1 Binary Gaussian Receptive Maps

Following the terminology in [7], we refer to the use of Gaussian derivatives for images analysis as Gaussian receptive maps. A local Gaussian scale space is computed as:

\[
I(x,y, \sigma) = G(x,y, \sigma) * I(x,y) = \frac{1}{2\pi \sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} + I(x,y)
\]

\[G^\sigma_n = \frac{\partial^k I(x,y, \sigma)}{\partial x^k \partial y^k} \]

Where \( \sigma \) is the size of the support in terms of the second moment (or variance), \( I \) is the image and \( * \) is the convolution operator. From the preceding equation, the steerable filter [4] response for the Gaussian derivative up to fourth order for an arbitrary orientation \( \theta \) could be defined as follows:

\[G^\sigma_1(\theta) = \cos(\theta)G^\sigma_1 + \sin(\theta)G^\sigma_0\]

\[G^\sigma_2(\theta) = \cos^2(\theta)G^\sigma_1 - 2\sin(\theta)\cos(\theta)G^\sigma_1 + \sin^2(\theta)G^\sigma_0\]

\[G^\sigma_3(\theta) = \cos^3(\theta)G^\sigma_1 - 3\sin(\theta)\cos^2(\theta)G^\sigma_1 - 3\sin^2(\theta)\cos(\theta)G^\sigma_0 + 3\sin^3(\theta)G^\sigma_0\]

\[G^\sigma_4(\theta) = \cos^4(\theta)G^\sigma_1 - 4\cos^3(\theta)\sin(\theta)G^\sigma_1 + 6\sin^2(\theta)\cos^2(\theta)G^\sigma_2 - 4\cos(\theta)\sin^3(\theta)G^\sigma_0 + 4\sin^4(\theta)G^\sigma_0\]

Then to compute Binary Gaussian Receptive Maps, a Local Binary Pattern (LBP) [2] is applied over each Gaussian derivative to assign a label to each pixel of the image by thresholding the \( 3 \times 3 \) neighborhood of each pixel with the center pixel value and considering the result as a binary or decimal number. The mathematical expression for this operation is:

\[BG^\sigma_n(\theta) = LBP(G^\sigma_n(\theta))\]

Where \( BG^\sigma_n(\theta) \) is the Binary Gaussian Receptive map, \( n \) is the order of the Gaussian derivative, \( \sigma \) is the scale factor, \( \theta \) is the steering angle and LBP is the Local Binary Pattern.

The Gaussian Receptive Maps could be efficiently computed at half-octave scales using a linear complexity \( O(N) \) algorithm for computing a Gaussian pyramid [3].

2 Tensorial Representation of \( BG^\sigma_n(\theta) \)

To compute the Tensorial representation, first we divide each \( BG^\sigma_n(\theta) \) of each image into non-overlapping rectangular sub-regions with a specific size. A set of histograms is then computed for each sub-region and finally each histogram is organized in four different 3-D tensors, where each tensor corresponds to an specific derivative order of the Binary Gaussian Receptive maps. The characteristic equation of \( T(BG^\sigma_n(\theta)) \) is shown as follows:

\[T(BG^\sigma_n(\theta)) \in \mathbb{R}_{N_0 \times N_{pos} \times N_{bins}} n = 1, 2, 3, 4\]

Where \( n \) and \( N_0 \) are the order and orientation angles for the gaussian derivatives respectively, \( N_{pos} \) is the number of non-overlapping positions in the map and \( N_{bins} \) is the number of bins used in the construction of each local histogram.

This representation may be projected to a more compact uncorrelated vectorial representation, by performing Multilinear Principal Component Analysis (MPCA) [6] over each tensor.

3 Optimal Configurations for Age Estimation

We have compared two possible algorithms for automatic age estimation using \( T(BG^\sigma_n(\theta)) \). The first algorithm concatenate the tensors \( T(BG^\sigma_n(\theta))_1, T(BG^\sigma_n(\theta))_2, T(BG^\sigma_n(\theta))_3 \) and \( T(BG^\sigma_n(\theta))_4 \) to form a 4-D tensor \( T \) as shown in the next equation:

\[T \in \mathbb{R}^{N_0 \times N_{pos} \times N_{bins}} n = 1, 2, 3, 4\]

MPCA is then applied to this tensor \( T \) to obtain a vector \( Y_F \in \mathbb{R}^m \).

\[Y_F = MPCA(T) Y_F \in \mathbb{R}^m\]

The second algorithm computes MPCA over each tensor separately to obtain the vectors \( y_1, y_2, y_3 \) and \( y_4 \), these vectors are concatenated to form a single vector \( Y_F \in \mathbb{R}^4 \times m \), where \( m \leq M \).

\[Y_F = [y_1 \ y_2 \ y_3 \ y_4] Y_F \in \mathbb{R}^{4 \times m} n = 1, 2, 3, 4\]

Finally, we use Relevance Vector Machines (RVM) [8] rather than Support Vector Machines (SVM) as a regression algorithm.

4 Experimental Evaluation

We have performed several experiments to compare different approaches for estimating age from facial images. Two publicly available databases have been used in our experiments: The FG-NET [1] database and the MORPH [5] database.

References