The CSI Journal on
Computer Science and Engineering
Vol. 20, No. 1, 2026
Pages 53–66
doi.org/10.22034/jcse.2025.548510.1062
Regular Paper

Fully Dispersed Haar-like Filters for Enhanced Facial Feature Extraction and Recognition

Zeinab Sedaghatjoo *, Hossein Hosseinzadeh, Ahmad Shirzadi

Faculty of Intelligent Systems Engineering and Data Sciences, Persian Gulf University, Bushehr, Iran.

Abstract:
Haar-like filters are well known for their simplicity, speed, and accuracy in various computer vision tasks. This paper proposes a novel algorithm to identify the optimal fully dispersed Haar-like filters for enhanced facial feature extraction. Unlike traditional Haar-like filters, the proposed filters allow pixels to move freely within an image, thereby capturing intricate local features more effectively. Extensive experiments on face detection and facial expression recognition demonstrate that the optimized filters can distinguish between face images and clutter with minimal error, thereby outperforming existing algorithms. By leveraging a dataset-driven approach to optimize filter weights, the proposed method achieves high accuracy in facial feature extraction, making it a promising tool for various computer vision applications. The MATLAB code corresponding to the proposed algorithm is available at https://github.com/Sedaghatjoo/fully-dispersed-Haar-like-filter.

Keywords: Feature extraction, Machine learning algorithms, Computer vision, Haar-like filters, Facial expression recognition

1 Introduction

Image processing and computer vision are two of the most extensive fields of this decade [1, 2]. Automatic face detection is one of the first computer vision applications [3, 4, 5]. The Viola-Jones face detector [6], developed by Paul Viola and Michael Jones in 2001, is a pioneering framework in machine learning for object detection, particularly face detection. It employs a method based on learning rigid templates, specifically utilizing boosted cascades of classifiers, which has inspired numerous advancements in the field of face detection [7, 8, 9, 10]. However, boosting-based face detection methods encounter two significant challenges: selecting appropriate features for extraction and determining the optimal learning algorithm to implement. In the Viola-Jones detector, Haar-like features serve as weak classifiers, and a robust classifier is formed by combining these weak hypotheses [6]. Haar-like features are local texture descriptors that quantify differences in the average intensity values between adjacent rectangular regions [11, 12]. Figure 1 illustrates several simple Haar-like filters that are commonly employed in feature extraction applications. One of the primary advantages of Haar-like features is their computational efficiency, which is enhanced by the use of integral images [13]. As illustrated in Figure 1, Haar-like filters are commonly represented as combinations of two or more rectangular regions composed of black and white pixels [6]. This figure showcases various examples of Haar-like filters, with the default weights clearly indicated on their respective rectangles. The feature value g for a Haar-like filter is defined as follows:

$$g(\mathbf{x}) = v_1\, m_1(\mathbf{x}) + v_2\, m_2(\mathbf{x}), \tag{1.1}$$

where $\mathbf{x}$ is an image, and m_1 and m_2 denote the mean intensities of the pixels within the black and white regions of the filter, respectively. The weights v_1 and v_2 correspond to these regions and are typically represented as two integers whose sum equals zero. Numerous algorithms are available for determining the optimal values of weights, including brute-force search (BFS) [6], genetic algorithms (GA) [14, 15, 16], and Fisher's linear discriminant analysis (FLDA) [17]. This paper focuses specifically on identifying

optimal Haar-like filters characterized by equal black and white regions. In this context, the weights are assigned as $v_1 = -1$ and $v_2 = 1$.

In [18], a novel type of Haar-like filter, termed the dispersed Haar-like filter, is introduced. Unlike traditional Haar-like filters, where the white and black regions are typically contiguous, this new filter features generally disjoint white and black components. The creation of the dispersed Haar-like filter begins with a matrix that is randomly initialized with values of 1, 0, and -1. Subsequently, optimization algorithms such as Differential Evolution (DE), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO) are employed to refine the pixel values. In this context, pixels assigned a value of 1 represent the white parts of the filter, while those assigned -1 correspond to the black parts. Pixels with a value of 0 do not contribute to the filter's functionality. The use of non-adjacent rectangles, as proposed in [18], enhances the flexibility of Haar-like filters, thus improving their robustness in face detection. According to the findings presented in [18], these filters significantly enhance detection rates for three main reasons:

- **Flexibility:** Traditional Haar-like filters possess a rigid structure, whereas the dispersed Haar-like filter offers greater adaptability.
- **Optimal Structure:** The configuration of the new filter can be optimized using various algorithms.
- **Local Feature Extraction:** The dispersed arrangement of pixels enables effective extraction of local facial features.

This paper investigates Haar-like filters and introduces a novel concept: fully dispersed Haar-like filters. A distinctive feature of these new filters is their adaptability, allowing pixels to reposition themselves dynamically to optimize the detection of local features. These filters mark a significant advancement within the Haar-like framework, merging ease of use with improved accuracy in feature detection. To mitigate the risk of overfitting, we develop an optimization algorithm that facilitates the creation of locally optimized dispersed Haar-like filters, specifically targeting regions of the face to enhance detection precision. Our approach is not limited to face detection; it also extends to facial expression recognition across seven distinct classes. The experimental results presented herein demonstrate the superior performance of the newly optimized Haar-like filters, affirming their effectiveness in practical applications. The structure of this paper is organized as follows: Section 2 reviews related works. Section 3 analyzes Haar-like filters and introduces our proposed fully dispersed Haar-like filters, alongside a groundbreaking optimization algorithm designed to enhance their performance. This algorithm is further elaborated in Section 4, detailing its role in creating locally optimized dispersed Haar-like filters for improved detection precision in specific facial regions. Section 5 presents additional experimental results that assess the accuracy and efficiency of the new Haar-like filters for both face detection and facial expression recognition, comparing these findings with those derived from the LBP and Viola-Jones algorithms. Finally, Section 6 provides a concise summary of our findings and suggests some directions for future research.

2 Related works

The Haar-like filters were applied for face detection by Viola and Jones [6] in 2000. The Viola and Jones method has been widely applied in digital cameras and image organization software since then [19, 20]. The literature is full of applications of Haar-like filters. For instance, a generalized Haar filter-based convolutional neural network is proposed in [21] for both vehicle and pedestrian detection. In [22], a smart cane function is developed by integrating face recognition features on the cane using Haar-like features and Eigenfaces. Additionally, in [23], an enhanced two-layer face detector composed of both Haar-like and multi-block local binary pattern (LBP) features is presented. In [24], a Haar-like local ternary co-occurrence pattern is designed for image retrieval applications. This pattern uses four different Haar-like filters to capture directional information in the image. In [25], a Haar-like descriptor based on the integral image and a multilayer perceptron-type classifier is proposed. Experimental results show that the system inherits advantages of the Haar-like descriptor and artificial neural networks in terms of robustness and speed. Haar-like filters are also applied in deep convolutional neural networks (DCNN) for feature extraction [26, 27]. A novel pupil detection pipeline is proposed in [28] that contains suitable Haar-like features for 2D-pupil detection. An algorithm is presented in [29] for constructing Haar-like bases on general discrete hierarchical trees. Also, Haar-like filters have been applied in several fields of data analysis, demonstrating their efficiency [30, 31, 32, 33].

In [34], the performance of three commonly used object detection approaches - Histogram of Oriented Gradients (HOG), Haar-like features, and LBP - is investigated, and a

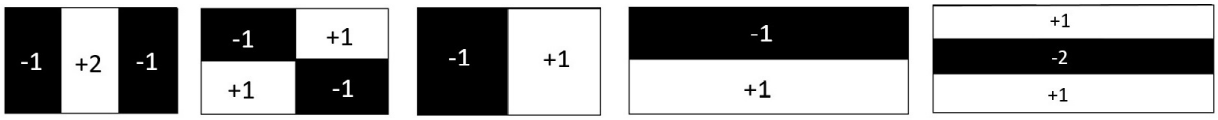

Figure 1: A wide illustration of multiple Haar-like filter configurations.

robust detection algorithm is proposed using a combination of the three different feature descriptors and AdaBoost cascade classification. Experimental results from [34] show that LBP features outperform the other two feature types with a higher detection rate. This indicates, although standard Haar-like features are significantly simple and fast in applications, they have some limitations in accuracy [35, 18]. To enhance the accuracy of Haar-like features, various variations have been proposed, such as joint Haar-like features [11], rotated Haar-like features [36, 37], block difference features [38], and Haar-like features with disjoint rectangles [39, 40, 18]. Additionally, in [18], the Haar-like features are subjected to the application of differential evolution (DE), genetic algorithm (GA), and particle swarm optimization (PSO) to accelerate the Viola-Jones classifier using Haar-like features. An extension of the Viola and Jones detection framework is presented in [39] by removing the geometry restriction of Haar-like filters. The authors of [39] propose a richer representation called scattered rectangle filters, which explore more orientations than horizontal, vertical, and diagonal Haar-like filters. Also, a novel type of filter named Non-Adjacent Rectangle (NAR) Haar-like filter is introduced in [40] to characterize the co-occurrence between facial landmarks and their surroundings. Traditional Haar-like features and NAR Haar-like features are then combined in [40] to form more powerful representations. Consequently, the non-adjacent rectangles proposed in [18] provide greater flexibility for Haar-like filters. However, the optimal positions of these rectangles are not clearly defined in [18], and the idea needs some modifications.

Therefore, a review of the literature on Haar-like feature extraction reveals that analytical studies in this field are inadequate, highlighting the need for optimal Haar-like filters. This work aims to address this gap by proposing an efficient algorithm that optimizes these filters to enhance their feature extraction capabilities for more accurate classification.

3 Optimal Haar-like filters

In the mathematical aspect, two-dimensional images in a dataset can be converted into one-dimensional vectors by arranging the pixels contained in the images. In this situation, an image of size 64×64 is transformed to a row vector of length 4096. Then, the black and white regions of a Haar-like filter can be represented by two vectors, $\mathbf{b}$ and $\mathbf{w}$, respectively, evaluated as:

$$\mathbf{b}(p) = \begin{cases} 1, & \text{if } p \in B, \\ 0, & \text{o.w.} \end{cases}, \quad \mathbf{w}(p) = \begin{cases} 1, & \text{if } p \in W, \\ 0, & \text{o.w.} \end{cases}, \tag{3.1}$$

in vector form, where p varies from 1 to the length of the vector obtained by image $\mathbf{x}$. Here, B and W denote the black and white regions of the filter, respectively. Then the feature value, g, can be expressed in vector form as

$$g(\mathbf{x}) = v_1\, m_1(\mathbf{x}) + v_2\, m_2(\mathbf{x}) = [m_1(\mathbf{x}), m_2(\mathbf{x})].\,[v_1, v_2]^T, \tag{3.2}$$

where m_1 and m_2 are computed as

$$m_1(\mathbf{x}) = \frac{1}{N_{\mathbf{b}}} \mathbf{b}.\,\mathbf{x}^T, \quad m_2(\mathbf{x}) = \frac{1}{N_{\mathbf{w}}} \mathbf{w}.\,\mathbf{x}^T. \tag{3.3}$$

such that $N_{\mathbf{b}}$ and $N_{\mathbf{w}}$ are the numbers of pixels in black and white regions of the filter, respectively. Note that, $N_{\mathbf{b}} + N_{\mathbf{w}}$ represents the total number of engaged pixels in the filter, denoted by N. The superscript T on vector $\mathbf{x}$ shows the transpose operator converts a row vector to a column one. To label image $\mathbf{x}$ as either the object or clutter, feature value g is compared to threshold θ, and the classification is performed by the classifier $h(\mathbf{x})$ as:

$$h(\mathbf{x}) = \begin{cases} +1 \ \text{(or object)} & \text{if } g(\mathbf{x}) > \theta, \\ -1 \ \text{(or clutter)} & \text{if } g(\mathbf{x}) < \theta. \end{cases} \tag{3.4}$$

Note that, the above equation can be abbreviated as

$$h(\mathbf{x}) = sign(g(\mathbf{x}) - \theta),$$

where $sign(.)$ is a function that returns the sign of a real number.

Traditional Haar-like filters are considered too simple, and various modifications have been proposed to enhance their performance, as documented in previous studies [11, 41, 18]. These modifications primarily vary in terms of orientation

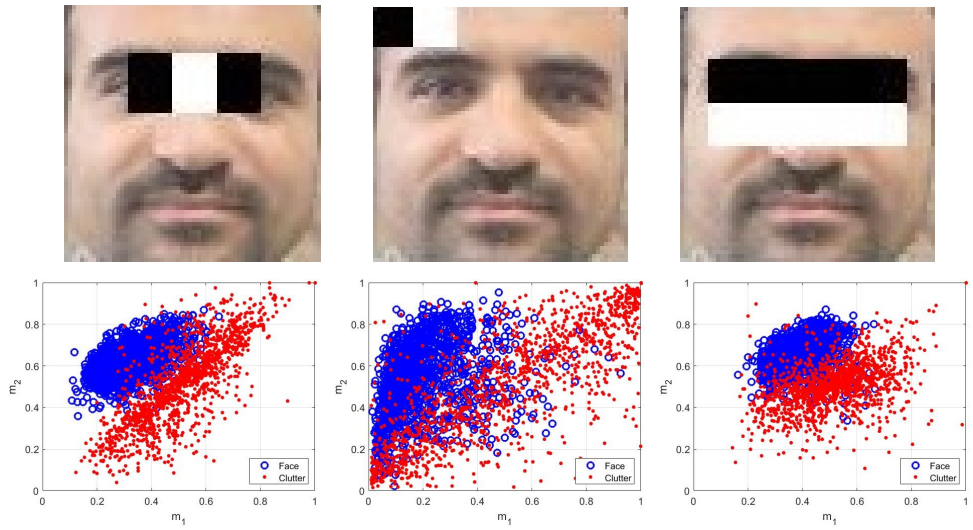

Figure 2: Three rectangular Haar-like filters imposed on some face and clutter images (first row), and the distribution of mean measurement for the images (second row). The first Haar-like filter characterizes the intensity difference between the eyes and the bridge of the nose. The second descriptor gauges the intensity contrast between the corner of the hairline and the left side of the forehead. The third filter measures the intensity difference between the eyes and the cheekbones area.

and the number of rectangles in relation to the template. In Figure 2, three rectangular Haar-like filters are depicted overlaying on facial and clutter images, along with the corresponding distribution of mean measurements (m_1, m_2). The mean measurements of the images were already studied in [41]. For better visual clarity, the number of rectangles in the Haar-like filters, depicted in Figure 2, is limited to three. The first Haar-like filter characterizes the intensity difference between the eyes and the region encompassing the bridge of the nose. The second descriptor gauges the intensity contrast between the corner of the hairline and the left side of the forehead. Finally, the third Haar-like filter measures the intensity difference between the eyes and the cheekbones area.

The second row of Figure 2 is dedicated to the distribution of mean measurements for 1410 face and 1410 clutter images. The face images are sourced from databases CFD [42], CFD-MR [43], and CFD-INDIA [44], where the facial regions were manually cropped and resized to images of size 64×64. This combined dataset is referred to as CFD-T in this paper. Additionally, the clutter images are extracted from a dataset that contains no human faces. Some representative face and clutter images from these databases are shown in Figure 3. The studied images are grayscale, and the light intensity of pixels in the images is normalized by dividing their values by 255. The distribution of the intensity histogram of images is equalized using the histogram equalizer function in Matlab software, called "histeq". Subsequently, m_1 and m_2 are calculated for the images, and their mean measurements are plotted in the second row of Figure 2. Similar to the methodology in [41], the points plotted on the (m_1, m_2) graph corresponding to face and clutter images are referred to as face and clutter points, respectively, in this paper. From Figure 2, it is evident that the face points are predominantly distributed on top of the clutter points, irrespective of the type and size of the Haar-like filters. Furthermore, from the figure, it can be observed that the face points exhibit a high degree of correlation with each other, while the clutter points are generally spread out in the region.

3.1 The filters based on the mean values

We look for the best position of the Haar-like rectangles that include human face features. Since the mean of the face images statistically is the best representation for the images, it may have the most important face features. Numerical experiments reveal that the mean of clutter images is also crucial for classification. Assume the face and clutter databases contain N_f and N_c vectorized face and clutter images of size 64×64, respectively. Then each image is converted to a vector of length 4096 after vectorization. The mean values of the face and clutter images are computed as follows:

$$\mathbf{m}_F = \frac{1}{N_f} \sum_{i=1:N_f} \mathbf{f}_i, \quad \mathbf{m}_C = \frac{1}{N_c} \sum_{i=1:N_c} \mathbf{c}_i, \tag{3.5}$$

where $\mathbf{f}_i$ and $\mathbf{c}_i$ represent the i-th vectorized face and clutter images, respectively. One can see that the image of $\mathbf{m}_F$ is

Figure 3: Some human frontal face (left) and clutter (right) images in the face and clutter databases, respectively.

also a face, while the image of $\mathbf{m}_C$ is a gray-scale neutral image with a light intensity of 0.5 when the light is normalized between 0 and 1. This value of $\mathbf{m}_C$ shows that clutter images may have no distinguishable features! If the black and white rectangles of a Haar-like filter are positioned over the corresponding black and white regions of the mean face image in a way that maximizes the feature value $g(\mathbf{m}_F)$ then the maximum value of

$$m_{g_F} = \frac{1}{N_f} \sum_{i=1:N_f} g(\mathbf{f}_i) = g(\frac{1}{N_f} \sum_{i=1:N_f} \mathbf{f}_i) = g(\mathbf{m}_F). \tag{3.6}$$

is obtained. Here, m_{g_F} represents the mean of feature values for the face images. From Equation (3.2), the value of m_{g_F} can be interpreted as the length of the projection of the vector $[m_1(\mathbf{m}_F), m_2(\mathbf{m}_F)]$ on $[v_1, v_2]$. Consequently, as m_{g_F} increases, the face points in the (m_1, m_2) graph are shifted toward the left-top corner.

Similar relationships can be established for the mean measurements of the clutter images, with the aim of shifting them towards the right-down corner in the (m_1, m_2) graph by decreasing the mean value of g for them. Then, if the black and white rectangles of a Haar-like filter are arranged such that $g(\mathbf{m}_C)$ is minimized, then the mean value of g for the clutter images, denoted by m_{g_C}, is also minimized. This is because:

$$m_{g_C} = \frac{1}{N_c} \sum_{i=1:N_c} g(\mathbf{c}_i) = g(\frac{1}{N_c} \sum_{i=1:N_c} \mathbf{c}_i) = g(\mathbf{m}_C). \tag{3.7}$$

From Equation (3.2), m_{g_C} is the projection of vector $[m_1(\mathbf{m}_C), m_2(\mathbf{m}_C)]$ on $[v_1, v_2]$. Note, the classifier h performs better when the Haar-like filter is configured to maximize m_{g_F} and minimize m_{g_C} simultaneously. This is equivalent to maximizing $m_{g_F} - m_{g_C}$. In this scenario, the distance between the mean values m_{g_F} and m_{g_C} is maximized, leading to better discrimination between face and clutter images. To achieve this goal, it is sufficient to define a new vector

$$\mathbf{m}_{FC} = \mathbf{m}_F - \mathbf{m}_C, \tag{3.8}$$

and find a Haar-like filter such that maximizes $g(\mathbf{m}_{FC})$. Therefore, since

$$\begin{aligned} s^* &= \max\{g(\mathbf{m}_{FC})\} \\ &= \max\{g(\mathbf{m}_F) - g(\mathbf{m}_C)\} \\ &= \max\{\frac{1}{N_f} \sum_{i=1:N_f} g(\mathbf{f}_i) - \frac{1}{N_c} \sum_{i=1:N_c} g(\mathbf{c}_i)\} \\ &= \max\{m_{g_F} - m_{g_C}\}, \end{aligned} \tag{3.9}$$

the distance between m_{g_F} and m_{g_C} is maximized when $g(\mathbf{m}_{FC})$ is maximized. Four disjoint Haar-like filters and the mean measurements of the face and clutter images corresponding to these filters are illustrated in Figure 4. These filters satisfy Equation (3.9), with different numbers of disjoint white and black blocks.

From Figure 4, it is evident that the first filter pushes the face points toward the left-top corner in the (m_1, m_2) graph. However, since the clutter images contain no feature, the clutter points are clustered around the center point $(0.5, 0.5)$. This observation remains valid for Haar-like filters with more blocks. Additionally, the fourth filter achieves a better separation of face and clutter points compared to the first filter. This effect reveals that Haar-like filters with finer blocks, resulting in larger s^*, lead to better classifiers.

The number of blocks can be increased up to the size of the filter if the blocks are shrunk to the pixels. Then, the blocks are replaced by the pixels in the filter in this situation. These replacements conclude the best Haar-like filter that one can find to disjoint face and clutter points. This filter is named *fully dispersed Haar-like filter* in this paper. A fully dispersed Haar-like filter based on the CFD-T dataset is shown in the last column of Figure 4 for $N_{\mathbf{b}} = N_{\mathbf{w}} = 256$ and the size of face and clutter images is 64×64. The filter can be easily constructed by sorting the pixels of the

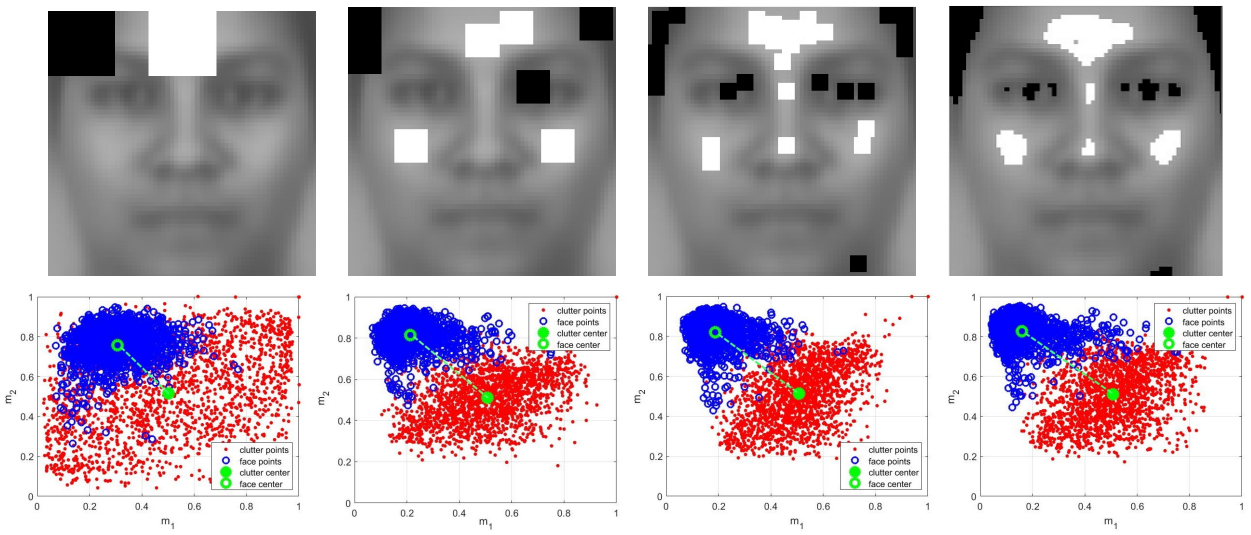

Figure 4: Graph of four disperse Haar-like filters (first row) and mean measurement of the images for them (second row). The filters satisfy Equation (3.9), with different numbers of disjoint white and black blocks. One can see, the filters with smaller blocks exhibit larger values of s^*, allowing for better separation between face and clutter points.

vector $\mathbf{m}_{FC}$ according to light intensity and selecting 256 lower and 256 upper ones as the black and white regions of the filter, respectively. The following algorithm outlines the process of creating a fully dispersed Haar-like filter with arbitrary size in Matlab software.

Algorithm 1 : Making a fully dispersed Haar-like filter.

define $\mathbf{m}_{FC}$ as Equation (3.8).
define $N_{\mathbf{b}}, N_{\mathbf{w}}$ and $N = N_{\mathbf{b}} + N_{\mathbf{w}}$.
initialize $\mathbf{b}$ and $\mathbf{w}$ as zero vectors of size $\mathbf{m}_{FC}$.
[$\mathbf{r}$, $\mathbf{o}$]=sort($\mathbf{m}_{FC}$).
$\mathbf{b}(\mathbf{o}(1{:}N_{\mathbf{b}}))$=1.
$\mathbf{w}(\mathbf{o}(\text{end-}N_{\mathbf{w}}{:}\text{end}))$=1.

It is notable that, by comparing Figure 4 and Figure 2, it is evident that dispersed Haar-like filters achieve superior separation of face and clutter points compared to conventional filters, where the white and black regions are joined. Thus, the optimal Haar-like filter will be disjoint, as is presented in the following section.

3.2 Optimal weights

Now, there is an important question: Can one distribute pixels of a dispersed Haar-like filter optimally? This optimization would result in less within-class variance of face and clutter points, leading to more accurate results. To optimize the filter, more delicate features should be extracted from the images and appended to the filter. The sought feature aims to decrease variances, particularly by incorporating features shared with misclassified images. It is important to note that the classifier (3.4) classifies images with the optimal value of θ, and we need only to disjoint the face and clutter points in the (m_1, m_2) graph. The impact of the misclassified images can be amplified by using weighted summations

$$\mathbf{m}_F = \sum_{i=1:N_f} w_i^f \mathbf{f}_i \ , \quad \mathbf{m}_C = \sum_{i=1:N_c} w_i^c \mathbf{c}_i \ , \tag{3.10}$$

instead of the mean values (3.5), where $w_i^f \geq 0, w_i^c \geq 0, \sum_{i=1:N_f} w_i^f = 1, \sum_{i=1:N_c} w_i^c = 1$, and higher weights are considered for the misclassified images. Four dispersed Haar-like filters are shown in Figure 5 based on some optimal weights. One can see that the filters separate the face and clutter points linearly. Then, classifier h is able to classify the face and clutter images, accurately for different values of N.

This shows that the weighted summations (3.10) are significantly more efficient than the mean values (3.8), provided that the weight vectors $\mathbf{w}^f = [w_1^f, w_2^f, ..., w_{N_f}^f]$ and $\mathbf{w}^c = [w_1^c, w_2^c, ..., w_{N_c}^c]$ are appropriately selected. The weights leading to the optimal filters are updated according to:

$$w_i^{f/c} \leftarrow \begin{cases} w_i^{f/c} + c & \text{if } \mathbf{x}_i \text{ misclassified,} \\ w_i^{f/c} + d & \text{o.w.,} \end{cases} \tag{3.11}$$

iteratively for trial image $\mathbf{x}_i$ where $c \geq d \geq 0$. To perform this update, we can use the sigmoid function as:

$$w_i^f \leftarrow w_i^f + \text{sigmoid}(-y_i), \tag{3.12}$$

$$w_i^c \leftarrow w_i^c + \text{sigmoid}(+y_i), \tag{3.13}$$

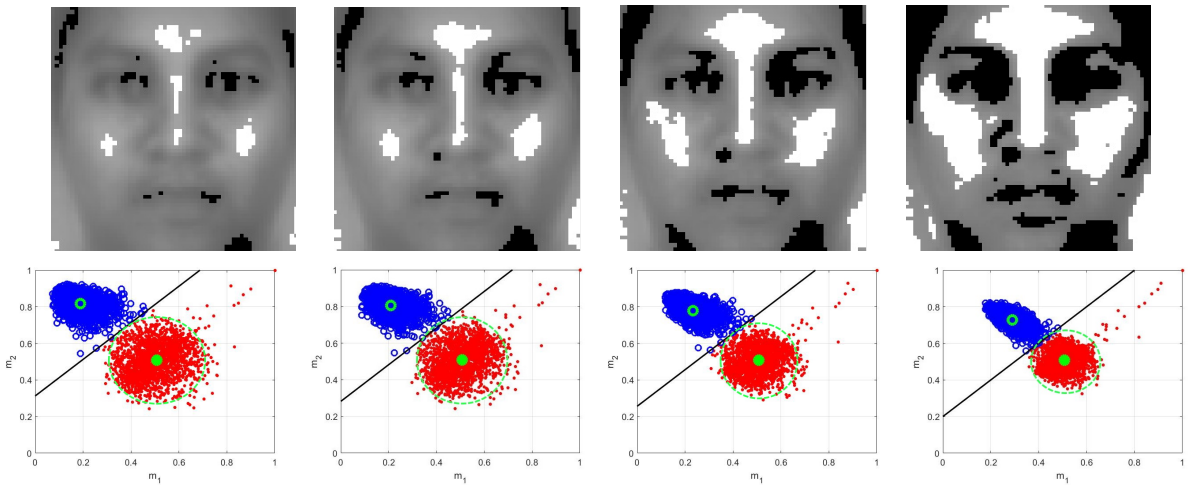

Figure 5: Four optimized fully dispersed Haar-like filters (first row) and mean measurement of the images for them (second row). The size of the filters is $N = 256, 512, 1024$ and 2048 for the first, second, third and fourth columns, respectively. From the second row of this figure, the optimized filters disjoint face and clutter images, linearly.

where $y_i = g(\mathbf{x}_i) - \theta$ and

$$\text{sigmoid}(y_i) = 1/(1 + \exp(-\varepsilon y_i)).$$

The sigmoid function transforms real values to the interval $(0, 1)$ softly when ε is small, and it transforms them to binary values $\{0, 1\}$ when ε tends to ∞. After updating the weights, we normalize them by dividing each by their summation as:

$$w_i^{f/c} \leftarrow \frac{w_i^{f/c}}{\sum_{i=1:N_f} w_i^{f/c}}. \tag{3.14}$$

Then, Algorithm 2 is suggested to derive optimal fully dispersed Haar-like filters.

Interested readers can find Matlab code corresponding to the new proposed algorithm at "https://github.com/Sedaghatjoo/fully-dispersed-Haar-like-filter". Note that the value of ε is very important in this algorithm, and it works like a learning rate in machine learning algorithms. Numerical experiments reveal that Algorithm 2 works well only if ε is selected carefully, especially when the number of data is larger than $10,000$. Large values of ε may lead to divergence and small ones may reduce the rate of convergence. When ε is large, pixels of the dispersed Haar-like filter may be distributed separately over the region without effectively capturing valuable face features. In this situation, we say that overfitting has occurred for the filter. Figure 6 displays images of two fully dispersed Haar-like filters trained on a dataset containing more than $20,000$ images, with over $9,000$ face and $11,000$ clutter images. The face images were collected from publicly available facial image datasets such as UTKFace and CDF-T (B-Datasets) [45]. The left filter in Figure 6 exhibits overfitting, while the right one does not. These filters are obtained by Algorithm 2 after 300 iterations for $\varepsilon \simeq \infty$ and $\varepsilon = 20$, respectively. The pixels of the first filter are dispersed widely across the region, while they are predominantly located on the face features for the second filter.

The filters indicated in Figure 5 are obtained by Algorithm 2 after at most 300 iterations. Although the filters lead to accurate results, it is advised not to use filters with less than or equal to $N = 128$ pixels for face detection, as they are very sensitive to preprocessing and may not effectively capture face features in the presence of noise in the data. Therefore, it is advisable to use filters with moderate sizes, e.g. $N = 512$, when the size of the trial image is 64×64.

The same approach can also be applied for general forms of disjoint Haar-like filters, as shown in Figure 4, to find the optimal location of their rectangles. However, Algorithm 1 is designed only for fully dispersed Haar-like filters, and a new, more complicated algorithm is required for them. Therefore, since the optimal filters are fully dispersed, we will only use fully dispersed Haar-like filters in the experimental results presented in the forthcoming section.

These local filters can also be applied in CNNs by focusing on specific regions of the human face. In a special case, 64 small filters are shown in Figure 8, where the face images in the CFD-T dataset are divided into 8×8 equal parts. The classification error for each filter is reported above it when it is applied for face detection. The results demon-

Algorithm 2 : The algorithm to find the best weights of data without over fitting.

define face vector $\mathbf{f}_i$ and initialize $w_i^f = 1/N_f$ for $i = 1, 2, ..., N_f$.
define clutter vector $\mathbf{c}_i$ and initialize $w_i^c = 1/N_c$ for $i = 1, 2, ..., N_c$.
define $\mathbf{m}_F$ and $\mathbf{m}_C$ as presented in (3.10).
set $\mathbf{m}_{FC} = \mathbf{m}_F - \mathbf{m}_C$
find vectors $\mathbf{b}$ and $\mathbf{w}$ by Algorithm 1.
find error of classifier h presented in Eq. (3.4).
initialize shape parameter ε.
while error of the classification is not zero **do**
    update the weights as (3.12).
    normalize the weights as (3.14).
    update $\mathbf{m}_F$ and $\mathbf{m}_C$ as presented in (3.10).
    update $\mathbf{m}_{FC}$ as $\mathbf{m}_{FC} = \mathbf{m}_F - \mathbf{m}_C$.
    find new vectors $\mathbf{b}$ and $\mathbf{w}$ by Algorithm 1.
    find error of classifier h presented in Eq. (3.4).
end while

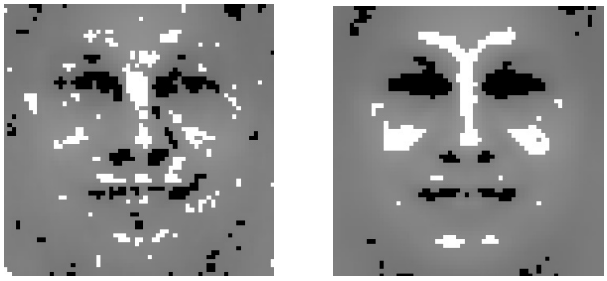

Figure 6: Images of two fully dispersed Haar-like filters. These filters are obtained by Algorithm 2 with shape parameters $\epsilon \simeq \infty$ and $\epsilon = 20$, respectively. The pixels in the left filter are distributed more freely across the image, whereas those in the right filter are primarily clustered around prominent features. The left filter leads to the overfitting while the right filter yields more stable results.

strate that these filters can reduce the error rate to less than 20% when they focus on critical areas of the face such as the eyes, eye corners, nose bridge, tip of the nose, and the center of the mouth. These regions are highlighted in Figure 8 in blue. In some cases, the error can be further reduced to 12%. These filters can be utilized in CNNs as convolutional filters in the first layer to extract prominent human face features. However, integrating these filters into a CNN presents two challenges; the consistency may deteriorate if the filter size becomes too large, and the accuracy may decline if the filter size is too small. So, the optimal size of the filter will be an open problem in this field.

4 Local dispersed Haar-like filters

Local dispersed Haar-like filters can also be obtained using Algorithm 2. These filters tend to yield improved results when a part of a face image is damaged or noisy. For instance, finding people wearing face masks in public places is a challenging task for which these filters can help. The proposed filters with respect to the dataset are displayed in Figure 7. The first three local filters in the figure are applied to the top, middle, and bottom parts of a human face. Obtained using Algorithm 2, the errors of classification with respect to these filters are displayed in Figure 7. The results demonstrate that the local filters reduce false positive and false negative errors to 3%, 6%, and 17%, respectively, after 400 iterations. Given the higher accuracy of the proposed local filters compared to the others, it is evident that the filter captures more important face features. It must be noted that the optimized global dispersed Haar-like filter, presented in the previous subsection, results in 2% error. This indicates that the first local filter performs nearly similar to the global one. The first local filter achieves high accuracy by including the eyes and the surrounding region. However, the filter can be modified by adding the bridge noise and cheeks to it. Then, the fourth local filter, presented in Figure 7, is obtained, where it leads to 2.5% error. Notably, the accuracy is reduced significantly for the third filter due to the presence of noise. Variations in facial features, such as the presence or absence of a beard, can act as noise for the third filter, leading to a significant reduction in accuracy.

These local filters can also be applied in CNNs by focusing on specific regions of the human face. In a special case, 64 small filters are shown in Figure 8, where the face images in the CFD-T dataset are divided into 8×8 equal parts. The classification error for each filter is reported above it when it is applied for face detection. The results demonstrate that these filters can reduce the error rate to less than 20% when they focus on critical areas of the face such as the eyes, eye corners, nose bridge, tip of the nose, and the center

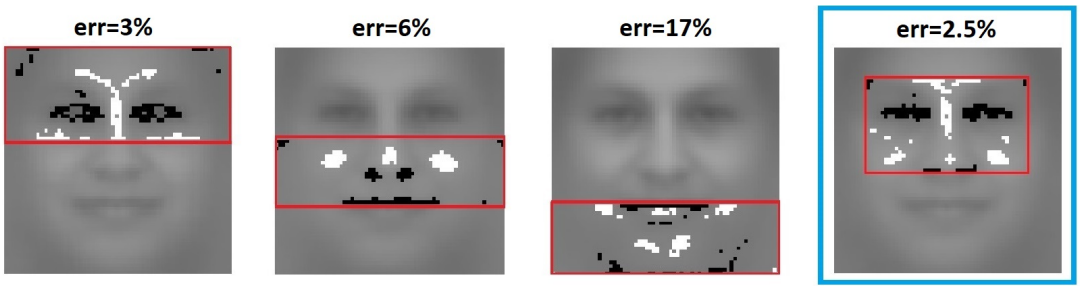

Figure 7: Four local Haar-like filters obtained by Algorithm 2. Errors of the classification with respect to the filters, presented on top of them, reveal that the fourth filter extracts Haar-like face features more efficiently than the others.

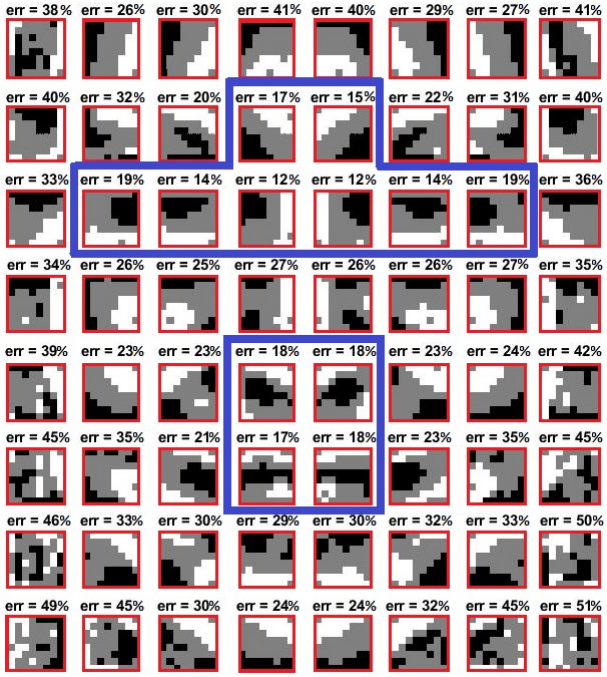

Figure 8: 64 local Haar-like filters can be applied in CNNs for feature extraction. Errors of the classification respect to the filters, presented on top of them, reveal some of them extract face features, efficiently.

of the mouth. These regions are highlighted in Figure 8 in blue. In some cases, the error can be further reduced to 12%. These filters can be utilized in CNNs as convolutional filters in the first layer to extract prominent human face features. However, integrating these filters into a CNN presents two challenges; the consistency may deteriorate if the filter size becomes too large, and the accuracy may decline if the filter size is too small. So, the optimal size of the filter will be an open problem in this field.

5 Experimental results

In this section, we apply the optimal Haar-like filter obtained through Algorithm 2 to face detection and facial expression recognition across various datasets. The algorithm is trained on 70% of a dataset and tested on the remaining 30% to assess its accuracy in face detection. In contrast, 80% of the dataset is used for training and 20% for testing in facial expression recognition due to the limited number of images available. The results are compared to the best outcomes achieved by the Local Binary Patterns (LBP) method [46] and the Viola-Jones algorithm [6], leading to some noteworthy conclusions.

5.1 Binary classification problem

Consider the AR Dataset [45] for face images. Algorithm 2 is employed to generate an optimal fully dispersed Haar-like filter and perform image classification. In Table 1, the accuracy of the classification is reported for seven iterations of the algorithm. The error is smaller than 1% for test images when the number of iterations exceeds 200. Similar results are also obtained for the CF dataset, as reported in the table. The images in AR and CF datasets have high resolution, and they are reshaped to 64×64 before feature extraction. Then, a dispersed Haar-like filter with $N = 512$ engaged pixels is applied for classification in Algorithm 2. Due to the sufficiently large size of the images, the Haar-like filters successfully extract facial features, resulting in accurate recognition of face images. However, the accuracy of the algorithm decreases when the resolution of trial images is lower. For example, the accuracy of the algorithm is less than 97% when the MIT dataset is used for feature extraction. The size of images in the dataset is 19×19. Table 1 presents the accuracy of Algorithm 2 for this dataset. The results are not sufficiently accurate.

Furthermore, the results of Algorithm 2 for AR and CF datasets exhibit some small jumps in certain iterations, indicating a lack of strong stability in the results. To improve stability, it is beneficial to enrich the databases with more images. In such a scenario, the algorithm focuses solely on

more general features while ignoring the noise in the data. Table 1 presents the accuracy of the algorithm for the UTK dataset, which includes more than $10,000$ images. It is notable that the AR and CF datasets include approximately 2000 and 1400 images, respectively. The algorithm's results are stable without jumps for iterations beyond 200. However, this stability might result in slightly reduced accuracy. One can observe a decrease in accuracy for the UTK dataset compared to the AR and CF datasets. The accuracy is approximately 98% for the UTK dataset after 200 iterations, whereas it is over 99% for the AR and CF datasets.

5.2 Multiclass classification problem

This subsection is devoted to applying optimal dispersed Haar-like filters for facial expression recognition. The CK dataset [47] contains 981 grayscale images of facial expressions, each measuring 48×48 pixels. These images are annotated into seven distinct emotional classes: anger (135 samples), contempt (54 samples), disgust (177 samples), fear (75 samples), happiness (207 samples), sadness (84 samples), and surprise (249 samples). Some sample images in the dataset are presented in Figure 9. Algorithm 2 is applied to extract the filters that distinguish between the facial expressions. Then, the Haar-like features are extracted from the filters and used for classification by the support vector machine (SVM) algorithm.

We used both Linear SVM and quadratic SVM for classification. Table 2 presents the accuracy of each class obtained through our experiments in facial expression recognition. The third and sixth columns of the table are related to the newly proposed Haar-like filters (denoted by Haar-like 2), and the second and fifth columns are devoted to a dispersed Haar-like filter obtained by Algorithm 1 (denoted by Haar-like 1). Additionally, the first and fourth columns of the table are devoted to the LBP method, in which 64 features are extracted by dividing the images into 4 equal parts and extracting 16 features in each part [46].

From the table, it can be observed that the new proposed Haar-like filters lead to more accurate results. Note that the accuracy of fear and sadness expressions is significantly lower than the accuracy of the others because the dataset has a low number of images in these classes. The results can also be compared with some well-known techniques for facial expression recognition presented in [48]. From Figure 10, it can be seen that the results obtained from the Haar-like filters are more accurate than the results of some techniques such as HOG, SDM, GF, and GL wavelet.

6 Conclusion and final remarks

Novel optimized fully dispersed Haar-like filters were introduced to extract Haar-like face features. These filters were constructed using one-pixel rectangles, making them fully dispersed Haar-like filters. Since pixels of the filters are allowed to move freely in the whole region of the image, the face features are captured efficiently. However, this freedom may lead to overfitting, which was prevented by controlling the learning rate. Numerical experiments show that false positive and false negative errors of classification were reduced to 5% for images with low resolution and to 2% when the resolution is 64×64 or higher. As directions for future research, the weights of white and black regions of the filters can be optimized to reduce the classification error. Also, optimization seems necessary when the sizes of the white and black regions are not equal. Moreover, using the proposed strategy, local filters can be derived to be used as weak classifiers within well-known classifier frameworks such as Viola-Jones and convolutional neural networks (CNN) for object detection. In this scenario, the local filters will efficiently extract local features, enhancing the efficiency of Haar-like filters in object detection. The authors plan to address this issue in future work.

References

[1] N. Jiang, B. Sheng, P. Li, and T.-Y. Lee, "Photohelper: portrait photographing guidance via deep feature retrieval and fusion," *IEEE Transactions on Multimedia*, vol. 25, pp. 2226–2238, 2022.

[2] T. Mansouri, M. Taha, K. Kiani, K. Kaveh, P. Shivakumara, R. Fadaeidehcheshmeh, and F. Razieh, "A novel topic modelling framework for automated surveying of computer vision research in prostate cancer detection," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 39, no. 03n04, 2025.

[3] M. Mansoorizadeh, N. M Charkari, and E. Kabir, "An expert system for emotion recognition from face image sequences," *The CSI Journal on Computer Science and Engineering*, vol. 4, no. 1, 2006.

[4] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: past, present and future," *Computer Vision and Image Understanding*, vol. 138, pp. 1–24, 2015.

Table 1: Accuracy of the Haar-like classifier (in percent) for a dispersed Haar-like filter obtained by Algorithm 2 in each iteration. Four datasets are considered for the face detection. One can see the accuracy is more than 99% in iteration=200 for AR and CF datasets while it is less than 98% for MIT and UTK datasets.

dataset	iteration						
	0	5	10	20	50	100	200
AR dataset	95.1%	97.1%	98.4%	99.0%	99.2%	99.4%	99.5%
CF dataset	95.5%	98.6%	99.0%	99.5%	99.6%	99.5%	99.6%
MIT dataset	95.8%	96.3%	96.4%	96.5%	96.7%	96.8%	96.9%
UTK dataset	88.0%	96.3%	97.0%	97.4%	97.6%	97.9%	98.0%

Figure 9: Some sample images in the CK dataset. The images show seven distinct emotional classes: anger, contempt, disgust, fear, happiness, sadness and surprise from left to right.

[5] K. A. Tychola, E. Vrochidou, and G. A. Papakostas, "Deep learning based computer vision under the prism of 3d point clouds: a systematic review," *The Visual Computer*, pp. 1–43, 2024.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1, pp. I–I, Ieee, 2001.

[7] A. Ghorbel, N. Ben Amor, and M. Abid, "Gpgpu-based parallel computing of viola and jones eyes detection algorithm to drive an intelligent wheelchair," *Journal of Signal Processing Systems*, vol. 94, no. 12, pp. 1365–1379, 2022.

[8] L. Ramana, W. Choi, and Y.-J. Cha, "Fully automated vision-based loosened bolt detection using the Viola-Jones algorithm," *Structural Health Monitoring*, vol. 18, no. 2, pp. 422–434, 2019.

[9] D. Matalov, S. Usilin, D. Nikolaev, and V. Arlazarov, "Application of cascade methods as a universal object detection tool," *Pattern Recognition and Image Analysis*, vol. 33, no. 4, pp. 685–698, 2023.

[10] B. Rehman, W. H. Ong, A. C. H. Tan, and T. D. Ngo, "Face detection and tracking using hybrid margin-based roi techniques," *The Visual Computer*, vol. 36, no. 3, pp. 633–647, 2020.

[11] T. Mita, T. Kaneko, and O. Hori, "Joint haar-like features for face detection," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, pp. 1619–1626, IEEE, 2005.

[12] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings. international conference on image processing*, vol. 1, pp. I–I, IEEE, 2002.

[13] C. H. Messom and A. L. Barczak, "Stream processing for fast and efficient rotated haar-like features using rotated integral images," *International journal of intelligent systems technologies and applications*, vol. 7, no. 1, pp. 40–57, 2009.

[14] D. D. Gomez, L. H. Clemmensen, B. K. Ersbøll, and J. M. Carstensen, "Precise acquisition and unsupervised segmentation of multi-spectral images," *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 183–193, 2007.

[15] B. Haddow and G. Tufte, "De goldberg. genetic algorithms in search optimization and machine learning. addison-wesley longman publishing co," in *Proceedings of the 2000 Congress on Evolutionary Computation CECOO*, 2010.

[16] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[17] R. Duda, P. Hart, D. Stork, and A. Ionescu, "Pattern classification, chapter nonparametric techniques," 2000.

Table 2: Accuracy of the SVM for face emotional recognition. Three different approaches are applied to extract the features; local binary pattern (LBP), dispersed Haar-like filter obtained by Algorithm 1 (Haar-like 1) and the optimal dispersed Haar-like filter obtained by Algorithm 2 (Haar-like 2).

	Accuracy (%)					
	Linear SVM			Quadratic SVM		
Method:	LBP	Haar-like 1	Haar-like 2	LBP	Haar-like 1	Haar-like 2
anger	51.7	83.3	89.3	74.3	94.4	96.3
contempt	57.7	75.9	88.6	55.0	93.2	94.1
disgust	69.3	93.5	92.2	96.4	96.1	96.9
fear	65.7	66.3	77.3	68.3	85.0	90.0
happiness	84.9	93.3	94.6	93.9	97.9	96.9
sadness	54.1	53.8	66.2	67.6	79.1	87.6
surprise	90.9	93.6	94.2	94.4	96.4	96.4
total	67.8	80.0	86.1	77.1	91.7	94.0

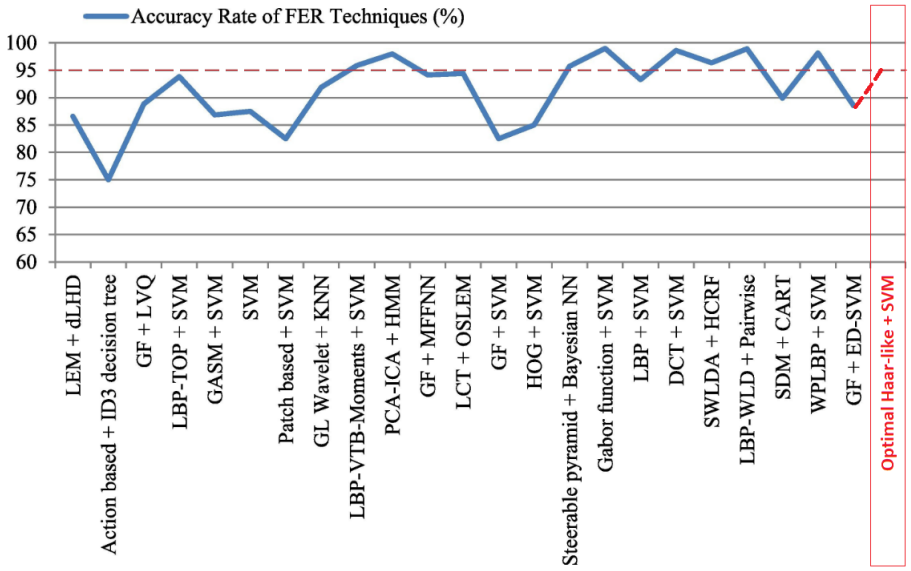

Figure 10: Accuracy rate of various face expression recognition techniques. The new proposed technique (Optimal Haar-like + SVM) is sufficiently accurate, despite its simplicity in application.

[18] M. Besnassi, N. Neggaz, and A. Benyettou, "Face detection based on evolutionary haar filter," *Pattern Analysis and Applications*, vol. 23, pp. 309–330, 2020.

[19] I. G. N. M. K. Raya, A. N. Jati, and R. E. Saputra, "Analysis realization of Viola-Jones method for face detection on CCTV camera based on embedded system," in *2017 International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics)*, pp. 1–5, IEEE, 2017.

[20] T. H. Obaida, A. S. Jamil, and N. F. Hassan, "Real-time face detection in digital video-based on Viola-Jones supported by convolutional neural networks.," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 12, no. 3, 2022.

[21] K. Lu, J. Li, L. Zhou, X. Hu, X. An, and H. He, "Generalized haar filter-based object detection for car sharing services," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1448–1458, 2018.

[22] G. I. Hapsari, G. A. Mutiara, and H. Tarigan, "Face recognition smart cane using haar-like features and eigenfaces," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 2, pp. 973–980, 2019.

[23] T. Dandashy, M. E. Hassan, and A. Bitar, "Enhanced face detection based on haar-like and MB-LBP features," *International Journal of Engineering and Management Research*, vol. 9, 2019.

[24] M. Agarwal and A. Singhal, "Directional local co-occurrence patterns based on haar-like filters," *Multimedia Tools and Applications*, pp. 1–15, 2022.

[25] B. Mohamed, A. Issam, A. Mohamed, and B. Abdellatif, "ECG image classification in real time based on

the haar-like features and artificial neural networks," *Procedia Computer Science*, vol. 73, pp. 32–39, 2015.

[26] J. Ai, R. Tian, Q. Luo, J. Jin, and B. Tang, "Multi-scale rotation-invariant haar-like feature integrated cnn-based ship detection algorithm of multiple-target environment in sar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10070–10087, 2019.

[27] M. Nagare, G. T. Buzzard, and C. A. Bouman, "Texture matching gan for ct image enhancement," *arXiv preprint arXiv:2312.13422*, 2023.

[28] L. Shi, C. Wang, H. Jia, and X. Hu, "Eps: robust pupil edge points selection with haar feature and morphological pixel patterns," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 35, no. 06, p. 2156002, 2021.

[29] R. Archibald and B. Whitney, "Haar-like wavelets on hierarchical trees," *Journal of Scientific Computing*, vol. 99, no. 1, p. 3, 2024.

[30] L. Zhang, J. Wang, and Z. An, "Vehicle recognition algorithm based on haar-like features and improved Adaboost classifier," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 2, pp. 807–815, 2023.

[31] X. Zheng, B. Zhou, M. Li, Y. G. Wang, and J. Gao, "MathNet: Haar-like wavelet multiresolution analysis for graph representation learning," *Knowledge-Based Systems*, vol. 273, p. 110609, 2023.

[32] L. Arreola, G. Gudiño, and G. Flores, "Object recognition and tracking using Haar-like features cascade classifiers: Application to a quad-rotor UAV," in *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, pp. 45–50, IEEE, 2022.

[33] M. Xie, Q. Su, L. Wang, Z. Zhang, B. Ding, X. Xu, and C. Wang, "The research of traffic cones detection based on haar-like features and adaboost classification," 2022.

[34] A. Arunmozhi and J. Park, "Comparison of HOG, LBP and Haar-like features for on-road vehicle detection," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 0362–0367, IEEE, 2018.

[35] K.-Y. Park and S.-Y. Hwang, "An improved haar-like feature for efficient object detection," *Pattern Recognition Letters*, vol. 42, pp. 148–153, 2014.

[36] S. Du, N. Zheng, Q. You, Y. Wu, M. Yuan, and J. Wu, "Rotated haar-like features for face detection with in-plane rotation," in *Interactive Technologies and Sociotechnical Systems: 12th International Conference, VSMM 2006, Xi'an, China, October 18-20, 2006. Proceedings 12*, pp. 128–137, Springer, 2006.

[37] M. Oualla, A. Sadiq, and S. Mbarki, "Rotated haar-like features at generic angles for objects detection," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pp. 351–355, IEEE, 2014.

[38] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[39] W. Zhang, R. Tong, and J. Dong, "Boosting 2-thresholded weak classifiers over scattered rectangle features for object detection.," *Journal of Multimedia*, vol. 4, no. 6, 2009.

[40] X. Zhao, X. Chai, Z. Niu, C. Heng, and S. Shan, "Context modeling for facial landmark detection based on non-adjacent rectangle (nar) haar-like feature," *Image and Vision Computing*, vol. 30, no. 3, pp. 136–146, 2012.

[41] S.-K. Pavani, D. Delgado, and A. F. Frangi, "Haar-like features with optimally weighted rectangles for rapid object detection," *Pattern Recognition*, vol. 43, no. 1, pp. 160–172, 2010.

[42] D. S. Ma, J. Correll, and B. Wittenbrink, "The chicago face database: A free stimulus set of faces and norming data," *Behavior research methods*, vol. 47, pp. 1122–1135, 2015.

[43] D. S. Ma, J. Kantner, and B. Wittenbrink, "Chicago face database: Multiracial expansion," *Behavior Research Methods*, vol. 53, pp. 1289–1300, 2021.

[44] A. Lakshmi, B. Wittenbrink, J. Correll, and D. S. Ma, "The india face set: International and cultural boundaries impact face impressions and perceptions of category membership," *Frontiers in psychology*, vol. 12, p. 627678, 2021.

[45] S. Y. Zhang, Zhifei and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[46] S. H. Khaleefah, S. A. Mostafa, A. Mustapha, and M. F. Nasrudin, "Review of local binary pattern operators in image feature extraction," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 23–31, 2020.

[47] 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*, pp. 94–101, IEEE, 2010.

[48] I. M. Revina and W. S. Emmanuel, "A survey on human face expression recognition techniques," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 6, pp. 619–628, 2021.

Zeinab Sedaghatjoo is a researcher in the Department of Mathematics at Persian Gulf University, Bushehr, Iran. Her research interests focus on machine learning, deep learning (including convolutional neural networks – CNNs), and mathematical methods for feature extraction in computer vision. She has developed novel approaches that combine numerical analysis with texture descriptors, such as local binary patterns (LBP), for facial analysis and image classification.
Email: z.sedaqatjoo@aut.ac.ir

Hossein Hosseinzadeh is a faculty member in the Department of Mathematics at Persian Gulf University. His research concentrates on computational mathematics and machine learning, with a particular emphasis on feature engineering, pattern recognition, and deep learning architectures for visual data analysis. He has collaborated on projects involving support vector machines (SVM), CNNs, and descriptor-based methods for efficient image representation.
Email: hosseinzadeh@pgu.ac.ir
Email: h_hosseinzadeh@aut.ac.ir

Ahmad Shirzadi is a faculty member in the Department of Mathematics at Persian Gulf University. His research concentrates on computational mathematics and machine learning, with a particular emphasis on feature engineering, pattern recognition, and deep learning architectures for visual data analysis. He has collaborated on projects involving support vector machines (SVM), CNNs, and descriptor-based methods for efficient image representation.
Email: shirzadi@pgu.ac.ir