Face Recognition Using Gray Level Weight Matrix (GLWM)

R.S. Sabeenian[1], M.E. Paramasivam[2], and P.M. Dinesh[3]

[1] Professor & Center Head Sona SIPRO
sabeenian@sonatech.ac.in
[2] Lecturer & Research Staff Sona SIPRO
sivam@sonatech.ac.in
[3] Research Associate, Sona SIPRO,
ECE – Department, Sona College of Technology, Salem, India
pmdineshece@live.com

Abstract. Face detection is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It also has several applications in areas such as content-based image retrieval, video coding, video conferencing, crowd surveillance, and intelligent human–computer interfaces. The purpose of this study is to propose a novel statistical face recognition system with improved performance, based on Gray Level Weight Matrix (GLWM). The process involved in GLWM is an improved version of the Local Binary pattern technique. It has been found out through experiments that the proposed GLWM is more efficient in face recognition.

Keywords: Local Binary Pattern; Gray Level Weight Matrix; Face Recognition.

1 Introduction

Face recognition technology has a variety of potential applications in information security, law enforcement and surveillance, smart cards, access control, etc. For this reason, this technology has received significantly increased attention from both the academic and industrial communities during the past 20 years. The main aim of face recognition is to identify or verify one or more persons from still images or video images of a scene using a stored database of faces. One of the major challenges encountered by current face recognition techniques lies in the difficulties of handling varying poses, i.e., recognition of faces in arbitrary in-depth rotations. Extensive efforts have been put into the research toward pose-invariant face recognition in recent years and many prominent approaches have been proposed. However, several issues in face recognition across pose still remain open. The human face is a dynamic object and has a high degree of variability in its appearance, which makes face detection a difficult problem in computer vision. The face recognition techniques across pose can be classified as general algorithms, 2D and 3D techniques for face recognition across pose and complex neural-network based 3D models. By "general algorithms", we mean these algorithms did not contain specific tactics on handling pose variations. They were designed for general purpose of face recognition equally handling all

V.V. Das, N. Thankachan, and N.C. Debnath (Eds.): PEIE 2011, CCIS 148, pp. 69–72, 2011.
© Springer-Verlag Berlin Heidelberg 2011

image variations (e.g., illumination variations, expression variations, age variations, and pose variations, etc.).The general algorithms can be further divided into two broad categories as Holistic approaches and Local approaches. Some of the commonly available Local approaches are Template matching [1], modular PCA [2], Elastic bunch graph matching [3], local binary patterns [4]. This study primarily focuses on improving the capability and universality of general face recognition algorithms (in particular the local binary pattern), so that image variations can be tolerated, rather than designing an algorithm that can eliminate or at-least compensate the difficulties brought by image variations.

2 Previous Method

2.1 Local Binary Pattern (LBP)

The Local Binary Pattern algorithm [4] was originally designed for texture descriptions. The method of local binary pattern for face recognition divides a given facial image into small regions and computes a description of each region using LBP operator. The operator assigns a label to every pixel of an image by thresholding the 3x3-neighborhood of each pixel with the center pixel value and transforming the result as a binary number. Then the histogram of the labels can be used as a texture descriptor. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa.

3 Proposed Method

In proposed method instead of thresholding the image transforming is made with neighborhood to a texture unit with the texture unit number under the ordering way as shown in figure1.

The transforming conditions

$$E_i = \begin{cases} 0 \text{ if } V_i < V_o \\ 1 \text{ if } V_i = V_o \\ 2 \text{ if } V_i > V_o \end{cases}$$

Where:

V_i = The center pixel value

V_o = The neighboring pixel value

Neighborhood (V_i)

62	85	92
29	40	36
67	36	66

$\longrightarrow$

Texture unit (E_i)

2	2	2
0	1	0
2	0	2

Fig. 1. Texture unit transformation

The values of the pixels in the transformed texture unit neighborhood are multiplied by the weights given to the corresponding pixels. Finally, the values of the eight pixels are summed to obtain a number for this neighborhood. This method considers the