

Facial Expression Recognition with Active Local Shape Pattern and Learned-Size Block Representations

Md Tauhid Bin Iqbal*, *Member, IEEE*, Byungyong Ryu*, Adín Ramírez Rivera, *Member, IEEE*, Farkhod Makhmudkhujaev, Oksam Chae, *Member, IEEE*, and Sung-Ho Bae, *Member, IEEE*

Abstract—Facial expression recognition has been studied broadly, and several works using local micro-pattern descriptors have obtained significant results. There are, however, open questions: how to design a discriminative and robust feature descriptor?, how to select expression-related most influential features?, and how to represent the face descriptor exploiting the most salient parts of the face? In this paper, we address these three issues to achieve better performance in recognizing facial expressions. First, we propose a new feature descriptor, namely Local Shape Pattern (LSP), that describes the local shape structure of a pixel's neighborhood based on the prominent directional information by analyzing the statistics of the neighborhood gradient, which allows it to be robust against subtle local noise and distortion. Furthermore, we propose a selection strategy for learning the influential codes being active in the expression affiliated changes by selecting them exhibiting statistical dominance and high spatial variance. Lastly, we learn the size of the salient facial blocks to represent the facial description with the notion that changes in expressions vary in size and location. We conduct person-independent experiments in existing datasets after combining above three proposals, and obtain an improved performance for the facial expression recognition task.

Index Terms—Local Shape Pattern, LSP, Orientlets, Active Codes, Learned-size Block Representation, Expression Recognition

1 INTRODUCTION

THIS recent years, Automatic Facial Expression Recognition Research (AFER) has put substantial impact on different areas of human-centric computing, such as emotion analysis, affective computing, and robot control [1]. Since different expressions can be characterized with the appearance changes of the face [2], [3], efficient representation of the expression-related appearance-features is a crucial task in AFER. However, due to the different facial traits and external noise factors, the representation of such features should be, simultaneously, discriminative and robust, which is challenging in practice. Moreover, describing the facial expressions using the most active regions on them is beneficial since not all the regions of the face are active in expression changes [4], [5], [6], [7], [8]. Nevertheless, selecting these active regions is challenging due to the diverse facial appearance and expressions of different individuals.

There are a number of feature descriptors available in the literature, where appearance-based descriptors are mostly

used. Such methods apply image-filters on the face, either globally to generate holistic features, or locally, to extract micro-level local features of the face image. Even though the global features, such as Eigenfaces and Fisherfaces [9] have been studied widely, local feature descriptors [2], [3], [10], [11] are popular due to their computational simplicity and illumination-robustness. Among the local descriptors, Local Binary Pattern (LBP) [10] is the most popular one, showing its robustness in monotonic illumination-variations. Later, edge-based descriptors have gained attention due to their superior performance over LBP in recognizing facial expressions [2], [3], [11], [12], [13]. Edge-based descriptors use top compass-mask responses to generate their codes, aiming at representing the principal edge direction of the local texture. Although this approach produces codes by extracting the consistent prominent direction of local edges, inconsistent codes may be produced in textures having multiple edge-directions, such as, corner, curve, branches, and more noticeable on flat regions. Hence, it may lose discriminating power. Moreover, these descriptors are prone to subtle local distortion and noise due to the use of small local regions. Such problems are addressed in recently proposed Neighborhood-aware Edge Directional Pattern (NEDP) [13], nevertheless NEDP may at times suffer to preserve the global shapes due to the inflexibility to explore a more wider region of neighborhood since the baseline coding scheme is confined to 3×3 neighborhood only.

Other existing works [14], [15] aim to reduce the effect of noisy codes from flat region by applying a threshold on primary edge response. This operation, however, eliminates a number of pixels from the face image. Therefore, the reduced number of samples introduce sampling errors

• *Equal Contribution.

• Md Tauhid Bin Iqbal, Oksam Chae and Sung-Ho Bae (Corresponding author) are with the Department of Computer Science and Engineering, Kyung Hee University (Global Campus), Yongin-si 17104, South Korea. Email: {tauhididq, oschae, shbae}@khu.ac.kr

• Byungyong Ryu is with POSCO ICT Center, Pangyo-ro 255, South Korea. Email: read100nm@khu.ac.kr.

• Adín Ramírez Rivera is with the Institute of Computing, University of Campinas, Campinas 13083, Brazil. Email: adin@ic.unicamp.br.

• Farkhod Makhmudkhujaev is with the Dept. of Information and Communication Engineering, Inha University, Incheon 22212, South Korea. Email: farkhodfm@inha.ac.kr.

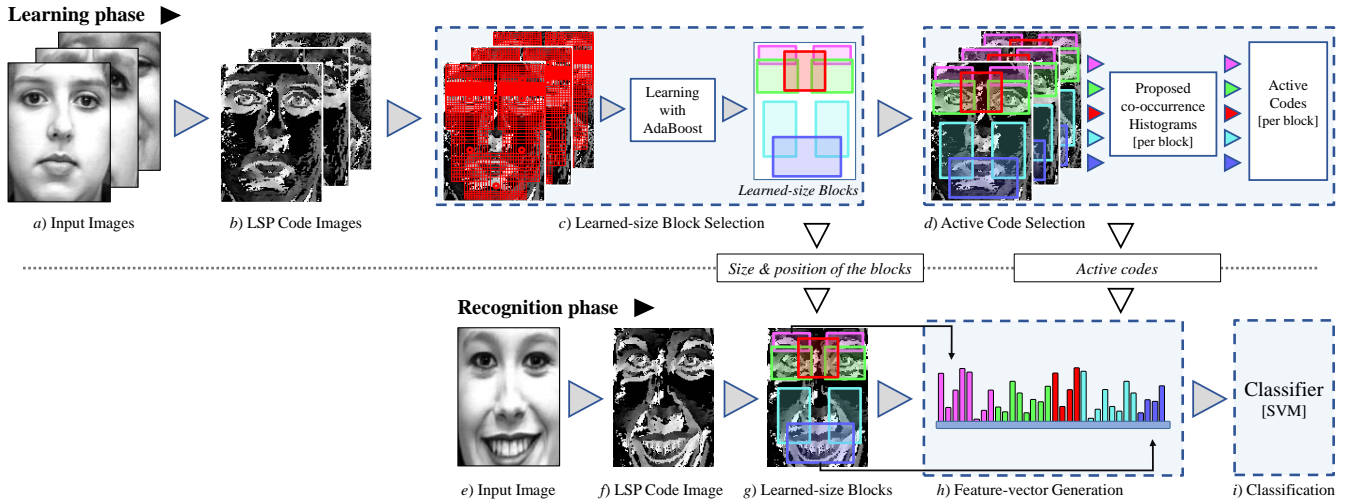


Fig. 1: The overall process of our proposal. [Learning phase] (a) A stream of input images and (b) corresponding LSP code image are shown. (c) Blocks of different sizes are set on the training images, from where optimal blocks are learned by AdaBoost. (d) Proposed co-occurrence histogram is applied to each learned blocks and active codes (per block) are selected. [Recognition phase] (e) An input image and (f) its LSP code image is shown. (g) The blocks are set on the image based on the size and position learned from the above learning phase. (h) A feature-vector for the input image is generated by combining the histogram of active codes from each of the learned-size blocks. (i) Feature-vector is sent to classifier (SVM) for the classification task.

while generating the code histogram of the face, as pointed by Ryu et al. [3]. These authors [3] tackle this issue by selecting a number of codes that are highly related to expression changes by accumulating the codes with statistical dominance. Using the accumulation alone, however, is disadvantageous too, since this strategy may select common featureless pixels as the active codes, and, thus, hinders the performance. Moreover, Ryu et al. [3] uses the same active codes for all the facial blocks to generate code histogram. It is evident that due to the diverse changes of expressions, facial components change differently and exhibit diverse shapes of features. Hence, applying same active codes over all the facial image may also not provide the maximum benefit.

Similarly, most methods still represent the description of the face with a collection of code histograms extracted from uniformly divided face blocks [2], [10], [11]. Since this uniform-grid histogram approach uses whole face image for the description, unnecessary person-specific information, such as hair, chin, background etc. are included in the feature description, leading to inappropriate expression-affiliated information. Most importantly, this approach cannot ensure consistent representation of the facial components in the same block since expression-changes may easily displace a facial component on a different block [4], [5]. Moreover, misalignment of the face may also lead to serious inconsistency of such representation. Some prior works [16], [17], however, use specific facial blocks with predefined fixed size and location. Nevertheless, location and size of the salient blocks usually vary person-to-person and may change under improper face-registration. Hence, such predefined block geometry results in poor performance in practice. Researchers tried to tackle this issue by learning the information of salient blocks using different learning techniques, such as multi task sparse learning [18]. Moreover, Zhang et al. [19] used multi-scale Gabor features to train with Adaboost to select the salient blocks. Nevertheless, the

position and size of the salient blocks in their work [19] vary highly when trained with different databases, pointing its inappropriateness in real-world scenario. Other works [5], [6] learn salient blocks from the facial image based on their dominant classification accuracies. Despite their promising results, the learned blocks are all uniform in sizes, for all the parts and expressions.

To make a summary to the above-mentioned limitations of existing works, i.e., the lack of discriminating power and robustness of local descriptors, inefficiency in preserving active feature information through codes, and the inconsistency in representing salient facial components, hinder the accuracy of the state-of-the-art methods. In this paper, we overcome these issues through the following proposals.

- First, we introduce a new feature descriptor, Local Shape Pattern (LSP), utilizing the statistics of neighboring pixels' gradient information to represent discriminating shapes on the local textures. The rationale behind the use of statistics (i.e., histogram of gradient orientations) of neighboring pixel is to incorporate the flexibility to explore a wider region of pixels, ensuring a consistent representation of the local texture's shape despite having subtle distortion and noise. Moreover, we apply preset structural restrictions on the gradients while accumulating them, which ensures the strong evidence of an edge boundary that passes through and simultaneously, avoids the futile noisy accumulations.
- We also propose a learned-size salient block selection strategy for the facial image, where we initially set different-sized blocks on the emotion-related facial components such as eyes, outer brows, brow, mouth, & side-mouths, and then learn the most salient blocks using Adaboost with our proposed feature descriptor yielding consistent expression-affiliated information. The highlight of using the learned-size blocks is that the learning structure around particular inter-

est points enhances the robustness and description capabilities of the final face descriptor while removing person-specific information and simultaneously, providing resistance against misalignment of facial images.

- Moreover, we improve the existing expression-related active code selection method [3] by incorporating a combined representation of spatial variance and statistical dominance information of the codes (within a co-occurrence histogram). Unlike previous methods [3], we learn distinct active codes for different significant facial regions (i.e., above-mentioned learned-size blocks), making it efficient to represent the most active expressive features for that particular region.
- Lastly, we present an efficient technique to represent the selected active LSP codes together with the learned-size blocks within its different sub-block divisions, achieving maximized performance.

We demonstrate the discriminability, robustness and efficiency of our above proposals on different existing facial expression recognition datasets. An overall flow of our method is depicted in Fig. 1.

2 LOCAL SHAPE PATTERN (LSP) DESCRIPTOR

In facial expression recognition, the appropriate definition of the local facial-texture shapes is key to represent the changes in appearance. Existing edge-based descriptors [2], [11], [12], [15] have shown that the directions of local edges can be used as a significant cue to define such shape changes. However, most of these works utilize principal orientations of the target pixel to represent local feature code. Such extraction of feature from single (target) pixel is often disadvantageous due to not considering individual directional information from other neighboring pixels, ending up with inconsistencies against fluctuations of local intensities and subtle noise, as pointed by Iqbal et al. [13].

Unlike the above line of works, we propose utilizing gradient information from the local neighborhood to have a wider perspective of the structure of underlying edge shape. Considering gradients from all the neighboring pixels may, however, arise noise in the shape description. Hence, we consider the orientation consistency of neighboring pixels with respect to the center (target) pixel to omit such random noisy variations of shape, as well as we analyze the local statistics of neighboring gradients in order to reliably extract the direction of underlying edge. The rationale behind this idea is that the neighboring pixels (falling) in the direction of the edge passing through the center pixel show near-similar gradient orientations, and accumulation of such orientations into histogram allows getting an estimate of the edge passing through local region. However, a simple accumulation may only provide a noisy estimation of edge-like structures, since not all neighboring pixels share similar gradient characteristics. Therefore, we propose to constrain neighboring orientations in case they do not comply with predefined template-orientations (we name them as *orientlets*) with respect to the center pixel while the accumulation process. In this way, only the selected orientations, showing correspondence to the solid edge structures, will

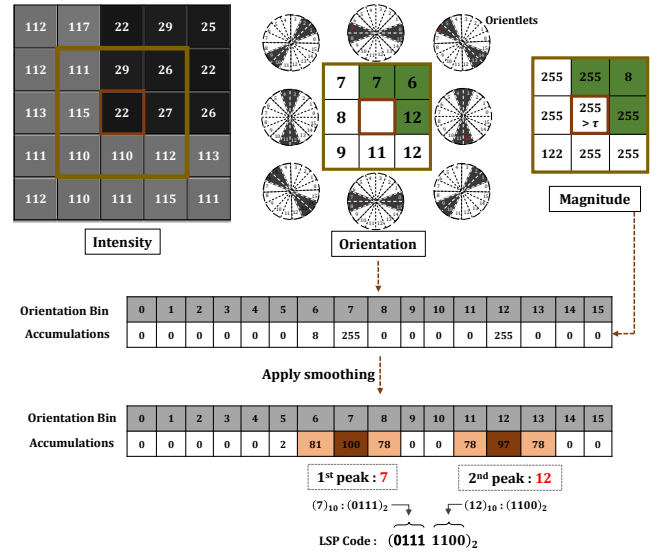


Fig. 2: LSP code computation process for a sample patch (3x3). Gradient orientations and magnitudes of the neighboring pixels are shown. Magnitudes are accumulated at the respective orientation bins when orientations comply with their orientlets (green color). Accumulations are smoothed and its peaks are selected as the principal directions. These peaks ($k = 2$) are concatenated to create the LSP code.

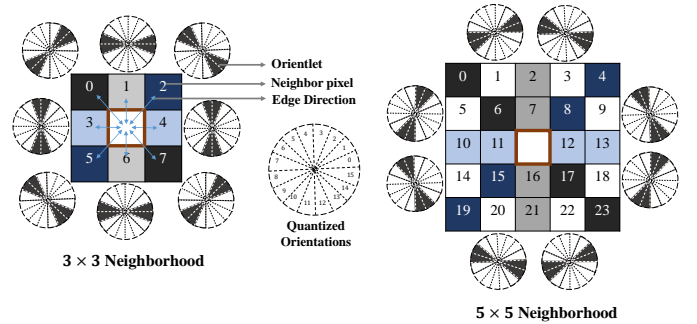


Fig. 3: Diagram of a set of orientlets (ideal orientations), $\mathcal{O}_p(q)$, for each neighbor, q , of a given pixel, p (marked in brown box). In this case we show the quantized orientations, $\hat{\theta}$, that helps define the orientlet. Note that neighbors in 5x5, colored similarly as in 3x3, contain same orientlets as shown in.

be accumulated, and dominant orientations will appear as peaks in the histogram. These peaks represent the direction of different edges going through the center pixel, which in turn represents the local shape structure of the neighborhood. Eventually, we represent the local shape structure by generating a feature code using the dominant k -orientation bins (peaks). An overview of the proposed code computation is shown in Fig. 2.

2.1 Defining Orientlets

We need to ensure whether the gradient orientation of a neighboring pixel (q) supports the existence of an edge going through this pixel and the center pixel (p). To investigate such supports, we define a set of ideal orientations for a neighboring pixel when it is connected with the center pixel by an edge. We call these ideal orientation templates *orientlets*. We detect these connections by analyzing the orientation-patterns of the pixels, since when an edge exists

between the neighboring pixel and the center pixel, the orientation of the neighbor pixel appears orthogonal to the direction of the edge [20], [21]. In Fig. 3, we provide a representative example of the orientlets for the neighbors within 3×3 and 5×5 region, where the orientlets are produced with 16 quantized orientations. The small blue-arrow lines connecting the neighbor pixels and the center pixel (in 3×3 Neighborhood) denote edge-segments passing through them. For such an edge (edge can be prolonged at practice), the ideal orientation at that neighbor appears orthogonal to the direction of the respective edge; i.e., for a vertical edge at neighbor 1, its possible orientlets are located horizontally in the figure. Note that for a particular neighbor, orientlets are shown in either direction because the edge may have two different contrasts, e.g., dark-to-bright and vice versa. One may vary the stride of orientlets in order to allow natural variations of edge; e.g., a stride of 2-bins is shown in this figure. Mathematically, for a neighbor q to have an edge with center pixel p , its orientlet set $\mathcal{O}_p(q)$ can be defined as

$$\mathcal{O}_p(q) = \bigcup_{\hat{\theta} \in [1, i]} (\hat{\theta}), \quad \text{where } \hat{\theta} \perp \mathcal{E}_p(q), \quad (1)$$

where, $\hat{\theta}$ is a value within i -quantized orientation bins. The set $\mathcal{O}_p(q)$ is comprised by $\hat{\theta}$ s orthogonal (denoted as \perp sign) to the direction of the edge, $\mathcal{E}_p(q)$, connecting p and q .

2.2 Coding Scheme

We like to generate the signature of underlying edge of a pixel by observing the accumulations of its neighboring orientations. Since neighbors falling in the direction of the edge show similar orientations, peaks in the histogram of these orientations will nicely reflect the directional axes of the edge. In the histogram, we accumulate the gradient magnitudes in the respective orientation-bin since magnitudes represent the strength of the edge and hence offer a reliable representation of such directional axes.

Formally, to generate the accumulation histogram H for a particular pixel p , we traverse its neighborhood, \mathcal{N}_p , and accumulate the gradient magnitude of each neighbor q in its respective orientation that belongs to its predefined orientlets, by

$$H_p(i) = \sum_{q \in \mathcal{N}_p} \gamma_p(q) \delta(\hat{\theta}(q), i), \quad \forall i, \quad (2)$$

where i is the bin of the histogram (within total quantized orientations). Function $\gamma_p(q)$ returns the magnitude $M(q)$ of neighbor q when $\hat{\theta}(q)$ belongs to $\mathcal{O}_p(q)$

$$\gamma_p(q) = \begin{cases} M(q) & \text{if } \hat{\theta}_p \in \mathcal{O}_p(q), \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

and, δ is the Dirac's delta function

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Note that we only allow an accumulate in H_p when the orientation of the neighbor belongs to its predefined orientlet-set. In this way, we ensure the accumulations of strong edges only that are going through the center pixel, which in turn, omits the inclusion of other outlier structures present in

the local neighborhood. In our experiments we utilize the widely used 3×3 Sobel kernels [22] to calculate the gradient magnitude and orientation. Although any other kernel can be used.

The peaks in histogram H_p represent the signature of edge boundaries passing through the center pixel, and hence we select top k -peaks to reliably represent such boundaries. Nevertheless, H_p may get affected by some noisy accumulations that may suppress other important peaks. To reduce the effect of such noise, we smooth the histogram with a $1 \times g$ Gaussian kernel G , by

$$\mathcal{H}_p = H_p * G. \quad (5)$$

Such smoothing operation helps generating sharp peaks in the histogram, leading to a clear representation of edge boundaries. Now, a bin i from this smoothed histogram \mathcal{H}_p is regarded as a peak if its accumulation is maximum than its preceding and succeeding accumulations; that is, it should be a local maximum within a predefined window. At first, we select a set of i indices (candidate peaks) those are strictly highest within a window of length $2a + 1$,

$$\mathcal{P}_p = \{i : \mathcal{H}_p(i) > [\mathcal{H}_p(j)], |i - j| < a\}, \quad i \neq j, \forall j, \quad (6)$$

where \mathcal{P}_p contains the set of candidate peaks, where the top peaks denote the principal directional axes of the edge. Therefore, we select top k -peaks based on their highest accumulations from \mathcal{H}_p ,

$$\mathcal{P}_p^k = \arg \max_k \{ \mathcal{H}_p(z) : z \in \mathcal{P}_p \}, \quad (7)$$

where $\arg \max$ operator returns k -values from the set \mathcal{P}_p based on their respective top accumulations from the histogram \mathcal{H}_p . We define this selected set of top k -peaks as \mathcal{P}_p^k . Finally, we generate our LSP code using these top k indices by concatenating their binary values through

$$\text{LSP}(p) = \begin{cases} \big\|_{k=1}^k (\hat{\mathcal{P}}_p^k)_2 & \text{if } M(p) > \tau \text{ and } \mathcal{P}_p^k \neq \emptyset, \\ \omega & \text{otherwise,} \end{cases} \quad (8)$$

where, $(\hat{\mathcal{P}}_p^k)_2$ is the binary representation of the k^{th} top direction from the set \mathcal{P}_p^k (7), $\|$ is the concatenation operator (of binary numbers), $M(p)$ is the gradient magnitude of the pixel p , τ is a threshold defined to avoid insignificant low responses from flat regions, and ω is a default code defined for low magnitude pixels (i.e. flat regions). We only generate a code for a pixel where its magnitude value passes the threshold, τ , to discard such featureless flat pixels. The threshold can be selected either adaptively or empirically; in our approach we follow an adapting way of selecting such threshold described by Iqbal et al. [13].

3 LEARNED-SIZE BLOCK SELECTION

As discussed in Section 1, traditional uniform-block representation of facial image often includes non-expressive features and at times shows its proneness to positional variations of facial components, limiting the overall recognition performance. Hence, we opt to use a learned-size block definition for our facial descriptor to cover only the important facial regions related to expressions along

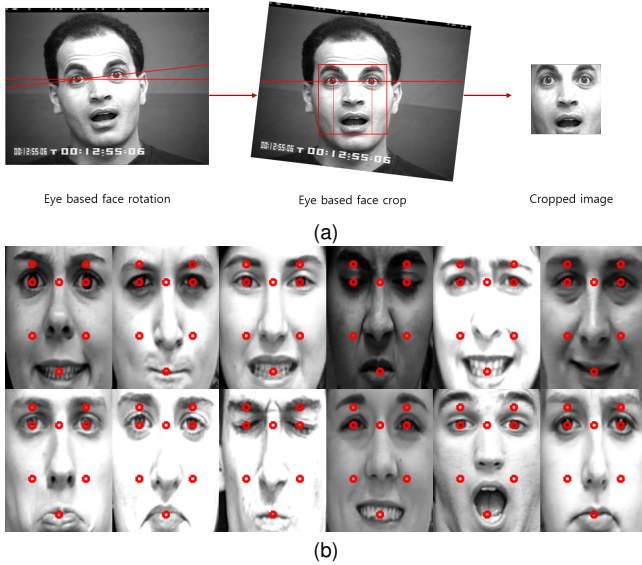


Fig. 4: Setting of reference-points. (a) An example of face-cropping based on eyes and mouth detection. (b) Example of automatically cropped faces, and their facial features (reference points) depicted with red circles that show that the interesting features are within the face for further analysis.

with their appropriate variations in size. According to the Emotion Facial Action Coding System (EMFACS) [23], eyes, outer brows, brow, upper nose, and mouth are significant features that change on facial expressions. Hence, we set learned-size blocks to include these six facial components approximately while allowing overlapping. In addition, we include the sides of the mouth as additional facial features for learned-size blocks as they also change greatly according to variations of expressions. An example of the location of the learned-size blocks in different facial regions (e.g., the left and right eyes, left and right outer brows, left and right sides of mouth, brow, and mouth blocks) can be found in Fig. 7(b).

3.1 Settings of Learned-Size Blocks

In order to set the desired blocks around the above-mentioned facial features, it is important to locate their accurate positions. For this, first we crop the facial region from the given image in such a way that those facial features are located at similar positions after cropping.

To perform this operation appropriately, we consider using eye and mouth-detection approach since consistent results have been shown in detecting eyes and mouth in the literature [24], [25]. Now having the locations of eyes and mouth, first, we compute the in-plane rotation angle of the eyes to align the face and then calculate the horizontal distance between the two eyes, and the vertical distance between the eyes & mouth to define rectangle parameters of the face. We now crop this rectangle (face) area and normalize it to 120×160 resolutions for further processing. Fig. 4(a) presents the overall process.

The above cropping strategy ensures the target facial features to appear at similar positions. Some examples from CK+ dataset [26] are given at Fig. 4(b), where it is shown that the reference points (red circles) are defined at similar positions among various individuals and expressions.

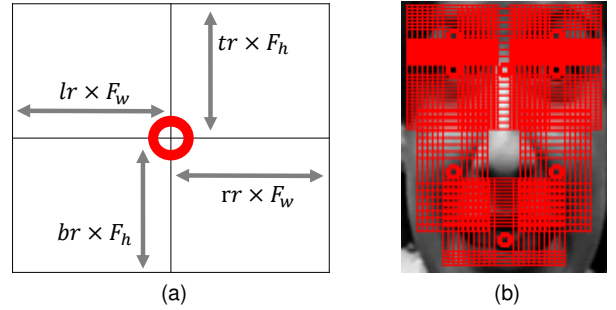


Fig. 5: (a) Definition of a learned-size block using the left, right, top, and bottom ratios (lr , rr , tr , and br , respectively) from the image width, F_w , and height, F_h , and the center of the reference point. (b) Possible block (45 194) over the reference points that are used for AdaBoost learning.

TABLE 1: Percentage of the image width, F_w , and height, F_h , that defines the x and y coordinates, respectively, for each facial reference point used on the learning phase of the learned-size blocks.

Reference Point	x	y	Reference Point	x	y
Right eye	0.25	0.25	Brow	0.5	0.25
Left eye	0.75	0.25	Mouth	0.5	0.875
Right outer-brow	0.25	0.125	Right side-mouth	0.25	0.625
Left outer-brow	0.75	0.125	Left side-mouth	0.75	0.625

Afterwards, we define eight reference points for the above-mentioned eight facial features that are relative to (cropped) image height and width, as detailed in Table 1. We now utilize these points to set the possible sizes of candidate blocks around the target facial features. For each block around such a reference point, we independently set left (lr), right (rr), top (tr), and bottom (br) size-ratios with respect to width (F_w) and height (F_h) of the face, as shown in Fig. 5(a). Such a setting having these size variables and reference point (as parameters of each block) allows each block to vary in sizes.

3.2 Optimal sizes of Learned-Size Blocks

In the above sub-section, we describe the strategy to set possible sizes of blocks around the reference feature-points. Among these candidate sizes, we now look for the most optimal block-size (for each reference point) that boost the overall accuracy. We observe that the size of each block should be sufficiently large to contain variations of each facial feature (regarding the change of expressions) while accommodating localization-errors of the reference point. However, a large increase of sizes will include person-specific information within the block while a small-sized block may still suffer against localization-errors and positional variation of features, raising uncertainty in the recognition performance.

Inspired by existing research utilizing *boosting* to learn a few of most efficient sub-regions [10], [27], we apply AdaBoost on the candidate blocks to choose the best set of blocks out of all candidates. According to above-mentioned block-settings, we define a total of 45 194 possible sizes (as shown in Fig. 5(b)) to generate as many different blocks to be used for training and testing the weak classifiers of AdaBoost. To perform training and boosting, LSP histogram is calculated from each block and then, utilized as a feature vector of each weak classifier. In previous works [10], [27],

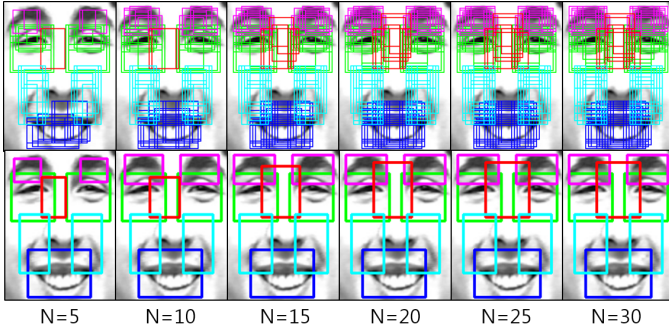


Fig. 6: Top row: blocks corresponding to top- N weak classifiers of AdaBoost for respective expressions. Bottom row: optimal sizes of learned-size blocks according to the top N weak classifiers ranging from 5 to 30.

mean-histograms were generated for each of the blocks in the training set, where distance from such mean-histograms were calculated to test weak classifiers in Adaboost. Instead of using such distance-based approach, we use SVM to train weak classifiers for our purpose. We divide available data into 10 partitions, where we first train SVM for each nine-partition, and then we apply the trained SVM to test weak classifiers on the remaining partition.

For the experiment, we use CK dataset with provided manual annotations. According to the number of available (six) expressions in CK, we generate six AdaBoost learning results for each reference point, on which candidate blocks with different sizes were set beforehand (shown in Fig. 5). We now select top N -weak classifiers (blocks) for each of the six results. Thus, we have a set of $8N$ blocks for each expression. We then determine the optimal size to contain all $8N$ blocks. Here, if the left and right symmetrical reference points (eyes, eyebrows, side-mouth) are learned differently, they are fitted to the larger size blocks for stable information extraction. Fig. 6 shows the selected blocks (by Adaboost) and their optimal sizes with N ranging from 5 to 30. The selected blocks are always the eight those contain the area of most discriminative weak classifiers calculated from AdaBoost.

4 ACTIVE LSP CODE SELECTION

In practice, not all the codes generated by local descriptors, including proposed LSP, are useful for a particular task (e.g., expression recognition). To be specific, some local patterns rarely occur in the images while some patterns belong to person-specific textures, and hence, they may arise unwanted ambiguity in the feature description. Similar considerations can also be found in previous work [28], [29]. To make LSP-based facial description highly useful, we propose to select active set of LSP codes those correspond highly to expression changes, contributing towards additional discrimination capability. Previous work on selecting active codes for facial expressions [3] utilizes whole facial image to select the such codes based on their highest occurrences, which are then applied to specific user-defined expression-affiliated regions (e.g., eye-brows, eyes, nose, and mouth) to generate final feature-vector. However, since all the facial parts do not correspond to expression changes,

codes selected from the whole face may still contain redundant codes.

Another drawback of Ryu et al.'s work [3] in selecting active codes is the sole use of occurrence value of the codes, which may at times include irrelevant codes. For example, facial image of old-aged person may contain person-specific information (e.g., wrinkles, furrows, etc.) that are somewhat irrelevant to expression changes, and a high occurrence of the codes at these textures may lead to select them as the active one. However, we observe such person-specific information occur sparsely (only in specific region for specific person), and hence the overall spatial spreadness (variance) of such codes are low. On the contrary, codes related to expression changes vary highly within the spatial positions, showing their high variance along with their high accumulations. Therefore, we look for the codes with high accumulation along with high variance to select them as the most active codes. Moreover unlike previous work [3] learning such codes from the whole face, we learn them from the salient learned-size blocks only to avoid codes from non-expressive features and apply the learned codes to the respective blocks, accordingly.

To elaborate it, we first calculate LSP codes (8) at each pixel within the learned-size blocks, and generate a co-occurrence histogram for each block. In the co-occurrence histogram, one axis contains all the LSP-code values while the other axis corresponds to the row-wise occurrence of them. Since most expression-related textures change horizontally while an expression being active, we observe a dominant vertical motion in the facial image, and hence, we analyze the row spreadness for the codes, instead of a joint space. We create this histogram by accumulating all the c valid LSP codes over the coded training data for the pixels within block \mathcal{B}_b as follows,

$$\text{CH}_b(r, c) = \frac{1}{s^2 T} \sum_{t=1}^T \sum_{p \in \mathcal{B}_b^r} \delta(\text{LSP}(p)_t, c), \quad (9)$$

where, $\text{CH}_b(r, c)$ denotes an entry in the co-occurrence histogram for the r -th row of b -th block (denoted as \mathcal{B}_b^r) for the LSP code c . δ function is the Dirack's delta (4), and the coding function LSP (8) is computed for each pixel, p , for r -th row within the block b of length s , and for every training image t (with a total of training images T).

Simply saying, the co-occurrence histogram CH_b represents the occurrence of the LSP codes along the spatial region of block \mathcal{B}_b , which, in turn, represents the distribution of each code for that interest block. We now use these distributions to select the specific codes that show high occurrence and spread (variance) within the block. Note that the accumulation is important to show the overall importance of the code while the variance is used to suppress the inclusion of person-specific information, as discussed earlier in this section.

In practice, to select the set of active codes within a block b , we define a score function, S , for each code, c ,

$$S_b(c) = \alpha_b(c) \sigma_b^2(c), \quad (10)$$

where $\alpha_b(c)$ and $\sigma_b^2(c)$ denote the spatial accumulation and variance of the code c , respectively. Among them, the

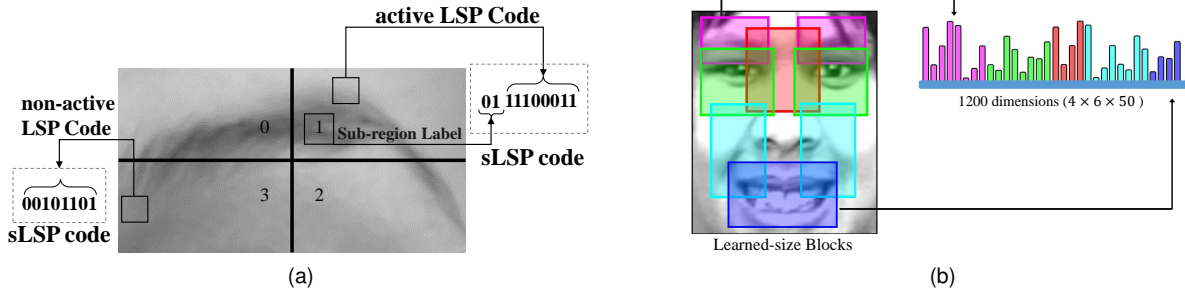


Fig. 7: Construction of the face-descriptor. (a) Division of an example eye-brow block, where spatiality is added to the active LSP codes by assigning labels (identifier) to the sub regions and then combining them together with the respective codes. (b) Face descriptor, \mathbb{F} , by concatenating the histograms of sLSP codes from each learned-size blocks, shown in different colors.

accumulation ($\alpha_b(c)$) is computed spatially by

$$\alpha_b(c) = \sum_r \text{CH}_b(r, c), \quad (11)$$

and the variance, $\sigma_b^2(c)$, is computed by

$$\sigma_b^2(c) = \frac{1}{s^2} \sum_r (r - \bar{r}_{b,c})^2 \text{CH}_b(r, c), \quad (12)$$

where the size of the histogram equals the length s of the block, and the mean coordinate is defined by

$$\bar{r}_{b,c} = \frac{1}{s} \sum_r r \text{CH}_b(r, c). \quad (13)$$

For each learned-size block, b , we select the n active codes as those with the highest n scores by

$$\mathcal{A}_b = \arg \max_c^n \{S_b(c)\}, \quad (14)$$

where the $\arg \max_c^n$ operator returns the set of arguments c that correspond to the top n maximum elements (scores) from the input set, and S_b is the score function (10) for a code. The set \mathcal{A}_b represents contains the active codes regarded as the most meaningful task-related codes (excluding redundant codes) for b -th block.

5 FACE DESCRIPTOR

Typically, the codes generated by local descriptors are pooled over uniformly divided facial regions using a histogram of code frequencies. Then, these histograms are used as the feature vector representing the input face image [2], [10], [11]. Instead, we make use of the learned-size facial blocks (Section 3) to generate the feature vector for our purpose. In fact, we include additional zoomed-in spatial information to the active LSP codes while generating the code-histogram of a facial block since such added spatiality in the facial description contributes towards performance boost, as showed in Ryu et al.'s work [3].

In order to generate facial descriptor, we obtain the learned-size blocks, as defined in Section 3. Further, we divide each block into sub-regions and assign a sub-region identifier (label) into LSP code by appending that unique label to it, as shown in Fig. 7(a), so that a zoomed-in spatial information is also added in the code. Formally, the spatial LSP code (sLSP) within a block b is

$$\text{sLSP}(p)_b = \begin{cases} L(p)_2 \parallel \text{LSP}(p)_2 & \text{LSP}(p) \in \mathcal{A}_b, \\ \text{LSP}(p) & \text{otherwise,} \end{cases} \quad (15)$$

where $L(p)$ is a location function that returns the identifier of the sub-region for the pixel p , LSP is the code function (8), \parallel is the concatenation operator (of the binary representation of the numbers, depicted by the subscript $_2$), and \mathcal{A}_b is the set of the learned active codes for the block b (14).

Specifically saying, to represent the facial image, we generate the sLSP code at every pixel for each of the learned-size block, b , and compute the histogram that represents it by

$$\mathbb{H}_b(c) = \sum_{p \in \mathcal{B}_b} \delta(\text{sLSP}(p), c), \quad \forall c, \quad (16)$$

where \mathcal{B}_b is the b -th (learned-size) block, δ is the Dirac's delta (4), sLSP is the spatial version of the codes (15), and c are all the possible sLSP codes. Finally, the face descriptor, is the concatenation of all the histograms of the learned-size blocks (as shown in Fig. 7(b)), that is,

$$\mathbb{F} = \parallel_{b=1}^B \mathbb{H}_b, \quad (17)$$

where B is the number of learned-blocks used (see Section 3.2 for the selection of the blocks and their number), and \parallel is the concatenation operation on the histogram vectors.

6 EXPERIMENTAL RESULTS

In this section, we present the experiments of our proposals for the recognition of facial expressions. We evaluate the performance of our proposals in different existing datasets, including CK+ [26], FACES [30], RaFD [31], BU-3DFE [32], Spontaneous-ISED [33], Spontaneous-NVIE [34], and GEMEP-FERA [35]. For the experiments, we crop the face region of the provided dataset images, according to the specification of Section 3.1 by using existing eyes and mouth detection methods [24], [25]. To conduct the experiments under person-independent protocol, we perform N -person cross-validation, commonly known as leave-one-person-out cross-validation, as done in recent works [3], [13]. This protocol trains the dataset images excluding the expression images of one person, and then performs the testing on the images of that particular person. The final result is produced by average result after repeating the process for N -persons. For classification, we use Support Vector Machine (SVM) with RBF kernel. Because SVM generates binary classification, multiclass classification is done with the one-against-one method.

6.1 CK+ Results

Extended Cohn-Kanade Facial Expression (CK+) dataset [26] includes 593 image sequences (from neutral to apex) of 123 subjects. Seven emotion categories, namely anger, contempt, disgust, fear, happiness, sadness and surprise, are included in 327 sequences. Similar to works presented in [2], [3], [10], [11], we select three most expressive image frames from 309 sequences with seven expression categories, having 981 images in total.

6.1.1 Optimal Parameters for LSP

There are several parameters of proposed LSP that needs to be selected for the experiments, including size of the neighborhood \mathcal{N} (2), the stride s for setting orientlets, and the number of principal responses k (8). We conduct several sets of experiments to find the optimal values of such parameters for using 7-class CK+ dataset with N -person cross validation. Since we perform these experiments to find the best parameters only, for the ease of experiments, we use the uniform blocks of the images of 6×7 size. Moreover, for the smoothing and peak-finding case, we use 3×3 window, respectively, found experimentally.

Due to the tight relation between the number of orientation-bins and the orientlets, we need to select a combined choice of these two parameters in order to have better accuracy. For this purpose, we run several experiments varying these two parameters. In Fig. 3, we show an example of different orientlets for $\mathcal{Q} = |\{\theta\}| = 16$ quantized bins. The number of orientation bins to be considered as the ideal orientations (1) for each edge-direction (i.e., neighboring pixel) will be called stride, s , in this experiment. For instance, Fig. 3 shows a representative example with $s = 2$ for a quantization of $\mathcal{Q} = 16$. In this experiment, we denote the combined representation of the number of quantized bins and the stride used to define the orientlet as \mathcal{Q}_s . We show the results in Table 2 for different neighborhoods, and with a fixed threshold, $\tau = 15$, and selecting $k = 2$ peaks. We observe that results of 16-bins with $s = 2$ are consistently better than the results of using either a wide number of bins, e.g., 32-bins, or small number of bins, e.g., 8. We found that putting a loose restrictions (e.g., 16_4 , 32_8) may get affected by neighboring noisy patterns, while putting too much tight restrictions (e.g., 32_2) may only consider very strong solid edge shape, ending up losing smooth and distorted edge shapes. Hence, we consider 16-quantized bins with $s = 2$ as a good combination for our purpose. Similarly, to test the best neighborhood size, we conduct our experiments twice with 3×3 and 5×5 , respectively. Results show that results for the 5×5 neighborhood is consistently better than 3×3 , and hence we use 5×5 as the optimal neighborhood size. For facial images, thus, a wider neighboring structure may be better than using smaller neighborhood.

The number of peaks, k (8), of the local histogram to represent the shape of the pixel is another very important parameter in our approach. Different number of peaks represent different shape-structures. For example, only one peak denotes the existence of a solid one-directional edge. Two peaks denote the directions of two prime edges going through the pixel, referring the existence of corner and curve-like textures. Likewise, considering more than two

TABLE 2: Recognition rates (%) for different orientlets using \mathcal{Q} quantized orientation bins with a stride s (\mathcal{Q}_s) for different neighborhoods \mathcal{N} . These results use a fixed threshold, $\tau = 15$ and $k = 2$ peaks.

\mathcal{Q}_s	8 ₁	16 ₂	16 ₄	32 ₂	32 ₄	32 ₈
$\mathcal{N}_{3 \times 3}$	90.25	90.32	89.50	89.11	89.70	88.94
$\mathcal{N}_{5 \times 5}$	90.41	91.13	90.62	89.86	90.51	89.54

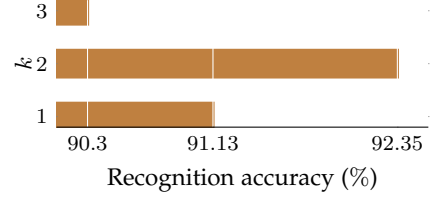


Fig. 8: Recognition accuracy (%) for different number of peaks (k).

peaks denote the occurrence of branch or complex junction-like textures, where more than two edge-directions can be found. However, to select the optimal number of peaks for our descriptor, we conduct experiments for different peaks, $k \in \{1, 2, 3\}$, and show the results in Fig. 8, where the best result is observed in $k = 2$. Due to the lack of complex junction-like structures, using $k = 3$ peaks is often redundant for facial image, whereas considering $k = 1$ peak may ignore important curve and corner textures; thereby, we use at most $k = 2$ peaks for LSP in the rest of the experiments.

6.1.2 Optimal Parameters for Active LSP Codes and Learned-Size Blocks

The proposed active LSP code and learned-size blocks have several parameters. Since the facial features contained in each learned-size block have different movements and code patterns according to facial expression changes, we have to find the optimal sub regions sizes (Fig. 7(a)), and number of active codes, n . Similar as before, for the optimal selection of such parameters, we conduct N -person cross validation on CK+ for different sets of parameters.

For the optimal parameters of sub blocks for each learned block, we defined 18 different sub regions by dividing the blocks corresponding to each facial reference point with sizes from $\{a \times b : a \in \{1, 2, 3\} \text{ and } b \in \{1, 2, 3, 4, 5, 6\}\}$. Next, we searched for the optimal sub regions sizes for each learned block by selecting the top $N = 15$ resultant blocks from AdaBoost (Fig. 6), and selecting top $n = 16$ active codes. Since the search space is too big if we do a full search, we set the division of all blocks to 3×3 sub regions, while varying just one block. For effectiveness, the eye, outer brow, and side mouth blocks paired left and right were set using the same sub region sizes. Table 3 shows test results of the optimal sub regions for each learned block. In the test, we found that learned-size blocks have the best results at 3×6 , 1×1 , 2×1 , 3×3 , and 3×4 for eye, outer brow, brow, mouth, and side mouth, respectively, according to contained facial features.

TABLE 3: Recognition accuracy (%) for LSP with learned-size blocks for 18 different sub regions in CK+ dataset. The experiments use the top $N = 15$ resultant blocks from AdaBoost, $n = 16$ top active codes, and 3×3 sub regions for other blocks (per experiment).

Sub blocks	Eye	Outer brow	Brow	Mouth	Side mouth
1×1	92.8	94.5	93.0	93.3	92.2
1×2	92.3	94.0	92.9	93.4	92.5
1×3	92.5	94.4	93.0	91.6	92.2
1×4	93.0	93.6	93.1	90.3	92.5
1×5	92.9	93.3	92.8	90.6	92.6
1×6	93.1	93.4	92.8	90.5	90.1
2×1	92.5	94.1	93.2	91.2	92.5
2×2	92.5	93.9	92.8	91.6	92.6
2×3	92.8	93.6	92.8	91.4	92.4
2×4	93.2	93.0	93.0	91.2	92.8
2×5	92.7	92.8	92.7	91.4	92.7
2×6	92.9	93.3	92.7	91.4	92.9
3×1	92.3	94.2	92.9	92.1	92.6
3×2	92.2	93.8	92.7	91.9	92.9
3×3	93.0	93.8	93.0	93.8	93.0
3×4	92.7	92.2	92.7	92.1	93.4
3×5	93.0	91.6	92.2	92.6	93.3
3×6	93.4	93.2	92.4	91.6	93.1

TABLE 4: Accuracy (%) of LSP in CK+ by varying the top N results obtained from AdaBoost for the optimal sizes of the learned-size blocks.

N	5	10	15	20	25	30
Accuracy (%)	93.3	94.8	94.8	94.8	94.8	94.9

After finding the optimal sub regions, we searched for the optimal top N blocks from AdaBoost by exploring $N \in [5, 10, \dots, 30]$ with $n = 16$ active codes, and the previously found optimal sub regions. Table 4 shows results of the optimal top N test. We found that the optimal sizes at $N = 30$. Finally, to find the optimal n of each learned block, we tested eight different n values from $[4, 8, \dots, 32]$ for each block, while setting $n = 16$ for other blocks. Since the optimal sub block of the outer brow one is 1×1 which means no sub region, it is not necessary to find the optimal n value for the outer brow. Thus, we excluded the outer brow blocks from this experiment. Similar to the previous test, we paired the left and right eyes and side mouths again for effectiveness. Table 5 shows the results. Based on these, we set $n = 20$ for the eye blocks, $n = 4$ for the brow block, $n = 24$ for the mouth block, and $n = 24$ for the side mouth blocks. From the three tests, we found the optimal parameters for active LSP and learned-size blocks. We used these optimal parameters in succeeding experiments.

TABLE 5: Recognition accuracy (%) for LSP in CK+ with learned-size blocks while varying n for each block, and setting the rest to $n = 16$.

n	Eye	Brow	Mouth	Side mouth
4	94.5	94.9	90.7	94.1
8	94.4	94.8	90.8	94.3
12	94.5	94.7	92.6	94.8
16	94.9	94.9	94.9	94.9
20	95.1	94.9	94.9	94.9
24	94.9	94.8	95.0	95.0
28	94.9	94.8	94.8	94.8
32	94.9	94.8	94.2	95.0

6.1.3 Efficacy of learned-size blocks and active codes

We experimentally validate the efficacy of the combined representation of proposed learned-size blocks and active codes against different other usages of the blocks.




First, we generate result for LSP code with uniform-size blocks (LSP+UB) extracted from the whole face. Second, we show the result for the proposed method, that is active LSP with leaned-size block representation (sLSP+LB) using the optimal parameters learned above. Thirdly, to show the efficacy of proposed active LSP codes over original LSP codes, we generate result for learned-size block representation when used with LSP code only. Fourthly, instead of using the learned blocks, we set square blocks of same size in the center of the learned blocks, to show the efficacy of the sizes of the blocks that are learned. Illustrative results are shown in Table 6 and we visualize that proposed learned-size blocks with active LSP code convincingly outperforms other combination of usages. As we observe from the given illustrations, uniform blocks cover the whole face and hence, consist of a bunch of blocks (especially on the cheek and forehead), having no expression-related information. The contribution of such redundant information is very minimal to the classifier rather creating unnecessary confusion at times, ending up with low accuracy. On contrary, proposed leaned-size blocks cover the regions contributing highly to the expression changes, and hence avoids unnecessary futile information, contributing to its higher accuracy. As we also observe, learned blocks vary in sizes according to the different facial components. Therefore, to verify the importance of the size of the learned blocks, we set square-size blocks around the center of the learned-size blocks (sLSP+LB^c). In this case, we keep the size of all the blocks same. We observe that some of the blocks (e.g., mouth block) fail short to include the respective facial components appropriately, while some blocks (e.g., eye-brow blocks) include redundant information of other facial components, and hence, result in performance-degradation. This, in turn, shows the efficacy of learning proper size of facial blocks in representing the salient expression information.

We also observe that the accuracy improves when learned-size blocks are used with active LSP codes instead of using LSP codes (LSP+LB). For this, we generate the result of LSP+LB using exactly the same way as described in Section 5, except instead of active LSP we use all the LSP codes. Since different blocks contain different texture characteristics, in our approach, we select specific codes (active codes) contributing highly to the expression changes for that block, contributing towards its higher accuracy. On the contrary, while using all the LSP codes, non-influential codes are also included in the feature-vector and hence arises unnecessary ambiguity to the classifier at time, ending up with lower accuracy.

6.1.4 Performance Under Noise and Misalignment

We analyze the performance of our proposed descriptor under noise and misalignment. For this, first, we randomly distribute zero-mean Gaussian noise to the 7-class images of CK+ dataset within the interval of $[0.08, 0.16]$ and $[0.16, 0.32]$ standard deviation. We now perform N -person cross-validation for different descriptors and present




TABLE 6: Recognition accuracy (%) of our method with different usage of facial blocks. Notations used in the caption are as, UB: uniform blocks over whole face, LB: learned-size blocks, LB^c: square-sized blocks positioned in the center of the learned-size blocks.

Method	LSP+UB	sLSP+LB	LSP+LB	sLSP+LB ^c
Result	93.78	95.13	93.02	90.32
Block				

the results at Table 7, where we observe that proposed LSP outperforms other descriptors under both the noise intervals. The key reason of such dominant result of LSP is the use of statistical information of neighboring pixels from a wider region that exploits a global shape of the texture, providing more stable structural information even under a noisy environment. Moreover, we select the salient peaks from the histogram of neighboring orientations to represent reliable edge boundaries, which, in turn, avoids the effect of noisy accumulations from single or fewer samples, as observed in many other existing descriptors. Note also that accuracy of LSP improves more when using with proposed active-LSP and learned-sized blocks. The active-LSP codes are generated in the pixels contributing to consistent expression-changes, which in turn, discards the textures having trivial effect on expressions producing futile information under the noise-effect. Moreover, the learned-size blocks removes insignificant region of the face that may generate redundant information under noise. All these factors perhaps cumulatively contribute towards such robust performance of our method.

In addition, we evaluate the performance of our method under the misaligned frontal faces. Such misalignment of facial image is very common in practice, which may occur due to the registration error in detecting face or mis-detection of different facial components (i.e., eye, nose, etc.). For our purpose, we artificially distort the alignment of the face by adding random Gaussian noise with zero mean and standard deviations within the interval of [0.5, 5.0] to the position of two eyes. We utilize the ground-truth eye-position information of CK+ to perturb the position of the eyes, generating the misaligned facial image. We now conduct N-person-cross-validation in the 7-classes of CK+ for different descriptors (with the uniform-block settings) and provide results in Table 7. We observe that proposed method achieves better accuracy than other descriptors under the misalignment of images. Most importantly, we observe a significant gain in performance of LSP when using learned-sized blocks. This mainly happens since same-sized uniform blocks often fail short to consistently preserve the information of same facial components in the same blocks [4], [5], as also discussed in previous subsection. We explicitly show it in Fig. 9(a), where we observe that some of the uniform blocks (marked in red) in misaligned image represent significantly different facial parts than that

TABLE 7: Performance of different descriptors under noise and misalignment.

Descriptors	Noise-variation		Misalignment
	[0.08, 0.16]	[0.16, 0.32]	
Example			
LBP	72.09	69.09	84.20
LDP	78.28	57.44	86.95
LDN	88.58	52.75	87.36
LPTP	91.64	77.37	90.01
PTP	91.03	82.16	89.91
LDTP	89.91	86.24	91.84
HOG	90.65	81.54	89.89
NEDP	91.67	85.26	91.34
LSP+UB	91.77	85.35	91.88
sLSP+LB	91.89	86.39	93.37

of the aligned image. On the contrary, learned-size blocks vary in sizes, and comparably bigger than the uniform-sized blocks to cover target facial components quite appropriately despite having registration error or misalignment of face. We also illustrate this in Fig. 9(b), showing that despite having misalignment, all the learned-size blocks, although except the right eye-brow, represent corresponding facial components similarly, as in the aligned image.

6.1.5 Comparison against state-of-the-art methods

We compare the performance of our method on CK+ dataset against other state-of-the-art methods. Apart from the baseline 7-class experiments [26], different methods evaluate their performance for different classes in CK+. For instance, due to the fewer samples of contempt class, some methods [13], [36] remove the images of contempt class and conduct 6-class experiments. Some other methods [41], [41], [42], [43], [44] add the first neutral expression frame of each video with the existing 7-classes and conduct 8-class experiment with the neutral expression-class. Therefore, in addition to the basic 7-class experiment, in order to compare our method with these methods, we also conduct separate experiments for 6-class and 8-classes, and provide the results at Table 8.

We compare the performance of proposed method against diverse existing methods, such as appearance-

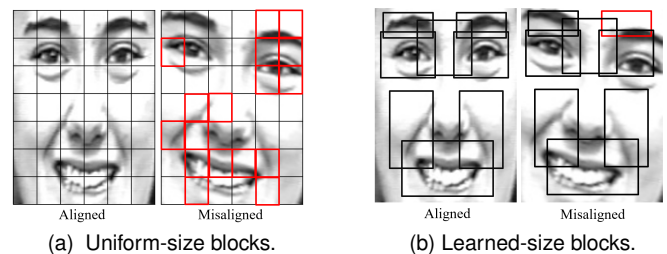


Fig. 9: Representation of uniform-size and learned-size blocks for an aligned and misaligned image. Blocks marked in red represent either completely or slightly different facial components under misalignment.

TABLE 8: Comparative person-independent recognition results (%) for different expression-classes of CK+. (Results with citations are taken from corresponding papers; others are generated by the authors.)

Methods	6-class	7-class	8-class
LBP	92.88	85.84	89.38
LDP	91.15	84.79	85.04
LDN	86.73	80.74	85.12
LPTP	92.34	91.64	86.20
PTP	91.69	91.03	88.99
LDTP [3]	94.97	94.19	91.03
NEDP [13]	93.78	92.97	90.03
ML-MV-G [36]	88.52	-	-
BDBN [†] [37]	96.70	-	-
LPQ+SRC [38]	-	80.78	-
ICV+CER [38]	-	92.34	-
SRC+IVR [39]	-	90.50	-
SPTS+CAPP [26]	-	83.30	-
MSR [40]	-	91.40	-
HOG [41]	-	-	89.53
Gabor [41]	-	-	88.61
SIFT [41]	-	-	86.39
Zero-biased CNN [†] [42]	-	-	81.80
FN2EN [†] [43]	-	-	88.70
AURF [†] [41], [43]	-	-	92.22
AUDN [†] [43], [44]	-	-	93.70
LSP+UB	95.22	93.78	91.50
sLSP+LB	96.77	95.13	93.81

†: Deep-learning method.

based methods, geometry-based methods, manifold-based approaches etc. Among them our method outperforms other appearance-based methods convincingly; for instance, LBP, LDP, LDN, LPTP, PTP, LDTP, NEDP for all the classes, while HOG, Gabor and SIFT for 8-class. Among the geometry-based approaches, we compare against a combined method of similarity-normalized shape (SPTP) and Canonical appearance feature (CAPP) [26] for 7-class problem, and observe better accuracy of our method. For 7-class, we also observe better performance of our method against a work that uses manifold based sparse representation (MSR) [40], and approaches dealing with intra-class variations [38], [39]. In addition, we compare against a work employing multi-layer multi-variable grouping (ML-MV-G) algorithm for expression analysis [36], where we also observe better accuracy of our method for 6-class recognition.

Comparison Against Deep-learning Methods: Recently, deep-learning based methods have shown dominating performances in facial expression recognition task. However, one critical issue for these methods in the available facial expression datasets is the unavailability of sufficient training data. To tackle this issue, recent methods increase their data using various data-augmentation strategies [42], [43] or pre-train the network using additional data from other domains [45]. For the sake of fair comparison with such methods, one has to show the results after adopting the above-mentioned strategies, as also suggested in previous work [13], [45]. However, such strategies do not fall within the scope of our work in this paper. Therefore, in our approach, we compare the deep-methods those use original training data for the result-generation under person-independent protocol. In this regard, we compare our results against some recent works, for example BDBN [37] for 6-class and Zero-biased CNN [42], FN2EN [43], AURF [41], AUFN [44] for 8-classes. Results can be found in Ta-

TABLE 9: Comparative recognition accuracy (%) for BU-3DFE. (Results with citations are taken from corresponding papers; others are generated by the authors.)

Methods	Accuracy (%)
LBP	56.20
LDP	61.30
LDN	56.50
LPTP	67.80
PTP	66.88
HOG	67.27
Geometric-based approach [46]	66.50
BDA/GMM [47]	68.28
LBP+LPQ+DCT+SIFT [48]	68.30
LGBP [49]	67.96
LGBP+LBP ^{ms} [49]	71.10
LDTP [3]	71.30
NEDP [13]	68.25
LSP+UB	72.87
sLSP+LB	73.91

ble 8, where we observe that some of the methods, e.g., BDBN [37], AURF [41], and AUFN [44] perform better than LSP when applied with uniform-size blocks (UB). Nevertheless, combined with the proposed active LSP (sLSP) and learned-size blocks (LB), our proposal achieves higher accuracies than all the above-mentioned deep-methods, showing its efficacy against these methods in such controlled experimental settings where deficiency of data is often observed.

6.2 BU-3DFE Results

The BU-3DFE dataset [32] provides 2400 face images of 100 subjects with six prototype emotions. The 2D-images of this dataset are rendered from 3D data, which are presented within four different intensity conditions with 512×512 resolution. Due to the variety of ethnic/racial ancestries and intensities of expressions, this dataset is considered to be a challenging one. Each image in this dataset is labeled with one of the six emotions (anger, disgust, fear, happiness, sadness, and surprise) with an emotion intensity from 01 to 04.

In BU3DFE, we achieve the highest accuracy than other given methods at Table 9. Along with different local descriptors, such as LBP, LDP, LDN, LPTP, PTP, HOG, LDTP, LGBP and NEDP, we also compare our method against a geometric-based approach proposed by Hu et al. [46], Bayes Discriminant Analysis via Gaussian Mixture Model (BDA/GMM) [47], and different fusion approaches, such as LBP+LPQ+DCT+SIFT [48] and LGBP+LBP^{ms} [49]. Interesting to observe that the accuracies of different descriptors, such as LBP, LDP, and LDN are quite low in this dataset. The black background behind the facial images of this dataset leads the above descriptors to generate insignificant meaningless patterns, yielding poor performance. In this case, LSP takes advantage of the thresholding (8) along with its embedded structural restriction scheme (1) to overcome such featureless pixels. The addition of learned blocks that preserve only the expressive components also contribute to ignore such meaningless textures.

6.3 FACES Results

FACES [30] is a set of images of naturalistic 171 subjects with six expressions: neutrality, sadness, disgust, fear, anger, and

TABLE 10: Comparative recognition accuracy (%) in FACES and RaFD datasets under age and gender variations.

Methods	FACES				RaFD		
	Overall	Young-aged	Middle-aged	Old-aged	Overall	Male	Female
LBP	91.52	92.67	86.16	81.29	93.46	92.95	91.11
LDP	89.57	91.38	88.09	80.70	93.02	91.67	88.83
LDN	88.69	89.37	87.79	80.12	92.90	88.89	86.67
LPTP	92.11	90.37	88.39	81.58	92.53	87.59	88.01
PTP	91.42	93.10	89.58	84.36	92.90	92.06	82.50
HOG	91.04	92.13	88.77	83.54	92.11	90.56	87.98
NEDP	92.15	92.45	89.79	84.85	94.08	92.10	90.22
LSP+UB	92.70	93.54	90.96	84.97	94.15	93.03	91.84
sLSP+LB	93.54	95.03	91.06	86.13	96.67	94.35	92.66

happiness. It contains a total of 2052 images comprising two sets of pictures per person and per facial expression. Among the total 171 subjects, 58 subjects are younger (19 ~ 31 years old), 56 are middle-aged (39 ~ 55 years old) and 57 are older (69 ~ 80 years old).

Expression recognition results for all the FACES images are presented in the first column of Table 10, which shows that the our descriptor achieves better performance against the existing local descriptors. It is worth mentioning that expression-images of this dataset possess significant variation in age. Thus, the higher result of the proposed descriptor in FACES shows its efficiency under the images of such variations. Moreover, Caroppo et al. [50] shows that the expression images of older subjects are hard to detect as their facial traits exhibit less differentiation among different expressions. Hence, to evaluate the performance of proposed method in such images, we conduct separate experiments for the given three age-groups, including young, middle-aged and old. We provide the results in Table 10, where we also observe better accuracy of proposed method than other descriptors in all the given age-groups. Nevertheless, similar to the findings in Caroppo et al.'s work [50], we also observe that recognition rate for the old-group images are comparably lower than the young-group. Older subjects possess more person-specific information, such as wrinkle, blemish etc., and show ambiguous expressions at times, making it difficult to preserve distinguishable expression-affiliated change information. Existing descriptors often fail to differentiate the codes from such person-specific regions, affecting the overall recognition accuracy. We tackle this issue by counting the spatial variance of the codes, which we consider while generating the active codes. Since such person-specific textures occur in specific facial area only, the position-wise (spatial) variance gets low despite the total occurrence is high. Therefore, the low variance of these codes make the overall score, as in (10), of these codes minimum, making it less likely to be included as the active codes, reducing the effect of such person-specific information. We can also observe this from the recognition rates of our method for the older groups, which is higher than other descriptors, showing its efficacy against other methods in recognizing the challenging expressions of older subjects.

6.4 RaFD Results

Radboud Faces Database (RaFD) [31] contains images of 67 subjects performing eight facial expressions (anger, disgust, fear, happiness, contemptuous, sadness, surprise and

neutral) with three gaze directions and five different face orientations. However, for our experiments, we use the frontal facial images with frontal gaze direction.

We test the performance of proposed method for RaFD dataset using all the images, and present comparative results in Table 10. We observe that proposed method achieves better accuracy than the other given methods. It is worth mentioning that expression-images of this dataset possess significant variation in human gender, which leaves the scope to evaluate performance under gender-diversity. Hence, we explicitly test the performance of proposed method for different gender, including male and female. For both the groups, we observe better accuracies of proposed method, suggesting its robust performance towards gender-invariant recognition of expressions.

6.5 Spontaneous-NVIE Results

Natural Visible and Infrared (NVIE) facial Expression dataset [34] provides a set of expression-sequences recorded by a visible and an infrared thermal camera, with illumination provided from three different directions. For our experiments, we collect the given peak expression frames from the spontaneous visible spectrum video sequences with frontal lighting from 100 subjects, as done in previous work [34], [38]. Moreover, among six given expressions, only three expressions, including disgust, fear and happiness were successfully induced by the emotional videos in most subjects. Thereby, these three expressions were used in the experiments by Wang et al. [34] and Lee et al. [38], which we also follow and conduct experiments for these three expression classes.

N -person-cross-validation results on NVIE dataset are given at Table 11, where we observe that proposed method successfully outperforms other existing methods. Besides comparing with different descriptors, we also compare our proposal against a method using intra-class variation [38], and different fusion method of PCA, LDA and KNN along with AAM, KNN and LDA from the baseline experiment [34]. Achieving higher accuracy in this dataset is particularly important since the images of NVIE are spontaneous in nature, providing the opportunity to judge the methods' performance in real world scenario. Nevertheless, the existence of glasses and uneven illumination also make the dataset challenging in practice. However, proposed method demonstrates the highest accuracy compared to other method, exhibiting its efficacy to recognize expressions under such challenging conditions.

TABLE 11: Comparative recognition accuracy (%) for NVIE (VIS) and ISED spontaneous datasets. (Results with citations are taken from corresponding papers; others are generated by the authors.)

Methods	NVIE results		ISED Results	
	Accuracy(%)		Accuracy(%)	F1-Score
LBP	71.22		76.47	0.68
LDP	71.92		74.61	0.67
LDN	72.63		75.85	0.65
LPTP	70.38		72.46	0.64
PTP	70.52		76.16	0.68
LDTP	72.26		76.88	0.69
HOG	70.33		75.66	0.67
NEDP	72.28		77.79	0.70
LBP+SRC [38]	59.50	-	-	-
LPQ+SRC [38]	62.17	-	-	-
Gabor+SRC [38]	65.00	-	-	-
ICV+CER [38]	70.33	-	-	-
PCA+LDA [34]	58.47	-	-	-
PCA+LDA+KNN [34]	65.25	-	-	-
AAM+KNN [34]	67.80	-	-	-
AAM+LDA+KNN [34]	61.86	-	-	-
LSP+UB	72.82		77.82	0.70
sLSP+LB	74.16		78.03	0.71

6.6 Spontaneous-ISED Results

The Indian Spontaneous Expression Database (ISED) [33] is a recently published dataset providing near-frontal spontaneous expressive images. ISED images has four different expressions including happiness, surprise, sadness and disgust. The peak expression frames of each videos and its corresponding emotion-label are given with the dataset. In total, the peak expression faces of 50 subjects from all 428 video clips are given, which were used in the baseline experiment [33]. Among the peak images, 227, 73, 48, and 80 images belong to the happiness, surprise, sadness, and disgust, respectively.

Using the given peak expression images, we conduct N-person-cross-validation experiments in ISED dataset, and provide the results at Table 11. Our proposal achieves the highest accuracy in ISED. However, the provided images per class in ISED are imbalances; for example, four classes of ISED, i.e., happiness, surprise, sadness, and disgust, contain 227, 73, 48, and 80 images, respectively. Therefore, the false-positives for the dominant class “happiness” are quite high compared to other classes, as we observe from our experiments. Since recognition accuracy does not count the false positives, it may end up with improper interpretation of result. Therefore, to get a better interpretation of the results, we add F1-scores [51] for different descriptors along with their recognition accuracies. F1-score is the harmonic mean of precision and recall; that is it takes into account true positives, as well as false positives and false negatives, which are important when considering often imbalanced databases. Results for F1-scores in Table 11 also show the superiority of our proposal against other methods, which in turn, points the efficacy of our proposal in case of having datasets with such imbalanced images.

6.7 GEMEP-FERA Results

The GEMEP-FERA emotion detection dataset [35] contains 134 videos of different expressions by 10 subjects. Videos were recorded while uttering meaningless words or a sustained vowel ‘aaa.’ Videos are of 720×576 resolution and

TABLE 12: Comparative recognition accuracy (%) for GEMEP-FERA. (Results with citations are taken from corresponding papers; others are generated by the authors.)

Methods	Person-independent	Person-specific	Overall
Baseline (LBP) [35]	44.0	73.0	56.0
PHOG+SVM [53]	66.7	69.0	67.0
PHOG+LPQ+SVM [53]	64.8	83.8	72.4
PHOG+LPQ+LMNN [53]	62.9	88.7	73.4
EAI+LPQ [52]	75.2	96.2	83.8
LDP	62.5	96.2	76.1
LDN	63.7	96.1	76.8
PTP	65.0	96.2	77.6
LSDP	62.5	96.1	76.0
LPTP	63.5	96.1	77.1
LDTP [3]	71.3	96.3	81.3
NEDP [13]	67.5	96.3	79.1
LSP+UB	70.7	98.0	80.8
sLSP+LB	72.0	98.5	82.1

they are categorized as one of the five expressions including anger, fear, joy, relief, and sadness. Training set consists of a total of 155 videos of 7 subjects and the test set comprises 134 videos of 6 subjects, half of which are not included in the training set.

In GEMEP-FERA [35], the baseline results were shown with LBP, where features were generated from all the frames followed by classifying each of them with SVM (using RBF-kernel). Label of an emotion for a video is determined by finding out the emotion occurring in maximum number. Following the same procedure, we also apply our method to recognize the emotion labels of GEMEP-FERA data, where we test our method for both the person-specific and person-independent protocol, as specified in the original paper [35]. Results in Table 12 show that our method consistently performs better than all the other given methods, except EAI+LPQ [52]. We note that EAI+LPQ is the rank-1 method in FERA Challenge [35] that uses complex avatar image generation and a complicated normalization process. Our method method, on the contrary, is much simple yet provides consistent performance, advocating for its overall efficacy.

7 CONCLUSION

This paper addressed three common open issues for facial expression recognition related to feature coding, including the design of an efficient descriptor, selecting the active expression-related features, and representing the facial image with learned-size blocks. The proposed method incorporates all three aspects, by creating a face descriptor from a set of blocks which size is learned for the facial expression task, and by using an adaptive set of codes that reflect the active expressive textures of the face. Our experiments show the robustness of the proposed methods on different existing datasets. It also shows that it outperforms existing similar methods on different scenarios (as shown by the datasets), which the existing methods cannot handle, exhibiting its overall efficacy in recognizing facial expressions.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea

(NRF) funded by the Ministry of Science, ICT & Future Planning (2018R1C1B3008159); in part by the São Paulo Research Foundation (FAPESP) under grant No. 2016/19947-6; and in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant No. 307425/2017-7.

REFERENCES

- [1] D. Consoli, "Emotions that influence purchase decisions and their electronic processing," *Annales Universitatis Apulensis: Series Oeconomica*, vol. 11, no. 2, p. 996, 2009.
- [2] A. Ramírez Rivera, J. Rojas Castillo, and O. Chae, "Local directional number pattern for face analysis: Face and expression recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1740–1752, 2013.
- [3] B. Ryu, A. Ramírez Rivera, J. Kim, and O. Chae, "Local directional ternary pattern for facial expression recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 6006–6018, 2017.
- [4] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 765–781, 2011.
- [5] S. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," *IEEE transactions on Affective Computing*, vol. 6, no. 1, pp. 1–12, 2015.
- [6] H. Mliki and M. Hammami, "Facial expression recognition using salient facial patches," *WSCG 2016*, 2016.
- [7] D. Ghimire, S. Jeong, J. Lee, and S. H. Park, "Facial expression recognition based on local region specific features and support vector machines," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 7803–7821, 2017.
- [8] Z. Shao, Z. Liu, J. Cai, Y. Wu, and L. Ma, "Facial action unit detection using attention and relation learning," *IEEE Transactions on Affective Computing*, 2019.
- [9] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [10] 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.
- [11] T. Jabid, M. H. Kabir, and O. Chae, "Robust facial expression recognition based on local directional pattern," *ETRI journal*, vol. 32, no. 5, pp. 784–794, 2010.
- [12] A. Ramírez Rivera, J. A. R. Castillo, and O. Chae, "Recognition of face expressions using local principal texture pattern," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 2609–2612.
- [13] M. T. B. Iqbal, M. Abdullah-Al-Wadud, B. Ryu, F. Makhmud-khujav, and O. Chae, "Facial expression recognition with neighborhood-aware edge directional pattern (nedp)," *IEEE Transactions on Affective Computing*, 2018.
- [14] M. T. B. Iqbal, B. Ryu, G. Song, and O. Chae, "Positional ternary pattern (PTP): An edge based image descriptor for human age recognition," in *Consumer Electronics (ICCE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 289–292.
- [15] M. T. B. Iqbal, M. Shoyaib, B. Ryu, M. Abdullah-Al-Wadud, and O. Chae, "Directional age-primitive pattern (DAPP) for human age group recognition and age estimation," *IEEE Transactions on Information Forensics and Security*, 2017.
- [16] K. Lekdioui, R. Messoussi, Y. Ruichek, Y. Chaabi, and R. Touahni, "Facial decomposition for expression recognition using texture/shape descriptors and svm classifier," *Signal Processing: Image Communication*, vol. 58, pp. 300–312, 2017.
- [17] M. Song, D. Tao, Z. Liu, X. Li, and M. Zhou, "Image ratio features for facial expression recognition application," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 779–788, 2010.
- [18] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas, "Learning active facial patches for expression analysis," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2562–2569.
- [19] L. Zhang and D. Tjondronegoro, "Facial expression recognition using facial movement features," *IEEE Transactions on Affective Computing*, vol. 2, no. 4, pp. 219–229, 2011.
- [20] L. Shapiro, *Computer vision and image processing*. Academic Press, 1992.
- [21] A. C. Bovik, *Handbook of image and video processing*. Academic press, 2010.
- [22] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Prentice Hall, 2008.
- [23] P. Ekman, E. Rosenberg, and J. Hager, "Facial action coding system affect interpretation dictionary (facsaid)," 1998.
- [24] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2D cascaded adaboost for eye localization," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2. IEEE, 2006, pp. 1216–1219.
- [25] R. Lienhart, L. Liang, and A. Kuranov, "A detector tree of boosted classifiers for real-time object detection and tracking," in *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, vol. 2. IEEE, 2003, pp. II–277.
- [26] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, 2010, pp. 94–101.
- [27] C. Shan, S. Gong, and P. W. McOwan, "Conditional mutual information based boosting for facial expression recognition," in *BMVC*, 2005.
- [28] S. Liao, M. W. Law, and A. C. Chung, "Dominant local binary patterns for texture classification," *IEEE transactions on image processing*, vol. 18, no. 5, pp. 1107–1118, 2009.
- [29] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 971–987, 2002.
- [30] N. C. Ebner, M. Riediger, and U. Lindenberger, "FACES—a database of facial expressions in young, middle-aged, and older women and men: Development and validation," *Behavior research methods*, vol. 42, no. 1, pp. 351–362, 2010.
- [31] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition and emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.
- [32] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*. IEEE, 2006, pp. 211–216.
- [33] S. Happy, P. Patnaik, A. Routray, and R. Guha, "The indian spontaneous expression database for emotion recognition," *IEEE Transactions on Affective Computing*, 2015.
- [34] S. Wang, Z. Liu, S. Lv, Y. Lv, G. Wu, P. Peng, F. Chen, and X. Wang, "A natural visible and infrared facial expression database for expression recognition and emotion inference," *IEEE Transactions on Multimedia*, vol. 12, no. 7, pp. 682–691, 2010.
- [35] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer, "The first facial expression recognition and analysis challenge," in *Face and Gesture 2011*. IEEE, 2011, pp. 921–926.
- [36] S. Taheri, Q. Qiu, and R. Chellappa, "Structure-preserving sparse decomposition for facial expression analysis," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3590–3603, 2014.
- [37] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014*, pp. 1805–1812.
- [38] S. H. Lee, W. J. Baddar, and Y. M. Ro, "Collaborative expression representation using peak expression and intra class variation face images for practical subject-independent emotion recognition in videos," *Pattern Recognition*, vol. 54, pp. 52–67, 2016.
- [39] S. H. Lee, K. N. K. Plataniotis, and Y. M. Ro, "Intra-class variation reduction using training expression images for sparse representation based facial expression recognition," *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp. 340–351, 2014.
- [40] R. Ptucha, G. Tsagkatakis, and A. Savakis, "Manifold based sparse representation for robust expression recognition without neutral subtraction," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, pp. 2136–2143.
- [41] M. Liu, S. Li, S. Shan, and X. Chen, "Au-aware deep networks for facial expression recognition," in *FG, 2013*, pp. 1–6.

- [42] P. Khorrami, T. Paine, and T. Huang, "Do deep neural networks learn facial action units when doing expression recognition?" in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 19–27.
- [43] H. Ding, S. K. Zhou, and R. Chellappa, "Facenet2expnet: Regularizing a deep face recognition net for expression recognition," in *Automatic Face & Gesture Recognition (FG 2017)*, 2017 12th IEEE International Conference on. IEEE, 2017, pp. 118–126.
- [44] M. Liu, S. Li, S. Shan, and X. Chen, "Au-inspired deep networks for facial expression feature learning," *Neurocomputing*, vol. 159, pp. 126–136, 2015.
- [45] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, and S. Yan, "Peak-piloted deep network for facial expression recognition," in *European conference on computer vision*. Springer, 2016, pp. 425–442.
- [46] Y. Hu, Z. Zeng, L. Yin, X. Wei, J. Tu, and T. S. Huang, "A study of non-frontal-view facial expressions recognition," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.
- [47] W. Zheng, H. Tang, Z. Lin, and T. S. Huang, "Emotion recognition from arbitrary view facial images," in *European Conference on Computer Vision*. Springer, 2010, pp. 490–503.
- [48] U. Tariq and T. S. Huang, "Features and fusion for expression recognition—a comparative analysis," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012 IEEE Computer Society Conference on. IEEE, 2012, pp. 146–152.
- [49] S. Moore and R. Bowden, "Local binary patterns for multi-view facial expression recognition," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 541–558, 2011.
- [50] A. Caroppo, A. Leone, and P. Siciliano, "Facial expression recognition in older adults using deep machine learning," in *Third Italian Workshop on Artificial Intelligence for Ambient Assisted Living, 2017*, pp. 30–43.
- [51] C. Van Rijsbergen, "Information retrieval. dept. of computer science, university of glasgow," URL: citeseer.ist.psu.edu/vanrijsbergen79information.html, vol. 14, 1979.
- [52] S. Yang and B. Bhanu, "Facial expression recognition using emotion avatar image," in *Face and Gesture 2011*. IEEE, 2011, pp. 866–871.
- [53] A. Dhall, A. Asthana, R. Goecke, and T. Gedeon, "Emotion recognition using phog and lpq features," in *Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011 IEEE International Conference on. IEEE, 2011, pp. 878–883.



Md Tauhid Bin Iqbal received his Bachelor degree (Dec 2012) in Information Technology from University of Dhaka, Bangladesh. He has received his Ph.D. degree (Feb 2019) in Computer Science & Engineering from Kyung Hee University, South Korea, where he is continuing his postdoctoral research as well. His current research interests include deep learning interpretation, facial analysis & recognition including expression, age & gender recognition.



Byungyong Ryu received the B.Sc. degree in 2010 and PhD degree in 2017 in computer engineering from Kyung Hee University, South Korea, where he had conducted his Post-doctoral research as well. He is currently working in POSCO ICT Center, Pangyo-ro, South Korea. His research interests include facial expression, age, and gender recognition using face images, and image enhancement and medical image processing in dentistry, and deep learning.

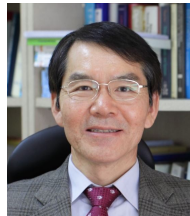


Adin Ramirez Rivera (S'12, M'14) received his B.Eng. degree in Computer Engineering from Universidad de San Carlos de Guatemala (USAC), Guatemala in 2009. He completed his M.Sc. and Ph.D. degrees in Computer Engineering from Kyung Hee University, South Korea in 2013. He is currently an Assistant Professor at the Institute of Computing, University of Campinas, Brazil. His research interests are video understanding (including video classification, semantic segmentation, spatiotemporal feature modeling, and generation), and understanding and creating complex feature spaces.



analysis & recognition.

Farkhod Makhmudkhujaev received the B.S. and M.S. degrees from Tashkent University of Information Technologies, Uzbekistan, in 2012, and 2014 respectively, and Ph.D. degree in computer science and engineering from Kyung Hee University, Rep. of Korea, in 2019. He is currently conducting his postdoctoral research at the Dept. of Information and Communication Engineering, Inha University. His current research interests include image synthesis using generative adversarial networks, and facial attribute



His current research interests include multimedia data processing environments, intrusion detection systems, sensor networks, and medical image processing in dentistry. Prof. Chae is a member of the SPIE, Korean Electronic Society (KES), and the Institute of Electronics, Information and Communication Engineers.

Oksam Chae (M92) received the B.Sc. degree in electronics engineering from Inha University, Incheon, South Korea, in 1977, and the M.S. and Ph.D. degrees in electrical and computer engineering from Oklahoma State University, Stillwater, in 1982 and 1986, respectively. He was a Research Engineer with Texas Instruments Image Processing Laboratory, Dallas, TX, from 1986 to 1988. Since 1988, he has been a Professor with the Department of Computer Engineering, KyungHee University, Gyeonggido, South Korea.



computer Science and Engineering, Kyung Hee University, Suwon, Republic of Korea. His current research interests include model compression, interpretation and architecture search for deep neural networks.

Sung-Ho Bae (M'17) received the B.S. degree from Kyung Hee University, Suwon, Republic of Korea, in 2011, and the M.S and Ph.D. degrees from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2012 and 2016, respectively. From 2016 to 2017, he was a Postdoctoral Associate with Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), MA, USA. Since 2017, he has been an Assistant Professor with the department of Com-