FACE RECOGNITION USING LDN CODE

J.K.Jeevitha[1] ,B.Karthika[2] ,E.Devipriya[3]

[1 2 3]*Assistant Professor, Department of Information Technology, Tamil Nadu, India*

Abstract - *LDN characterizes both the texture and contrast information of facial components in a compact way, producing a more discriminative code than other available methods. An LDN code is obtained by computing the edge response values in 8 directions at each pixel with the aid of a compass mask. Image analysis and understanding has recently received significant attention, especially during the past several years. At least two reasons can be accounted for this trend: the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after nearly 30 years of research. In this paper we propose a novel local feature descriptor, called Local Directional Number Pattern (LDN), for face analysis, i.e., face and expression recognition. LDN characterizes both the texture and contrast information of facial components in a compact way, producing a more discriminative code than other available methods.*

Key Words: LDN, Compact mask, Median Noise.

1.INTRODUCTION

Face analysis has a varied range of applications, namely biometric authentication, surveillance, human-computer interaction, and multimedia management. Due to the endless possibility of its application and the interest generated, research and development in automatic face analysis which consist of face recognition and expression recognition follows naturally.

A good descriptor should have a high variance among classes i.e. between different persons or expressions, but little or no variation within classes i.e. same person or expression in different conditions. These descriptors are used in several areas, such as, facial expression and face recognition. An image may be defined as a two-dimensional function, f(x,y), where x and y are the spatial coordinates, and the amplitude of f at any couple of coordinates (x, y) is called the intensity or a gray level of the image at that point. When x, y, and an amplitude values of f are all finite, separate quantities, we call the image a digital image.

Face and expression recognition has been one of the fast developing areas due to its wide range of applications such as emotion analysis, biometrics, image retrieval and it is one of the area on which lot of research has been carried by

solving the problems occurring in recognition of the face expressions under different conditions and numerous other variations. Local Directional Number Pattern (LDN) acts as a face descriptor for recognizing robust faces and encodes the information related to structural and intensity variations of the face's texture. The structure of a local neighborhood is encoded by analysing its directional information.

This paper represents a method for face and facial expression recognition more efficiently and robust as compared to the existing methods. It is a novel encoding scheme, named as, Local Directional number pattern (LDN) encodes efficiently into a compact code by taking advantage of different structural face textures.

2.LITERATURE REVIEW

Face recognition is one of the most successful applications of image analysis and understanding, face recognition has recently received important attention, especially during the past few years. There are two common approaches to extract facial features: geometric-feature-based and appearance based methods. The performance of the appearance-based methods is excellent in constrained environment but their performance degrades in environmental variation. The face and expression features are recognized in different applications in different conditions.

I.Kotsia and I.Pitas, [5] proposed facial expression recognition in facial image sequences are presented. The user has to manually place few Candide grid nodes to face landmarks depicted at the first frame of the image sequence under assessment. The grid-tracking and deformation system is used based on deformable models, tracks the grid in successive video frames eventually, as the facial expression evolves, in anticipation of the frame that corresponds to the greatest facial expression intensity. The geometrical displacement of selected certain Candide nodes, defined as the dissimilarity of the node coordinates between the first and the greatest facial expression intensity frame also used as an input to a novel multiclass Support Vector Machine (SVM) system of classifiers that are used to recognize either the six basic facial expressions or a set of chosen Facial Action Units (FAUs).

M. Pantic and L. J. M. Rothkrantz, [6] proposed the Face Expression Recognition and Analysis: The State of the Art in this automatic face and expression recognition the characteristics of an ideal system, Databases that have been used and the advances made in terms of their standardization and a detailed summary of the state of the art and discusses facial parameterization using FACS Action Units (AUs) and MPEG-4 Facial Animation Parameters (FAPs) and the recent advances in face detection, tracking, feature extraction methods. Observations have also been offered on emotions, expressions and facial features, conversation on the six prototypic expressions and the recent studies on expression classifiers.

L.Wiskott,J.-M.Fellous,N.Kuiger and C. von der Malsburg, [15] proposed Face Recognition by Elastic Bunch Graph Matching it present a system for recognizing human faces from single images out of a large database containing one image per person. The task is difficult because of image variation in terms of position, expression, size, and pose. The system collapses most of this variance by extracting concise face descriptions in the form of image graphs. In these, fiducial points on the face (eyes, mouth, etc.) are described by sets of wavelet components (jets). Image graph extraction is based on a novel approach, the bunch graph, which is developed from a small set of sample image graphs. Recognition is based on a simple comparison of image graphs. We statement recognition experiments on the FERET database as well as the Bochum database, as well as recognition across pose.

T. Ahonen, A. Hadid, and M. Pietik¨ainen, [1] proposed a Face Description with Local Binary Patterns: Application to Face Recognition in that the face image is divided into several regions from which the LBP feature distributions are extracted and concatenated into an enhanced feature vector to be used as a face descriptor. The act of the proposed method is assessed in the face recognition problem under different challenges.

3. EXISTING SYSTEM

Local binary pattern (LBP) is a nonparametric descriptor, which efficiently summarizes the local structures of images. In recent years, it has aroused increasing interest in many areas of image processing and computer vision and has shown its
effectiveness in a number of applications, in particular for facial image analysis, including tasks as diverse as face detection, face recognition, facial expression analysis, and demographic classification. As a typical application of the LBP approach, LBP-based facial image analysis is extensively reviewed, while its successful extensions, which deal with various tasks of facial image analysis, are also highlighted.

The original LBP operator labels the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel. Each pixel is compared with its eight neighbors in a 3 × 3 neighborhood by subtracting the center pixel value; the resulting strictly negative values are encoded with 0, and the others with 1. For each given pixel, a binary number is obtained by concatenating all these binary values in a clockwise direction, which starts from the one of its top-left neighbor. The corresponding decimal value of the generated binary number is then used for labeling the given pixel. The derived binary numbers are referred to be the LBPs or LBP codes.

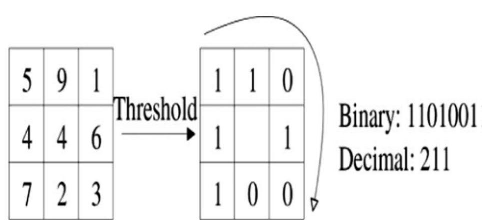

Fig.3.1: Example of the basic LBP operator

In above Fig.3.1 each pixel is compared with its eight neighbors in a 3 × 3 neighborhood by subtracting the center pixel value; the resulting strictly negative values are encoded with 0, and the others with 1.

One limitation of the basic LBP operator is that its small 3 × 3 neighborhood cannot capture dominant features with large-scale structures. To deal with the texture at different scales, the operator was later generalized to use neighborhoods of different sizes [9]. A local neighborhood is defined as a set of sampling points evenly spaced on a circle, which is centered at the pixel to be labeled, and the sampling points that do not fall within the pixels are interpolated using bilinear interpolation, thus allowing for any radius and any number of sampling points in the neighborhood.

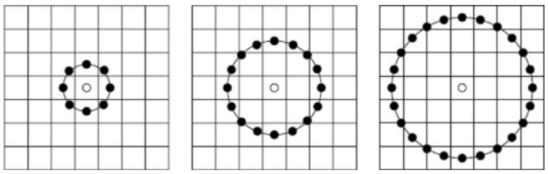

Fig. 3.2: Examples of the ELBP operator. The circular (8, 1), (16, 2), and
(24, 3) neighborhoods.

In above Fig.3.2 some examples of the extended LBP (ELBP) operator, where the notation (P, R) denotes a neighborhood of P sampling points on a circle of radius of R.

3.1Enhancing the discriminative probability

The LBP operator defines a certain number of patterns to describe the local structures. To enhance their discriminative capability, more patterns or information could be encoded. Jin et al. [10] modified the LBP operator to describe more local structure information under certain circumstances. Specifically, they proposed an improved LBP (ILBP), which compares all the pixels (including the central pixel) with the mean intensity of all the pixels in the patch. For instance, the LBP(8,1) operator produces only 256 (28) patterns in a 3 ×3 neighborhood, while ILBP has 511 patterns (29 − 1, as all zeros and all ones are the same). Later, ILBP was extended to use the neighborhoods of any size instead of the original 3 ×3 patch [6]. Almost at the same time, a similar scheme was used to extend CT to modified CT [3], namely, modified LBP (MLBP) in [12]. A mean LBP [2] is presented, which is similar to ILBP, but without considering the central pixels.

Yang and Wang [15] proposed Hamming LBP to improve the discriminative ability of the original LBP. They reclassified non uniform patterns based on Hamming distance, instead of collecting them into a single bin as LBPu2 does. In the Hamming LBP, these non uniform patterns are incorporated into existing uniform patterns by minimizing the Hamming distance between them; for example, the nonuniform pattern (10001110)2 is converted into the uniform one (10001111)2 , since their Hamming distance is one. When several uniform patterns have the same Hamming distance with a nonuniform pattern, the one with the minimum Euclidian distance will be selected.

4. PROPOSED WORK

Local directional number encodes the directional information of the face's textures in a dense way, producing a more discriminative code than current methods. We figure the structure of each micro-pattern with the support of a compass mask that extracts directional information, and encode such information using the well-known direction indices and sign .which allows us to distinguish among similar structural patterns that have different intensity transitions. Then we divide the face into several regions, and take out the distribution of the LDN features from them. Then, concatenate these features into a feature vector, and we use it as a face descriptor.

In which descriptor performs consistently under illumination, noise, expression, and the time lapse variations. The proposed Local Directional Number Pattern (LDN) is a six bit binary code assigned to each pixel of an input image that represents the structure of the texture and its intensity transitions. As previous research, edge magnitudes are largely insensitive to lighting changes. Accordingly, we create our pattern by computing the edge response of the neighborhood using a compass mask, and by captivating the top directional numbers, that is positive and negative directions of those edge responses. We show this coding scheme.

The positive and negative responses give valuable information of the structure of the neighborhood, as they expose the gradient direction of bright and dark areas in the neighborhood. Thus, this distinction, between dark and bright responses, allows LDN to discriminate between blocks with the positive and the negative direction swapped (which is equivalent to swap the bright and the dark areas of the neighborhood) by generating a different code for each case, while other methods may mistake the swapped regions as one. also, these transitions occur repeatedly in the face, for example, the top and bottom edges of the eyebrows and mouth have different intensity transitions.

4.1 Local Directional Number (LDN)

The code is generated using LDN, by analyzing the edge response of each mask, {M0,...,M7}, that represents the edge significance in its respective direction, and by combining the dominant directional numbers. Given that the edge responses are not equally important; the presence of a high negative or positive value signals a prominent dark or bright area. Hence, to encode these prominent regions, we implicitly use the sign information, as we assign a fixed position for the top positive directional number, as the three most significant bits in the code, and the three least significant bits are the top negative directional number.

$LDN(x,y)=8i_{x,y}+j_{x,y}$

where (x, y) is the central pixel of the neighborhood being coded, ix,y is the directional number of the maximum positive response, and jx,y is the directional number of the minimum negative response defined by:

$i_{x,y} = \text{argmax}\{II^i(x,y) | 0 \le i \le 7\}$

$j_{x,y} = \text{argmax}\{II^j(x,y) | 0 \le i \le 7\}$

where Пi is the convolution of the original image, I, and the ith mask, Mi , defined by:

$II^i = I * M^i$

To create the LDN code, a compass mask is used to compute the edge responses. I Proposed code is analyzed using two different asymmetric masks: Kirsch and derivative-Gaussian. Both masks operate in the gradient space, which reveals the face structure. Furthermore, explore the use of Gaussian smoothing to stabilize the code in presence of noise by using the derivative-Gaussian mask. The Kirsch mask is rotated apart to obtain the edge response in eight different directions. This indicate the use of this mask to produce the LDN code by LDNK. Moreover, inspired by the Kirsch mask , use the derivative of a skewed Gaussian to create an asymmetric compass mask that is used to compute the edge response on the smoothed face.
This mask is strong against noise and illumination changes, while producing strong edge responses. Therefore, given a Gaussian mask defined by:

$G_\sigma(x,y)= 1/2\pi\sigma^2 \exp(-x^2 + y^2/2\pi\sigma^2)$

where x, y are location positions, and is the width of the Gaussian bell; this defines the mask as:

$M_\sigma(x,y) = G_\sigma{}'' (x+k,y) * G_\sigma (x,y)$

where is the derivative of Gσ with respect to x, σ is the width of the Gaussian bell, * is the convolution operation, and k is the offset of the Gaussian with respect to its centre. A compass mask is generated, {M0σ ,...,M7σ}, by rotating Mσ 45° apart, in eight different directions. Thus, a set of masks is obtained.

Architecture Diagram

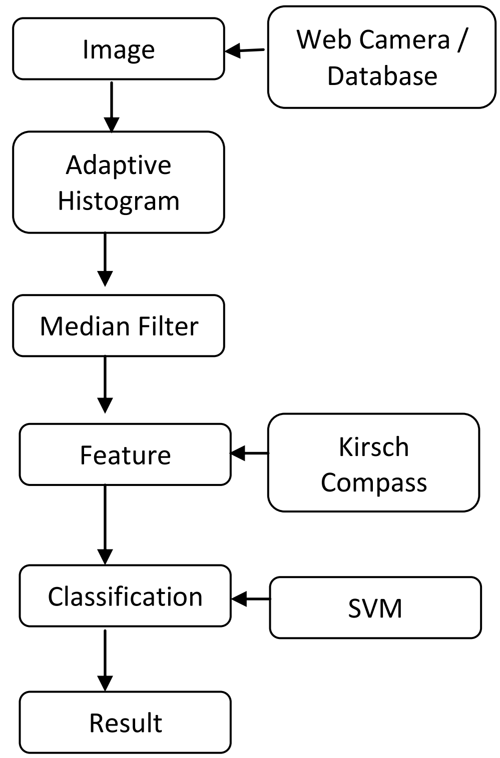

Fig:4.1.1 Image acquisition

Fig4.1.1 explain Image acquisition Image acquisition in image processing can be broadly defined as the action of retrieving an image from some source, usually a hardware-based source, so it can be passed through whatever processes need to occur afterward. Performing image acquisition in image processing is always the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed and is the result of whatever hardware was used to generate it, which can be very important in some fields to have a consistent baseline from which to work. One of the ultimate goals of this process is to have a source of input that operates within such controlled and measured guidelines that the same image can, if necessary, be nearly perfectly reproduced under the same conditions so anomalous factors are easier to locate and eliminate.

Depending on the field of work, a major factor involved in image acquisition in image processing sometimes is the initial setup and long-term maintenance of the hardware used to capture the images. The actual hardware device can be anything from a desktop scanner to a massive optical telescope. If the hardware is not properly configured and aligned, then visual artifacts can be produced that can complicate the image processing. Improperly setup hardware also may provide images that are of such low quality that they cannot be salvaged even with extensive processing. All of these elements are vital to certain areas, such as comparative image processing, which looks for specific differences between image sets.

One of the forms of image acquisition in image processing is known as real-time image acquisition. This usually involves retrieving images from a source that is automatically capturing images. Real-time image acquisition creates a stream of files that can be automatically processed, queued for later work, or stitched into a single media format. One common technology that is used with real-time image processing is known as background image acquisition, which describes both software and hardware that can quickly preserve the images flooding into a system.

There are some advanced methods of image acquisition in image processing that actually use customized hardware. Three-dimensional (3D) image acquisition is one of these methods. This can require the use of two or more cameras that have been aligned at precisely describes points around a target, forming a sequence of images that can be aligned to create a 3D or stereoscopic scene, or to measure distances. Some satellites use 3D image acquisition techniques to build accurate models of different surfaces.

Of course, such extraction programs will rarely be perfect on their first execution and, in the course of debugging them, it is often helpful to have a perfectly replicable input. This can help narrow the scope of a search for bugs or errors. In a vision application this replicable input may be accomplished by saving a single stream of data and piping it into subsequent program executions.

4.2.Median Filtering

The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image). Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise.

Fig4.2.1: The image before filtering and after filtering

In above Fig.4.2.1 the main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal.

For 1D signals, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median, see median for more details.

4.3. Image noise

It is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable by-product of image capture that adds spurious and extraneous information.

The original meaning of "noise" was and remains "unwanted signal"; unwanted electrical fluctuations in signals received by AM radios caused audible acoustic noise ("static"). By analogy unwanted electrical fluctuations themselves came to be known as "noise". Image noise is, of course, inaudible.The magnitude of image noise can range from almost imperceptible specks on a digital photograph taken in good light, to optical and radioastronomical images that are almost entirely noise, from which a small amount of information can be derived by sophisticated processing.

4.4. Compass mask

We use the gradient space, instead of the intensity feature space, to calculate our code. The former has more information than the later, as it holds the associations among pixels implicitly. Also, due to these relations the gradient space reveals the underlying structure of the image. Accordingly, the gradient space has more discriminating power to discover key facial features.

In addition, we explore the use of a Gaussian to smooth the image, which makes the gradient estimation more stable. These operations make our method more robust; similarly previous research used the gradient space to calculate their code. Consequently, our method is robust against illumination due to the gradient space, and to noise suitable to the smoothing.

4.5 Blurring and noise reduction

Filters are most commonly used for blurring and for noise reduction. Blurring is used in pre processing steps, such as removal of small details from an image prior to large object extraction.

5.CONCLUSION

The project describes an encoding scheme called LDN which is computed from the edge response value using a compass mask. LDN takes advantage of the structure of the face's textures and encodes it efficiently into a compact code. It uses directional information that is more stable against noise than intensity, to code the different patterns from the face's textures. LDN, implicitly, uses the sign information of the directional numbers which allows it to distinguish similar texture's structures with different intensity transitions, e.g., from dark to bright and vice versa.

REFERENCES

[1]T.Ahonen, A. Hadid, and M. Pietik¨ainen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[2]H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under Bayesian framework," in Proc Int. Conf. Image Graph., 2004, pp. 306–309.

[3]H.Jin, Q. Liu, X. Tang, and H. Lu, "Learning local descriptors for face detection," in Proc. IEEE Int. Conf. Multimedia Expo., Jul. 2005, pp. 928–931.

[4] I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," IEEE Trans. Image Process., vol. 16, no. 1, pp. 172–187, Jan. 2007.

[5] S. Liao and A. C. S. Chung, "Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude," in Proc. Asian Conf. Comput. Vis., 2007, pp. 672–679.

[6]T.Ojala, M. Pietik¨ainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. [7]M. [10]Pantic and L. J. M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pp. 1424–1445, Dec. 2000.Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, Jul. 2002.

[7] P.Penev and J. Atick, "Local feature analysis: A general statistical theory for object representation," Network: Computation in Neural Systems, vol. 7, no. 3, pp. 477–500, 1996.

[8] J.Ruiz-del-Solar and J. Quinteros, "Illumination compensation and normalization in eigenspace-based face recognition: A comparative study of different pre-processing approaches," Pattern Recog. Lett., vol. 29, no. 14, pp. 1966–1979, 2008.

[9] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," Image and Vision Computing, vol. 27, no. 6, pp. 803–816, 2009.

[10] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 775–779, Jul. 1997.

[11] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no. 7, pp. 775–779, Jul. 1997.

[12] H. Yang and Y. Wang, "A LBP-based face recognition method with Hamming distance constraint," in Proc. Int. Conf. Image Graph., Aug., 2007, pp. 645–649.

