Research Journal of Applied Sciences, Engineering and Technology 8(22): 2265-2271, 2014
DOI:10.19026/rjaset.8.1228
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 13, 2014 Accepted: September 20, 2014 Published: December 15, 2014

Research Article

Face Recognition System Based on Sparse Codeword Analysis

[1]P. Geetha, [1]E. Gomala and [2]Vasumathi Narayanan
[1]Anna University, Chennai,
[2]St.Joseph's College of Engineering, Old Mamallapuram Road, Kamaraj Nagar, Semmencherry, Chennai, Tamil Nadu 600119, India

Abstract: In recent times, large-scale content-based face image retrieval has grown up with rapid improvement and it is an enabling technology for many emerging applications. Content based face image retrieval is done by computing the similarity between images in the databases and the input/query face image. Content based face image retrieval systems retrieves the image only using low level features therefore the retrieval rate is low in this system. To improve the retrieval rate sparse codeword based scalable face image retrieval system is developed. This system uses both low level features and high level human attributes. The proposed system has several stages to retrieve the images; 1. Low level features are extracted using LTP descriptor and utilize the automatically detected high level human attributes such as hair, Gender and race. 2. Sparse codeword techniques are applied on the low level features and attributes to generate the codeword. 3. The third stage is an indexing; in the indexing attribute embedded inverted indexing method is used. Using the methods mentioned above, face image retrieval system has achieved promising retrieval result. Experiment is conducted on different dataset such as pub fig, LFW and FERET. Among those dataset LFW dataset achieve higher performance.

Keywords: Content based image retrieval, face image search, high level features, sparse coding

INTRODUCTION

Now a day the popularity of social networks like Face book, Twitter, Flickr are mostly used by the people. Many of them use human face images to their profile. Maximum of the user use the celebrities image (Krishnan *et al*., 2011). In Photo search by face positions and facial attributes on touch devices the images attribute is represented in the outline.

CBIR refers to techniques used to index and retrieve images from databases based on their visual content. Visual content is typically defined by a set of low level features extracted from an image that describe the colour, texture and/or shape of the entire image. It is an enabling technology for many applications such as automatic face annotation and crime investigation.

Face images are taken from Labelled Faces in the Wild dataset (Huang *et al*., 2007) and the images are frontal with up to about 20° of pose changes, such that the five face components (e.g., eyes, nose and mouth) are visible in a given face image.

Figure 1 shows example online celebrity face images with various poses, expressions and illumination.

Face retrieval task is closely related to face recognition task. Human attribute are high level descriptions about a person. Using this attribute many authors have achieved promising result in different

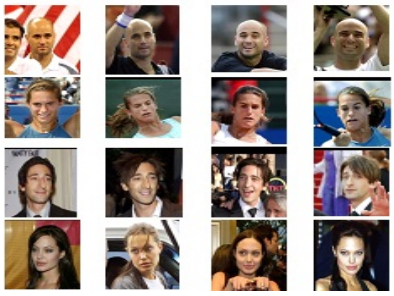

Fig. 1: Example online celebrity face images

applications such as face verification and face identification (Kumar *et al*., 2011; Nayar *et al*., 2009). Human attributes only contain limited dimension. Human attributes are represented as vector points. It does not work with large indexing method suffers slow response.

So far, numerous approaches using texture features and indexing have been developed and successfully applied to Face Image Retrieval System. The content of the image is information that can be derived from the image itself such as colours, shapes or textures. There are ample of indexing methods are used for large scale database.

The idea of our work is instead of using only low level features and human annotation (tagging) here

Corresponding Author: P. Geetha, Anna University, Chennai, India

using high level features that is people attributes to construct sparse codeword for the face image retrieval task.

A learning framework is proposed to automatically find describable visual attributes (Kumar *et al*., 2011). Automatically detected human attributes have been shown promising in different applications recently. A Bayesian network approach to utilize the human attributes for face identification (Scheirer *et al*., 2011).

It is proposed to use relative attributes (Parikh and Grauman, 2011) and proposed multi-attribute space (Scheirer *et al*., 2011) to normalize the confidence scores from different attribute detectors for similar attribute search. To reduce the quantization loss, adopt GIST feature with locality sensitive hashing for face image retrieval (Torralba *et al*., 2003).

A component-based local binary pattern (LBP), is used as a well known feature for face recognition (Ahonen *et al*., 2004) combined with sparse coding and partial identity information to construct semantic codeword's for content-based face image retrieval.

Recently, sparse coding has shown promising results on many different applications such as image denoising and classification. A machine learning framework using unlabeled data to learn basis of sparse coding for classification task is proposed (Raina *et al*., 2007) and further improved it by applying on SIFT descriptors (Yang *et al*., 2009) along with spatial pyramid matching (Lazebnik *et al*., 2006). Although using sparse coding combined with inverted indexing results in an efficient retrieval framework, it does not take advantage of using identity information. Taking advantages of the effectiveness and simplicity of LBP feature with the superior characteristics of sparse coding on face images.

Earliest methodologies for facial land mark detection were based on Principal Component Analysis and Linear Discriminate Analysis (LDA)/ Fishers linear discriminate and Active Shape model. Unlike subspace models, geometric feature based recognition techniques are invariant to similarity transformations, robust enough to withstand pose, illumination and expression changes in face. Selection of such features, which contribute towards robust recognition in case of transformation, has been studied extensively (Milborrow and Nicolls, 2008). A cascaded object detector is used to detect the face using Haar features (Viola and Jones, 2001).

Contributions of the proposed work describes below:

- Low level features are extracted using LTP descriptor and automatically detected high level features are used to construct the semantic codeword.
- Methods used in this study are to improve the content based face image retrieval.
- The Indexing method used in this study is attribute embedded inverted indexing is to improve the retrieval result.

The Experiment is conducted and demonstrates the performances of the proposed work in LFW dataset.

MATERIALS AND METHODS

This section, describes the overview of Sparse Codeword based scalable face image retrieval system and then explains the contribution in this study. The main contribution is in the Low Level Feature

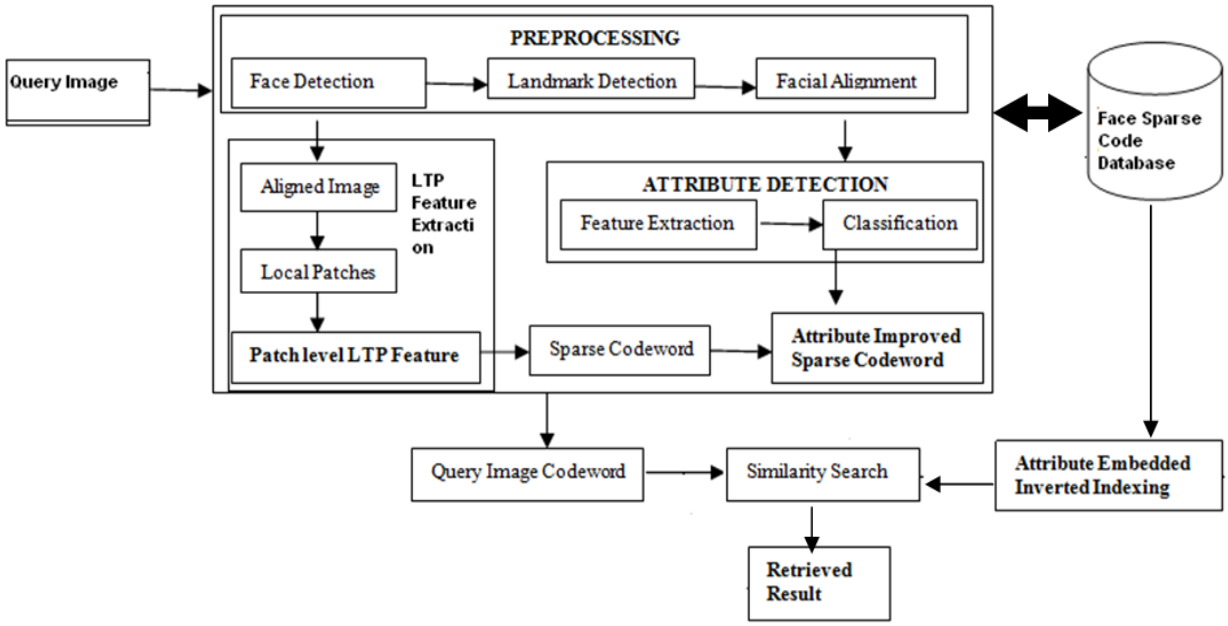

Fig. 2: Frame work of proposed work

Extraction. The low level features are extracted using LTP and then two methods are used in this study. They are Attribute Enhanced Sparse Codeword and Attribute Embedded Inverted Indexing. The overall System Architecture is illustrated in Fig. 2.

Pre-processing: The pre-processing steps are explained in the following sections.

Face detection: Retrieval system begins with face detection. Previous works for face detection uses Viola-Jones face detector, Okao detector. The methodology used in this study is skin color based face detection (Phung *et al*., 2002). RGB-YCrCb-HSV model is applied to detect human skin region. A structuring element of morphological filters is used to minimize noises in the face. Skin regions are represented as white pixels.

Template image is created to take the mean value of all images. Template image is matched with the morphological operation performed image. The average of the image being tested must be subtracted to eliminate biasing toward brighter areas. Face is detected by using skin region and automatically cropped the image.

Landmark detection and alignment: The detected face images are in different pose and lighting condition. The next pre-processing step is to align the face image. For alignment facial landmarks are to be detected. Active shape model is applied to locate different facial landmarks on the image. The landmarks are detected using Active Shape Model. For each lanmark point pb procrustes analysis is done to get an smooth shape.Matrix is constructed with all contour points. Eigen value and Eigen vector is find from an matrix points.The method used for finding Eigen value and Eigen vector is PCA.

Active Appearance Model AAM is used for texture analysis and shape. The AAM is start with a shape.

In our work 68 landmark points are detected. Using these facial landmarks, barycentre coordinate based mapping process is applied to align every face with the face mean shape. Using this landmark points face images are aligned.

Attribute detection: Attributes can be thought of as functions aim that map images I to real values aligned images. Large positive values of aligned images indicate the presence or strength of the attribute, while negative values indicate its absence.

Consider the attribute "gender." If images I1 and I2 are of males and image J is of a female, the gender function *ai* should map the males to positive values and J to a negative value, even if I1 and I2 differ in other respects such as lighting, pose, age, expression. Attribute Detection use the framework proposed (Kumar *et al*., 2011) to find the different attribute scores. They use the two classifications Method namely Attribute classifier and smile classifier, are binary classifiers. In both the method SVM with RBF kernel are used.

The SVM with RBF Kernel give the highest performance for attribute detection. The classifier output is a binary score. In the attribute detection framework they use 73 different attribute. In our work only few attribute scores are used. They are Gender, Eyes open or closed, Nose shape, Wearing Eye glass, Teeth visible, hair color, mouth open or closed etc. The attribute score is a binary value i.e., one or zero. Using this attribute score codeword is generated using sparse coding techniques.

Feature extraction: Many methodologies were used to extract the low level features. In previous work the method used to extract the low level feature is Local Binary Pattern (LBP). Local binary pattern is a 2-valued (binary) code. The LBP operator idea is based on just two bit values either 1 or 0. LBP method describes the local texture pattern with a binary code. It is built by thresholding a neighbourhood pixel with a radius by the grey value of its canter. As a result of this LBP method each pixel is replaced by binary patterns. This is applied to each pixel of the image and the feature vector is a histogram with all the LBP numbers.

The method used in this study is Local Ternary pattern (LTP). LTP is a 3-valued texture operator. LTP is an extension of LBP. Instead of a thresholding that is based only on the central pixel value of the neighbourhood, the user will define a threshold say t and any pixel value within the interval of $-t$ and $+t$, thus assigns the value 0 to that pixel, while the user assigns the value 1 to that pixel if it is above this threshold and a value -1 if it is below it when compared to the central pixel value.

LTP is a 3-valued texture operator. LTP is an extension of LBP. Instead of a thresholding that is based only on the central pixel value of the neighbourhood, the user will define a threshold say t and any pixel value within the interval of $-t$ and $+t$, thus assigns the value 0 to that pixel, while the user assigns the value 1 to that pixel if it is above this threshold and a value -1 if it is below it when compared to the central pixel value.

The Aligned face image is given as an input to this method. Before performing an LTP operation, extract the component in the given image. The components in the face image are eyes, nose, mouth and forehead etc. In this study five facial components are considered. They are two eyes, two mouth corners and nose tips. After detecting the facial components LTP operation is applied. The algorithm works as follows:

Step 1: From the Aligned face image, Components are extracted. Five facial components are extracted in this study.

Step 2: 7×5 grids are extracted from the facial component. Grids are square patch. Totally 175 grids from the five facial components are extracted.

Step 3: For each square patch LTP operation is applied:

$$LTP(i) = \begin{cases} 1 & p_i - p_c \geq t \\ 0 & p_i - p_c < t \\ -1 & p_i - p_c \leq -t \end{cases} \quad (1)$$

The above equation is a LTP operation, where *t* is a user specified threshold, *pi* is a pixel value in the neighbourhood and *pc* is the central pixel value.

Step 4: After setting a threshold value, there will be a negative and positive values.

Step 5: LTP values are divided in to two LTP channels, upper LTP (LTPU) and lower LTP (LTPL).

Step 6: The LTPU is obtained by replacing the negative values in the original LTP by zeros.

Step 7: The LTPL is obtained in two steps: first, replaced all the value of 1's in the original LTP to be zeros then changed the negative values to be 1's. LTP feature descriptor as our local feature.

Step 8: After obtaining local feature descriptors, quantize every descriptor into codeword's using attribute-enhanced sparse coding.

Attribute enhanced sparse coding: Attribute enhanced sparse coding is generated by considering human attributes and the feature descriptor values. ASC is applied to all patches in a single image to find different codeword's and combine all those codeword's together to represent the image. To generate the sparse codeword for a face image retrieval it should follow the below equation:

$$\min_{D,V} \sum_{i=1}^{n} \left\| x^{(i)} - DV^{(i)} \right\| + \lambda \left\| x^{i} \right\| \quad (2)$$

In the above mentioned equation $x^{(i)}$ is the LTP features extracted from a patch of face image, D is a dictionary which contains the codeword, images and the sparse value. V is the sparse representation of the image patches. The above mentioned equation consists of two main parts one is dictionary and another is the sparse representation.

Sparse coding is applied for all patches in an image after finding $v^{(i)}$ for each image patch; consider nonzero entries as codeword of image. The above process to 175 different spatial grids separately, so codeword from different grids will never match. Accordingly, we can encode the important spatial information of faces into sparse coding.

Dictionary selection (ASC-D) is important in order to consider the human attributes in the sparse representation. For a single human attribute the dictionary is divided into two different subsets, images with positive attribute scores use one of the subset and images with negative attribute scores will use the other. By doing these, images with different attributes will surely have different codeword. For the cases of multiple attributes, divide the sparse representation into multiple segments based on the number of attributes and each segment of sparse representation is generated depending on single attribute. This study considers only the single attribute. The above goal can be achieved by solving:

$$\min_{D,V} \sum_{i=1}^{n} \left\| x^{(i)} - DV^{(i)} \right\| + \lambda \left\| x^{i} \right\| + AS \quad (3)$$

where, AS is the attribute score obtained by a face image.

Attribute embedded inverted indexing: Attribute Enhanced Sparse codeword is used to generate the codeword of the image based on the human attribute and feature descriptor value. This Attribute Embedded Inverted Indexing utilizes the human attribute and feature descriptor values, constructs the indexing structure to retrieve the face image. A sparse coding technique is applied in ASC to construct the codeword.

After computing the sparse representation using Attribute enhanced sparse codeword use that codeword.

The similarity between two images is computed by taking the intersection between the codeword's as generated by the above method i.e., ASC. The Similarity score equation is given in Eq. (4):

$$S(i,j) = \left\| C^{(i)} \cap C^{(i)} \right\| \quad (4)$$

where, $c^{(i)}$ is the codeword for an given image, $c^{(i)}$ is the codeword for an dictionary image. By considering the human attributes in inverted indexing structure. The similarity score is modified and the threshold value is set is mentioned in Eq. (5):

$$S(i,j) = \left\{ \left\| C^{(i)} \cap C^{(j)} \right\| if\ h\left(C^{(i)}, C^{(j)}\right) \leq \mathrm{T} \right. \quad (5)$$

In the above equation the threshold value should not be too large and too small. If the threshold value is too large or too small the retrieval result is unsatisfactory. The threshold value is 0.1 produce satisfactory results. Attribute embedded inverted index

is built using the original codeword and the binary attribute signatures associated with all database images. The image ranking according to codeword is efficiently computed using inverted index by simply doing a XOR operation to check the hamming distance before updating the similarity scores.

RESULTS AND DISCUSSION

Experimental settings: Data set: The images are taken from LFW datasets for these retrieval systems. In this study 800 face images among 500 peoples in the dataset are taken. The retrieval system tested 12 images of 10 person totally 120 images as a query set. Figure 1 shows some example images from the dataset. Throughout the experiments, mean average precision (MAP), time, Precision at 5 (P@5) as a perform measurement (Table 1).

Compared methods: Retrieval system uses several different methods and the methods are compared and performance is analyzed. This study includes state of the art face recognition features. The methods are *LTP:* concatenated 59-dimension uniform LTP features computed from 175 local patches i.e., totally 10325 dimensions. *LBP:* concatenated 59-dimension uniform LBP features computed from 175 local patches i.e., totally 10325 dimensions. *ATTR*: Different human attributes scores computed by the method described in Torralba *et al*. (2003); *SC-LBP*: the sparse representation computed from LBP features combined with inverted indexing. Similar methods are used in Kumar *et al*. (2011); *SC-LTP*: the sparse representation computed from LTP features combined with inverted indexing. *ASC*: attribute-enhanced sparse representation uses face recognition features and attribute score. *AEI*: attribute-embedded inverted indexing uses inverted indexing based on the sparse codeword.

Performance: Sparse coding parameter setting and performance: In sparse coding, there is a need to decide the size of dictionary *K* and sparsity parameter λ. In this experimental work different dictionary size has been chosen from 100 to 400 with λ range from 10-4 to 10-1. It is found that if λ is properly set (around 10-1 to 10-2) the performance affects little by the size *K*. When λ is big (around 100), it penalizes the nonzero entry so much that all the entries in sparse representation become zero, therefore, the MAP drops to zero. When λ is small (around 10-4), the performance drops faster with smaller dictionary size. When λ is chosen properly, the performance of sparse coding exceeds baselines because the baselines with high dimensional features suffer from the curse of dimensionality. Note that in Fig. 3, the component based baseline (BC) outperforms, this is because that grids from component based baseline have many overlaps in the central face area where contains more information, Experiment uses $K = 100$, $\lambda = 0.1$ as our default parameters (Table 1).

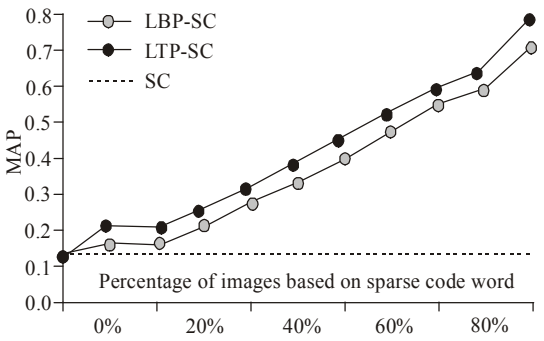

Fig. 3: Performance of sparse coding with LBP and LTP under different percentage of images

Table 1: Performance based on low level features with sparse codeword

	Performance measures		
Used methods	MAP (%)	P@5 (%)	Time (s) (%)
LTP	13.6	49.9	1.01
LTP-SC	12.8	46.8	0.01
LBP	11.9	43.61	1.52
LBP-SC	13.4	45.8	0.03
ATTR	11.6	37.8	0.04

Attribute enhanced sparse coding performance: The parameter used in the attribute enhanced sparse coding is the dictionary selection K. Different attributes values will produce different codeword. For a single human attribute dictionary is divided in to two parts, images with positive attribute scores and images with negative attribute scores. First half of the dictionary have +1 attribute score and use them to represent images with the positive attribute; the other half of the dictionary is assigned with -1 to represent images with the negative attribute. The experimented is conducted using single attributes.

The attributes used in this study are Gender (G) Male or female, wearing earrings, beard and must ache. Hair colour Black hair, Brown hair and Grey hair. Races are Asian, Indian and Caucasian. And other attributes such as teeth visible, nose size, eye opened or closed, nose size and shape. Using these ASC achieve relative performance.

When combining these attributes for retrieval system would produce better performance. Table 2 shows Attribute Enhanced Sparse codeword performance. The methods used in this study achieve salient performance (Table 3) using only several informative human attributes while still maintaining the scalability of the system. Some attributes will have negative effects on the performance.

Table 2: Performance based on single attribute. The mean average precision of single attribute used in ASC.

Attributes	ASC (MAP) (%)
Male	16.1
Female	15.2
Black hair	13.4
Brown hair	13.9
No eyewear	15.7
Eyeglasses	15.9
Wearing Earrings	13.1

Table 3: Overall performance for the retrieval system combining ASC and AEI. By combining this method achieve salient performance in retrieving

Methods	MAP (%)	P@5 (%)	P@10 (%)
SC	16.1	46.8	36.2
ASC	18.3	57.2	45.3
AEI	14.3	49.0	37.5
ASC+AEI	18.6	57.3	45.5

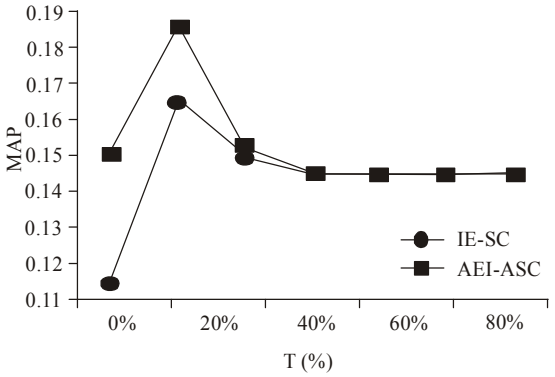

Fig. 4: The results of attribute-embedded inverted indexing in LFW using different threshold

Fig. 5: The top two images are query images; The left two column images are ranked result of first query image; The right two columns are ranked result of second query images; SC and ASC are the methods used for retrieval; Scalable face image retrieval system result based on SC and ASC by setting an appropriate threshold (T) value; The threshold value is 0.1

Attribute embedded inverted indexing: Attribute Embedded Inverted Indexing (AEI) performance is based on the threshold (T) value. When T is large the performance is similar to Sparse Codeword (SC) because in SC attribute signature are not considered. So choosing the threshold is very important. When T is small the relevant images are displayed based on the query set and the performance is improved. When T is too small the relevant images are not displayed. The performance will drop dramatically and shown in Fig. 4. By using the attribute listed in the Table 3 the performance is further improved compared to the attributes such as Smiling, Mouth Closed.

Result: Figure 5 shows some query example using SC and ASC. The red boxes indicate the false positive and the number denoted below each images is the rank of the images and the number in the Square bracket indicates rank by SC.

CONCLUSION

Attribute Enhanced Sparse codeword and Attribute Embedded Inverted Indexing methods utilize automatically detected human attributes to significantly improve content-based face image retrieval combining low-level features and automatically detected human attributes for content-based face image retrieval. Attribute- enhanced sparse coding exploits the global structure and uses several human attributes to construct semantic-aware codeword's. Attribute-embedded inverted indexing further considers the local attribute signature of the query image and still ensures efficient retrieval. The proposed indexing scheme can be easily integrated into inverted index, thus maintaining a scalable framework. Current methods treat all attributes as equal. Methods to dynamically decide the importance of the attributes and further exploit the contextual relationships between them.

REFERENCES

Ahonen, V., A. Hadid and M. Pietikainen, 2004. Face recognition with local binary patterns. Proceeding of the European Conference on Computer Vision, pp: 469-481.

Huang, G.B., M. Ramesh, T. Berg and E. Learned-Miller, 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report, University of Massachusetts, Amherst, MA, USA, pp: 07-49.

Krishnan, B., M. Veerasamy and G. Nagammapudur, 2011. Effect of weight assignment in data fusion based information retrieval. Int. Arab J. Inf. Techn., 8(3): 244-250.

Kumar, N., A.C. Berg, P.N. Belhumeur and S.K. Nayar, 2011. Describable visual attributes for face verification and image search. IEEE T. Pattern Anal., 33(10): 1962-1977.

Lazebnik, S., C. Schmid and J. Ponce, 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp: 2169-2178.

Milborrow, S. and F. Nicolls, 2008. Locating facial features with an extended active shape model. Proceeding of the European Conference on Computer Vision, pp: 504-513.

Nayar, S.K., A.C. Berg, P.N. Belhumeur and N. Kumar, 2009. Attribute and simile classifiers for face verification. Proceeding of the International Conference on Computer Vision, pp: 202-207.

Parikh, D. and K. Grauman, 2011. Relative attributes. Proceeding of the IEEE International Conference on Computer Vision, pp: 503-510.

Phung, S.L., Bouzurdom, A. and D. Chai, 2002. A novel skin color model in YCbCr color space and its application to human face detection. Proceeding of the IEEE International Conference on Image Processing, 1: 289-292.

Raina, R., A. Battle, H. Lee, B. Packer and A.Y. Ng, 2007. Self-taught learning: Transfer learning from unlabeled data. Proceeding of the 24th International Conference on Machine Learning (ICML, 2007), pp: 759-766.

Scheirer, W., N. Kumar, K. Ricanek, T.E. Boult and P.N. Belhumeur, 2011. Fusing with context: A Bayesian approach to combining descriptive attributes. Proceeding of the International Joint Conference on Biometrics Compendium, pp: 1-8.

Torralba, A., K.P. Murphy, W.T. Freeman and M.A. Rubin, 2003. Context based vision system for place and object recognition. Proceeding of the IEEE International Conference on Computer Vision, 1: 273-280.

Viola, P. and M. Jones, 2001. Rapid object detection using a boosted cascade of simple features. Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 1: 511-518.

Yang, J., K. Yu, Y. Gong and T. Huang, 2009, Linear spatial pyramid matching using sparse coding for image classification. Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp: 1794-1801.