

## Article

# Human Emotion Recognition Based on Spatio-Temporal Facial Features Using HOG-HOF and VGG-LSTM

Hajar Chouhayebi <sup>1,\*</sup> , Mohamed Adnane Mahraz <sup>1,\*</sup>, Jamal Riffi <sup>1</sup>, Hamid Tairi <sup>1</sup> and Nawal Alioua <sup>2</sup>

<sup>1</sup> Laboratory of Computer Science, Signals, Automation and Cognitivism (LISAC), Department of Computer Science, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco; riffi.jamal@gmail.com (J.R.); htairi@yahoo.fr (H.T.)

<sup>2</sup> LMC, Polydisciplinary Faculty, Department of Mathematics and Computer Science, Cadi Ayyad University, Safi 46000, Morocco; nawal.alioua@uca.ac.ma

\* Correspondence: hajar.chouhayebi@usmba.ac.ma (H.C.); adnane\_1@yahoo.fr (M.A.M.); Tel.: +212-659-315-232 (H.C.)

**Abstract:** Human emotion recognition is crucial in various technological domains, reflecting our growing reliance on technology. Facial expressions play a vital role in conveying and preserving human emotions. While deep learning has been successful in recognizing emotions in video sequences, it struggles to effectively model spatio-temporal interactions and identify salient features, limiting its accuracy. This research paper proposed an innovative algorithm for facial expression recognition which combined a deep learning algorithm and dynamic texture methods. In the initial phase of this study, facial features were extracted using the Visual-Geometry-Group (VGG19) model and input into Long-Short-Term-Memory (LSTM) cells to capture spatio-temporal information. Additionally, the HOG-HOF descriptor was utilized to extract dynamic features from video sequences, capturing changes in facial appearance over time. Combining these models using the Multimodal-Compact-Bilinear (MCB) model resulted in an effective descriptor vector. This vector was then classified using a Support Vector Machine (SVM) classifier, chosen for its simpler interpretability compared to deep learning models. This choice facilitates better understanding of the decision-making process behind emotion classification. In the experimental phase, the fusion method outperformed existing state-of-the-art methods on the eNTERFACE05 database, with an improvement margin of approximately 1%. In summary, the proposed approach exhibited superior accuracy and robust detection capabilities.

**Keywords:** human emotion recognition; HOG-HOF; facial expression recognition; deep learning; visual geometry group; long short term memory; support vector machine; histograms of oriented gradients; histogram of optical flow



**Citation:** Chouhayebi, H.; Mahraz, M.A.; Riffi, J.; Tairi, H.; Alioua, N. Human Emotion Recognition Based on Spatio-Temporal Facial Features Using HOG-HOF and VGG-LSTM. *Computers* **2024**, *13*, 101. <https://doi.org/10.3390/computers13040101>

Academic Editor: Lucia Maddalena

Received: 6 March 2024

Revised: 30 March 2024

Accepted: 3 April 2024

Published: 16 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As is commonly acknowledged, human communication heavily relies on speech, complemented by body language to emphasize specific aspects and convey emotions [1]. In human contact, facial expressions are among the strongest, most natural, and simplest ways to convey intentions and feelings. While emotions can also be conveyed through voice, text, and other mediums, the face remains the most prominent [2]. In 1974, Mehrabian [3] demonstrated that around 50% of individuals communicate information through facial expressions in daily interactions, with approximately 40% relying on both voice and facial cues, while the remaining 10% express themselves through words. This preference is attributed to the face containing numerous effective emotional features, offering advantages in data collection [3].

In the last few years, the progression of computer vision, machine learning, and facial expression recognition has evolved into a captivating and challenging field of study. This technology holds significant importance and is applied across various domains such as driver safety, medicine, and human-computer interaction. In the realm of human-computer

interaction, particularly with Intelligent Personal Assistants (IPAs) communicating through natural language, effective communication can be enhanced by integrating facial expression recognition. In medical contexts, individuals facing physical or psychological challenges may encounter difficulties expressing emotions conventionally, making emotion recognition technology a valuable solution for effective communication [4]. In terms of safety, facial expression recognition plays a crucial role in identifying the emotional state of a driver. Through non-invasive monitoring, it enables prompt and effective evaluation to determine if a driver might engage in risky behavior, thus helping to avert potential hazards. Additionally, it assists in the continuous monitoring and prediction of fatigue levels and attention, contributing to accident prevention [5].

Emotions are dynamic and evolve over time, necessitating models that can effectively capture temporal dependencies in facial expressions. Many current methods may lack the sophistication to model these temporal dynamics accurately. In recent years, amidst advancements in human research and the fast evolution of related areas, facial expression recognition algorithm remains a focal point of investigation. However, challenges persist in emotion recognition algorithms. First, many feature extraction methods still resemble traditional manual approaches, proving ineffective in extracting features. Second, overcoming the challenge of effectively reducing the residual generalization error within emotion recognition algorithms significantly impacts the accuracy and robustness of the system. On the other hand, deep learning architectures offer a promising solution by enabling the extraction of significantly more particular features compared to typical machine learning algorithms. This capability leads to the extraction of more robust features, thereby enhancing the clarity of facial expression identification [6]. Emotion recognition algorithms can be broadly classified into two categories: the typical approach of emotion recognition and deep learning-based approaches [7]. The performances of these two primary types are examined in Section 2.

The proposed fusion approach focuses on evaluating facial information to detect seven emotional categories: disgust, sadness, anger, happiness, scared, neutrality, and surprise. Initially, Residual Network (ResNet) [8] was employed for face detection, followed by the utilization of a combination of two architectures, VGG19 [9] and LSTM [10], to obtain the corresponding vector descriptor. In the second phase, we integrate appearance Histograms of Oriented Gradients (HOG) [11] and motion descriptors Histogram of Optical Flow (HOF) [12] for parallel temporal segmentation and recognition. The introduced HOG-HOF descriptors serve to recognize variations in facial appearance. The main contribution of this paper lies in the use of a spatio-temporal descriptor vector HOG-HOF, employing manual feature extraction concatenated with a descriptor vector from a pre-trained VGG-19 model and the LSTM. The second contribution is the utilization of a general descriptor vector based on the interaction of the two elements from the two descriptor vectors via the MCB method. This unique combination results in the development of a robust system capable of achieving high-accuracy facial expression recognition.

The contribution of the methodology lies in its integration of traditional algorithms with modern techniques, mitigating the weaknesses of each approach when used in isolation. Traditional algorithms may struggle to capture the complexity of facial expressions or emotion recognition due to their reliance on handcrafted features. Conversely, deep learning methods might be sensitive to variations in facial expressions caused by factors such as lighting conditions or facial occlusions. By combining these approaches, the methodology seeks to leverage the strengths of each while offsetting their respective limitations, ultimately enhancing the overall effectiveness of emotion recognition systems.

Validation of this system was carried out using the eNTERFACE05 dataset [13], demonstrating that our system surpasses state-of-the-art approaches in automatic facial expression recognition in video sequences.

The subsequent sections of the article are structured as follows: Section 2 engages in a discussion on recent state-of-the-art approaches. Section 3 presents the details of the proposed approach. The performance of the proposed approach on a public dataset is

analyzed in Section 4. Finally, conclusions of this research approach and future perspectives are provided in Section 5.

## 2. Related Works

In this section, we present a brief literature assessment on human emotion recognition to inform our model choice. We examine three methods such as dynamic texture-based methods, deep learning methods, and transfer learning methods.

### 2.1. Dynamic Texture-Based Methods

Facial Expression Recognition (FER) systems can be classified into two primary types according to feature representations: static image and dynamic sequence. In static-based approaches [14,15], the feature representation is exclusively derived from the spatial information within the current single image. On the other hand, dynamic-based approaches [16,17] consider the temporal relationships among contiguous frames in the input emotion sequence. The commonly used local spatio-temporal descriptors are inspired by SIFT [18]: each local video volume is partitioned into blocks, and for each block, outputs are aggregated (optical flow or oriented gradients). The final descriptor is a concatenation of the aggregated outputs from multiple adjacent blocks.

Three-dimensional gradient solutions were produced by monitoring directed gradients in the temporal dimension, as reported by Scovanner et al. [19] and Kläser et al. [20]. The suggestion to aggregate HOG and HOF was made by both Dalal et al. [21] and Laptev et al. [22]. In addition, the calculation of variations in optical flow, or Motion Boundary Histogram (MBH), was suggested by [21]. Gradient states in three dimensions were produced by measuring directed gradients in the temporal dimension, as first demonstrated by Kläser et al. [20] and Scovanner et al. [19].

To improve emotion recognition, multimodal algorithms, Corneanu et al. [23] have integrated additional modalities, such as audio and physiological methods, on top of these two vision-based approaches. Although facial expression recognition from visible images of faces alone can produce interesting results, adding other modalities to a comprehensive structure can provide complementary information and enhance robustness.

In [24], the focus was on elaborating facial micro-expression identification through a fusion-based deep learning approach and improved optical flow. Meanwhile, Sadeghi and Raie [25] investigated the extraction of features for facial expression recognition influenced by human vision. The method comprised convolution using Gabor filters, cropping the human face from the source image, and spatial normalization applied to the reduced image. After segmenting the resulting coded Gabor filter convolution matrix into several blocks, the feature vectors were created by compiling the histograms of each block separately. In [26], the focus was on emotion recognition through facial expressions. The researchers collected the region of interest (ROIs) from various facial components, including the face, mouth, left chin, right chin, forehead, and eyes. They also identified the eye's center of mass. Utilizing multiple Local Binary Patterns (LBPs), features were extracted from the face, eyes, and nose. The study concluded by combining MLBP-SIFT features, which were then input into a SVM classification system.

Lakshmi et al. [27] employed a modified HOG and LBP. The Viola–Jones face detection method was utilized for detecting the face region. Subsequently, a Butterworth high-pass filter enhanced the detected region to identify the eyes, nose, and mouth regions. Then, the suggested modified HOG and LBP feature descriptors were used to extract features from these identified regions.

### 2.2. Deep Learning Methods

While traditional methods of facial recognition that rely on manually generated features have made tremendous progress, researchers have increasingly turned to the deep learning approach over the past decade due to its high capacity for automatic recognition.

In this context, we highlight recent studies in FER that introduce deep learning methods aimed at achieving enhanced detection.

Cai et al. [28] introduce a novel architecture, combining convolutional neural network (CNN) with Sparse Batch Normalization (SBP). This network employs two successive convolution layers at the outset, followed by max pooling and SBP. To address the issue of overfitting, dropout is applied in the middle of three fully connected layers. In their study, Agrawal and Mittal [29] investigate the impact of varying CNN settings for rate recognition on the FER2013 database. Initially, each image has a defined size of 64 by 64 pixels. They experiment with different filter sizes and numbers, as well as the optimizer type selected (SGD, Adam) within a basic CNN architecture. This CNN comprises two successive convolution layers, with the second layer serving as max pooling, followed by a softmax function for classification. To tackle the occlusion of the face challenges, Li et al. [30] propose a novel CNN method. Initially, the data are fed into the VGGNet network, followed by the application of a CNN technique incorporating an attention mechanism, referred to as ACNN.

Variations in facial expressions during emotional states are examined by Kim et al. [31], who also suggest a spatio-temporal conception that blends CNN and LSTM. The CNN first learns the spatial features of the emotional state's facial expressions across all frames. Then, it applies the LSTM to keep the whole sequence of these spatial properties. Furthermore, a novel architecture known as Spatio-Temporal Convolutional with Nested LSTM is introduced by Yu et al. [32]. Three neural network sub-networks form the base of this architecture: temporal LSTM, which keeps the temporal dynamic; convolutional LSTM, which models multiple-level features; and 3DCNN, which extracts spatio-temporal data. Yolcu et al.'s [32] suggestion was to use three CNNs with a similar structure to identify the key facial features. Every CNN is made to identify a particular facial feature, like the eye, eyebrow, or mouth. The images go through a resizing phase and key-point recognition for faces before being sent into the CNNs. In order to detect facial expressions, a second type of CNN is created by combining the obtained famous face with the initial image.

Chouhayebi et al. [33] introduce a recognition of facial expression approach that amalgamates two approaches. Firstly, they propose a descriptor named Histogram of Oriented Gradients from Three Orthogonal Planes (HOG-TOP) to extract dynamic textures from video sequences, effectively characterizing changes in facial appearance. Additionally, a deep learning-based model is employed, using a transfer learning approach that has been already trained on ImageNet, the VGG19, to obtain visual details from a video. They leverage LSTM to model the temporal flow of visual features, and to integrate these spatio-temporal features derived from both visual models, they apply the MCB method, resulting in a more robust vector descriptor.

Hu et al. [34] propose a method for emotion recognition that integrates visual input with a bidirectional recurrent unit within an attentional convolutional neural network. Initially, they pre-trained log-mel spectrograms using a ResNet-based neural network to extract speech features. Then, they combined these voice features with static facial appearance features obtained from a CNN. Subsequently, the combined vocal and facial appearance features were further merged with facial geometric features to create hybrid features.

Priyasad et al. [35] introduce a deep learning approach aimed at leveraging and combining textual and acoustic data for facial expression classification. They utilized a SincNet layer to extract the audio features. In terms of text processing, they used Bidirectional Recurrent Neural Network and DCNN. Before fusing, they applied self-attention to both feature vectors that were obtained from the two networks, which allowed identification of informative regions from each feature vector.

### 2.3. Transfer Learning Methods

Chowdary et al. [36] introduce a facial emotion recognition system employing transfer learning methodologies. The approach utilizes pre-trained convolutional neural networks, including MobileNet, Inception V3, ResNet50, and VGG19, all initially trained on the

ImageNet database, for the task of facial emotion recognition. Li et al. [37] introduced an enhanced facial emotion recognition system. They employ a multi-layer neural network trained through transfer learning, utilizing the ResNet-101 deep network for effective feature extraction. The trained deep neural network adeptly abstracts data features layer by layer, ultimately extracting the necessary features for task completion. The integration of these features occurs through the full connection layer, succeeded by the utilization of a classifier to accomplish the final recognition task.

Priyasad et al. [38] introduced an automated emotion recognition system utilizing both audio and visual modalities. Frame-level facial features are captured using VGG19 models, and their temporal distribution at a segment level is captured by LSTM. Simultaneously, auditory features are extracted from Mel Frequency Cepstral Coefficients (MFCC) using a separate VGG19 model. The spatial-temporal features extracted from visual and audio methods are combined through a neural network based on attention.

### Result and Limitation of Existence Methods

Deep learning methods, particularly convolutional neural networks (CNNs), have achieved remarkable success in facial expression recognition. They have demonstrated superior accuracy in extracting discriminative features from facial images, leading to improved recognition performance. Despite their robustness to certain variations, deep learning models may struggle to generalize well to unseen conditions, such as extreme poses, low-resolution images, or non-frontal faces. This limitation can affect their performance in real-world applications.

Analyzing dynamic textures involves complex algorithms for motion detection, tracking, and feature extraction. These processes can be computationally intensive and may require sophisticated techniques such as optical flow estimation or spatio-temporal filtering, leading to increased computational overhead.

Overcoming these limitations highlights ongoing challenges in the field that demand special attention. Here are some possible reasons why the existing approaches may not be robust enough:

**Limited Dataset Size:** The datasets used in these studies may not be large or diverse enough to capture the full range of emotional expressions encountered in real-world scenarios.

**Feature Representation:** The features extracted from facial expressions or audio signals may not effectively capture the underlying emotional cues, leading to suboptimal performance.

Addressing these challenges may require exploring novel methodologies, collecting more extensive and diverse datasets, refining feature extraction techniques, and improving model architectures. Additionally, concatenating these two examined models is the best solution to enhance model generalization capabilities, and is a crucial step toward advancing the state-of-the-art in facial expression recognition.

### 3. The Proposed Approach

Our proposed approach consists of three basic parts, as shown in Figure 1: a visual feature extractor using deep learning techniques, a spatio-temporal feature extraction using the combined HOG-HOF descriptor (integrating both HOG and HOF descriptors), and a fusion algorithm.

The VGG19 model is chosen for its effectiveness in feature extraction from images. Its deep architecture allows it to capture hierarchical representations of visual features, which are crucial for recognizing facial expressions.

LSTM networks are well suited for modeling sequential data, making them suitable for capturing temporal dependencies in video sequences. In the context of emotion recognition, LSTM can learn the dynamics of facial expressions over time.

HOG-HOF descriptors are effective in capturing both spatial and temporal information in video data. They provide a compact representation of motion patterns and texture variations in facial expressions, which can complement the features extracted by deep learning models.

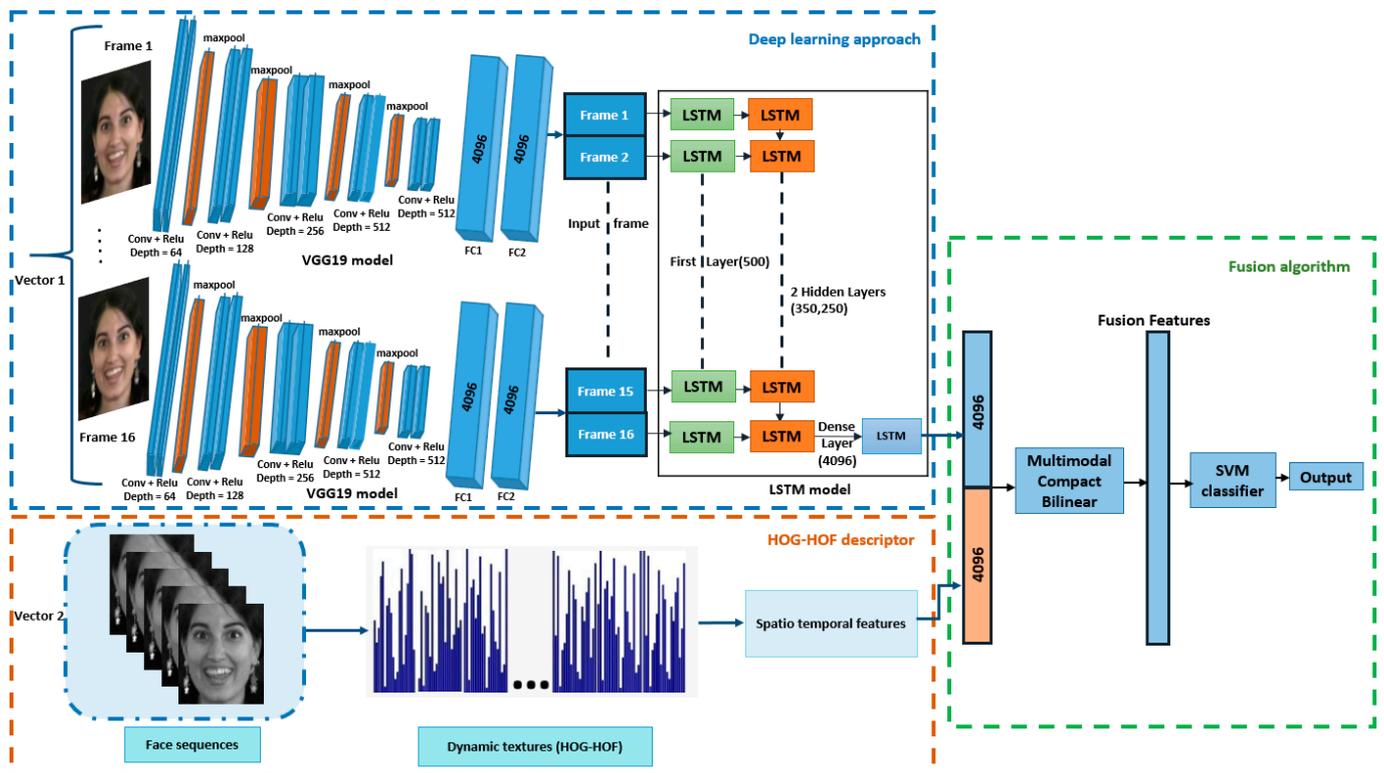


Figure 1. Our architecture.

### 3.1. Deep Learning Part

We employ the same approach as in [33] for the deep learning section. They employed the ResNet [8] structure for face detection, the VGG19 framework [9] for face analysis and the LSTM model [10] to produce the most expressive vector, which had a length of 4096.

According to Chouhayebi et al. [33], 2 factors influenced the choice of the 16 most emotional frames in the video: first, a review of every video in the database showed that, out of a total of 16 frames, the most emotive frames were found between frame 26 and 42. Second, a comparison of the distinction between emotion and feeling revealed that emotions are quick movements, whereas feelings can linger for a long time.

In light of these two considerations, we decided to decrease the computing duration of each video by enhancing the expressiveness of the video. We opted for 16 frames showcasing the highest expressiveness from each video in the dataset. This selection spanned frames 26 to 42, a priori. First, the frontal region of the human face was recognized using the ResNet model [8], which also recorded the expression in every video frame. The images that ResNet had detected were then collected for examination by the VGG19 [9], which had been previously trained using ImageNet [39]. Next, we took feature representations with a size of 4096 from the last layer of this fully connected architecture (FC7) and stacked them to create a feature set ( $16 \times 4096$ ). After that, a many-To-One LSTM [10] is used to process the VGG19 features, producing only a 4096 descriptor vector. Next, the feature vector obtained from the HOG-HOF descriptor [40] is combined with this vector.

#### 3.1.1. ResNet

ResNet is a structure for deep learning that incorporates residual learning. The key innovation of ResNet is the use of residual blocks, which contain shortcut connections allowing for the direct flow of information from one layer to another. These shortcuts help to address the vanishing gradient problem, enabling the training of very deep neural networks.

In the context of facial expression recognition, ResNet architecture is often employed to identify and extract features from facial images, contributing to the overall recognition performance. The model's ability to capture intricate patterns and hierarchical representa-

tions makes it suitable for complex tasks like facial expression analysis. Figure 2 shows the basic block of a ResNet model.

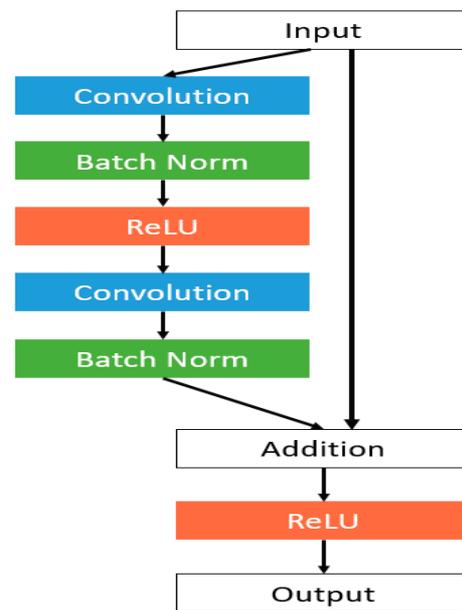


Figure 2. ResNet basic block.

### 3.1.2. VGG19

VGG19 is a deep convolutional neural network architecture that has been widely used for image classification tasks, including facial expression recognition. In our methodology, we utilized the FC7 layer for the vector descriptor, comprising 4096-D elements. This descriptor serves as the input for our LSTM model. Figure 3 shows the architecture of the VGG19 model.

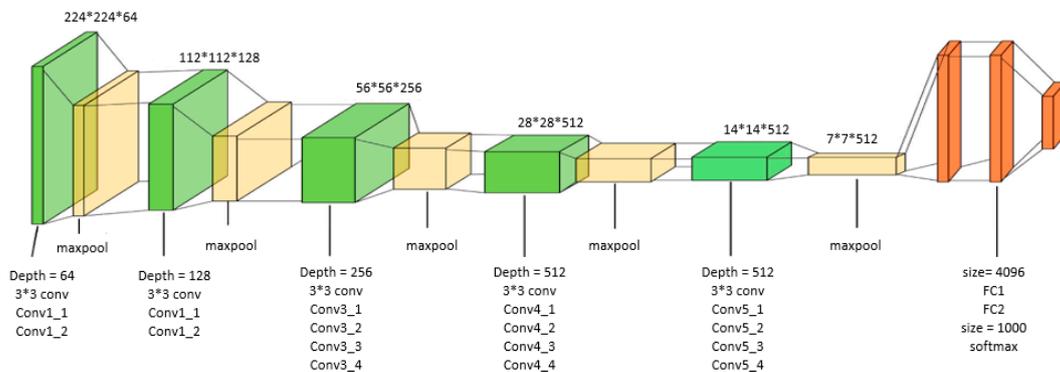


Figure 3. VGG-19 network architecture.

### 3.1.3. LSTM

The details for the specific LSTM architecture include its ability to address the vanishing gradient problem in traditional RNNs, allowing the capture of long-range dependencies in sequential data. LSTM has memory cells with input, output, and forget gates, enabling effective information retention and retrieval. The input gate controls the flow of new information, the forget gate manages the removal of unnecessary information from the cell, and the output gate regulates the information output. The equations governing the behavior of an LSTM cell are as follows:

Input Gate

$$i_t = \sigma(W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + b_i) \quad (1)$$

where:

- $x_t$  is the input at time step  $t$
- $h_{t-1}$  is the output of the previous time step.
- $W_{ix}$ ,  $W_{ih}$ , and  $b_i$  are the weight matrix and bias vector for the input gate, respectively.
- $\sigma$  is the sigmoid activation function.

Forget Gate

$$f_t = \sigma(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + b_f) \quad (2)$$

where:

- $W_{fx}$ ,  $W_{fh}$  and  $b_f$  are the weight matrix and bias vector for the forget gate, respectively.

Cell State Update

$$h_t = \tanh(W_{cx} \cdot x_t + W_{ch} \cdot h_{t-1} + b_c) \quad (3)$$

where:

- $W_{cx}$ ,  $W_{ch}$  and  $b_c$  are the weight matrix and bias vector for updating the cell state.

Cell State Update

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4)$$

where:

- $C_t$  is the cell state from the previous time step.

Output Gate

$$o_t = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \quad (5)$$

where:

- $W_{ox}$ ,  $W_{oh}$  and  $b_o$  are the weight matrix and bias vector for updating the cell state.

Hidden State Update

$$h_t = O_t \cdot \tanh(C_t) \quad (6)$$

The many-to-one architecture is often used in tasks like video classification, where the goal is to classify the entire video based on the information contained in its frames. Each frame is considered as a time step in the sequence, and the LSTM learns to capture temporal dependencies and patterns across the frames. Figure 4 shows the architecture of Many-to-One LSTM model.

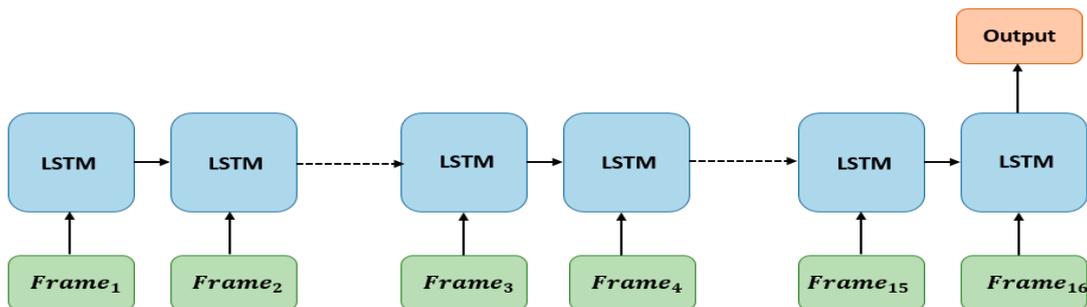
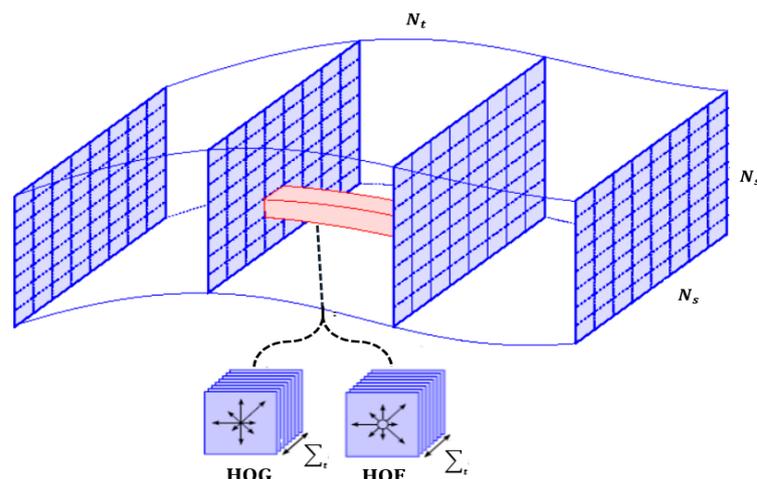


Figure 4. Many-To-One LSTM model.

In the context of facial expression recognition, LSTM can effectively model the temporal dependencies between frames in a video sequence, allowing for nuanced understanding of dynamic changes in facial expressions over time. This makes LSTM a suitable choice for capturing the temporal aspects crucial for emotion recognition in sequential data like videos.

### 3.2. HOG-HOF Part

In this paper, we combine HOF and HOG to capture both spatial and temporal information, respectively. Figure 5 illustrates the trajectory description of HOG and HOF descriptors.



**Figure 5.** Trajectory description of HOG and HOF descriptors.

The 72-bin descriptor known as the Histogram of Oriented Gradients [13] describes the local appearance. The authors suggest defining a pattern of cells  $n_x \times n_y \times n_t$  in the immediate space–time area (the default options:  $3 \times 3 \times 2$ ) and computing the 4-bin histogram of directed gradients for each cell in the grid. The 90-bin descriptor, known as the Histogram of Optical Flow [13], describes the local motion. The authors suggest defining a pattern of cells  $n_x \times n_y \times n_t$  (with the default values of  $3 \times 3 \times 2$ ) around the covering space–time region and computing the 5-bin histogram of optical flow for each cell in the grid. The HOF and HOG descriptors are combined to create the 162-bin HOG-HOF descriptor.

The local spatio-temporal features aim at representing a video by detecting and describing small video volumes. Most approaches are extensions of successful methods for images. For instance, for selecting regions that are robust to capturing conditions, Laptev extends the Harris [41] cornerness criterion to videos. In a similar spirit, the Hessian detection introduced by Willems et al. [42] is an extension of the well-known blob detection in images. For descriptors, Kläser et al. [20] extend the successful Histogram of Oriented Gradient [11] to video volumes by quantizing the gradient angles in 3D. It is also common to extract HOF [22] or MBH [43], depending on the optical flow’s gradient.

### 3.2.1. HOF Descriptor

In this section we present the spatio-temporal feature descriptor, called HOF, which captures the moving patterns by choosing regions in the video. These regions are the most important points, Dlib [44], on the face. The process of extracting the HOF descriptor involves the following steps:

1. **Optical Flow Computation:** Compute the optical flow between consecutive frames of a video sequence. Optical flow represents the motion of objects by analyzing the displacement of pixels between frames.
2. **Dense Grids:** Define dense grids over the optical flow field, covering the entire image or region of interest. In our case, we use facial landmarks to track specific points on the face. For example, facial landmarks could include points on the eyes, nose, and mouth.
3. **Histograms of Orientation:** For each grid cell or facial landmark point, calculate histograms of the orientation of optical flow. Divide the optical flow orientations into bins and accumulate the counts of optical flow orientations within each bin.
4. **Spatial Blocks:** Organize the image or region of interest into spatial blocks, where each block contains multiple grid cells or facial landmark points. This step introduces spatial relationships into the descriptor.

5. Histograms within Blocks: Compute histograms of optical flow orientations within each spatial block. Similar to the grid cell histograms, quantize orientations into bins and accumulate counts.
6. Concatenation: Concatenate the histograms from all spatial blocks merging into just one feature vector. This feature vector represents the overall optical flow-based representation of facial expression dynamics.
7. Normalization: Normalize the concatenated feature vector to enhance robustness against variations in lighting and contrast and facial pose.

The HOF descriptor is particularly useful in capturing motion patterns in video sequences and leverages the temporal information captured by optical flow to discern subtle changes in facial expressions, making it suitable for dynamic facial expression analysis in video sequences.

This histogram is based at the optical flow:

The optical flow approaches aim to estimate the motion between the two images at all positions (in our case, the two successive frames of video) at times  $t$  and  $t + \Delta t$ . The optical flow equation is usually written as a single, two-variable equation. To aid in the flow estimating process, all optical flow approaches include new criteria. Also, it characterizes the apparent velocities of brightness patterns in an image, providing crucial insights into the spatial organization and its rate of change [45]. Selected for its ability to reveal facial expressions through movement direction and amplitude, optical flow serves as the scene description in our study.

The Horn–Schunck (HS) optical method, as initially proposed P. Horn and G. Schunck [45], is employed, integrating a global smoothness constraint to compute optical flow. The HS method combines a data term assuming constancy in some image property with a spatial term modeling the expected flow variation across the image [46].

Farneback’s method [47], named after Gunnar Farneback, is an algorithm used for dense optical flow estimation. It is commonly employed in computer vision tasks to calculate motion between frames in a video sequence. This method computes both the magnitude and direction of motion at each pixel in the image, providing a dense flow field. The algorithm involves polynomial expansion and is known for its efficiency in capturing complex motion patterns in various computer vision applications [47].

In our work, the two-dimensional sequences of images’ [12] optical flow are generated as a global energy functional

$$E = \int \int \left[ (I_x u + I_y v + I_t)^2 + \alpha (\|\nabla u\|^2 + \|\nabla v\|^2) \right] dx dy \quad (7)$$

In Equation (1):  $I_x$ ,  $I_y$ , and  $I_t$  represent the derivatives of frame intensity values in the  $y$ ,  $x$ , and  $t$  dimensions, correspondingly.  $u$  and  $v$  represent the optical flow’s components, and  $\alpha$  acts as a regularization constant.

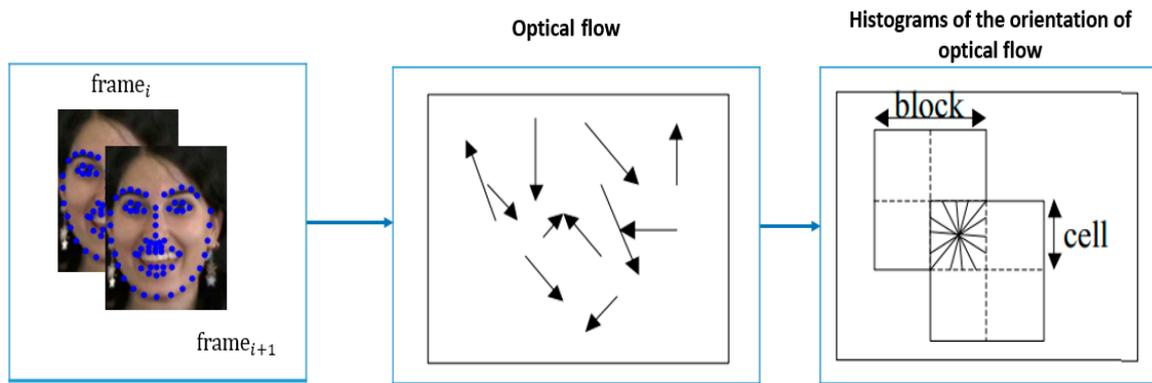
Given its dependence on neighboring values, the solution requires iteration when the neighboring pixels are updated. Within an iterative structure, The expression for the optical flow is [12]:

$$u^{k+1} = u^k - \frac{I_x (I_x u^k + I_y v^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \quad (8)$$

$$v^{k+1} = v^k - \frac{I_y (I_x u^k + I_y v^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \quad (9)$$

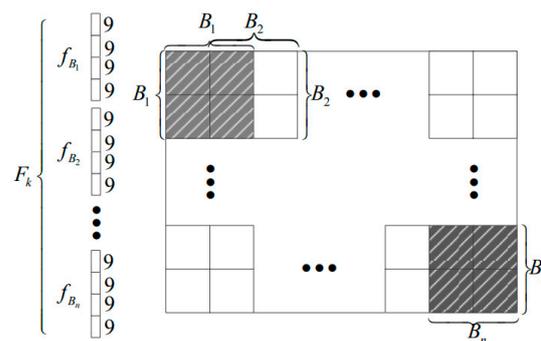
In this formulation,  $k$  is the iteration of the algorithm. For this study, a single time step was considered, allowing computations based on just two adjacent images. In this paper, we suggest HOF. Unlike the descriptor in [21], which considers differential optical flow, we calculate our HOF in dense optical flow grid. The HOF descriptor is generated over dense and overlapping grids of spatial blocks. Optical flow orientation features are extracted at a fixed resolution and consolidated into a high-dimensional feature vector [48].

The HOF descriptor in Figure 6 illustrates a  $2 \times 2$  cell configuration, with our work adopting a rectangular cell. Within this configuration,  $u$  and  $v$  fields, or horizontal and vertical optical flows, contribute to  $n$  oriented bins in the  $0^\circ$ – $360^\circ$  range. Each pixel computes a weighted vote for an edge orientation histogram channel, considering the optical flow element's orientation that is centered on it. The orientation of the histogram is determined by stacking these votes into orientation bins across small spatial regions, with the optical flow magnitudes of the cell serving as voting weights.



**Figure 6.** Compute the optical flow orientation histograms.

Each block's HOF feature is computed, and the resulting unified vector referred to as  $F_k$  for frame  $k$  is combined. The HOF feature is calculated within each block and then concatenated into a unified vector represented as  $F_k$  for frame  $k$ . As depicted in Figure 7, each frame's HOF creates a vector with the dimensions  $n\text{-blocks} \times n\text{-bins}$ . For the  $0^\circ$ – $360^\circ$  range, the orientation bins are divided into nine equal parts. HOF computation involves an overlapping fraction between two neighboring blocks. In Figure 7, the overlapping fraction of two adjacent blocks is 50 percent. A block consists of  $bh \times bw$  cells, where  $bh$  and  $bw$  represent the number of cells in the  $y$  and  $x$  directions, respectively. Note that smaller cell or block sizes may increase the computation time for the HOF feature.

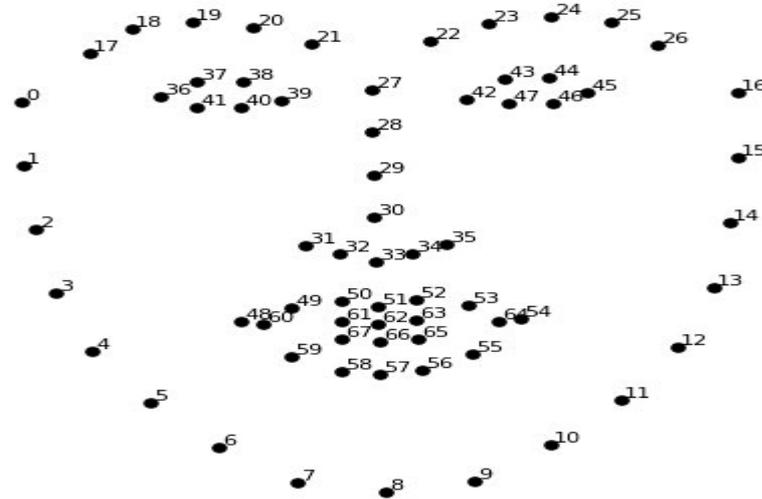


**Figure 7.** Histograms showing the optical flow feature's orientation for the frame  $k$ .

In this paper, the HOF feature vector has a length of 36 and we computed the optical flow histograms for the most prominent Dlib [44] points across 16 frames (see Figure 8). The resulting descriptor vector has a size of 3072, achieved by concatenating the histograms from all the selected points.

Each histogram has a length of 12, so for each frame, we have  $12 \times 16 = 192$ , where 16 is the numbers of selected points. And, the final length of the vector is  $192 \times 16 = 3072$ , where 16 is the number of frames. Our dense trajectory implementation utilizes the optical flow method from [47], striking a favorable balance between speed and accuracy [43]. Extracting the HOF descriptor for facial expression recognition involves computing the optical flow in dense grids, using Farneback's method, and generating histograms of flow orientations

over spatial blocks. The orientation features are then aggregated into a high-dimensional feature vector. The process is repeated for consecutive frames, resulting in a descriptor vector for each frame. These vectors are concatenated to form a monolithic vector for further analysis.



**Figure 8.** The 68 points recognized by dlib library.

### 3.2.2. HOG Descriptor

The initial descriptor employed in the feature extraction block is HOG (see Figure 9). It can be considered one of the commonly used appearance-based descriptors. Dalal and Triggs [11] introduced it, and it has been applied in the context of pedestrian detection. The generation of this descriptor involves the following steps:

- 1 Gradient Computation: Compute the image's gradient to identify edges and intensity changes. This is typically achieved using convolution with specific filters like Sobel filters.

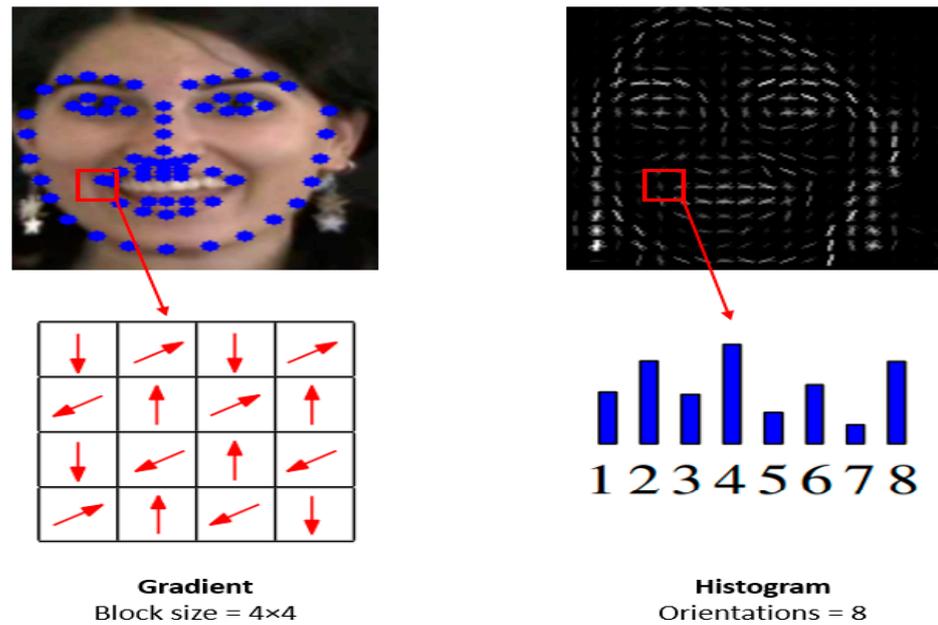
$$G_x = \frac{\partial I}{\partial x}, G_y = \frac{\partial I}{\partial y} \quad (10)$$

- 2 Gradient Magnitude and Orientation: Calculate the magnitude ( $M$ ) and orientation of the gradient at each pixel ( $\theta$ ). The magnitude represents the strength of the gradient, while the orientation indicates the direction of the gradient.

$$M = \sqrt{G_x^2 + G_y^2} \quad (11)$$

$$\theta = \arctan2(G_y, G_x) \quad (12)$$

3. Cell Division: Divide the image into cells, which are small regions that will be used to accumulate gradient information. Typically, cells are square and can vary in size.
4. Histograms within Cells: Create a gradient orientations histogram for each cell. The orientations are quantized into bins, and the histogram captures the distribution of gradient orientations within the cell.
5. Block Normalization: Organize cells into blocks, which are larger regions consisting of multiple cells. Blocks typically overlap, and within each block, normalize the histograms to account for variations in lighting and contrast.
6. Concatenation: To create a single feature vector, concatenate each block's normalized histogram.
7. Final Normalization: Normalize the concatenated feature vector to ensure robustness to varying illumination and contrast conditions.



**Figure 9.** The process of extracting the HOG descriptor.

These steps collectively result in the HOG descriptor, a feature vector that captures important information about the local gradient patterns in the image. This descriptor is commonly used for facial expression recognition and other computer vision tasks.

In this section, we extract the most important points, Dlib [44], on the face and then we apply the HOG descriptor to each facial chosen point. The resulting descriptor vector has a size of 1024, achieved by concatenating the histograms from all the selected points. Each histogram has a length of 8, so: For each frame we have  $8 \times 8 = 64$ , where 8 is the number of selected points. The final length of the vector is  $64 \times 16 = 1024$ , where 16 is the number of frames.

### 3.2.3. Combination of HOG and HOF

We combine HOG and HOF features to provide a more comprehensive representation of facial expressions by capturing both spatial and temporal aspects and to create a hybrid feature representation as shown in Figure 10. We concatenate the HOG and HOF vectors to obtain a final vector descriptor with length of 4096. Here is a breakdown of the calculation of the HOG-HOF algorithm:

Input:

- Video sequence  $V$  which contains  $N$  frames with the same height  $H$  and width  $W$ .
- Each frame contains  $D$  numbers of Dlib points, with the height  $y$  and width  $x$ .

HOG Descriptor:

- Calculate the gradient magnitude  $G_x$  and  $G_y$  and the orientation  $\theta$  at each  $D$  in the frame.
- For each point  $D$ , create a histogram of gradient orientations vector  $V_{HOG_D}$ . Typically, each histogram has nine bins covering 0 to 180 degrees.
- Concatenate all normalized histograms  $V_{HOG_D}$  to form the HOG descriptor for the frame:  $V_{HOG_N}$ .

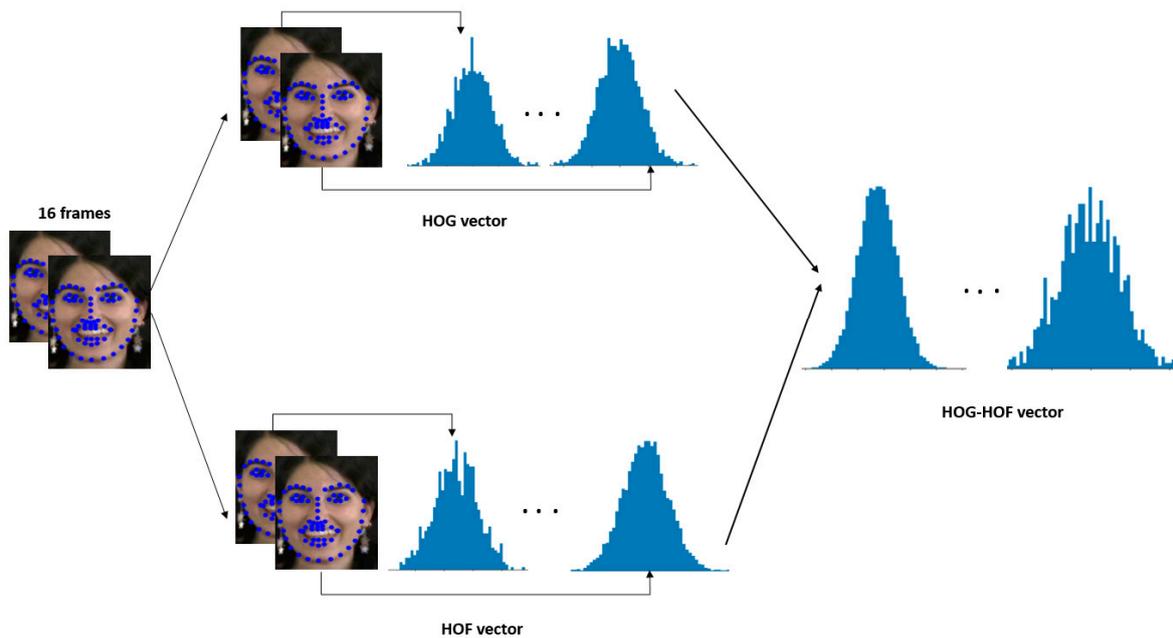
HOF Descriptor:

- Calculate the optical flow HS between two consecutive frames  $N$  and  $N + 1$  in a video sequence.
- For each frame that contains  $D$  numbers of Dlib points, create a histogram of optical flow directions vector  $V_{HOF_D}$ .

- Concatenate all normalized histograms  $V_{HOF_D}$  to form the HOF descriptor for the frame:  $V_{HOF_N}$ .  
Output:
- Concatenate the HOG descriptor vector computed for the all frames  $V_{HOG_{allframes}}$  with the HOF descriptor vector computed from the optical flow between all consecutive frames  $V_{HOF_{allframes}}$  to create the HOG-HOF vector denoted as  $V_{HOG-HOF}$ :

$$V_{HOG-HOF} = V_{HOG_{allframes}} + V_{HOF_{allframes}} \quad (13)$$

Length HOF vector is 3072, the HOG length vector is 1024, and the HOG-HOF length vector is  $3072 + 1024 = 4096$ .



**Figure 10.** HOG-HOF Vector: the combination of HOG and HOF vectors.

In summary, HOG and HOF can be complementary in capturing spatial and temporal information, respectively, leading to a more robust facial expression recognition system, especially when dealing with video sequences. The combination of these features allows the model to understand both the static facial features and the dynamic changes in expressions over time.

### 3.3. Fusion Part

In this part, we aim to combine two descriptors HOG and HOF to obtain good results. We introduce two fusion methods that Chouhayebi et al. [33] has used. The initial method involves a simple concatenation of the HOG-HOF vector and the LSTM vector, resulting in a final vector with a length of 8192.

#### MCB Fusion

The second approach employs the multimodal compact bilinear pooling (MCB) algorithm [49] to concatenate the two vectors.

The MCB module is a neural network component that is used to fuse information from multiple modalities (e.g., vision and language) in a compact and effective way. This module is particularly useful for multimodal tasks where information from different sources needs to be combined for improved performance. The MCB module operates on the principle of compact bilinear pooling. Bilinear pooling is a method used to capture second-order statistics of feature vectors, and it has proven effective in various computer vision tasks.

However, directly applying bilinear pooling to combine features from multiple modalities can lead to a significant increase in the number of parameters, making it computationally expensive.

The MCB module addresses this issue by introducing a compact bilinear pooling strategy. It efficiently computes the outer product of the input feature vectors using tensor factorization techniques, reducing the computational cost while maintaining the expressive power of bilinear pooling. This allows the model to capture rich interactions between different modalities in a more efficient manner. The key equation for MCB is:

$$z = W_1 \cdot \text{Vect}(X_1) \odot W_2 \cdot \text{Vect}(X_2) \quad (14)$$

where:

- $X_1$  and  $X_2$  are the feature matrices from different sources,
- $\text{Vect}()$  denotes vectorization of a matrix,
- $W_1$  and  $W_2$  are projection matrices,
- $\odot$  represents the element-wise product.

The MCB module has found applications in tasks such as emotion recognition and other multimodal learning scenarios where the fusion of information from different modalities is crucial for accurate predictions. The structure of the MCB module involves the LSTM and the HOG-HOF vectors as its input. Utilizing the projection function for Count Sketch for dimensionality reduction, the two different vectors are combined in the Fourier space, resulting in the greater-order vector. Here is an outline of the algorithm for the multimodal compact bilinear pooling operation:

Input:

- Input features from different modalities: let  $x_1$  and  $x_2$  be the input feature vectors from two different modalities.

Random Projection:

- Randomly project the input feature vectors  $x_1$  and  $x_2$  into high-dimensional spaces using random projection matrices  $W_1$  and  $W_2$ . These matrices are randomly generated and are shared across different samples.
- Let  $y_1 = W_1 * x_1$  and  $y_2 = W_2 * x_2$  be the projected feature vectors.

Element-wise Product:

- Compute the element-wise (Hadamard) product of the projected feature vectors  $y_1$  and  $y_2$ :

$$y = y_1 \odot y_2 \quad (15)$$

Compact Bilinear Pooling:

- Perform Compact Bilinear Pooling (CBP) on the element-wise product  $y$ .
- CBP is computed by taking the Fast Fourier Transform (FFT) of the element-wise product  $y$ , followed by  $FFT^{-1}$  operation.
- The resulting compact bilinear pooled feature vector is denoted as  $z$ .

Output:

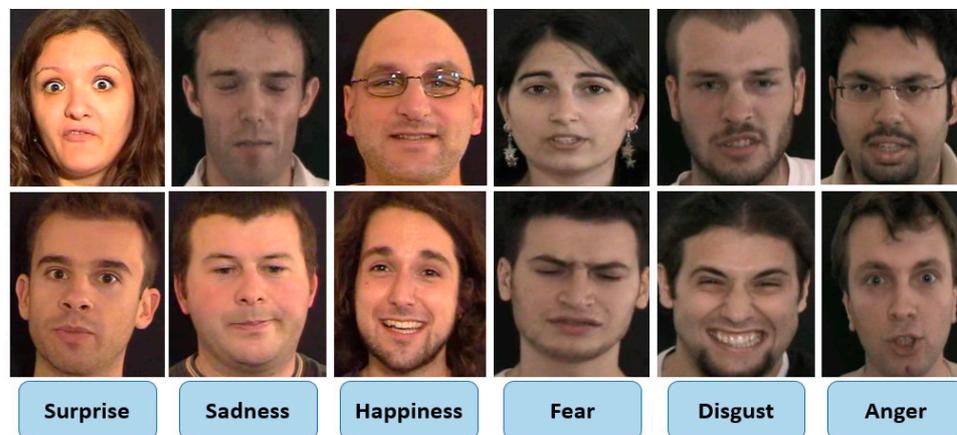
- The final output of the multimodal compact bilinear pooling operation is the compact bilinear pooled feature vector  $z$ .

#### 4. Experiment Studies and Result Analysis

This section examines the proposed method for concatenating the VGG19-LSTM model with the HOG-HOF descriptor. We execute tests at least 3 times for the classification vector. We will provide the average accuracy of the experiments.

#### 4.1. The Database Selected

The experiments were carried out utilizing the eNTERFACE05 [13] database (see Figure 11), a dataset specifically designed for audio-visual data. This database has been utilized in [33,38], allowing for a comparison of our results with previous studies.



**Figure 11.** Some facial images extracted from the eNTERFACE05 database.

The eNTERFACE05 database is a facial expression database commonly used in the field of facial expression recognition research. It was created as part of the eNTERFACE05 international challenge, which focused on facial expression (Angry, Disgusted, Happy, Sad, Scared, and Surprised) analysis. The database contains 1290 video sequences of facial expressions performed by multiple subjects in various scenarios. The video frame size is  $720 \times 576 \times 3$ . This database contains 42 individuals. Among them, 81% were men and the remaining 19% were women. A total of 31% wore glasses, while 17% of the individuals had a beard. Each video sequence lasts nearly 4 s and is in RGB color. Researchers use this database to develop and evaluate facial expression recognition algorithms.

Link for the eNTERFACE05 database: [50].

#### 4.2. The Classifier Selected

After evaluating both the SVM and multilayer perceptron (MLP) classifiers, it was observed that MLP requires significantly more time for training on a given dataset, and the achieved results are comparatively less powerful. Given the importance of speed in our context, the preferable choice is to utilize SVM.

The main application of SVM is in multiple classification tasks, where the objective is to classify input data points into one of multiple categories. Additionally, SVM is a versatile algorithm known for its effectiveness in high-dimensional spaces and robustness in scenarios with a clear margin between classes. Here are some details about using SVMs for facial expression recognition:

- **Feature Vector:** Extracted features are transformed into a feature vector, which serves as input to the SVM classifier. This vector represents the unique characteristics of facial expressions in the dataset.
- **Training:** SVMs are trained using a labeled dataset where each sample is associated with a specific facial expression label. The SVM can identify the optimal hyperplane for separating the various classes.
- **Kernel Trick:** SVMs frequently convert the input features into a space with more dimensions by using the kernel approach, making it easier to find a hyperplane that separates the classes.
- **Cross-Validation:** Cross-validation is essential to assess the SVM's generalization performance. It involves splitting the dataset into training and testing sets multiple times to evaluate the model's robustness.

- Evaluation Metrics: Common evaluation metrics for facial expression recognition using SVMs include accuracy, precision, recall, and F1 score.

Precision: Precision measures the accuracy of the positive predictions made by the classifier. It is calculated as the ratio of true positive predictions to the total number of positive predictions made by the classifier.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (16)$$

Recall (Sensitivity): Recall measures the ability of the classifier to correctly identify all positive instances in the dataset. It is calculated as the ratio of true positive predictions to the total number of actual positive instances in the dataset.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (17)$$

F1 score: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, making it a useful metric for imbalanced datasets. It is calculated as follows:

$$F1\text{-score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (18)$$

#### 4.3. Experimental Results and Analysis

To evaluate the performance of our FER system, we present the results on the eN-TERFACE05 dataset. The two phases that comprise our models are extraction and fusion of features.

Using the eNTERFACE05 dataset as a reference, we evaluated the accuracy of every component in our suggested approach for FER during the first stage of feature extraction. Our goal was to evaluate the accuracy of each model separately and identify the best combination that provides the highest score.

We used two methods in the fusion stage: one used a simple vector concatenation and the other used the MCB method. To choose the most effective approach, we compared the results of each technique.

The SVM classifier was employed to train models for visual feature extraction, with the approaches presented by Priyasad et al. [38] and Chouhayebi et al. [33] considered as reference models.

Priyasad et al. [38] employed the ResNet and VGG19 architectures to extract facial features, resulting in 4096-dimensional vectors, followed by the LSTM architecture to integrate visual and acoustic features. Chouhayebi et al. [33] adopted the same visual features as Priyasad et al. [38] and concatenated them with HOG-TOP features to obtain the most expressive vector using the SVM classifier.

##### 4.3.1. HOG-HOF Method Examination

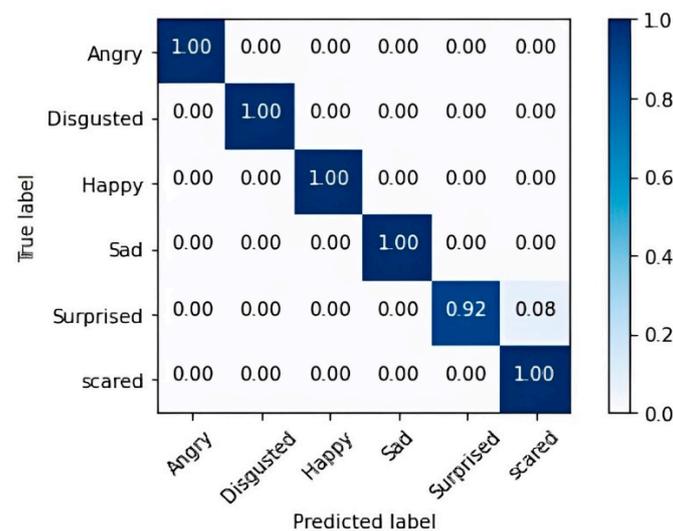
To optimize the performance of HOG-HOF descriptor we employ the SVM classifier with linear kernel using k-fold cross-validation to obtain an impressive score of 98.03%.

We use k-fold cross-validation with  $k = 10$  to evaluate the performance of our proposed method and its generalization capability, while also helping to mitigate the risk of overfitting by providing a more reliable estimate of the model's performance on unseen data. The choice of  $k = 10$  for k-fold cross-validation is based on the fact that our dataset is large. Therefore, it is a common practice aimed at balancing computational efficiency and obtaining reliable estimates of model performance. The cross-validation process in our case involves splitting the dataset into  $N$  (10) parts, where  $N$  is typically chosen based on the desired level of validation. Each part is then used iteratively for training and validation, and the model's performance is evaluated. Additionally, the model is tested separately on a completely unseen dataset to assess its generalization ability.

The accuracy for each emotion is detailed in Table 1, and the confusion matrix is presented in Figure 12. This accuracy represents the mean obtained through k-fold cross-validation.

**Table 1.** Score of HOG-HOF descriptor (Unit = %).

Emotion	Precision	Recall	F1-Score
Angry	100.00	100.00	100.00
Disgusted	100.00	100.00	100.00
Happy	100.00	100.00	100.00
Sad	100.00	100.00	100.00
Surprised	100.00	92.30	96.00
Scared	85.71	100.00	92.30
Accuracy	-	-	98.03
Macro avg	97.61	98.71	98.05
Weighted avg	98.31	98.03	98.07



**Figure 12.** The confusion matrix for the HOG-HOF descriptor.

Our algorithms' execution time is linear in terms of the number of training images, frames in a new video, pixels in a single image, and histogram size (number of spatial cells multiplied by orientation bins). The reliance on the size of the filtering zone for each pixel is logarithmic since sorting is required.

#### 4.3.2. LSTM Architecture Examination

We evaluate the performance of the VGG19-LSTM algorithm and optimize its hyper parameters to achieve better results by using SVM classifier with linear kernel utilizing k-fold cross-validation as shown in Table 2. For the LSTM model, we selected the following components: LSTM layer, dense layer, activation, and optimizer. The ReLU activation function tends to generate more diverse and informative representations in the hidden layers of the network compared to saturating activation functions like tanh or sigmoid. The Adam optimizer is less sensitive to the choice of hyperparameters compared to other optimization algorithms like SGD. It generally performs well across a wide range of tasks and hyperparameter settings, making it a popular choice for neural network training. We chose 2000 epochs because we observed through multiple training runs that the model stabilizes after around 1800 epochs.

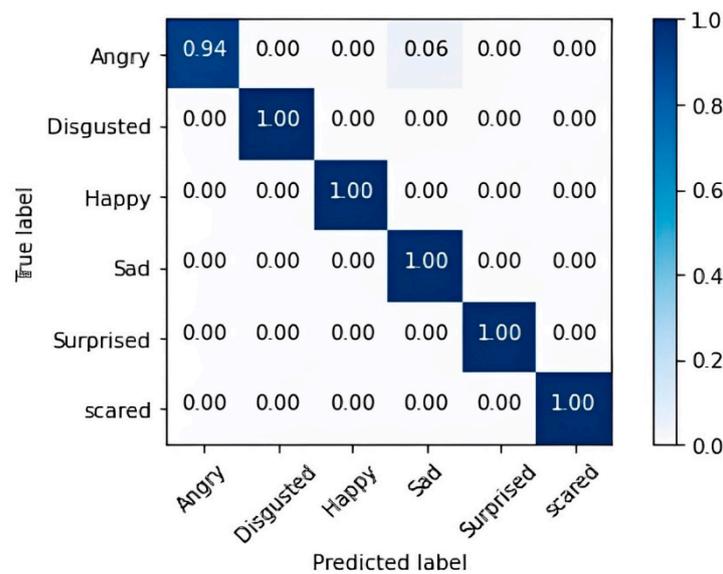
**Table 2.** List of LSTM model hyperparameters.

Model Hyperparameter Name	Value
LSTM layer 1	500
LSTM layer 2	350
LSTM layer 3	250
Dense layer	256
Activation Function	Relu
Dense layer	4096
Activation Function	Relu
Epochs	2000
Optimizer	adam
DropOut ratio	0.2

Every model used in this method was tested separately before being integrated to provide the most effective approach. Following the integration of the ResNet for face detection, VGG19 architecture, and LSTM algorithm, the results of the LSTM approach are presented, achieving a score of 98.30%, as illustrated in Table 3. The corresponding confusion matrix is depicted in Figure 13. This accuracy represents the average obtained using k-fold cross-validation.

**Table 3.** Score of ResNet, VGG19, and LSTM (Unit = %).

Emotion	Precision	Recall	F1-Score
Angry	100.00	94.11	96.97
Disgusted	100.00	100.00	100.00
Happy	100.00	100.00	100.00
Sad	93.33	100.00	96.55
Surprised	100.00	100.00	100.00
Scared	100.00	100.00	100.00
Accuracy	-	-	98.30
Macro avg	98.88	99.02	98.92
Weighted avg	98.41	98.30	98.30

**Figure 13.** The confusion matrix for ResNet, VGG19, and LSTM approach.

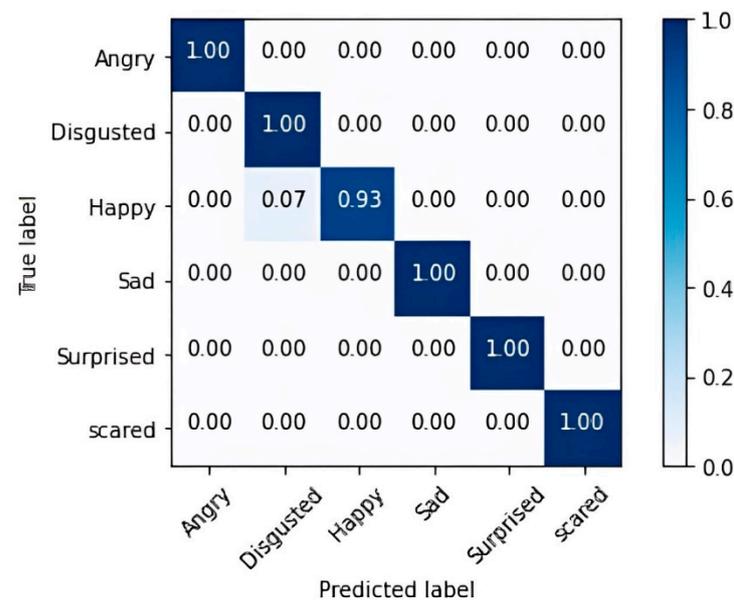
#### 4.3.3. Methods Vectors Merged

For optimal outcomes, we integrated the LSTM model-based approach with the HOG-HOF descriptor-based approach, employing two distinct methods.

In the initial fusion approach, a basic concatenation of two descriptor vectors was employed to generate an 8192-size output vector. The efficacy of this method is showcased in Table 4, utilizing the SVM classifier using k-fold cross-validation with  $k = 10$ . The confusion matrix is displayed in Figure 14, where 98.24% is the average score obtained using k-fold cross-validation.

**Table 4.** The score of the basic concatenation of the two vectors (Unit = %).

Emotion	Precision	Recall	F1-Score
Angry	100.00	100.00	100.00
Disgusted	87.50	100.00	93.33
Happy	100.00	92.85	96.29
Sad	100.00	100.00	100.00
Scared	100.00	100.00	100.00
Surprised	100.00	100.00	100.00
Accuracy	-	-	98.24
Macro avg	97.91	98.81	98.27
Weighted avg	98.46	98.24	98.27



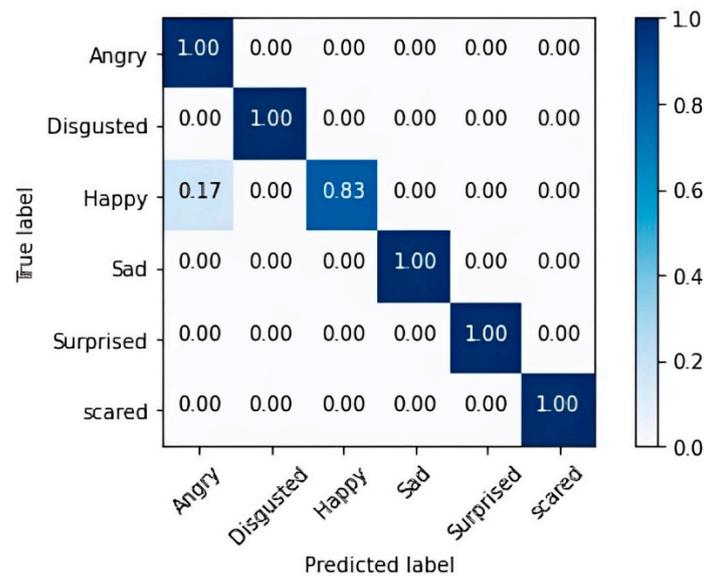
**Figure 14.** The confusion matrix for the basic concatenation of two vectors.

In the second fusion method, the MCB method was employed with a vector size of 16,000. The choice of 16,000 as the size of the vector descriptor is based on MCB parameters.

Table 5 shows this method's accuracy with the SVM classifier using k-fold cross-validation with  $k = 10$ , and Figure 15 presents the confusion matrix with a 97.14% score.

**Table 5.** The score of the concatenation of the two vectors using MCB algorithm (Unit = %).

Emotion	Precision	Recall	F1-Score
Angry	85.71	100.00	92.30
Disgusted	100.00	100.00	100.00
Happy	100.00	83.33	90.90
Sad	100.00	100.00	100.00
Scared	100.00	100.00	100.00
Surprised	100.00	100.00	100.00
Accuracy	-	-	97.14
Macro avg	97.61	97.22	97.20
Weighted avg	97.55	97.14	97.12



**Figure 15.** The confusion matrix resulting from MCB for the concatenation of vectors.

This accuracy denotes the average acquired through k-fold cross-validation.

After evaluating all kernel types of the SVM classifier, assessing their performance metrics, and computing the corresponding  $p$ -values (which indicate statistical significance with a value of 0.03), we conclude that the linear kernel outperforms others for emotion recognition classes.

After analyzing the results of all emotion recognition accuracies, especially for “Disgusted”, “Sad”, and “Scared”, we obtained recognition accuracies above 0.98, while “Happy”, “Surprised”, and “Angry” had relatively lower accuracies, although still above 0.88. This suggests that emotions such as “Surprised” and “Angry” may have subtle facial expressions that are easily confused with other emotions, making them more challenging to classify accurately.

Comparison analysis using State-of-the-Art methods:

We conducted a study to compare our method with other related works as shown in Table 6.

**Table 6.** Proposed method comparison with other state-of-the-art approaches in terms of accuracy function using the eNTERFACE05 database.

Method	Accuracy
Chouhayebi et al. [33]	96.89%
Hu et al. [34]	89.65%
Priyasad et al. [35]	80.51%
Priyasad et al. [38]	97.75%
Proposed method	98.24%

The first reference we considered is proposed by Chouhayebi et al. [33]. They combined two methods: the first one was based on VGG19 and LSTM models, and the second one was based on the HOG-TOP algorithm for appearance features. The combination of these methods was achieved using two different techniques (MCB and simple concatenation). They employed the SVM classifier using k-fold cross-validation on the same eNTERFACE05 database.

In our comparative study, the second reference utilized was proposed by Hu et al. [34]. They provide an approach to emotion identification that combines visual input with an attentional convolutional neural network’s bidirectional recurrent unit. To extract speech characteristics, they first trained log-mel spectrograms in a neural network based on ResNet. Second, voice characteristics are combined with static face appearance features that CNN

collected. Geometric face traits are extracted using a sequence of closed recurrent units equipped with attention mechanisms. After that, merged vocal appearance elements and face geometric features are further combined to create hybrid features.

Priyasad et al. [35] propose a deep learning approach to integrate and leverage audio and textual information for facial expression classification. Their methodology involves extracting auditory data from raw audio. Subsequently, a deep convolutional neural network is employed in their process.

Priyasad et al. [38] employed the LSTM architecture for extracting the visual features, which were concatenated with acoustic features using the MFCC model. They utilized basic concatenation of the two vectors and employed the SVM classifier using k-fold cross-validation on the eNTERFACE05 database.

Lakshmi et al. [27] utilize the modified HOG and the LBP. The Viola–Jones face detection method was used to detect the face region. A Butterworth high pass filter was then used to enhance the discovered region and identify the eyes, nose, and mouth region. Second, the suggested modified HOG and LBP feature descriptor is utilized to extract features from the identified selected regions. The obtained characteristics of these three areas are concatenated to obtain a vector descriptor by using CK+ dataset.

Upon reviewing existing methodologies, it becomes evident that traditional models face limitations as they heavily depend on manually crafted features, potentially failing to capture the intricate nuances of facial expressions or emotional cues within the dataset. Furthermore, deep learning approaches may exhibit sensitivity to facial expression variations induced by factors like lighting conditions, occlusions, head poses, and ethnic diversity, thereby resulting in diminished performance across diverse scenarios. Hence, our proposal advocates for the fusion of these two methodologies, constituting the primary contribution of our approach. We employed a spatio-temporal descriptor vector, combining HOG and HOF descriptors, alongside a descriptor vector derived from a pre-trained VGG-19 model and LSTM. Our secondary contribution lies in utilizing a generalized descriptor vector, integrating interactions from both descriptor vectors' elements.

After comparing our proposed method with the other related works, we observe that the combination of HOG-HOF and LSTM models offers a superior solution for emotion recognition compared to existing methods with an impressive score of 98.24%. This underscores the effectiveness of our approach in addressing the challenges associated with this task and highlights its potential for various real-world applications.

Our method outperforms the methods listed below. Table 7 presents a comprehensive comparison of the several methods employed in our study. In conclusion, it can be said that the strength of our novel method lies in the fusion of two different types of methods. One is based on facial texture in a spatio-temporal space, while the other method focuses on the use of the most relevant and recent methods for emotion recognition. The choice of the fusion method has also played a crucial role in achieving a higher score.

**Table 7.** Comprehensive comparison of the various methods.

Method	Accuracy
ResNet VGG19 LSTM	98.30%
HOG-HOF	98.03%
Fusion with basic concatenation	98.24%
Fusion with MCB algorithm	97.14%

Our approach achieved a significantly higher accuracy rate compared to existing methods. Specifically, we observed that our combined model outperformed other approaches in terms of both accuracy and robustness in facial expression recognition tasks.

By combining manual features, such as HOG and motion descriptors like HOF, with deep learning models like VGG-19 and LSTM, the proposed methodology addresses the limitations of both traditional and deep learning approaches. The integration of these

diverse techniques allows for a more comprehensive representation of facial expressions, capturing both spatial and temporal dynamics effectively.

Moreover, the utilization of the MCB method for integrating the descriptor vectors further enhances the model's discriminative power and robustness. This approach enables the extraction of rich features from the concatenated descriptor vectors, facilitating improved classification performance.

Overall, the methodology proposed in this work represents a significant advancement in facial expression recognition by combining the strengths of traditional and deep learning approaches while mitigating their respective limitations. This integrated approach results in a more robust and accurate system, demonstrating superior performance compared to traditional algorithms alone.

In the future, our aim is for our proposed approach to seamlessly transition into practical applications with tangible real-world effects, particularly in human–computer interactions. For instance, within virtual assistant technology, the system could dynamically adjust its responses according to the user's emotional cues, fostering empathetic and tailored interactions.

## 5. Conclusions

This research introduces a novel algorithm that merges two methodologies for facial expression recognition. The initial method employs deep learning models to extract visual features from video sequences. Specifically, the ResNet model is utilized for face extraction, and the VGG19 model is employed to generate a vector serving as an input for the LSTM (Many-To-One) model. This LSTM model constructs one descriptor vector of size 4096. This vector is combined with a second descriptor vector extracted using the HOG-HOF descriptor, which integrates appearance descriptor and motion descriptor HOF for spatial–temporal video processing. The resulting combined vector is subjected to SVM classification, utilizing two concatenation methods: MCB and simple concatenation. The results obtained with a score of 98.24% using the eNTERFACE05 database indicate that our fusion of the two methods yields better scores compared to state-of-the-art methods. We expect that the results of this study will set the groundwork for advancements in emotion recognition and related research. Integrating multiple modalities such as facial expressions, speech, gestures, and physiological signals can lead to more comprehensive and accurate emotion recognition systems. Future research could focus on developing sophisticated fusion techniques to combine information from different modalities effectively and using several datasets to improve our approach.

**Author Contributions:** Conceptualization, H.C.; Supervision, M.A.M., J.R., H.T. and N.A.; Writing—original draft, H.C.; Writing—review and editing, H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** We used the eNTERFACE05 dataset available online: [www.interface.net/interface05/docs/results/databases/project1\\_database.zip](http://www.interface.net/interface05/docs/results/databases/project1_database.zip)

**Acknowledgments:** The authors employ AI tools within the manuscript to detect and rectify errors related to spelling, grammar, punctuation, and overall clarity.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Efraty, B.; Huang, C.; Shah, S.K.; Kakadiaris, I.A. Facial landmark detection in uncontrolled conditions. In Proceedings of the 2011 International Joint Conference on Biometrics (IJCB), Washington, DC, USA, 11–13 October 2011. [CrossRef]
2. Ding, J.; Chen, K.; Liu, H.; Huang, L.; Chen, Y.; Lv, Y.; Yang, Q.; Guo, Q.; Han, Z.; Ralph, M.A.L. A unified neurocognitive model of semantics language social behaviour and face recognition in semantic dementia. *Nat. Commun.* **2020**, *11*, 1–14. [CrossRef]
3. Anagnostopoulos, C.N.; Iliou, T.; Giannoukos, I. Features and classifiers for emotion recognition from speech: A survey from 2000 to 2011. *Artif. Intell. Rev.* **2015**, *43*, 155–177. [CrossRef]

4. Dobs, K.; Isik, L.; Pantazis, D.; Kanwisher, N. How face perception unfolds over time. *Nat. Commun.* **2019**, *10*, 1–10. [[CrossRef](#)] [[PubMed](#)]
5. Kumar, M.P.; Rajagopal, M.K. Detecting facial emotions using normalized minimal feature vectors and semi-supervised twin support vector machines classifier. *Appl. Intell.* **2019**, *49*, 4150–4174. [[CrossRef](#)]
6. Kim, Y.; Lee, H.; Provost, E.M. Deep Learning for Robust Feature Generation in Audiovisual Emotion Recognition. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; University of Michigan Electrical Engineering and Computer Science: Ann Arbor, MI, USA, 2013; pp. 3687–3691.
7. Mellouk, W.; Handouzi, W. Facial emotion recognition using deep learning: Review and insights. *Procedia Comput. Sci.* **2020**, *175*, 689–694. [[CrossRef](#)]
8. Huang, Y.Y.; Wang, W.Y. Deep residual learning for weakly-supervised relation extraction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 27 September 2017; pp. 1803–1807. [[CrossRef](#)]
9. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
10. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM--A tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv* **2019**, arXiv:1909.09586, 1–42.
11. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
12. Wang, T.; Snoussi, H. Histograms of optical flow orientation for abnormal events detection. In Proceedings of the 2013 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), Clearwater Beach, FL, USA, 15–17 January 2013; pp. 45–52. [[CrossRef](#)]
13. Martin, O.; Kotsia, I.; Macq, B.; Pitas, I. The eNTERFACE'05 Audio-Visual emotion database. In Proceedings of the ICDEW 2006—22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, GA, USA, 3–7 April 2006; pp. 2–9. [[CrossRef](#)]
14. Liu, P.; Han, S.; Meng, Z.; Tong, Y. Facial expression recognition via a boosted deep belief network. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1805–1812. [[CrossRef](#)]
15. Mollahosseini, A.; Chan, D.; Mahoor, M.H. Going deeper in facial expression recognition using deep neural networks. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016. [[CrossRef](#)]
16. Zhang, F.; Zhang, T.; Mao, Q.; Xu, C. Joint Pose and Expression Modeling for Facial Expression Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3359–3368. [[CrossRef](#)]
17. Zhao, X.; Liang, X.; Liu, L.; Li, T.; Han, Y.; Vasconcelos, N.; Yan, S. Peak-piloted deep network for facial expression recognition. In *Computer Vision—ECCV 2016, Lecture Notes in Computer Science (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*; Springer: Cham, Switzerland, 2016; Volume 9906, pp. 425–442. [[CrossRef](#)]
18. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
19. Scovanner, P.; Ali, S.; Shah, M. A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th ACM International Conference on Multimedia, Augsburg Germany, 25–29 September 2007; pp. 357–360. [[CrossRef](#)]
20. Kläser, A.; Marszałek, M.; Schmid, C. A spatio-temporal descriptor based on 3D-gradients. In Proceedings of the BMVC 2008—British Machine Vision Conference 2008, Leeds, UK, 1 September 2008. [[CrossRef](#)]
21. Dalal, N.; Triggs, B.; Schmid, C. Human Detection Using Oriented Histograms of Flow and Appearance to cite this version: Human Detection using Oriented Histograms of Flow and Appearance. In *Computer Vision—ECCV 2006, Lecture Notes in Computer Science, European Conference on Computer Vision*; LNIP; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3952, pp. 428–441.
22. Laptev, I.; Marszałek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7. [[CrossRef](#)]
23. Corneanu, C.A.; Simón, M.O.; Cohn, J.F.; Guerrero, S.E. Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect-Related Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1548–1568. [[CrossRef](#)]
24. Li, R.; Tian, J.; Chua, M.C.H. Facial expression classification using salient pattern driven integrated geometric and textual features. *Multimed. Tools Appl.* **2019**, *78*, 28971–28983. [[CrossRef](#)]
25. Sadeghi, H.; Raie, A.A. Human vision inspired feature extraction for facial expression recognition. *Multimed. Tools Appl.* **2019**, *78*, 30335–30353. [[CrossRef](#)]
26. Sharma, M.; Jalal, A.S.; Khan, A. Emotion recognition using facial expression by fusing key points descriptor and texture features. *Multimed. Tools Appl.* **2019**, *78*, 16195–16219. [[CrossRef](#)]
27. Lakshmi, D.; Ponnusamy, R. Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders. *Microprocess. Microsyst.* **2021**, *82*, 103834. [[CrossRef](#)]
28. Cai, J.; Chang, O.; Tang, X.L.; Xue, C.; Wei, C. Facial Expression Recognition Method Based on Sparse Batch Normalization CNN. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 9608–9613. [[CrossRef](#)]
29. Agrawal, A.; Mittal, N. Using CNN for facial expression recognition: A study of the effects of kernel size and number of filters on accuracy. *Vis. Comput.* **2020**, *36*, 405–412. [[CrossRef](#)]

30. Li, Y.; Zeng, J.; Shan, S.; Chen, X. Occlusion Aware Facial Expression Recognition Using CNN with Attention Mechanism. *IEEE Trans. Image Process.* **2019**, *28*, 2439–2450. [[CrossRef](#)]
31. Kim, D.H.; Baddar, W.J.; Jang, J.; Ro, Y.M. Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition. *IEEE Trans. Affect. Comput.* **2019**, *10*, 223–236. [[CrossRef](#)]
32. Yolcu, G.; Oztel, I.; Kazan, S.; Oz, C.; Palaniappan, K.; Lever, T.E.; Bunyak, F. Facial expression recognition for monitoring neurological disorders based on convolutional neural network. *Multimed. Tools Appl.* **2019**, *78*, 31581–31603. [[CrossRef](#)]
33. Chouhayebi, H.; Mahraz, M.A.; Riffi, J.; Tairi, H. A dynamic fusion of features from deep learning and the HOG-TOP algorithm for facial expression recognition. *Multimed. Tools Appl.* **2023**, *83*, 32993–33017. [[CrossRef](#)]
34. Hu, Z.; Wang, L.; Luo, Y.; Xia, Y.; Xiao, H. Speech Emotion Recognition Model Based on Attention CNN Bi-GRU Fusing Visual Information. *Eng. Lett.* **2022**, *30*, 427–434.
35. Priyasad, D.; Fernando, T.; Denman, S.; Sridharan, S.; Fookes, C. Attention Driven Fusion for Multi-Modal Emotion Recognition. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3227–3231. [[CrossRef](#)]
36. Chowdary, M.K.; Nguyen, T.N.; Hemanth, D.J. Deep learning-based facial emotion recognition for human–computer interaction applications. *Neural Comput. Appl.* **2023**, *35*, 23311–23328. [[CrossRef](#)]
37. Li, B. Facial expression recognition via transfer learning. *EAI Endorsed Trans. e-Learn.* **2021**, 169180. [[CrossRef](#)]
38. Priyasad, D.; Fernando, T.; Denman, S.; Sridharan, S.; Fookes, C. Learning Salient Features for Multimodal Emotion Recognition with Recurrent Neural Networks and Attention Based Fusion. In Proceedings of the 15th International Conference on Auditory-Visual Speech Processing, Melbourne, Australia, 10–11 August 2019; pp. 21–26. [[CrossRef](#)]
39. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2010; pp. 248–255. [[CrossRef](#)]
40. Konečný, J.; Hagara, M. One-shot-learning gesture recognition using HOG-HOF features. *J. Mach. Learn. Res.* **2014**, *15*, 2513–2532. [[CrossRef](#)]
41. Harris, C.; Stephens, M. A Combined Corner and Edge Detector. In Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, 1 September 1988; pp. 23.1–23.6. [[CrossRef](#)]
42. Wang, H.; Ullah, M.M.; Kläser, A.; Laptev, I.; Schmid, C. Evaluation of local spatio-temporal features for action recognition. In Proceedings of the British Machine Vision Conference, BMVC 2009, London, UK, 7–10 September 2009. [[CrossRef](#)]
43. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [[CrossRef](#)]
44. King, D.E. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.* **2009**, *10*, 1755–1758.
45. Horn, B.K.P.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. [[CrossRef](#)]
46. Sun, D.; Roth, S.; Black, M.J. Secrets of optical flow estimation and their principles. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2432–2439. [[CrossRef](#)]
47. Farneb, G. Two-Frame Motion Estimation Based on polynomial expansion. *Lect. Notes Comput. Sci.* **2003**, *2749*, 363–370.
48. Dalal, N.; People, F.; Interaction, V.H. Finding People in Images and Videos. Ph.D. Thesis, Institut National Polytechnique de Grenoble-INPG, Grenoble, France, 2006; p. 150.
49. Fukui, A.; Park, D.H.; Yang, D.; Rohrbach, A.; Darrell, T.; Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. In Proceedings of the EMNLP 2016—2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 457–468. [[CrossRef](#)]
50. eNTERFACE05. Available online: [www.enterface.net/enterface05/docs/results/databases/project1\\_database.zip](http://www.enterface.net/enterface05/docs/results/databases/project1_database.zip) (accessed on 6 March 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.