ph.D. degree in 2010 from Nanyang Technological University, Singapore. She is currently a post-doctoral research fellow at Nanyang Technological University, Singapore. Her research interests are neural networks and control.

Please cite this article as: R. Venkatesh Babu, et al., Human action recognition using a fast learning fully complex-valued classifier, Neurocomputing (2012), http://dx.doi.org/10.1016/j.neucom.2012.03.003