Face Identification and Face Classification Using Computer Vision and Machine Learning Algorithm

Suhas Athani(✉) and C.H. Tejeshwar

B.V. Bhoomaraddi College of Engineering and Technology, Hubli, India
suhas2012athani@gmail.com, chtejeshwar4@gmail.com

Abstract. This paper aims to address a mechanism to recognize and classify the faces present in the videos. The number of videos are collected from the different sources. Each of the videos are divided into frames which are images. Key frames are extracted and stored in a separate directory. From the extracted key frames, faces present in them are detected. These set of faces detected will be used as a training data-set which are stored in normal directory. The features from each of the face images are extracted by making use of Local Binary Patterns (LBP). The feature values will further be used to train the Support Vector Machine (SVM) classifier. In the testing phase, the same process is carried out and SVM makes the prediction based on the training data-set to which class the face belongs to. The effectiveness of the proposed system's prediction is demonstrated by collecting different video data-set and the accuracy of the model is comparable.

The proposed system aims to classify two persons and accuracy of the model is calculated using K-fold cross validation.

Keywords: Key frame extraction · Local binary patterns (LBP) · Support vector machine (SVM)

1 Introduction

In this paper, face of the person present in a video is identified and further prediction is made whether the face of a person belongs to class-1 or class-2 with the help of linear based support vector machine (SVM) classifier. We are classifying two persons; Class-1 means a face of first person while class-2 means the second person. A single person can exhibit many facial expressions like happiness, sadness and so classification of faces with different facial expressions, illuminations in a video is a challenging problem. To address this topic, the video is divided into frames. Key frames are extracted. If a face is present in an extracted key frame it is identified and features are extracted from the respective key frames. The feature extracted values is used to build a model for prediction of the class labels (class-1 or class-2) using SVM. The classification of faces into different classes find its applications in the areas of content based retrieval, face detection, face recognition and face tracking [1].

In [2], authors propose a new pattern classification method called Nearest Feature Line (NFL), which has been shown to yield good results in face recognition and audio

R. Silhavy et al. (eds.), *Artificial Intelligence Trends in Intelligent Systems*,
Advances in Intelligent Systems and Computing 573, DOI 10.1007/978-3-319-57261-1_28

classification and retrieval. In [3], author extended the NFL method to video retrieval. Unlike conventional methods such as NN and NC, the NFL method takes into consideration of temporal variations and correlations between key-frames in a shot. The main idea is to use the lines passing through the consecutive feature points in the feature space to approximate the trajectory of feature points. In [4], the idea of Eigen face is introduced, which is one of earliest successes in the face recognition research and successfully applied the texture descriptor, local binary pattern (LBP), on the face recognition problem. In [5], author proposes the use of sparse representation derived from training images for face recognition. The method is proved to be robust against occlusions for face recognition.

In [6], some researches also use a reference set to improve the accuracy of face recognition and retrieval and used attribute classifiers, SVM classifiers trained on reference set, for face verification. Methods for off-line recognition of hand printed characters have successfully tackled the problem of intra-class variation due to differing writing styles. However, such approaches typically consider only a limited number of appearance classes, not dealing with variations in foreground/background color and texture [7].

The rest of the paper is organized as follows. Section 2 describes proposed approach. The sequential approach of key frame extraction is discussed in Sect. 3. The statistical feature extraction using LBP are detailed in Sect. 4. Classification using SVM is explained in Sect. 5. The results are mentioned on the considered data-set in Sect. 6 and K-fold cross validation is used to obtain the accuracy of the model. Finally we conclude in Sect. 7.

2 Proposed System

The proposed system Fig. 1, consists of three main modules. They are key frame extraction, feature extraction using LBP, classification using SVM classifier. The system also involves two phases, training and testing phase. In the training phase, training videos are divided into frames. Only the key frames are extracted for which sequential key frame technique is used. These key frames (images) are stored in a separate directory. Furthermore, faces present in the images are detected from which features are extracted using LBP and for the convenience histogram is plotted to store the count of each LBP value. In the next phase, a model is built using SVM classifier using training data-set. During training phase, labels are assigned to the classes considered. For class-1, label provided is 1 and for class-2, label provided is 0. In testing phase, separate testing data-set is considered which are unlabeled. In this phase, SVM classifier makes a prediction whether the face belongs to class-1 or class-2. Class-1 is named as face1 and class-2 is named as face2. Thus, linear SVM is used to classify two persons.

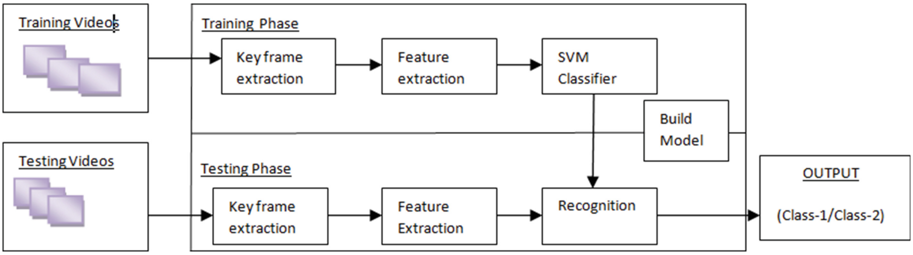


Fig. 1. System model showing the steps involved in proposed system

3 Key Frame Extraction

Video contains massive quantity of information at different levels in terms of scenes, shots and frames. The objective here addressed is the removal of redundant data which makes further processing easier. So, key frame extraction is the fundamental step in any of the video retrieval applications. Key frames are the frames which provide the summarized information of the complete video. Key frames are selected based on their uniqueness when compared to their subsequent frames. Dissimilarity between the frames must be computed in order to detect the key frames. The proposed system makes use of sequential comparison method wherein the first extracted key frame is compared with all other frames. This process is carried out until a different key frame is obtained. The sequential key frame extraction method is easy to implement as it has low computational complexity [8, 9].

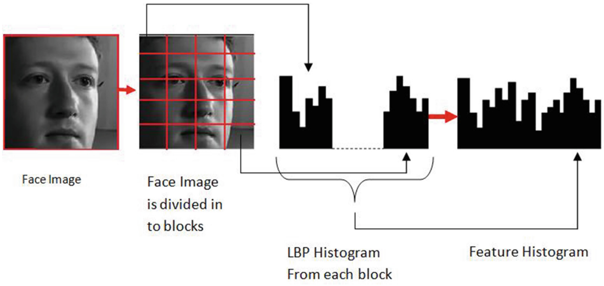
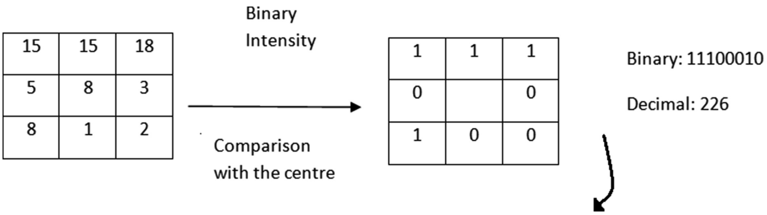

Face Image
Face Image is divided in to blocks
LBP Histogram From each block
Feature Histogram

Fig. 2. Phases involved in Linear Binary Pattern

4 Feature Extraction Using LBP

Local Binary Pattern (LBP) is widely used in the field of computer vision as a type of optical identity. This algorithm is helpful and serves as a powerful feature for texture classification. Steps carried out in LBP:

1. LBP looks at 9 pixels at a time.
2. It looks at 3 × 3 pixels and particularly interested at the central pixel.
3. For example, if the central pixel is 8 as shown in Fig. 3.



15	15	18
5	8	3
8	1	2

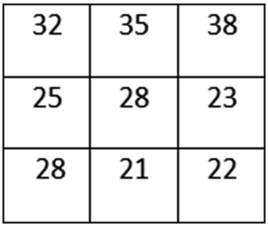
1	1	1
0		0
1	0	0

Fig. 3. Conversion of 1-byte of data into decimal carried during LBP computation

4. The central pixel is compared with the neighboring 8 pixels. If the neighboring pixel value is greater than or equal to central pixel value, then we assign 1 or else 0.
5. The binary values from the Fig. 3 are noted down as 11100010. This binary value will be converted into a decimal number which will be used to train the system. Here the binary values are noted in a clockwise manner [10] (Fig. 2).

The main advantage about LBP is that it is illumination invariant. If the lighting in the image is increased, the pixel values will also rise but the relative difference between the pixels will remain the same.

Consider the Fig. 4, 32 is greater than 28, so LBP value remains the same irrespective of illumination variation.



32	35	38
25	28	23
28	21	22

Fig. 4. Example for illumination invariant behavior of LBP

Consider Fig. 5, it is an image which is divided into 9 blocks. LBP also helps to detect the edges in a face like outline of mouth, eyelids. In the Fig. 5, three 1's and next 0, this transition indicates there are edges. By this, it is easy to make out the dark areas and light areas in the face. So basically there is a conversion from high dimensional space into a low dimensional space that only encodes relative intensity values and in doing so encodes edges.

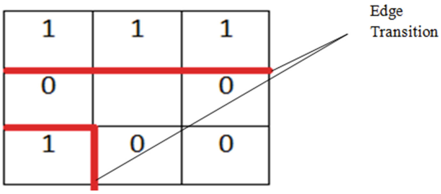


Fig. 5. Example to show transition of values helps to detect edges in LBP

5 Classification Using SVM

Classification is a process which involves separation of classes based on extracted features. Classes formed will be different from each other. The classifier is built by considering regularity patterns of training data-set. There are different classification algorithms like k-nearest neighbors, decision tree learning and support vector machine which are majorly used in various types of classification. K-nearest neighbor algorithm (kNN) uses the k-nearest neighbors to build the selection of class assignment straight from the training example. There are many algorithms like C4.5 to construct decision trees to predict the class of the input. SVM uses the concept of hyperplanes to predict the class of the input [11]. In the proposed system, SVM is used for classification of two persons.

Support Vector Machine (SVM) makes use of hyperplane that acts as boundary which divides two classes. The position of the hyperplane has to be decided for better classification. In the Fig. 6, the circles represent the features belonging to class C_1 (class-1) and triangles represent the features belonging to class C_2 (class-2). The position of hyperplane as shown in the Fig. 6, is not desirable because it gives a large bias in favor of class C_2 whereas it puts penalty with class C_1.

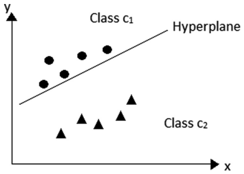


Fig. 6. Undesirable positioning of hyperplane

The reason is that, an interspace which is represented as black region in the Fig. 7, is given to the class C_2 and margin for class C_1 is less.

The classifier provides more appropriate results if the position of hyperplane is as shown in the Fig. 8, which is at an equal distance from the two classes. For Support Vector Machine, 1 (class-1) and 0 (class-2) are assigned as labels for two classes [12]. The circles and triangles that lie on the line as shown in Fig. 8 are support vectors. Decision function for a Support Vector Machine (SVM) is completely identified by

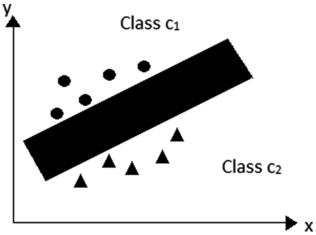


Fig. 7. Undesirable positioning of hyperplane

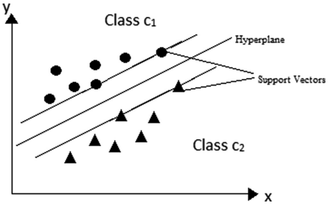


Fig. 8. Desirable positioning of hyperplane

batch of information which defines the location of hyperplane. Support Vector Machine (SVM), specifically explains the norm about selection boundary which is best, when it is distant off from whichever data instant. Presently, the classes are class-1 and class-2, and the features are the statistical texture features extracted for each image. Basically the SVM's use hyperplanes to separate different classes through an ideal function committed from training data as follows:

$$g(x) = ax + b \tag{1}$$

$ax + b = 0$, represents a hyperplane in multi-dimensionality

a – vector perpendicular to the hyperplane
b – position of the hyperplane in the d-dimensional space

For every feature vector X, the linear function $ax + b$ has to be computed.

Considering two classes C_1 and C_2, a feature vector x_1, if function lies on the positive side of the hyperplane then,

$$g(x_1) = ax_1 + b > 0, \quad x_1 € C_1 \tag{2}$$

If function lies on the negative side of the hyperplane then,

$$g(x_1) = ax_1 + b < 0, \quad x_1 € C_2 \tag{3}$$

and if function lies on the hyperplane then,

$$g(x_1) = ax_1 + b = 0 \tag{4}$$

Classifier has to be trained initially i.e., to find a and b. Supervised learning is used for training. The calculated a and b values are taken and for every sample $ax + b$ is checked. If a sample from class C_1 is chosen and if the value of $ax + b$ is not greater than zero, then the values of a and b are modified in such a way that the position of hyperplane is so modified that particular x taken from C_1 is moved to positive side of the hyperplane.

The aim of SVM takes care of maintaining maximal distance between the feature vectors of two separating classes [13]. For every x_i, we can have y_i which represents the belongingness (± 1). Therefore, $y_i(ax_i + b)$ is always greater than zero irrespective of the class. Now $ax_i + b = \gamma$ where γ is the margin which is the measure of distance of x_i from the separating plane. Considering $ax + b = 0$, the distance of a point x from the hyperplane is given by,

$$\frac{ax + b}{||a||} \geq \gamma \tag{5}$$

where, $||a||$ tells the orientation of the plane.

$$ax + b \geq \gamma||a|| \tag{6}$$

such that $\gamma||\mathrm{a}|| = 1$.

hence,

$$ax + b \geq 1 \; if \; x \epsilon C_1 \tag{7}$$

$$ax + b \geq 1 \; if \; x \epsilon C_2 \tag{8}$$

we have,

$$y_i(ax_i + b) \geq 1 \tag{9}$$

therefore, we can conclude that,

if $y_i(ax_i + b) > 1$, then $\mathrm{x_i}$ is not a support vector. And, if $y_i(ax_i + b) = 1$, then x_i is a support vector.

SVM is a linear machine whose design is greatly influenced by the position of support vectors. The distance of the point x_i from the plane has to be maximized.

From Eq. 5, $(ax + b)$ should be maximized and $||a||$ should be minimized.

From Eq. 9, it can be observed that $y_i(ax_i + b) \geq 1$ acts as a constraint. This constraint problem can be converted to un-constraint one by using Lagrange's multiplier.

we have,

$$L(a, b) = \frac{1}{2} \tag{10}$$

6 Results and Discussion

A set of 100 images of both class-1 and class-2 was taken from the key frames. The collected images were then cropped manually to be of same size. The data-set has 40 class-1 images and 40 class-2 images which are named as c1 to c80. The remaining 10 images are used for testing. All the images classified as class-1 are named as face1 and images classified as class-2 are named as face2. They are named likely because images named as face1 is the face of one person whose face is different from the images named as face2.

The Fig. 9, gives training image data-set which has been utilized to build SVM model and also depicts which all images are of class-1 and class-2 (Fig. 10).

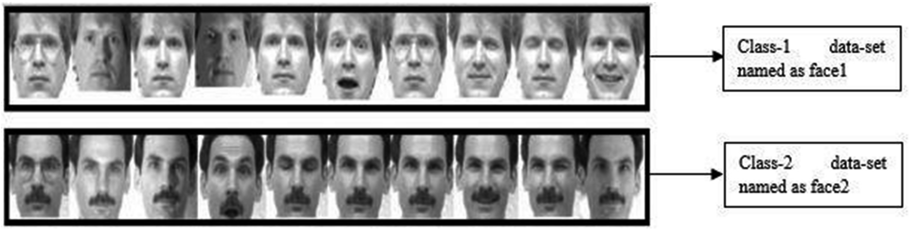

Class-1 data-set named as face1
Class-2 data-set named as face2

Fig. 9. Training data-set

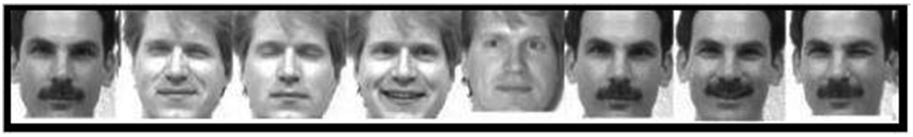
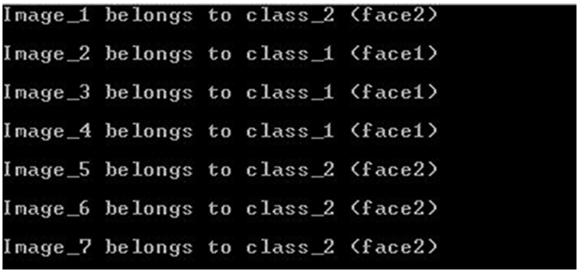


Fig. 10. Testing data-set

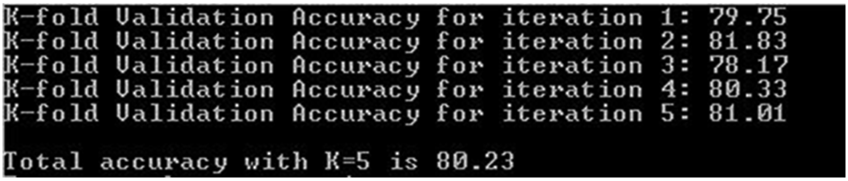
The Fig. 11, is the output which shows the prediction made by the SVM classifier. The output contains the image name and it's classification as belonging to class-1 (face1) or class-2 (face2).



```
Image_1 belongs to class_2 (face2)
Image_2 belongs to class_1 (face1)
Image_3 belongs to class_1 (face1)
Image_4 belongs to class_1 (face1)
Image_5 belongs to class_2 (face2)
Image_6 belongs to class_2 (face2)
Image_7 belongs to class_2 (face2)
```

Fig. 11. Results of image classification using SVM classifier

The classifier built can be evaluated by evaluation technique such as k-fold cross validation. Cross-validation is a model validation technique for estimating the performance of an independent data-set. In K-fold cross validation, entire data-set is partitioned into initial 80 images for the training phase and rest 20 images for the testing phase. Likewise, the process is continued for complete data-set. Finally, the exactness is studied by computing mean of 5 repetitions. During the process of K-fold cross-validation, number of folds considered are 5. The accuracy is estimated to be 80.23%. The pseudo code is as follows:



```
K-fold Validation Accuracy for iteration 1: 79.75
K-fold Validation Accuracy for iteration 2: 81.83
K-fold Validation Accuracy for iteration 3: 78.17
K-fold Validation Accuracy for iteration 4: 80.33
K-fold Validation Accuracy for iteration 5: 81.01

Total accuracy with K=5 is 80.23
```

Fig. 12. Results of K-fold cross validation

The Fig. 12, describes the result of the same as method in the Algorithm 1, where the accuracy of classification for the first, second, third, fourth and fifth iteration are namely, 79.75%, 81.83%, 78.17%, 80.33% and 81.01%. The complete accuracy is calculated as follows:

$$Total\,accuracy = \frac{Accuracy\,of\,(iteration\,1 + iteration\,2 + iteration\,3 + iteration\,4, + iteration\,5)}{5} \tag{11}$$

```
divide train data into 5 parts
for i = 1 to 5
train SVM using 80 images
compute accuracy using 20 images
end for
compute average accuracy of the 5 runs and express it in terms of percent
```

Algorithm 1: Usage of K-fold Cross Validation in Calculating Accuracy of the model

7 Conclusion

In this paper, faces of humans are detected and classified using LBP and SVM algorithms. The key frames are extracted from the videos. From these key frames, faces are identified and features are extracted using LBP. The classification of faces is carried out using linear SVM classifier. Further, work will be considered for more than two classes, where the implementation and processing can be carried out in a parallel and distributed manner. This is required because the time requirement for SVM increases with an increase in the number of classes.

References

1. Li, S.Z., Jain, A.K.: Handbook of Face Recognition. Springer, London (2005)
2. Li, S.Z., Lu, J.: Face recognition using the nearest feature line method. IEEE Trans. Neural Netw. **10**(2), 439–443 (1999)
3. Zhano, L., Qi, W., Li, S.Z., Yang, S.-Q., Zhang, H.J.: Key-frame extraction and shot retrieval using nearest feature line (NFL). Technical report, China (2000)
4. Chen, B.-C., Chen, C.-S., Hsu, W.H.: Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. IEEE Trans. Multimed. **17**(6), 804–815 (2015)
5. Naik, R.K., Lad, K.B.: A review on side-view face recognition methods. Int. J. Innov. Res. Comput. Commun. Eng. **4**(3), 2984–2991 (2016)
6. David, H., Athira, T.A.: Improving the performance of SVM detection. In: Fourth International Conference on Advances in Computing and Communications, Kochi, 27—29 August 2014
7. Pallabi, P., Thuraisingham, B.: Face recognition using multiple classifiers. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006), May 2006
8. Athani, S., Tejeshwar, C.H.: Performance analysis of key frame extraction using SIFT and SURF algorithms. Int. J. Comput. Sci. Inf. Technol. **7**(4), 2136–2139 (2016)
9. Athani, S., Tejeshwar, C.H.: Content-based text retrieval using image processing techniques. Int. J. Comput. Sci. Inf. Secur. **14**(11), 556–561 (2016)
10. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
11. Hall, L.O., Goldgof, D.B., Felatyev, S., Smarodzinava, V.: Horizon detection using machine learning techniques. In: Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA 2006) (2006)
12. Pujari, J.D., Yakkundimath, R., Byadgi, A.S.: Classication of fungal disease symptoms affected on cereals using colour texture features. Int. J. Signal Process. Image Process. Pattern Recogn. **6**(6), 321–330 (2013)
13. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Knowl. Discov. Data Min. **2**, 121–167 (1998)