

Human Action Recognition in Compressed Domain using PBL-McRBFN Approach

Badrinarayanan Rangarajan
School of Computer Engineering
Nanyang Technological University, Singapore
Email: badrinarayanan@ntu.edu.sg

Venkatesh Babu Radhakrishnan
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore
Email: venky@serc.iisc.ernet.in

Abstract—Large variations in human actions lead to major challenges in computer vision research. Several algorithms are designed to solve the challenges. Algorithms that stand apart, help in solving the challenge in addition to performing faster and efficient manner. In this paper, we propose a human cognition inspired projection based learning for person-independent human action recognition in the H.264/AVC compressed domain and demonstrate a PBL-McRBFN based approach to help take the machine learning algorithms to the next level. Here, we use gradient image based feature extraction process where the motion vectors and quantization parameters are extracted and these are studied temporally to form several Group of Pictures (GoP). The GoP is then considered individually for two different bench mark data sets and the results are classified using person independent human action recognition. The functional relationship is studied using Projection Based Learning algorithm of the Meta-cognitive Radial Basis Function Network (PBL-McRBFN) which has a cognitive and meta-cognitive component. The cognitive component is a radial basis function network while the Meta-Cognitive Component(MCC) employs self regulation. The McC emulates human cognition like learning to achieve better performance. Performance of the proposed approach can handle sparse information in compressed video domain and provides more accuracy than other pixel domain counterparts. Performance of the feature extraction process achieved more than 90% accuracy using the PBL-McRBFN which catalyzes the speed of the proposed high speed action recognition algorithm. We have conducted twenty random trials to find the performance in GoP. The results are also compared with other well known classifiers in machine learning literature.

Index Terms—H.264/AVC, Human action recognition, Compressed domain video analysis, Motion vectors, Quantization parameters, Meta-cognitive learning, radial basis function network, PBL-McRBFN

I. INTRODUCTION

Computer Vision is one of the most widely researched areas in which Human action recognition plays a vital role. Everyday researchers encounter various problems and come up with innovative solutions to tackle scale, appearance, illumination, orientation variations, and occlusions. Speed of recognition has not been given ample thrust in the evaluations. The sole aim of this paper is to take action recognition to a whole new level, where recognition rate too is considered as a benchmark evaluation criteria. Monitoring at higher resolution, low bandwidth usage, reduced storage requirements, faster frame rates, and better video quality makes H.264/AVC [1] the ideal video standard for real-time applications. In a typical video transmission

system, raw video input from the camera is compressed and streamed across a network targeting a decoder to reconstruct the source video. Bandwidth is the limiting factor over which bit-rate adjustment has an upper hand. Motion Vectors (MV) are the indication of how motion was compensated while exploiting the temporal frame redundancy during encoding. Quantization Parameter (QP) has a major role in regulating the bit rate during the encoding process. QP values can be adjusted to maintain a constant bit rate within the allowed channel bandwidth. Hence, real-time encoders heavily depend on varying the QP values to control the trade-off between video quality and compression. In our study, we consider three types of features. The first two are Quantization-parameter Gradient Image (QGI) spread and QGI projection profile. The third feature, histogram of magnitude-weighted orientation of MVs, obtained at sub-macro block resolution (4 x 4) provides important cues regarding the dynamics of the action. The task is to find the functional relationship between the features and action labels. Recently, Meta-Cognition based learning has been claiming its position among the other algorithms. Meta-cognitive Radial Basis Function Network (McRBFN) emulates the Nelson and Narens model of human meta-cognition, and has 2 components, namely, a cognitive component and a meta-cognitive component. Projection Based Learning algorithm of the Meta-cognitive Radial Basis Function Network (PBL-McRBFN) has a radial basis function network with Gaussian activation function at the hidden layer which is the cognitive component of McRBFN and a self-regulatory learning mechanism which is its meta-cognitive component. McRBFN begins with zero hidden neurons, adds and prunes neurons until an optimum network structure is obtained. The self-regulatory learning mechanism of McRBFN uses the best human learning strategy, namely, self-regulated learning to decide what-to-learn, when-to-learn and how-to-learn in a meta-cognitive framework. The meta-cognitive learning algorithm has been implemented in neuro-fuzzy [1]–[3], complex-valued [4], [5] and neural networks [6]. Among these implementations, PBL-McRBFN [7] is computationally efficient and provides better generalization performance. Hence, we employ PBL-McRBFN to approximate the functional relationship between the proposed features and action labels. The results show a superior performance of PBL-McRBFN in recognizing human actions. Recently it has been shown in neural network literature that

learning algorithms employing meta-cognition (what-to-learn, when-to-learn and how-to-learn) provides better generalization performance. The paper is structured in the following way. Related works on activity recognition in pixel and compressed domains are presented in section 2. In section 3 we discuss in detail the proposed approach. Section 5 presents the experimental results and discussions. Concluding remarks and future works are stated in section 6.

A. Related Work

A lot of research has been reported till date in recognizing human actions in pixel domain. However, none of the algorithms in pixel domain have claimed recognition speed above 25 frames per second (fps). Applications demanding low memory utilization and faster speeds cannot employ those techniques because of the higher computational load. Pixel domain approaches [8], [9], [10], [11], [12], [13], [14] offer better classification at the cost of higher computational complexity and hence very few have achieved real-time recognition. Sadek et al. [15] uses a finite set of features directly derived from difference frames of the action snippet for activity recognition which works at 18fps. Yu et al. [16] presented action recognition using semantic texton forests which work on video pixels generating visual codewords. The algorithm works at 10 to 20 fps. Li et al. [17] perform a luminance field manifold trajectory analysis based solution for human activity recognition which gives results in real time.

Compressed domain approaches on the other hand are more feasible for real-time applications owing to their faster performance. But, very less amount of research has been done in recognizing human actions in compressed videos. Ozer, Wolf and Akansu [18] proposed a hierarchical approach for human activity detection and recognition in MPEG compressed sequences. Body parts were segmented out and Principal Component Analysis was performed on the segmented motion vectors prior to classification. Another notable work was put forward by Babu et al. [19] in MPEG compressed domain. The features extracted from motion vectors were fed to Hidden Markov Model (HMM) for classification. Babu et al. [20] later proposed a method using motion flow history and motion history image in MPEG compressed domain. Yeo et al. [21] proposed a high-speed algorithm for action recognition and localization in MPEG streams based on computing motion correlation measure by utilizing differences in motion direction and magnitudes. However, the algorithm cannot handle scale variations. The proposed method, on the other hand is robust to scale changes and attains slightly higher accuracy than [21]. Their method calculated optical flow from the MVs. After the formation of optical flow, the approach is equivalent to any pixel-based algorithm and hence is not a pure compressed domain approach. Manu et al. proposed a rapid human action recognition approach in H.264 [22] compressed domain using motion vectors and quantization parameters with SVM classifier [23].

In this paper, we utilize the features proposed in [23] and learn the functional relationship using Projection Based Learn-

ing algorithm of the Meta-cognitive Radial Basis Function Network (PBLMcRBFN) which has a cognitive and meta-cognitive component. The proposed framework can handle sparse information in compressed video domain and performs rapid action recognition with comparable recognition rate to other pixel domain counterparts.

B. QGI Feature Extraction

Here, the QGI based features are extracted and the corresponding projection profiles are created. First, the QP delta (difference in QP values of adjacent macroblocks) and MV values are extracted by partial decoding of the input H.264 bit-stream. QGI is then formed using the QP delta values which is then utilized to generate the QGI projection profile and spread features. Also, motion accumulated projection profile and histogram of magnitude-weighted orientation of motion vectors are formed as features using the extracted MVs. All the individual features, after weighting, are combined to form the final feature vector.

C. Action Representation

In this work, the action is represented in the following three ways: (i) *QP Gradient Image*, (ii) *Motion Accumulation* and (iii) *Magnitude-Weighted-Orientation of Motion Vectors*. Features extracted from these representations are used for classifying the actions.

Quantization Parameter Gradient Image (QGI) : H.264 allows to vary the value of QP on a macroblock (MB) basis to achieve better compression. Encoder maintains fixed QP values for MBs with low information and vice versa. Adjacent MBs in raster-scan order will have very small change in QP values, if they are part of the static background. Higher the delta value, higher is the edge information, which is a direct measure of action occurrence in the respective MB pair. The MBs present at the action area will have non zero QP-delta and will be classified as action-MBs. Spurious MBs with no action information can be removed by connected component analysis. This step is continued for all P frames and provides initial action segmentation with MB level precision. To incorporate temporal content, we propose a weighted accumulation of the filtered action-MBs over time. We define forward (F_a) and backward accumulation factors (B_a) as the number of frames accumulated in forward and backward temporal directions respectively. These parameters can be selected empirically and we fixed F_a and B_a as 2 frames each. Even in real-time applications, buffering of frames is possible which validates the credibility of forward accumulation. The action-edge accumulated output for current frame $P_{acc}(T)$ is given by:

$$P_{acc}(T) = \sum_{k=0}^{B_a} w(T-k)P(T-k) + \sum_{k=1}^{F_a} w(T+k)P(T+k) \quad (1)$$

where $P(T-k)$ and $P(T+k)$ denotes the output of initial filtering stage for a frame which is k -frames farther from the current frame $P(T)$ in reverse and forward temporal

directions respectively. Similarly, $w(T - k)$ and $w(T + k)$ denotes the respective weights assigned which are linearly distributed across past and future frames. To generate the final QGI for a bunch of frames, the action-edge MBs of each frame are again accumulated temporally. Specifically, for a k -frame bunch starting from frame t , the QGI is defined as:

$$QGI(t, k) = \sum_{j=0}^{k-1} P_{acc}(t + j) \quad (2)$$

QGI located at $[l, m]$ can be represented as $QGI(t, k, l, m)$.

Motion Accumulation : The magnitude and orientation of motion vector of a sub-macroblock (4x4 block in H.264/AVC) and location $[l, m]$ in frame t and with horizontal and vertical components u and v respectively are given by :

$$M_t(l, m) = \sqrt{u^2 + v^2} \quad (3)$$

$$A_t(l, m) = atan(v/abs(u)) \quad (4)$$

where, $atan(x)$ returns the inverse tangent (arctangent) of x in radians. For real elements of x , $atan(x)$ is in the range $[-\pi/2, \pi/2]$. We employed the absolute value of the horizontal component of motion vector to attain left-right symmetry of actions. The magnitude of motion vector of each sub-macroblock is accumulated temporally over k frames as given by:

$$MA_t(l, m) = \sum_{j=0}^{k-1} M_{t+j}(l, m) \quad (5)$$

The accumulated motion is used to form the horizontal and vertical projection profiles (*I-D2*) to represent each action.

Magnitude-Weighted-Orientation of Motion Vectors : Magnitude-weighted orientation of a motion vector in frame t at location $[l, m]$ is given by:

$$A'_t(l, m) = A_t(l, m) \cdot M_t(l, m) \quad (6)$$

where, $' \cdot '$ indicates element-wise multiplication.

D. Feature Extraction

1) *QGI Features* : There are two different QGI based features extracted for our analysis namely QGI Spread Feature and QGI Projection Profile.

QGI Spread feature : It can be observed from the QGIs that the horizontal (HS) and vertical spread (VS) of action has clue regarding the same. The VS is more for actions like jack and pjump. In case of hand waving actions the spread is localized to the upper torso and hence the VS will be less. Actions wave1 and wave2 can be distinguished by HS, even though the VS are comparable. Actions like bend, walk and run have comparable VS. But, bend can be distinguished from run or walk by adding HS too as a feature. The spread feature can distinguish between actions like bend, jack, pjump, wave1 and wave2. But the confusion between skip, side, run, walk and jump remains as the spreads in both dimension are comparable. Spread is normalized with maximum of frame height or width (in MBs) before adding to the final feature vector.

QGI Projection Profile (QGIP) feature : The horizontal and vertical QGI projection profiles of each action-bunch of k frames starting from frame t , for a video with frame width w and frame height h (in pixels) is given by:

$$QGIP_{horiz}(t, k) = \sum_{j=1}^{w/16} QGI(t, k, l + j, m) \quad (7)$$

$$QGIP_{vert}(t, k) = \sum_{i=1}^{h/16} QGI(t, k, l, m + i) \quad (8)$$

2) *Motion Accumulated Projection Profile (MAP) Feature*: The motion-accumulated projection profiles are given by:

$$MAP_{horiz}(t, k) = \sum_{j=1}^{w/4} MA_t(l + j, m) \quad (9)$$

$$MAP_{vert}(t, k) = \sum_{i=1}^{h/4} MA_t(l, m + i) \quad (10)$$

Horizontal and vertical profiles, after sub-sampling and normalization, are concatenated to form the projection feature. In our experiments we took only one in four samples of each MAP profile to minimize the feature vector length.

3) *Histogram of Magnitude-Weighted Orientation of MVs* : Each action has a typical set of motion vectors with more frequency of occurrence, which may differ in magnitudes but are similar in orientations. For example, hand-waving actions will have a typical set of repeating MV orientations in the waving-area. To utilize this property, histogram can be used. If raw histograms are employed, actions like running and walking will get confused as the orientation of those actions will be more or less the same. In those cases, the speed of action need to be utilized . The magnitude of the MVs will be more for actions performed at higher speeds. Hence, the histogram of orientations weighted by magnitudes is used as a feature.

II. PBL-MCRBFN CLASSIFIER OVERVIEW

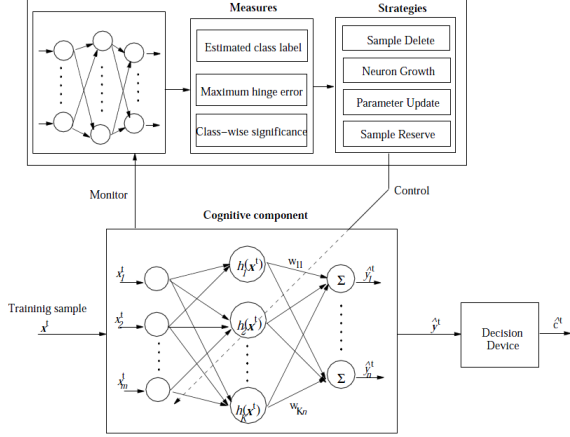
Consider a set of training data samples, where $\mathbf{x}^t \in \mathfrak{R}^m$ is an m -dimensional input of the t^{th} sample, and $c^t \in (1, n)$ is its class label, and n is the total number of classes. The objective of McRBFN is to approximate the function $\mathbf{x}^t \in \mathfrak{R}^m \rightarrow \mathbf{y}^t \in \mathfrak{R}^n$. McRBFN architecture is same as in [24] and has two components, namely the cognitive component and the meta-cognitive component as shown in Figure 1. We discuss these components in the following sections.

A. Cognitive component of McRBFN

The cognitive component of McRBFN is a radial basis function network. The hidden layer of the network holds the Gaussian activation function. For a given input \mathbf{x}^t , the predicted output \hat{y}_j^t is given as

$$\hat{y}_j^t = \sum_{k=1}^K w_{kj} h_k^t, \quad j = 1, \dots, n \quad (11)$$

Fig. 1. McRBFN Classifier
Meta-cognitive component



where w_{kj} is the weight connecting the k^{th} hidden neuron to the j^{th} output neuron and h_k^t is the response of the k^{th} hidden neuron to the input \mathbf{x}^t is given by

$$h_k^t = \exp\left(-\frac{\|\mathbf{x}^t - \mu v_k^l\|^2}{(\sigma_k^l)^2}\right) \quad (12)$$

where $\mu v_k^l \in \mathbb{R}^m$ is the center and $\sigma_k^l \in \mathbb{R}^+$ is the width of the k^{th} hidden neuron. Here, the superscript l represents the corresponding class of the hidden neuron.

Projection based learning algorithm helps to calculate the optimal network output parameters. For t consecutive samples, the error function is

$$J(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^t \sum_{j=1}^n \begin{cases} 0 & \text{if } y_j^i \hat{y}_j^i > 1 \\ (y_j^i - \hat{y}_j^i)^2 & \text{otherwise} \end{cases} \quad (13)$$

The resultant optimal output weights \mathbf{W}^* corresponding to the error function $J(\mathbf{W}^*)$ is represented as

$$\sum_{k=1}^K a_{kp} w_{kj} = b_{pj}, \quad (14)$$

which is a system of linear equation in the form of $\mathbf{W}^* = \mathbf{A}^{-1}\mathbf{B}$. Here, a_{kp} is the projection matrix and b_{pj} is the output matrix. The derivation of this equation can be found in [24].

B. Meta-cognitive component of McRBFN

The meta-cognitive component models the dynamics of the cognitive component, its corresponding knowledge measures and the self-regulated thresholds. During the learning process, the meta-cognitive component monitors the cognitive component and updates its dynamic model of the cognitive component. The meta-cognitive component uses predicted class label (\hat{c}^t), maximum hinge loss (E^t), confidence of classifier ($\hat{p}(c^t|\mathbf{x}^t)$) and class-wise significance (ψ_c) as the measure of knowledge in the new training sample. The meta-cognitive component builds two sample-based and two neuron-based learning strategies using the knowledge measures and the self-regulated thresholds. One of these strategies is chosen

for the new training sample such that the cognitive component learns them accurately and achieves better generalization performance.

The meta-cognitive component knowledge measures are defined as shown below:

Predicted class label (\hat{c}^t): $\hat{c}^t = \arg \max_{j \in 1, \dots, n} \hat{y}_j^t$

Class-wise Significance (ψ_c): The class-wise distribution portrays as the key impact factor in calculating the performance of the classifier. Let K^c be the number of neurons associated with the class c , then class-wise spherical potential or class-wise significance (ψ_c) is defined as

$$\psi_c = \frac{1}{K^c} \sum_{k=1}^{K^c} h(\mathbf{x}^t, \mu v_k^c) \quad (15)$$

C. Learning strategies

The meta-cognitive component has four learning strategies, which directly addresses the basic principles of self-regulated human learning namely, *what-to-learn*, *when-to-learn* and *how-to-learn* and selects the best strategy for the new training sample.

Sample delete strategy is given by

$$\hat{c}^t == c^t \text{ AND } \hat{p}(c^t|\mathbf{x}^t) \geq \beta_d \quad (16)$$

where β_d is the deletion threshold. If β_d is close to 1 then all samples participates in the learning which may result in over-training. Please refer [7] for further understanding on deletion thresholds.

Neuron growth strategy is given by

$$(\hat{c}^t \neq c^t \text{ OR } E^t \geq \beta_a) \text{ AND } \psi_c(\mathbf{x}^t) \leq \beta_c \quad (17)$$

where β_c is the knowledge measurement threshold and β_a is the self-adaptive addition threshold. If β_c is chosen closer to 0, then very few neurons will be added to the network. If β_c is closer to 1, then the resultant network may contain neurons with poor generalization ability. Based on the nearest neuron distances, the new hidden neuron center and width parameters are determined for the different overlapping/non-overlapping conditions as in [24].

When a neuron is added to McRBFN, the output weights are initialized as:

The size of matrix \mathbf{A} is increased from $K \times K$ to $(K+1) \times (K+1)$

$$\mathbf{A}^t = \left[\begin{array}{c|c} \mathbf{A}_{K \times K}^{t-1} + (\mathbf{h}^t)^T \mathbf{h}^t & \mathbf{a}_{K+1}^T \\ \hline \mathbf{a}_{K+1} & a_{K+1, K+1} \end{array} \right] \quad (18)$$

Please refer to the neuron growth architecture on [24] to further understand the output weights.

Parameters update strategy: The present (t^{th}) training sample helps to update the output weights of the cognitive component ($\mathbf{W}_K = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K]^T$) if the following condition is satisfied.

$$c^t == \hat{c}^t \text{ AND } E^t \geq \beta_u \quad (19)$$

where β_u is the self-adaptive update threshold. If β_u is closer

to 50% of E^t , then fewer samples will be used and most of the samples will be dragged to the end of the training sequence. The resultant network will not approximate the function accurately. If a lower value is chosen, then all samples will be used in updating without any change in the training sequence.

The $\mathbf{A} \in \mathbb{R}^{K \times K}$ and $\mathbf{B} \in \mathbb{R}^{K \times n}$ matrices updated as

$$\mathbf{A}^t = \mathbf{A}^{t-1} + (\mathbf{h}^t)^T \mathbf{h}^t; \mathbf{B}^t = \mathbf{B}^{t-1} + (\mathbf{h}^t)^T (\mathbf{y}^t)^T \quad (20)$$

Finally the output weights are updated as

$$\mathbf{W}_K^t = \mathbf{W}_K^{t-1} + (\mathbf{A}^t)^{-1} (\mathbf{h}^t)^T (\mathbf{e}^t)^T \quad (21)$$

where \mathbf{e}^t is hinge loss for t^{th} sample. **Sample reserve strategy:** the current sample is pushed to the rear end of data stream. If the new training sample does not satisfy either the sample deletion or the neuron growth or the cognitive component parameters update criterion. Since McRBFN modifies the strategies based on the knowledge in the current sample, these samples may be used in later stage. Next, we shall evaluate the action recognition ability of PBL-McRBFN on two standard benchmark datasets.

III. PERFORMANCE EVALUATION

In this paper, we evaluate the performance of PBL-McRBFN on Weizman [25] action recognition datasets. Even though huge datasets like UCF101 [26], HMDB51 [27] etc. are available, with more than 50 classes of actions each, the classification accuracies reported in those databases is very less, even in pixel domains. It will be even more difficult to classify the actions using the sparse compressed domain cues. Also, Weizman more to surveillance-type scenarios since camera motion is very less or even negligible.

For performance evaluation, group of pictures (GOP) based features are extracted (refer table I). The extracted features are input to PBL-McRBFN and the performance is recorded for 10 random trials by two leaving persons for testing. This is further randomized by creating twenty different trials for comparison. Table I provides the list of all the datasets that are used in our experiments. For comparison, In all the experiments, we consider SVM [28]. The parameters of these algorithms are optimized as explained in their respective references. Weka tool [29] has been used to evaluate these algorithms. Next, we shall describe the Weizmann dataset.

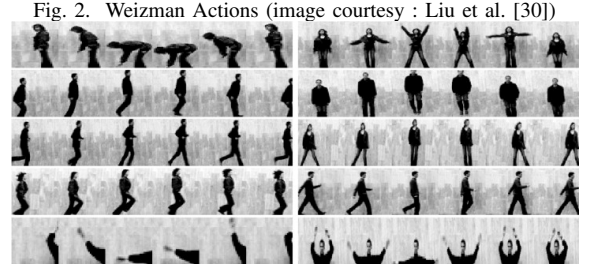
TABLE I
EXPERIMENTAL SETUP

Feature Extraction	Size
Group of Pictures	(20, 25, 30, 35, 40)

A. Weizmann Dataset

Weizmann dataset consists of 10 different actions viz., walk, run, jump, gallop sideways, bend, one-hand waving, two-hands waving, jump in place, jumping jack and skip. The actions are performed by 9 subjects. There are a total of 93 video

sequences. Five different groups of pictures are analyzed as shown in table I. Weizmann dataset has a total of 5532 frames. The classification speed achieved is 2186fps with a standard deviation of 7 frames (refer figure 2).



B. Effect of person independent feature extraction in Weizmann dataset

1) *GOP*: The results for GOPs of sizes (20, 25, 30, 35, 40) are provided in the table II. The table provides the performance evaluation of PBL-McRBFN with respect to the accuracy of best possible outcome and mean accuracy. Here the best possible outcome is the maximum accuracy attained in the corresponding experiment. The performance of PBL-McRBFN is compared against the standard SVM classifier. From the table, it could be observed that PBL-McRBFN performs significantly better when compared to SVM. In particular, the following can be observed:

TABLE II
PBL_McRBFN vs SVM

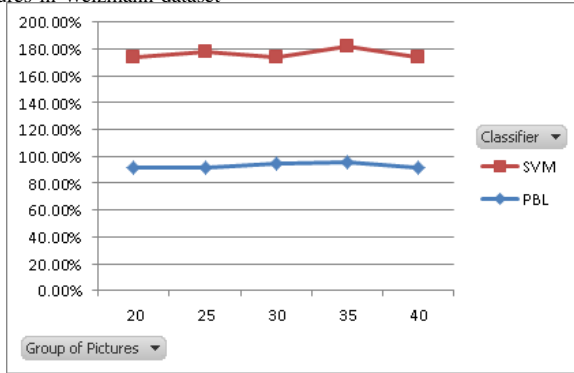
Group of Pictures	Classifier	Mean	Best Possible Outcome
20	SVM	73.91%	82.61%
20	PBL	90.06%	91.30%
25	SVM	79.13%	86.96%
25	PBL	89.56%	91.30%
30	SVM	76.60%	78.94%
30	PBL	90.06%	94.74%
35	SVM	75.45%	85.96%
35	PBL	91.30%	95.65%
40	SVM	73.91%	82.61%
40	PBL	87.58%	91.30%

- In all cases, the mean accuracy for SVM is close to 75% whereas for PBL-McRBFN, it is close to 90%.
- For GOP_35, PBL-McRBFN achieves 95%. In random validation, mean accuracy for SVM drops by 10% whereas PBL-McRBFN drops by 4%. PBL-McRBFN performs atleast 16% better than SVM.
- Increase in the size of GOP from 20 to 35, increases the overall performance of PBL-McRBFN.

Figure 3 shows the overall performance of PBL-McRBFN across all the group of pictures.

From the experiments carried out, it is evident that PBL-McRBFN performs better in comparison to SVM. The meta-cognitive learning strategies employed in McRBFN has helped the network avoid over-training and as a result, attain better generalization.

Fig. 3. Testing Performance of PBL-McRBFN taken across all the Group of Pictures in Weizmann dataset



IV. CONCLUSION AND FUTURE WORK

In this paper, we have tested the QGI based compressed domain features. The extracted QGI and MAP features are grouped in pictures of sizes 20, 25, 30, 35 and 40. These are fed on the state-of-the-art PBL-McRBFN classifier. The classification employs meta-cognition based sequential self-regulatory learning approach in which the meta-cognitive component is used to specify when to learn and what sample to learn. This generalization achieves a better performance in classification problems. In our analysis we have concluded that PBL-McRBFN outperforms SVM providing atleast 10% to 15% more accuracy than the latter. This work will be followed by classifying adaptive actions and occlusions.

REFERENCES

- [1] N. S. K. Subramanian, S. Suresh, "A meta-cognitive neuro-fuzzy inference system (mcfis) for sequential classification problems," in *IEEE Trans. on Fuzzy Systems*, 2013.
- [2] K. Subramanian and S. Suresh, "Human action recognition using meta-cognitive neuro-fuzzy inference system," *International Journal of Neural Systems*, vol. 22, no. 6, 2012.
- [3] —, "A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system, applied soft computing," *Applied soft computing*, vol. 12, no. 11, pp. 36703–3614, 2012.
- [4] R. Savitha, S. Suresh, and N. Sundararajan, "Metacognitive learning algorithm for a fully complex-valued relaxation network," *Neural Networks*, vol. 24, pp. 309–318, 2012.
- [5] —, "Meta-cognitive learning in a fully complex-valued radial basis function network," *Neural Computation*, vol. 24, no. 5, pp. 1297–1328, 2012.
- [6] R. Savitha, S. Suresh, and H. J. Kim, "A meta-cognitive learning algorithm for an extreme learning machine classifier," *Cognitive Computation*, July 2013.
- [7] G. S. Babu and S. Suresh, "Meta-cognitive rbf networks and its projection based learning algorithm for classification problems," *Applied Soft Computing*, vol. 13, no. 1, pp. 654–666, 2013.
- [8] C.-A. Lin, Y.-Y. Lin, H.-Y. M. Liao, and S.-K. Jeng, "Action recognition using instance-specific and class-consistent cues," in *International Conference on Image Processing, 2012*.
- [9] S. M. Amiri, P. Nasiopoulos, and V. C. Leung, "Non-negative sparse coding for human action recognition," in *International Conference on Image Processing, 2012*.

- [10] B. Wu, C. Yuan, and W. Hu, "Human action recognition based on a heat kernel structural descriptor," in *International Conference on Image Processing, 2012*.
- [11] X. Zhang, Z. Miao, and L. Wan, "Human action categories using motion descriptors," in *International Conference on Image Processing, 2012*.
- [12] Y. Wang and G. Mori, "Max-margin hidden conditional random fields for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 872–879.
- [13] R. V. Babu, S. Suresh, and R. Savitha, "Human action recognition using a fast learning fully complex-valued classifier," *Neurocomputing*, vol. 89, pp. 202–212, 2012.
- [14] R. V. Babu, R. Savitha, S. Suresh, and B. Agarwal, "Subject independent human action recognition using spatio-depth information and meta-cognitive RBF network," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 2010–2021, 2013.
- [15] B. M. S. Sadek, A. Al-Hamadi and U. Sayed, "A fast statistical approach for human activity recognition," *International Journal of Intelligence Science*, vol. 2 No. 1, pp. 9–15, 2012.
- [16] T.-H. Yu, T.-K. Kim, and R. Cipolla, "Real-time action recognition by spatiotemporal semantic and structural forests," in *BMVC, 2010*, pp. 1–12.
- [17] Z. Li, Y. Fu, T. Huang, and S. Yan, "Real-time human action recognition by luminance field trajectory analysis," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 671–676.
- [18] B. Ozer, W. Wolf, and A. Akansu, "Human activity detection in mpeg sequences," in *Human Motion, Proceedings. Workshop on*, 2000, pp. 61–66.
- [19] R. V. Babu, B. Anantharaman, K. R. Ramakrishnan, and S. H. Srinivasan, "Compressed domain action classification using hmm," *Pattern Recognition Letters*, vol. 23, pp. 1203–1213, 2002.
- [20] R. V. Babu and K. R. Ramakrishnan, "Recognition of human actions using motion history information extracted from the compressed video," *Image and Vision Computing*, vol. 22, no. 8, pp. 597–607, August 2004.
- [21] C. Yeo, P. Ahammad, K. Ramchandran, and S. Sastry, "High-speed action recognition and localization in compressed domain videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1006–1015, Aug. 2008.
- [22] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.
- [23] M. Tom and R. V. Babu, "Rapid human action recognition in H.264/AVC compressed domain for video surveillance," in *Visual Communications and Image Processing, 2013*, pp. 1–6.
- [24] G. Sateesh Babu, R. Savitha, and S. Suresh, "A Projection Based Learning in Meta-cognitive Radial Basis Function Network for Classification Problems," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 2907–2914.
- [25] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *The Tenth IEEE International Conference on Computer Vision*, 2005, pp. 1395–1402.
- [26] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.
- [27] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.
- [28] C.-C. Chang and C.-J. Lin, "Libsvm - a library for support vector machines," 2001, the Weka classifier works with version 2.82 of LIBSVM. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.
- [30] C. Liu and P. C. Yuen, "Human action recognition using boosted eigenactions," *Image and Vision Computing*, vol. 28, no. 5, pp. 825–835, 2010.