Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 746 – 755

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Video-Based Face Recognition and Face-Tracking using Sparse Representation Based Categorization

Shruthi Nagendra[a,*], R. Baskaran[a] and S. Abirami[b]

[a]Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai
[b]Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai

Abstract

Face recognition system is used in order to automatically identify a person from an image or a video source. The recognition task is performed by obtaining facial features from an image of the subject's face. The main objective of video-based face recognition is to identify a video face-track of famous personalities using a large dictionary of still face images, while rejecting unknown individuals Existing methods use probabilistic models on a frame-by-frame basis to identify faces which is computationally expensive when the data size is large. To overcome this drawback, the proposed regularized sparse representation classification (RSRC) algorithm uses ℓ^2 minimization approach instead of conventional ℓ^1 minimization method and obtains a single coefficient vector for all frames. Since second order minimization is used, more sparsity ratios are achieved and the residual error over the frames are reduced. The proposed algorithm is compared with the existing methods and the experimental results prove that, due to minimal error better classification accuracy and high confidence value are achieved.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Face recognition; Minimization approach; Residual error; ℓ^2 minimization approach and sparsity ratio.

1. Introduction

Face recognition has become an exploring domain in recent years due to increase the security demands and its law enforcement applications. It is used to automatic detection of a person with any images or video frame. It identifies the facial features by extracting features, from an image of the subject's face and analyses the relative position, size, shape of the eyes, nose, cheekbones and jaw of the detected person. These obtained features are utilized to search for corresponding matching features in other images. Facial recognition system spans across several applications such as biometric systems, immigration checking, e-voting, banking domain and in the gaming industry.

A typical approach in face recognition is video-content based searching[4]. A retrieval system should return all videos containing specific actors upon a user's request. For example, in YouTube, where a cast list may not be available, the visual content plays a major role in achieving this task successfully. But the main disadvantage is the availability of annotated video face tracks. Existing video face recognition methods is inclined to perform classification on a frame-by frame basis[8,9] and combine those predictions using an appropriate metric. Use of ℓ^1 minimization way in this fashion

*Corresponding author.
E-mail address: shruthi9223@gmail.com

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.088

is very computationally expensive. Other common issues in a face recognition system are illumination change between images, occlusions, pose variations and lighting conditions.

The current video based recognition algorithms use probabilistic models[12] or mutual subspace learning to track and identify faces. These models detect faces from a video and impose motion to obtain tracked faces. The obtained tracked faces are presented in a group. The sparsity ratio of the group is small, owing to absence of a single coefficient vector for all the tracked faces. Thus lower sparsity values lead to higher rate of misclassification. In this paper, skin tone is invoked as a feature to perform face tracking as it does not pose many variations due to illumination changes. To obtain higher sparsity and accurate prediction, the proposed method uses regularized SRC algorithm to calculate mean for each test track and performs a joint optimization over all faces on the track at once using ℓ^2 minimization approach.

2. Related Work

In recent years, many well-established methods have been suggested to solve face recognition problems in various domains. The current video face recognition techniques are classified into one of three categories: key frame based, temporal model based, and image-set matching based techniques.

Key frame based methods perform a prediction on the identity of each key-frame in a face track followed by a probabilistic fusion or majority voting to select the best match. Kaihua Zhang[3] uses a key-frame selection in a database with still images collected from the Internet. They learn a model over this dictionary by learning key faces via clustering. These cluster centres are compared to test frames using a nearest – neighbor search followed by majority, probabilistic voting to make a final prediction. In A. Yilmaz[10] the two-dimensional appearance of the face image is treated as a vector by scanning the image in lexicographical order, with the vector dimension being the number of pixels in the image. In the Eigen – face approach S. Li[7], all face images consist of a distinctive face subspace. This subspace is linear and spanned by the eigenvectors. But PCA does not achieve accuracy in terms of recognition since the creation of the face subspace does not significantly discriminate between humans.

The temporal model based methods learns the temporal, facial dynamics of the face throughout a video. In this method Z. Kal[11] performs Hidden Markov Models (HMM) which uses image training library by imposing motion information on it to train a HMM. It probabilistically generalizes a still – image library to do video-to-video matching. Training these models is computationally expensive, especially when the dataset size is large. S. Oron[8] uses all available information, to identify the cast rather than the facial information alone. It uses a manifold for known characters which successfully cluster input frames. In P. Viola and M. Jones[6] formulate tracking as a probability density propagation problem and the algorithm provides verification results. However, no systematic evaluation of recognition was done. The major challenge with this approach, is to find such region of interest.

Image-set matching based methods allows the modelling of a face track as an image set. C. Bao[1] uses LDML which performs a mutual subspace distance where each face track is modelled in their own subspace from which a distance is calculated between each. They are helpful with clean data, but the methods are very sensitive to the variations inherent in video face tracks. LDML is very computationally expensive and focuses more on learning relationships within the data, whereas we directly relate the test track to the training data. Ng[5] uses Sparse Representation-based Classification in which a given test image can be represented by a linear combination of images from a large dictionary of faces. The main idea is adding sparsity, since a test face track can be reconstructed from a subset of training faces of the same class. A straightforward adaptation of this method would perform estimation on each frame and combine the results probabilistically. However, ℓ^1 minimization is well known for be computationally expensive. A constrained optimization reduces the computation to a single ℓ^2 minimization over the average face track. To obtain this many color channels and binary patterns is reduced.

Thus by performing a survey on related concepts we conclude that, in key frame based methods single coefficient vector is not considered for modelling training and testing data. Most of the face recognition algorithms do not consider tracked faces as a group due to absence of correlation among the frames. If correlated frames were considered the prediction accuracy could have been improved. The main contribution of this paper is to use a single feature vector for obtaining the correlated vectors across frames and to reduce the prediction error during classification.

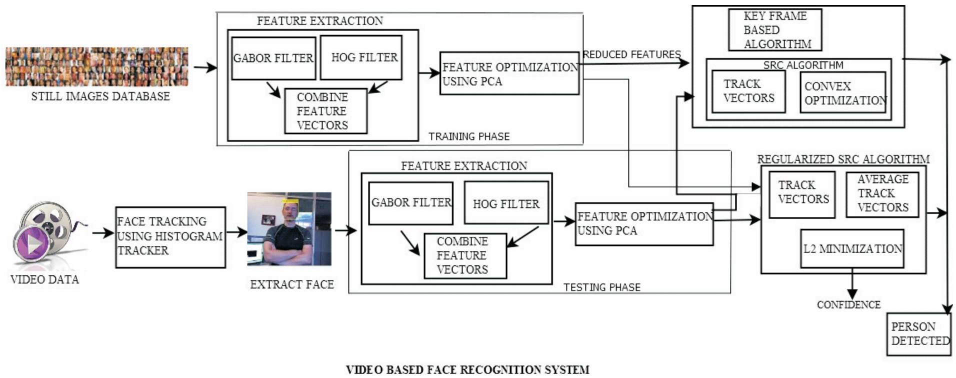

Fig. 1. Video-based face recognition system.

In summary, the aim of this paper is

- To identify a video face-track using a large dictionary of still face images.
- To perform face detection and face-tracking using histogram based tracker algorithm.
- To solve Regularized SRC algorithm using single coefficient vector to obtain more appropriate sparse solutions and higher classification accuracy.
- To detect known personalities and reject unknown individuals.

3. Video Based Face Recognition System

3.1 Face detection and tracking using histogram based tracker

In Fig. 1, a test video dataset is used to detect faces and track them across the frames. After face detection, it obtains facial features using hog and Gabor filters. The filters use edge detection and orientation values and extract, facial features from images.The obtained features are in large dimensions [32, 272 dimensions]. These dimensions are optimized to 1536 dimensions for each feature vector using PCA. The optimized features are given as a test input to the algorithm. The training features are available from a gallery of still images. The outcome of this algorithm is to identify the person of interest from the test face track and recognize them if they are found in the training images. It also specifies the corresponding movie in which the personality has acted and confidence measure. In this paper, the proposed algorithm provides better prediction and classification accuracy when the residual error over the frames is minimized. The assumption made in this algorithm is all the images from the face track belong to the same person. So a high degree of correlation among the sparse coefficient vectors can be anticipated. This assumption holds good for a single face video dataset. Owing to the similarity between the faces in each track, nearly the same coefficient vector is to be recovered from each frame. Hence a single coefficient vector can be used for all frames and the sum squared residual error over the frames can be minimized using ℓ^2 minimization approach.

For face detection, in a video, an object detector is required to detect the location of the face in a frame Vision. Cascade Object detector is a function in MATLAB which is used for face detection. The cascade object detector uses viola-jones algorithm as a classifier for detecting face in the video. After detecting face in a single frame, to detect faces in successive frames step function is used.

To perform face – tracking a feature is needed to analyse the different facial movements in consecutive frames. In this paper, skin tone is invoked as a feature to perform tracking. It is used because it does not vary when the object moves or when background is affected by color or illumination changes. The cascade object face detector uses Vision. Histogram Based Tracker for tracking face in sequential frames. Histogram based tracker uses eye coordinates and nose box to obtain a histogram of pixel values. When the location of the face is known to the tracker places a bounding box around the face. It determines the face, if it is always in the scene using a score value. In this paper, a score value of 0.4 is used as a threshold to detect faces. If the value is less than 0.4 then we need to find the face again.

```
Input: Set of Images
Output: Set offeature vectors
  1.  Pre-processing step
        •  Load the images and convert them from RGB ->Grayscale
        •  Resize the image to 160 * 160 dimensions.
  2.  Initialize gamma, theta, lambda values to obtain rotation values in Gaussian filter
  3.   Compute x_theta, y_theta ,gb(xy);
  4.        for x=1:160
  5.           for y=1:160
  6.  x_theta=im_resize(x,y)*cos(theta)+im_resize(x,y)*sin(theta);
  7.  y_theta= im_resize(x,y)*sin(theta)+im_resize(x,y)*cos(theta);
  8.  gb(x,y)= exp(-(x_theta.^2/2*bw^2+ gamma^2*y_theta.^2/2*bw^2))*
      cos(2*pi/lambda*x_theta+psi).
  9.   end for;
  10. end for;
  11. Save gb(feature vector);
```

Algorithm 1. Gabor filter.

3.2 Feature extraction using HOG and gabor filter

3.2.1 Gabor filter

In Gabor filter, the size of the input image is chosen in such a way that the filter results have always the same size after subsampling.

The description for the above algorithm is as follows. Initially load all the training images and convert all the RGB color channel in images to grayscale to obtain a single sample in each pixel. Grayscale images have only black and white color channels where black has high intensity information and white color has the least. All the images are resized to $160 * 160$ dimensions to obtain same number of feature vectors for all the images. Initialize the variables to calculate edge detection and rotation values. Calculate the frequency and simulation to obtain the feature vector.In the best resolution channel, Gabor filters extract fine image structures of a small image region. In the lower resolution levels coarse image structures can be extracted over large regions.

3.2.2 HOG filter

The HOG filters are used as a feature descriptor to extract features from image. The descriptor counts the occurrences of gradient orientation and magnitude in localized portions of an image.

Initially load all the training images and convert the RGB color channels in the images to grayscale to obtain a single sample value in each pixel. Grayscale images have only black and white color channels. Black has high intensity information and white color has the least. All the images are resized to $160*160$ dimensions to obtain same number of feature vectors for all the images. Perform orientation binning to obtain rotation values. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. Identify the descriptor blocks to perform normalization and compute histogram and concatenate histogram of all cells. In this paper, SVM classifier is used to map data into defined output categories.

3.3 Feature optimization using PCA

The extracted features from Gabor and hog filters are present in large dimensions. Thus computation of features in large dimensions produces redundant features and reduces the prediction accuracy. Feature vectors[12] obtained using PCA convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. Steps for feature optimization using PCA is given below.

3.4 Key frame based technique

Key-frame method[7] performs a prediction on the identity of each key-frame in a face track followed by a probabilistic fusion or majority voting to select the best match. They learn a model over dictionary by learning key

Input: Set of Images
Output: Set of feature vectors

1. Pre-processing step
   - Load the images and convert them from RGB->Grayscale
   - Resize the image to 160 * 160 dimensions.
2. Initialize gamma, theta, lambda values to obtain rotation values in Gaussian filter
3. Initialize Gaussian filter;
4. RPI=Remove_reduntant_pixels (Pimage);
5. FV=Initialize feature_vector (RPI);
6. For each block
7. Compute Angle, Magnitude, Histogram;
8. end for;
9. For each cell in a block
10. Compute Angle, Magnitude, Histogram;
11. end for;
12. Rotation_values=Grad_Orientation binning (image);
13. Return (rotation_values);
14. Set threshold= [0,180];
15. If threshold>10 && Threshold<180
16. Obtain histogram channel;
17. End if;
18. Normalize= Normalize_values for each block;
19. Concatenate feature vectors;
20. Remove infinity values;
21. Save hog (feature vector);

Algorithm 2. HOG filter.

1. Input: N x d matrix X.
2. Output: N x m matrix Y.
3. Centralize the data x.
4. Select a normalized direction in d-dimensional space along which the variance in X is maximized
5. Select the matrix A , where each row is an eigenvector of $E = \frac{1}{n} XX^T$
6. Find another direction m along which variance is maximized.
7. Repeat this procedure until m vectors are selected.
8. If A is a square matrix, a non-zero vector v is an eigenvector of A if there is a scalar λ such that Av =λv.
9. Calculate the eigenvectors of the covariance matrix (orthonormal).
10. Select m eigenvectors that correspond to the largest m eigenvalues to be the new basis.

Algorithm 3. Principal component analysis (PCA).

1. Input: N x d matrix X.
2. Output: N x m matrix Y.
3. Centralize the data x.
4. Select a normalized direction in d-dimensional space along which the variance in X is maximized
5. Select the matrix A , where each row is an eigenvector of $E = \frac{1}{n} XX^T$
6. Find another direction m along which variance is maximized.
7. Repeat this procedure until m vectors are selected.
8. If A is a square matrix, a non-zero vector v is an eigenvector of A if there is a scalar λ such that Av =λv.
9. Calculate the eigenvectors of the covariance matrix (orthonormal).
10. Select m eigenvectors that correspond to the largest m eigenvalues to be the new basis.

Algorithm 4. Key frame based technique.

faces via clustering. These cluster centres are compared to test frames using a nearest – neighbour search followed by majority probabilistic voting to make a final prediction.

1. Input: Training images A and test face track Y.
2. Output: Identify the person and confidence measure.
3. Compute the features ỹ=Ry and a=RA and normalize ỹ and columns of â to unit length.
4. Solve the convex optimization problem

$$Min \; \| X_1 \| \text{ Subject to } \| y - aX \|^2 \leq \varepsilon$$

   - Compute the residual errors for each class

$$R_i(y) = \| Y - a\phi(X) \|^2 \text{ For } i=1, 2, 3 \ldots. k$$

5. Using Sparsity index obtain the confidence measure
6. When a test track is given identity the personality from a set of training images and recognize them.

Algorithm 5. Sparse representation based classification (SRC) algorithm.

- Input: Training images P and test face track F
- Output: Identify the person and confidence measure
- Normalize the columns of P to have unit ℓ^2-norm.
- Compute mean for each the track

$$F = \sum_{m=1}^{M} \frac{f_m}{M}$$

Where M is the length of the track.

- Normalize the track features to have unit ℓ^2-norm.
- All images in a face track belong to the same person, hence high degree of correlation is present amongst the sparse coefficient vectors.
- Hence we obtain a single coefficient vector x which determines the linear combination of training images P to obtain the feature vector of given person.
- Solve the ℓ^2 minimization problem.

$$X^2 = \text{argmin} \| F - Px \|_2^2 + \lambda \| x \|_2$$

- Compute the prediction error for each class

$$PE(y) = \| F - P_j x_j \|^2$$

- Using Sparsity index obtain the confidence measure

$$SI = \frac{K.\max_j \| x_j \| / \| q \| - 1}{K - 1} \in [0,1]$$

  Wherex$_j$ is the recovered coefficients from the test track, K denotes the number of classes and q is the global solution of recovered coefficients of class j.
- When a test track is given identity the personality from a set of training images and recognize them.

Algorithm 6. Regularized SRC algorithm (RSRC).

3.5 Sparse representation based classification algorithm

In SRC algorithm[11], the training gallery is given from a set of dictionary images and test face track is obtained from video data. Initialize the sparsity weight parameter λ. In this paper λ value is given as 0.001.

The description of the algorithm is as follows. Initially the training sample vector and test face track are given as an input. We are required to normalize the feature vectors to unit length to solve the convex optimization equation. After normalization obtain the minimum coefficient vector for each track in a frame. Once the vectors are generated, calculate the residuals for each class and using sparsity index evaluate the residual distribution to predict the classes. The main drawback of using convex optimization is it uses minimum coefficient vector for each frame in test face track. Minimum coefficient vector may sometime lead to improper residuals during distribution which may lead to misclassification. Another drawback of SRC algorithm is the evaluation of each face track takes about an average of 25 minutes which is time – consuming.

3.6 Regularized SRC using ℓ^2 minimization approach

The Regularized SRC algorithm is a combination of frame-by-frame method and image set matching method. Input for hybrid (RSRC) algorithm is a set of training gallery is given from a set of dictionary images (public fig dataset) and test face track (movie trailer dataset) is obtained from video data. Initialize the sparsity weight parameter λ. In this paper λ value is given as 0.01.The test face is given as $F = [f_1, f_2, f_M]$ where M represents the length of the track. X is the coefficient vector used in all frames. The confidence measure specifies as how well the residuals are distributed across classes.

The description of the algorithm is as follows. Initially the training sample vector and test face track are given as an input. Calculate the mean for each test face track. We are required to normalize the track vectors to unit length to solve the minimization equation. Since all the frames are correlated, each frame produces nearly the same coefficient. Thus single coefficient vector is used for all the frames in a track, the misclassification error is considerably reduced. Once the vectors are generated, calculate the residuals for each class using sparsity index evaluate the residual distribution to predict the classes.

4. Implementations and Results

4.1 Dataset used

A. Movie trailer dataset

Movie trailer face dataset is built using 101 movie trailers from YouTube from the 2010 release year that contained celebrities present in the supplemented PublicFig+10 dataset which is used in training. PubFig+10 consists of 34,522 images and Movie Trailer Face Dataset has 4,485 face tracks, which we use to conduct experiments on several algorithms. It allows to test larger scale face identification, as well as each algorithms ability to reject unknown identities. In this paper we have used 300 images from PubFig+10 as training samples and 61 face tracks as test face track samples to evaluate the algorithms.

B. Tools used

MATLAB (matrix laboratory) is a multi-paradigm fourth-generation programming language. Developed by Math Works, MATLABallows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. It is used for a range of applications, including signal processing and communications, image and video processing, test and measurement, computational finance, and computational biology. In this paper, MATLAB 2014a version has been used.

4.2 Empirical results

A. Performance metrics

Precision

Precision is defined as ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved.

$$P = \frac{A}{A + C} * 100\% \tag{1}$$

where, A is number of irrelevant records retrieved C is number of number of relevant records retrieved.

Recall

Recall is defined as ratio of the number of relevant records retrieved to the total number of relevant records in the database.

$$R = \frac{A}{A + B} * 100\% \tag{2}$$

where, A is number of relevant records retrieved B is number of relevant records not retrieved.

Average precision

Averaging the precision values from the rank positions where a relevant document was retrieved and set precision values to be zero for the not retrieved documents.

$$AveP = \frac{\sum_{r=1}^{N} (P(r) * \text{rel}(r))}{\text{Number of relevent documents}} \tag{3}$$

Accuracy

In this paper we need to identify whether the given test face track is present in the database or not. If it is present, it recognizes the subject and its corresponding movie in which he/she has acted. Accuracy is used as a statistical measure of how well a binary classification test correctly identifies or excludes a condition.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where, TP-True positive, TN-True negative, FP-False positive, FN-False negative.

4.3 Performance evaluation on SRC model

The performance for SRC model is assessed using precision and recall of the test data. The public fig dataset consists of 300 images which are used for training. 61 face tracks from movie trailer dataset as a test track in the test set. Feature vectors are obtained from test images are used to check whether the given test track is present in the dictionary of still images or not. If it is present, it recognizes the face track and displays the actor's name and the corresponding movie name in which he/she is acted. The straightforward application of SRC on a frame, by- frame basis. It computes SRC on each frame, approximately by 25 minutes per track which is time consuming. It obtains 58.3% average precision and 23.2% recall. In terms of timing, the pre-processing steps of tracking runs identically for SRC at 20 fps and feature extraction runs at 30 fps. For identification, SRC on a single frame takes 100 milliseconds.

4.4 Performance evaluation on regularized SRC (RSRC) model

The performance for regularized SRC (RSRC) model is evaluated using precision and recall of the test data. The public fig dataset consists of 300 images which are used for training. 61 face tracks from movie trailer dataset as a test track in the test set. Feature vectors are obtained from test images are used to check whether the given test track is present in the dictionary of still images or not. If it is present, it recognizes the face track and displays the actor's name and the corresponding movie name in which he/she is acted. The regularized SRC is a combination of key frame

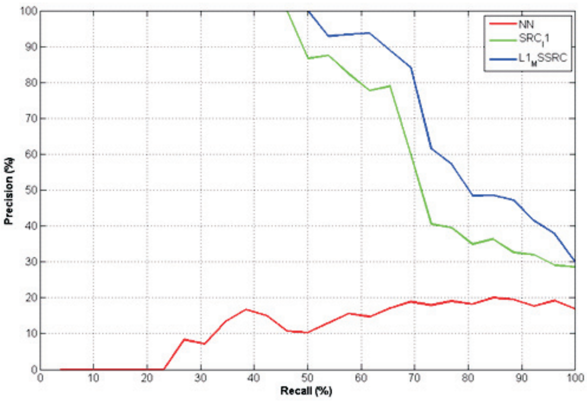

Fig. 2. Precision and recall graph of key frame, SRC and regularized SRC algorithms.

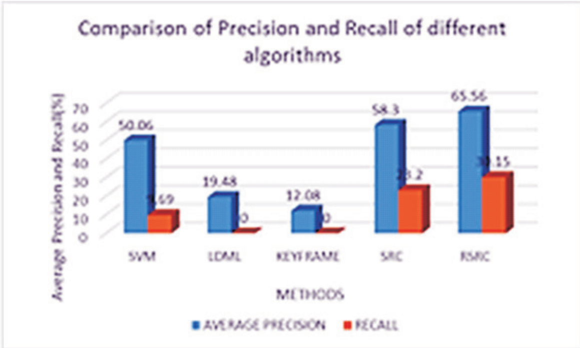

Fig. 3. Comparison of different methods.

Table 1. Average precision and recall for movie trailer dataset.

Algorithm	Average precision	Recall at 90% precision
Key-Frame	12.16	0.00
SRC	58.3	23.2
Regularized SRC	65.56	30.23
SVM	50.06	9.69
LDML	19.48	0.00

method and image set matching method. Since single coefficient vector is utilized to all the frames the residual error over the frames is minimized. Regularized SRC computes on each frame approximately by 1.5 minutes per track. It obtains 65.56% average precision and 30.15% recall. In terms of timing, the pre-processing steps of tracking runs identically for regularized SRC at 20 fps and feature extraction runs at 30 fps. For identification, SRC on a single frame takes 20 milliseconds per frame. Figure 2 show that experimental results prove Regularized SRC provides better classification than SRC algorithm due to the usage of single coefficient vector. A comparison with different existing methods is given in Fig. 3.

5. Conclusion and Future Work

In this paper a fully automatic end to-end system for video face recognition is presented, which includes face tracking and identification of persons from both still images of the known dictionary and video for recognition. A novel Regularized SRC (RSRC) algorithm is proposed which uses a single coefficient vector in all frames which are correlated unlike other existing algorithms and hence performs a joint optimization using all of the available image data to perform video face recognition. Furthermore, the proposed method excels the existing methods of rejecting unknown identities outperforming the average precision by 7%. In future, the effect of selecting key – frames, or less noisy frames, the area of domain transfer for transferring knowledge from the still-image domain to the videos can be considered.

References

[1] C. Bao, Y. Wu, H. Ling and H. Ji, Real Time Robust l1 Tracker using Accelerated Proximal Gradient Approach, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830–1837, March (2012).

[2] Enrique G. Ortiz, A. Wright and M. Shah, "Face Recognition in Movie Trailers via Mean Sequence Spare Representation-Based Classification, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, September (2013).

[3] Kaihua Zhang, Lei Zhang and Ming-Hsuan Yang, Fast Compressive Tracking, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pp. 1–13, October (2014).

[4] Lianyang Ma, Xiaokang Yang and Dacheng Tao, Senior Member IEEE Person Reidentification over Camera Networks using Multi – Task Distance Metric Learning, *IEEE Transactions on Image Processing*, vol. 23, no. 8, August (2014).

[5] Ng and M. Jordan, On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes, In *Advances in Neural Information Processing Systems*, pp. 841–848, July (2002).
[6] P. Viola and M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, pp. 511–518, August (2005).
[7] S. Li and Z. Zhang, Float Boost Learning and Statistical Face Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, November (2004).
[8] S. Oron, A. Bar-Hillel, D. Levi and S. Avidan, Locally Order Less Tracking, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1940–1947, January (2012).
[9] A. Weingessel and K. Hornik, Local PCA Algorithms, *IEEE Transactions in Neural Networks*, vol. 11, no. 6, pp. 1242–1250, November (2000).
[10] Yilmaz, O. Javed and M. Shah, Object Tracking: A Survey, *ACM Computing Surveys*, vol. 38, no. 4, June (2006).
[11] Z. Kalal, J. Matas and K. Mikolajczyk, Pn Learning: Bootstrapping Binary Classifiers by Structural Constraints, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–56, April (2010).
[12] Zhaowen Wang, Jianchao Yang and Nasrabadi Huang, A Max-Margin Perspective on Sparse Representation-Based Classification, *IEEE International Conference in Computer Vision*, pp. 1217–1224, December (2013).