

Integral Local Binary Patterns: a Novel Approach Suitable for Texture-Based Object Detection Tasks

Eanes Torres Pereira, Herman Martins Gomes
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, PB, Brazil
Email: {eanes,hmg}@dsc.ufcg.edu.br

João Marques de Carvalho
Departamento de Engenharia Elétrica
Universidade Federal de Campina Grande
Campina Grande, PB, Brazil
Email: carvalho@dee.ufcg.edu.br

Abstract—This work is concerned with the proposition and empirical evaluation of a new feature extraction approach that combines two existing image descriptors, Integral Histograms and Local Binary Patterns (LBP), to achieve a representation that exhibits relevant properties to object detection tasks (such as face detection): fast constant time processing, rotation, and scale invariance. This novel approach is called the Integral Local Binary Patterns (INTLBP), which is based on an existing method for calculating Integral Histograms from LBP images. This paper empirically demonstrates the properties of INTLBP in a scenario of texture extraction for face/non-face classification. Experiments have shown that the new representation added robustness to scale variations in the test images - the proposed approach achieved a mean classification rate 92% higher than the standard Rotation Invariant LBP approach, when testing over images with scales different from the ones used for training. Moreover, the INTLBP dramatically reduced the required processing time when searching patterns in a face detection task.

Keywords-Integral Histograms, Local Binary Patterns, Integral LBP, Face Detection

I. INTRODUCTION

One of the most used object detection techniques in images is the sliding window technique [1]. A window of predefined size is slid over the input image with a certain step of displacement. At each slide step, the underlying sub-image is cropped and given as input to an image classifier. This type of processing is very time consuming. In order to overcome the problem of computing features at each step, integral representations have been proposed. Two types of integral representations usually employed are the Integral Image [2] (used to extract Haar-like features) and the Integral Histograms [3]. The basic characteristic of both is to precompute the features before sliding the windows - in this way the features are computed only once.

The majority of face detection systems have two main modules: feature extractor, and feature classifier. Among all the feature extractors used to face detection one that has gained attention of the Computer Vision community is the Local Binary Pattern (LBP) [4]. By means of center-surround processing of pixel neighborhoods, the LBP achieve good rotation invariance. However, the processing of

LBP features at each step of window sliding approach may be time consuming. As the LBP codes are grouped in a histogram, the experiments presented in this paper demonstrate that precomputing LBP features using a combination of the original LBP method and the Integral Histogram representation yields a faster feature processing method, suitable to fast face detection, among other tasks. The experiments also demonstrate that using all LBP codes without grouping them by similar uniformity produce better results in terms of precision and processing time than the traditional approach of grouping the similar uniformity codes in the same labels during histogram creation.

Feature classifiers may use Neural Networks, Support Vector Machines (SVM), Bayesian inference, or a combination of these, among other techniques. A very prominent face detection method is the combination of Adaboost and Integral Image representation, first proposed by Viola and Jones [2]. However, the faces to be detected may be in many poses and orientations, and all the methods that use some variation of the Viola and Jones [2] method must create many different classifiers, one for each pose variation, to tackle the problem of face rotations and varying poses. Since LBP features are invariant to in-plane rotation, a frontal face detector that uses this method does not need to create a combination of many classifiers to detect faces in particular orientations. Within the above context, this paper is concerned with the proposition of a new feature extraction method, Integral LBP, that exhibits relevant properties to object detection tasks (such as face detection): fast constant time processing, rotation, and scale invariance. In the next section, some relevant related work is discussed in the context of feature extraction for face detection, which is the case study chosen in this work to evaluate the proposed method.

II. RELATED WORK

The Local Binary Pattern (LBP) operator has been widely used to represent, detect and recognize faces [5], as well as other object classes [6]. Some reasons for this popularity are the operator simplicity and robustness to variations on

illumination and orientation. LBP was proposed by Ojala et al. [7] as a histogram with N levels ($N = 2^P$, P is the number of neighbors in a predefined mask). The LBP operator was extended in many ways, one of those extensions is the rotation invariant LBP (LBP^{ri}) [4]. In the LBP^{ri} there are only $P + 2$ levels in the histogram. To represent faces using this operator, it is necessary that the spatial disposition of face elements is coded in the extracted feature vectors. In order to achieve that coding, one approach is to divide the face image in regions and extract LBP histograms from each one of those regions, as proposed by Zhang et al. [8]. After that, all histograms are concatenated to form a single feature vector representing the face image.

Although the method proposed by Ojala et al. [4] claims to be gray-scale invariant, another improvement was performed on LBP in order to give it invariance to illumination. Jin et al. [9] proposed a variation of the LBP method, the Improved Local Binary Patterns (ILBP). The authors claim and demonstrate by empirical evaluation, that their method is invariant to illumination. The main difference from the original LBP proposed by Ojala et al. [7] is that instead of using the gray value of the center pixel as the value for thresholding, the ILBP method uses the average of gray values in the mask. In order to test their method, some experiments were performed on face detection. The modeling of the face and non-face classes has been done by using multivariate gaussian modeling over images obtained from FERET database (4000 images). The tests were performed over images obtained from YaleB face dataset (2340 images) and from MIT-CMU testset (47 images). The experiments showed that the method achieved false positive rates of 2.99×10^{-7} and true positive rates over 90%.

Another option to Haar-like features are scale invariant local features due to its high level of invariance to in-plane transforms [10]. It was this high degree of invariance that motivated the approach proposed by Toews and Arbel [10]. They proposed a method to detect, locate and classify, in terms of gender, facial images. The extraction of invariant features is performed using the SIFT technique (Scale Invariant Feature Transform). The appearance models are learned from probabilistic models that use data distribution and bayesian inference, these models are known as OCI (Object Class Invariant) [11]. The majority of experiments were performed over the Color Feret Face Database [12], therefore the results on face detection are not representative due to the well-behaved backgrounds. The strength of the approach is on face recognition and facial expression and gender classification. Besides, the SIFT feature extraction is typically very slow.

There is not yet a face detection approach that is robust on the majority of variations that a face could suffer, such as variations on illumination, facial expression, orientation and viewpoint. What has been used to tackle this great variability problem is to train many classifiers separately,

each one to a type of variability. For instance, Li and Zhang [13] used a pyramid of classifiers, at each pyramid level specific classifiers are trained for specific in-plane face rotations. The usage of many classifiers on a single system may deteriorate the face detection performance. In order to overcome this drawback, the present paper proposes a new approach to represent face images for detection: the Integral Local Binary Pattern. This approach is derived from two well established image representations the Integral Histograms [3] and the Local Binary Patterns, and may be applied to other object detection applications, besides face detection.

The remaining of this paper is organized as follows. The next section describes in some detail the LBP and Integral Histogram methods, and how the proposed method (Integral LBP) combines the two previous methods to achieve better and faster results in the task of face detection. In the Section IV, the experiments are described and discussed. Finally, Section V presents the conclusions and proposals of future work.

III. ROTATION AND SCALE INVARIANT IMAGE REPRESENTATION

This section presents the Integral Local Binary Pattern (INTLBP) approach. The major properties of this representation are its robustness to in-plane rotation (inherited from the Original LBP descriptor), scale changes, and the constant time needed to obtain vector features from any location in an image (inherited from Integral Image Histograms). The experimental results shown in Section IV demonstrate that the constant time feature extraction from any location may be used to create real time face detection systems. This section is divided in three subsections, Local Binary Patterns Descriptor, Integral Histogram Representation, and Integral Local Binary Patterns, respectively.

A. Local Binary Patterns Descriptor

To understand the Integral LBP approach, one must first understand the original multiresolution LBP approach. The aim of this section is to give an overview of the original LBP method. A LBP code is obtained from a thresholding of pixel gray levels, in this thresholding a mask is passed over the image and the center pixel is compared with its neighbors, if the neighbor gray value is greater than the center gray value a 1 is attributed to the code, otherwise a 0 is attributed. The thresholding process is illustrated in Figure 1, and the obtained code for this example is 11011010.

The LBP extension proposed by Ojala et al. [4] is named rotation invariant uniform patterns ($LBP_{P,R}^{ri}$), where P is the number of neighbors and R is the radius from center to neighbor. The $LBP_{P,R}^{ri}$ code has the ability to deal with patterns with invariance to orientation. This ability comes from the observation that there are some codes, called uniform codes, which can be grouped by unique

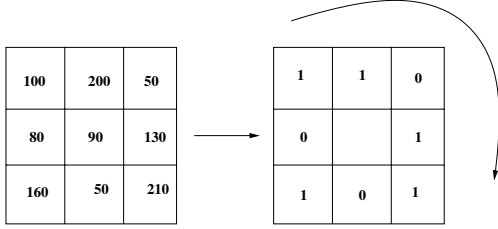


Figure 1. LBP code extraction

labels due to their similar characteristics. For instance, the following codes have a same uniformity, which is equal to 2: 10011111, 11001111, 11100111, 11110011.

Ojala et al. [4] proposed a measure of LBP code similarity based on the number of transitions from 1 to 0 or vice-versa in the codes. The measure establishes that a code is uniform if it has the number of transitions less or equal to 2. To extract the LBP codes and to perform the uniformity measure, the Equations 1 and 2 are used, as proposed by Ojala et al. [4].

$$LBP_{P,R}^{ri2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \quad (1)$$

where P and R correspond to the number of neighbor pixels and coverage radius, respectively. The g_p and g_c correspond to the grey levels of a neighbor pixel and the central pixel.

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (2)$$

In Equation 1, the codes with uniformity measure less or equal to 2 are summed together, and the non-uniform codes are grouped on the label $P + 1$. In the uniformity Equation 2, the differences between the gray value of the center pixel is compared with the values of its neighbors (the s function verifies the signal of the difference between its values).

The number of different codes should be 2^P , but grouping codes with similar uniformity the number of codes is reduced to $P + 2$. The non-uniform codes are grouped in a unique label, named miscellaneous. After the LBP uniform codification of the image, a histogram with $P + 2$ bins is determined using the extracted uniform codes. The generated histogram is used to represent the image, and can be given as input to classifiers in learning tasks. For the particular task of face image learning, the disposition of face components is described by the subdivision of the image in regions. Different histograms are extracted for each region and the histograms are grouped in an unique feature vector. For instance, if the image is divided in 9 regions and $P = 4$, the final feature vector will have 54 elements ($9 \times (P + 2)$).

Therefore, besides obtaining invariance to rotation, the LBP operator is also a very reduced way of representing the texture of images.

In order to perform quantizations in any resolution and obtaining as a result a multiresolution operator, Ojala et al. [4] proposed the creation of feature vectors composed by features extracted using variations of P and R . However, there are some criticisms about those analysis. First, the histograms extracted for different values of P and R must be statistically independent. That independence, generally, is not guaranteed. Besides, the number of elements in the final histogram may be large. For instance, if the image is divided into 9 regions and the values of P equal to 4, 8, 16, and 24, there will be a total of elements equal to $(4 + 2) + (8 + 2) + (16 + 2) + (24 + 2) * 9 = 540$ elements in the feature vector. Therefore, the number of elements in the final vector is dependent on the number of resolutions from which the LBP codes are extracted, and that number may be large and slow down the processing. Finally, the grouping of similar uniform codes in unique labels may induce the lost of accuracy in the results, as will be experimentally demonstrated in the Subsection IV-A.

B. Integral Histograms

A common technique used to detect faces in images is to slide a window with predefined size which is resized until certain value. At each step of this sliding process, the features are extracted from the image region inside the window and they are used as input to a classifier previously trained for that type of patterns. The problem with the sliding window technique comes from the time needed to compute the features at each step.

The Integral Image representation [2] overcomes the processing time problem by precomputing all the possible summations of pixel gray values before the passage of the sliding window. At each step, only few accesses to a precomputed matrix are needed and the summation is done in a constant time for any scale and position. However, there are some criticisms about the usage of differences between summations of gray values in adjacent image regions. Balas and Sinha [14] argue that a collection of edge fragments is a simple way of image representation, but the local processing performed by the edge fragment extraction restricts the generalization strength of the features in adapting to small changes in illumination. Besides, the edge maps implicitly ignore the majority of image information modified by geometrical transformations in the image. Those problems lead to the search for new image representations that could tackle some of them. Wang et al. [15] argues that the best compromise between distributional structure and the retaining of good image properties for class estimation are histograms. Within the context of real time face detection, Integral Histograms [3] is a new technique that is receiving great attention.

For an image with n columns and m rows, an Integral Histogram [3], [15] is represented by a matrix with $(L+1) \times (C+1)$ rows and N columns (n corresponds to the number of bins on each composing histogram). The value of each matrix element can be computed using the Equation 3 [15].

$$H_{x,y}[N] = \sum_{x' \leq x, y' \leq y} \delta(x', y') \quad (3)$$

where $\delta(x', y') = 1$, or $\delta(x', y') = 0$, if the value of the pixel (x, y) belongs or not to the n -th bin of the histogram. As in the Integral Image representation, the histogram of any location in the image may be extracted using a few recurrences of the Equation 4 [15].

$$\begin{aligned} H_{x,y}[n] &= H_{x-1,y}[n] + h_{x,y}[n], \\ h_{x,y}[n] &= h_{x,y-1}[n] + \delta(x, y), \\ n &= 1, \dots, N \end{aligned} \quad (4)$$

For all the columns at any coordinate (x, y) of the histogram matrix, one may obtain the histogram of the region from (x, y) to left and above this point. If the Integral Histogram be precomputed in face detection system, the histograms of any sliding window may be fastly computed and in constant time for any location.

C. Integral Local Binary Patterns

The approach proposed in this paper may be used by any object detection system that requires texture feature extraction by LBP and a sliding window method to crop sub-images for classification, given the generality of the LBP descriptor. The following discussions are focused on face detection because this is a problem that usually needs real time processing.

Although the LBP extraction is reasonably fast, if it must be performed thousands of times, the total processing time may be very high. As mentioned before, the majority of face detection algorithms use the sliding window method for searching faces in images. To understand the time processing problem, consider a image of resolution 320×240 pixels. If the window size is 20×20 and the sliding step is 5, the features will be extracted more than 2600 times for just one scale. If each extraction takes $1/2600$ seconds, the passage of the window in that scale will take 1 second. Therefore, the extraction of features at each sliding of the window is unpractical for real time processing.

The proposed approach is to preprocess the LBP histograms, and at each sliding of the window, to access the precomputed matrix of LBPs. Therefore, this approach has two major stages. First, a LBP image is created. The LBP image is composed by replacing the values of the pixels by the corresponding LBP codes. Second, an Integral Histogram is extracted from the LBP image. Performing those preprocessing stages before sliding a window expedites the LBP extraction and improve the object detection process

in two aspects: rotation and scale invariance. The rotation invariance comes from the original LBP, and the scale invariance comes from the original Integral Histogram.

There is an important difference between the rotation invariant LBP (LBP^{ri}) proposed by Ojala et al. [4] and the rotation invariant LBP used in the proposed approach. As was explained in Subsection III-A, the LBP^{ri} groups similar uniform patterns in the same label. In the approach proposed here, all the uniform patterns are used separately to form the final histogram. For instance, for $P = 8$, the LBP^{ri} histogram would have 10 different bins. The INTLBP would have 128 bins. Although, the number of bins is higher, the final time needed for processing an image is lower than using LBP^{ri} . As will be explained in Section IV, the usage of all uniformity values as different labels yields better accuracy in the results.

IV. EXPERIMENTS AND RESULTS

Two types of experiments were performed: the first one, presented on Subsection IV-A, compares the INTLBP with the traditional LBP^{ri} with regards to scale variations; and the second one, presented on Subsection IV-B, compares the overall time requirements of the two methods. All the experiments were performed on a *Core 2 Duo Intel Centrino* processor with 2GB of RAM. The software was produced using GNU g++ on Linux Ubuntu operating system, no user processes were executing during the experiments.

A. Performance Evaluation Experiments Considering Different Scales

Face images have been acquired from three different sources: the database of facial images collected under the FERET program (2505 frontal images) [12], [16], the Cohn-Kanade AU-Coded Facial Expression Database (8785 frontal images) [17], [18], [19], and the Extended Yale Face Database B (704 frontal images) [20], [21]. The non-face images were extracted from images found in the World Wide Web. All the images were cropped, resized, visually verified, and divided in two sets: training and testing. The training set was composed by 6602 face images and 24636 non-face images, the testing set was composed by 6605 face images and 36187 non-face images. The number of non-face images was higher than the number of face images in order to represent the disparity of existing patterns on real images: in an image there are much more non-face patterns than faces. Four image resolutions were tested: 29×35 , 58×70 , 116×140 , and 232×280 . A rectangular crop was used instead of a squared one, because the cropping process aimed to get all the head, including forehead and chin lines.

The first step in the experimentation was to extract LBP features from the images. Two different implementations of $LBP_{P,R}^{r,i2}$ were used, one similar with that proposed by Ojala et al. [4], and another with the modifications proposed in this paper. The images were divided in 25, 16, 9 and 4 regions

with equal dimensions. For all the experiments the values of P and R were 4 and 2, respectively. Other values of P greater than 4 were not tested because the number of elements in the final vector would be very large compared with the number of pixels in the lower dimension images (29×35 , corresponding to 1015 pixels). Besides that, as the best results of classification were achieved dividing the images into 25 regions, in the lower resolution (29×35) the size of each region is 5×7 pixels, thus it does not permit the use of a high values of P and R . For example, if the combination of parameters $P = 8$ and number of regions equals to 25 had been used, the number of elements in the final vector would be 6400 (this number corresponds to $(2^8 \times 25)$). All the histograms were normalized, the normalization provides a certain robustness to variations of scale, as it will be shown in the experiments. The number of elements in the final feature vector relies on the number of histograms bins and on the number of neighbors of the center pixel in the mask. Therefore, for the original LBP algorithm using 25 regions and $P = 8$, the total number of elements would be 250 (this number comes from $(8 + 2) \times 25$).

For the learning and classification experiments, the LIB-SVM library [22] was used to generate Support Vector Machine (SVM) models. A relevant issue regarding the generation of SVM models is the choice of parameters. In the experiments presented in this subsection, the procedure proposed by Hsu et al. [23] was adopted. That procedure is performed by training and testing different pairs of parameters C and γ over an RBF kernel SVM using cross validation. The best pair of parameters is used to train the final SVM model.

The first set of results is shown on Table I. That table condenses the results for the experiments using images that were divided on 25 subregions. It may be seen that for all face image resolutions, the results for non-face classifications are practically the same (differing by less than 1%). In all the result tables presented in this section, the columns F and NF contain the correct classification rates for faces and non-faces, respectively. The lines MEAN and STD contain the mean and standard deviation calculated column by column from the statistics above those lines. Associated with the correct classification rates are the F-measures (FM) corresponding to the harmonic mean to precision and recall that is obtained from Equation 5.

$$FM = \frac{2 \times TP}{2 \times TP + FN + FP} \quad (5)$$

where TP , FP , and FN are the true positive, false positive, and false negative rates, respectively.

There are only two cases on which the non-face classification using 16 bins overcome the classification using 6 bins by a value higher by more than 1%. That occurred for training resolution of 232×280 and testing resolutions 29×35 and 58×70 . Among all the 16 tests, only in 3 cases the face

classification rates using 6 bins were higher than the face classification using 16 bins. However, the differences among those higher values and the corresponding ones using 16 bins were very low, near 1%. There were some results using 16 bins much higher than the corresponding ones using 6 bins. For example, when using training resolution of 58×70 and testing resolution of 29×35 , in this case the difference was higher by more than 30%. In some tests, null classification rates had occurred because of great disparities between the scales of testing and training images. However, the majority of the tests presented robustness to variations of scale. For all the training resolutions, the performed experiments using images divided on 25 regions achieved values of FM higher to $INTLBP$ than to LBP^{ri} . In the training resolution 58×70 , the mean of FM values was 2.13 times higher to $INTLBP$ than to LBP^{ri} .

Table I
COMPARISON OF RESULTS FOR 16 BINS ($INTLBP$) AND 6 BINS (LBP^{ri}), WHEN DIVIDING THE IMAGES INTO 25 REGIONS.

SCALE		$INTLBP$			LBP^{ri}		
TRAIN	TEST	F	NF	FM	F	NF	FM
232 × 280	29×35	0.00	99.95	0.00	0.05	98.38	0.09
	58×70	25.51	99.97	40.64	26.77	98.64	41.79
	116×140	97.03	99.99	98.49	88.84	99.26	93.72
	232×280	99.79	99.99	99.89	99.44	99.81	99.62
MEAN		55.58	99.98	59.76	53.78	99.02	58.80
STD		50.55	0.02	48.47	48.06	0.64	46.98
116 × 140	29×35	0.00	99.99	0.00	0.08	99.80	0.16
	58×70	90.96	99.99	95.26	87.77	99.50	93.24
	116×140	99.33	99.99	99.66	97.53	99.59	98.55
	232×280	93.93	99.98	96.86	78.38	99.83	87.80
MEAN		71.06	99.99	72.95	65.94	99.68	69.94
STD		47.50	0.01	48.66	44.60	0.16	46.72
58 × 70	29×35	32.58	100.00	49.15	0.68	99.89	1.35
	58×70	99.86	99.99	99.92	97.94	99.78	98.85
	116×140	92.63	99.98	96.16	31.17	99.84	47.47
	232×280	68.99	99.98	81.64	2.89	99.97	5.61
MEAN		73.52	99.99	81.72	33.17	99.87	38.32
STD		30.31	0.01	23.10	45.36	0.08	45.40
29 × 35	29×35	99.15	99.97	99.56	97.94	99.57	98.74
	58×70	32.19	99.99	48.70	0.03	99.88	0.06
	116×140	0.03	99.99	0.06	0.00	99.99	0.00
	232×280	0.00	100.00	0.00	0.00	99.99	0.00
MEAN		32.84	99.99	37.08	24.49	99.86	24.70
STD		46.73	0.01	47.55	48.96	0.20	49.36

On Table II, the results of tests using images subdivided on 16 regions are showed. Similarly to the results on Table I, the non-face classification results are very near for both 16 and 6 bins tests. The face classification results using 6 bins were higher than the corresponding ones using 16 bins on only two cases, both occurred when the training resolution was 116×140 and the testing resolutions were 29×35 and 58×70 , respectively. Using this particular number of 16 regions, some extremely higher results occurred when using 16 bins, when compared with the corresponding scales using 6 bins. For example, for training resolution of 29×35 and testing resolution of 58×70 , the results with 16 bins were higher by more than 92%. For this set of results, in all the cases on which a resolution was tested over images of resolutions twice as higher than the training resolution, the face classification results were higher by more than 90% when using 16 bins. That *multiscale adaptability* does not occur in any case when using 6 bins, on the contrary, there are some cases on which the classifications results are very

degraded for this type of tests. For example, the results using 6 bins, training resolution of 29×35 and testing resolution of 58×70 achieved results lower than 90% when comparing with the corresponding ones using 16 bins. For this number of regions, the means of FM values were almost two times higher to *INTLBP* compared with *LBP^{ri}* in two cases, training resolutions 58×70 and 29×35 .

Table II
COMPARISON OF RESULTS FOR 16 BINS (*INTLBP*) AND 6 BINS (*LBP^{ri}*), WHEN DIVIDING THE IMAGES INTO 16 REGIONS.

SCALE		<i>INTLBP</i>			<i>LBP^{ri}</i>		
TRAIN	TEST	F	NF	FM	F	NF	FM
232	29×35	0.00	99.69	0.00	0.00	99.15	0.00
×	58×70	14.70	99.87	25.60	11.36	98.71	20.17
×	116×140	94.29	99.98	97.05	83.06	98.91	90.21
280	232×280	99.74	99.98	99.86	97.32	99.56	98.42
MEAN		52.18	99.88	55.63	47.94	99.08	52.20
STD		52.16	0.14	50.56	49.36	0.36	49.44
116	29×35	0.00	99.70	0.00	0.18	99.79	0.36
×	58×70	81.38	99.90	89.68	83.39	99.35	90.62
×	116×140	99.18	99.98	99.58	96.02	99.18	97.56
140	232×280	92.93	99.98	96.33	71.19	99.54	82.95
MEAN		68.37	99.89	71.40	62.69	99.47	67.87
STD		46.17	0.13	47.78	42.89	0.2619	45.40
58	29×35	50.33	99.87	66.90	27.89	99.81	43.55
×	58×70	99.41	99.98	99.69	96.99	99.56	98.25
70	116×140	98.05	99.96	99.00	26.45	99.42	41.64
	232×280	83.85	99.94	91.19	5.97	99.75	11.24
MEAN		82.91	99.94	89.20	39.33	99.64	48.67
STD		22.83	0.05	15.36	39.73	0.18	36.22
29	29×35	97.96	99.96	98.95	97.71	99.39	98.54
×	58×70	93.54	99.93	96.63	0.97	99.78	1.92
×	116×140	29.10	99.95	45.06	0.00	99.95	0.00
35	232×280	8.55	99.99	15.75	0.00	99.99	0.00
MEAN		57.29	99.96	64.10	24.67	99.78	25.12
STD		45.23	0.03	40.71	48.69	0.27	48.96

Only four tests with 6 bins achieved results higher than the corresponding ones using 16 bins for face classification when dividing the images on 9 regions, as may be seen on Table III. The average results for face classification were very similar when the training resolution was 232×280 : 48.38% for 16 bins and 48.61% for 6 bins. In three tests, the results using 16 bins were higher than the corresponding tests using 6 bins by a factor more than 40%. It may be inferred from the results of Tables III and IV that the usage of less regions reduces accuracy. For instance, in all the results with 16 bins, using 25 and 16 regions, the accuracy of face classification tests when the training resolution was the same of test resolution achieved results higher by more than 99%. However, for the same cases using the lower quantities of regions none of the tests achieved 99% of accuracy. Another important observation is that the differences in accuracy among the tests with 16 and 6 bins became even more evident when a few number of regions were used. In only one case the mean of FM values was higher for *LBP^{ri}*, it occurred when the images were divided into 9 regions and the training resolution was 232×280 , in all the other experiments the *INTLBP* achieved higher means of FM values. Besides that, the higher robustness to scale variations of the *INTLBP* approach may be perceived when the images are divided into a lower quantity of regions. For instance, when the images are divided into 4 regions, the means of FM values to *INTLBP* are much higher than the corresponding means of *LBP^{ri}*.

Table III
COMPARISON OF RESULTS FOR 16 BINS (*INTLBP*) AND 6 BINS (*LBP^{ri}*), WHEN DIVIDING THE IMAGES INTO 9 REGIONS.

SCALE		<i>INTLBP</i>			<i>LBP^{ri}</i>		
TRAIN	TEST	F	NF	FM	F	NF	FM
232	29×35	0	99.68	0.00	0.82	98.59	1.60
×	58×70	8.99	99.82	16.47	21.68	97.98	35.05
×	116×140	85.74	99.94	92.29	75.35	98.38	85.16
280	232×280	98.80	99.93	99.36	96.58	99.27	97.90
MEAN		48.38	99.84	52.03	48.61	98.56	54.93
STD		51.09	0.12	51.10	44.82	0.54	44.72
116	29×35	0.00	99.65	0.00	1.18	99.52	2.32
×	58×70	79.38	99.87	88.44	81.35	98.93	89.19
×	116×140	96.33	99.93	99.30	96.02	98.69	97.32
140	232×280	85.97	99.89	92.40	66.83	99.31	79.79
MEAN		65.42	99.84	69.74	61.35	99.11	67.16
STD		44.17	0.13	46.66	41.84	0.37	43.81
58	29×35	50.16	99.83	66.73	10.93	99.53	19.62
×	58×70	96.96	99.89	98.40	96.43	99.17	97.77
×	116×140	95.08	99.68	97.32	58.38	99.22	73.36
70	232×280	73.37	99.61	84.45	24.97	99.61	39.84
MEAN		78.89	99.75	86.72	47.68	99.38	57.65
STD		21.94	0.13	14.76	38.11	0.22	34.74
29	29×35	96.41	99.92	98.13	92.25	98.42	95.19
×	58×70	80.39	99.85	89.06	2.63	98.86	5.07
×	116×140	19.77	99.88	32.98	0.00	99.61	0.00
35	232×280	0.59	99.96	1.17	0.00	99.90	0.00
MEAN		49.29	99.90	55.33	23.72	99.20	25.06
STD		46.30	0.05	46.20	45.70	67.87	46.81

Table IV
COMPARISON OF RESULTS FOR 16 BINS (*INTLBP*) AND 6 BINS (*LBP^{ri}*), WHEN DIVIDING THE IMAGES INTO 4 REGIONS.

SCALE		<i>INTLBP</i>			<i>LBP^{ri}</i>		
TRAIN	TEST	F	NF	FM	F	NF	FM
232	29×35	0.00	99.93	0.00	4.29	97.20	8.01
×	58×70	59.64	99.82	74.63	63.60	95.71	75.76
×	116×140	86.37	99.66	92.52	74.46	96.06	83.48
280	232×280	95.53	99.71	97.57	91.20	97.67	94.25
MEAN		60.39	99.78	66.18	58.39	96.66	65.40
STD		43.04	0.12	45.20	37.81	0.93	38.94
116	29×35	16.65	99.73	28.48	4.04	98.89	7.68
×	58×70	87.86	99.73	93.41	80.61	97.46	88.03
×	116×140	92.57	99.71	96.00	89.58	96.97	93.02
140	232×280	76.88	99.83	86.85	62.53	98.29	76.14
MEAN		68.49	99.75	76.19	59.19	97.90	66.22
STD		35.18	0.05	32.04	38.45	0.85	39.66
58	29×35	73.41	99.90	84.62	17.18	99.01	29.08
×	58×70	92.43	99.72	95.93	91.23	97.67	94.27
70	116×140	90.55	99.56	94.82	56.72	97.36	71.18
	232×280	69.39	99.65	81.76	40.53	98.78	57.19
MEAN		81.45	99.71	89.28	51.42	98.21	62.93
STD		11.73	0.14	7.15	31.11	0.81	27.26
29	29×35	93.85	99.55	96.60	87.54	95.90	91.36
×	58×70	33.32	99.90	49.95	1.48	97.83	2.86
×	116×140	1.61	99.96	3.17	0.00	98.90	0.00
35	232×280	0.03	99.98	0.06	0.00	99.61	0.00
MEAN		32.20	99.85	37.45	22.26	98.06	23.55
STD		43.87	0.20	45.56	37.70	1.61	45.22

The results presented on Tables I, II, III, and IV confirm one of the main hypothesis of the presented approach: using all the LBP codes, instead of a reduced amount of codes labeled by similar uniformities, improves the classification results and gives to the LBP approach a considerable robustness to scale variations.

B. Measurement of Processing Time

There are two major approaches for sliding windows over an image in object detection. The first approach is to fix the initial size of a window and, at each cycle of sliding, upsize (or downsize) the dimensions of the window. The second approach creates a pyramid of images, each one corresponding to a downsizing over the original image, and the window sliding is performed over each level of the pyramid using a fixed window size for all the levels. In order to evaluate the time performance of extracting LBP features

using the LBP^{ri} and $INTLBP$, two types of experiments were performed. The first type used a scale varying sliding window over 100 images with resolution of 320×240 pixels. The second type used a sliding window method in a pyramid fashion using the same images of the previous experiments.

The number of windows analyzed over an image is a function of two main parameters: the scale factor, and the step size. The scale factor determines the degree of upsizing over the window. For instance, a common used scale factor is 1.1. Therefore, if the initial window is 29×35 the next size will be 31.9×38.5 , and the resolution will be increased until the resolution achieves the lower size of the image dimension. The window must be displaced on a predefined step, the step size will influence the accuracy of the results and the speed of processing. The measurement of processing time may be influenced by many factors. Because of this, the experiments described in this subsection were performed over 100 images aiming to give an average estimate for the processing time. For the two types of experiments, the scale factors were 1.1, 1.5, and 1.9, the step sizes were from 5 to 25 incremented by 5 at each iteration.

The results for the experiments that considered scale variation of the sliding windows are showed in Figure 2. This figure plots the number of windows versus the processing times for the integral and traditional approaches. It is possible to see that the curves are nearly linear and that the LBP^{ri} curve is noticeably steeper than the $INTLBP$ curve, which indicates that $INTLBP$ is less sensitive to variations on the number of windows when compared to LBP^{ri} . The slopes of the $INTLBP$ and LBP^{ri} curves are approximately 2.87×10^{-5} and 32.92×10^{-5} , respectively. From the above it is possible to imply that the $INTLBP$ will achieve even better processing time results when a great number of windows is used. Therefore, in considering that the accuracy of the classification is better when more windows are used in the sliding window approach, then $INTLBP$ may be considered a better choice for feature extraction in terms of classification accuracy and processing time.

The experiments using image pyramids obtained the same number of windows for certain combinations of parameters. Five groups of three experiments had the same number of windows with different sets of parameters. The image pyramids were formed by four levels, each level corresponding to resolutions of 100%, 75%, 50%, and 25% of the original image. In the case of image pyramids (Figure 3), the processing time of $INTLBP$ was better than that of LBP^{ri} only in the experiments with higher number of images (approximately above 10^5 windows). The explanation for this behavior comes from the fact that at each iteration the LBP image must be created four times, and the LBP image creation is the most expensive part of the $INTLBP$ algorithm. However, as it was observed in the experiments showed in Figure 2, $INTLBP$ curve is also much less steep

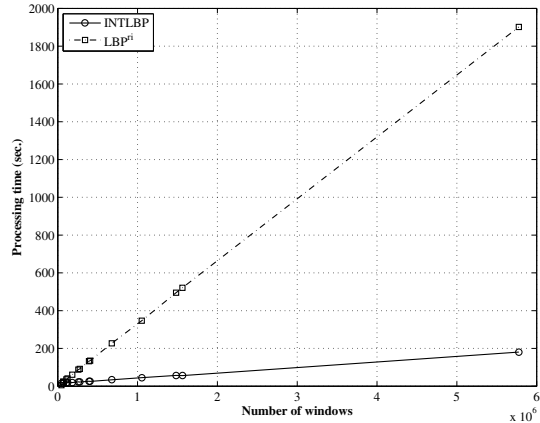


Figure 2. Processing times versus number of windows, considering 100 images of 320×240 pixels and using the sliding window method with scale variations.

than the corresponding LBP^{ri} curve in Figure 3 (slopes of 2.05×10^{-5} and 20.96×10^{-5} , respectively). In summary, the results in Figures 2 and 3 emphasize the better results of $INTLBP$ when compared to the LBP^{ri} method.

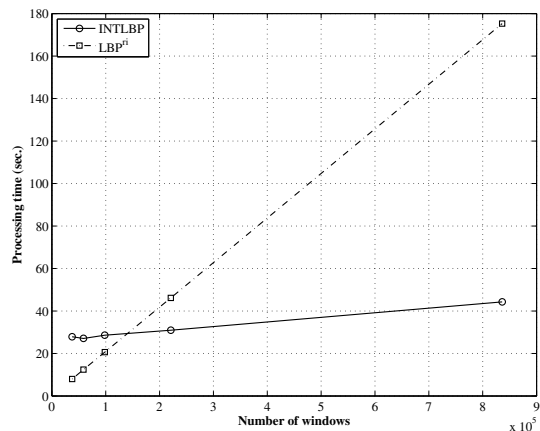


Figure 3. Processing times versus number of windows, considering 100 images of 320×240 pixels using the pyramid sliding window method.

The experiments presented in this subsection have shown that the Integral LBP representation for extracting texture features from images improves the quality of results in two main aspects. First, the Integral LBP representation adds robustness to scale variations in the test images. Moreover, the integral representation improves the processing speed when searching patterns in a general object detection task, such as face detection.

V. CONCLUSIONS AND FUTURE WORK

The Integral LBP approach for texture extraction has experimentally proven to be a better choice to face detection than the Rotation Invariant LBP approach proposed by Ojala et al. [4]. The face detection using $INTLBP$ is faster and may

be formed using a cascade of few SVM models to achieve scale invariance. The most used methods, like Adaboost plus Haar-like features, use many cascades of classifiers, and for each orientation or face view-point a different classifier must be learned, differently from LBP where only one classifier may handle in-plane variations of patterns. As future work, the INTLBP will be applied on a cascade fashion for face detection using very few stages. The few stages will be achieved because the INTLBP is robust to scale and orientation variations and, therefore, does not need the learning of different classifiers for many different scales and orientations.

ACKNOWLEDGMENT

Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office [12], [16]. This research has also used the Extended Yale Face Database B [20], [21] and the Cohn-Kanade AU-Coded Facial Expression Database [17].

REFERENCES

- [1] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008, pp. 1–8.
- [2] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [3] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 829–836.
- [4] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.
- [5] A. Hadid, "The local binary pattern approach and its applications to face analysis," in *First Workshop on Image Processing Theory, Tools and Applications*, 2008, pp. 1–9.
- [6] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proceedings of the Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, 2009, pp. 1–8.
- [7] T. Ojala, M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [8] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block lbp representation," in *ICB07, Lecture Notes in Computer Science*, 2007, pp. 11–18.
- [9] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved lbp under bayesian framework," in *Third International Conference on Image and Graphics (ICIG'04)*, 2004.
- [10] M. Toews and T. Arbel, "Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1567–1581, 2009.
- [11] —, "Detection over viewpoint via the object class invariant," in *International Conference on Pattern Recognition*, 2006, pp. 765–768.
- [12] P. J. Philips and H. Moon, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [13] S. Z. Li and Z. Zhang, "Floatboost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, 2004.
- [14] B. J. Balas and P. Sinha, "Dissociated dipoles: Image representation via non-local comparisons," Tech. Rep., 2003.
- [15] H. Wang, P. Li, and T. Zhang, "Proposal of novel histogram features for face detection," in *ICAPR, Lecture Notes in Computer Science*, 2005, pp. 334–343.
- [16] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The feret database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.
- [17] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, 2000, pp. 46–53.
- [18] Y. li Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 1–19, 2001.
- [19] J. F. Cohn, A. J. Zlochower, J. Lien, and T. Kanade, "Automated face analysis by feature point tracking has high concurrent validity with manual face coding," *Journal of Psychophysiology*, vol. 35, no. 1, pp. 1–23, 1999.
- [20] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
- [21] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
- [22] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using the second order information for training svm," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.
- [23] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Tech. Rep., 2009.