ITC 1/54

Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 44-63
DOI 10.5755/j01.itc.54.1.36444

Optical-Flow Based Symmetric Feature Extraction for Facial Expression Recognition

Received 2024/02/23 | Accepted after revision 2024/10/17

HOW TO CITE: Zeraatkar, M. A., Joloudari, J. H., Rajesh, K. N. V. P. S., Gaftandzhieva, S., Hussain, S. (2025). Optical-Flow Based Symmetric Feature Extraction for Facial Expression Recognition. *Information Technology and Control*, 54(1), 44-63. https://doi.org/10.5755/j01.itc.54.1.36444

Optical-Flow Based Symmetric Feature Extraction for Facial Expression Recognition

Mohammad Ali Zeraatkar
Department of Computer Engineering, Islamic Azad University, Tehran, Iran; e-mail: mmdalix@gmail.com

Javad Hassannataj Joloudari
Department of Computer Engineering, Faculty of Engineering, University of Birjand, Birjand, Iran;
Department of Computer Engineering, Babol Branch, Islamic Azad University, Babol, Iran;
Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran;
e-mail: javad.hassannataj@birjand.ac.ir

Kandala N. V. P. S. Rajesh
School of Electronics Engineering, VIT-AP University, Vijayawada, India; e-mail: rajesh.k@vitap.ac.in

Silvia Gaftandzhieva
Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", Plovdiv, Bulgaria;
e-mail: sissiy88@uni-plovdiv.bg

Sadiq Hussain
Examination Branch, Dibrugarh University, Dibrugarh 786004, Assam, India; e-mail: sadiq@dibru.ac.in

Corresponding author: sissiy88@uni-plovdiv.bg

Facial expression analysis is one of the most essential tools for behavior interpretation and emotion modeling in Intelligent Human-Computer Interaction (HCI). Although humans can easily interpret facial emotions, computers have great difficulty doing so. Analyzing changes and deformations in the face is one of the methods through which machines can interpret facial expressions. However, maintaining great precision while being accurate, stable, and quick is still challenging in this field. This research presents an innovative and novel method to fully automatically extract critical features from a face during a facial expression to address this issue. Various machine learning models are used on these features to analyze emotions. We used the optical flow algorithm to extract motion vectors divided into sections on the subject's face. Finally, we calculate a new vector using each section and its symmetric section. The final features produce a state-of-the-art accuracy of over 98% in emotion classification in the Extended Cohen-Kanade (CK+) facial expression dataset. Furthermore, we proposed an algorithm to filter the most important features with an SVM classifier and achieved an accuracy of over 97 % by only looking at 15% of the face area.

KEYWORDS: Facial Expression Recognition, Optical Flow Algorithm, Feature Extraction, Emotion Recognition, Extended Cohen-Kanade (CK+) Dataset.

1. Introduction

Facial expression recognition (FER) systems play a significant role in machine interaction and perceiving human intentions in any social interaction between humans and machines, where emotion recognition is essential. To better understand humans' underlying thoughts and emotions in different situations, it is necessary to recognize human facial expressions [30], incl. in education for identifying the students' emotions during online learning sessions and help teachers change their teaching strategies in virtual learning environments and engage students [7, 33, 49, 58].

Although FER is very easy and intrinsic for humans, and some pieces of evidence show humans can easily recognize emotion even from different cultures [51], it is a very tough task for machines. Many FER systems are designed with vast methods to satisfy such a need. However, FER is still desirable in many fields of machine learning and computer vision because of the many challenges machines face for FER. The research in this field is focused on two main approaches. The first approach is to design techniques to extract or build dense information feature vectors and, simultaneously, very brief in dimensions. The second approach assumes designing machine learning models that leverage such extracted features for FER with the most accuracy possible in the least time and computation power. Because the FER Classifier is highly dependent on extracted features, its computation intensity and reaction time are dictated by architecture and the dimensions of its input features. Many recent research in the field focused on analyzing the importance of designing elegant and rich features; for example, Zamir et al. [46] investigated the areas of interest in the face used by classifiers such as C5.0, CRT, QUEST, CHAID, Deep Learning, and Discriminant algorithms, and concluded the eyes and mouth are the most influential parts. Nguyen et al. [39] explored the patterns of emotional regulation in collaborative learning, used Artificial Intelligence to examine learners' associated emotions and emotional synchrony in regulatory activities, and proposed an approach to provide empirical evidence on the application of technologies in assessing emotional regulation in synchronous computer-support collaborative learning.

Generally, facial expressions are one of the most essential components to investigate human emotions in Human-Computer Interaction (HCI) systems. Studies have demonstrated that most human communication uses facial matter [24, 51]. Thus, in such systems, a camera captures the human face. Researchers used various techniques such as Gabor filters, local binary patterns (LBP), convolutional neural networks (CNN), and histogram of oriented gradients (HOGs) to analyze the captured video [6, 44] or image [45] for interpreting extreme and subtle [26, 45] emotions.

Despite the solutions proposed by researchers, maintaining high accuracy while being quick and spontaneous is still a challenging task. As a result, the need for heavy computation and processing delays in some high-accuracy models became problematic, rendering them useless for some real-time applications that require instant processing.

In this research, we will offer a new feature extraction method based on motion vectors on the face area for extracting essential characteristics from facial tissue deformations during a facial expression sequence. This paper focused on designing a novel feature set that may generalize the model with fewer dimensions. We have analyzed how the symmetricity of the human face can lead to the design of symmetric features for both left and right parts of the face without losing meaningful information in prediction but with improvement in dimensionality reduction, which will be translated to get faster at the time of classification.

The main contributions of this study are as follows:

- Proposed a new feature extraction method based on the symmetricity of the face to extract the underlying facial information for face expression detection.
- The proposed approach attained an accuracy exceeding 98% on the CK+ dataset [29], surpassing the previous state-of-the-art method by a margin of 3%.
- This study introduces a feature selection technique capable of preserving the model's accuracy above 97%, leveraging only 15% of the facial area.
- Employing an optical flow algorithm to utilize sequence data of facial expression results in robust performance.
- Achieving a low latency of 36ms in Deep Models And as low as 0.2ms for SVM models during the classification phase, exhibiting the effectiveness of the proposed feature processing method in real-world applications.
- High generalization because of innovative face segmentation method and proposed symmetric merge of vectors.
- Two different approaches followed for the data preparation for time-series and non-time-series models.

The subsequent sections of this paper are structured as follows. Section 2 delves into an extensive review of relevant literature, contextualizing our proposed objective within the broader landscape of existing research. Section 3 elucidates the methodology of our proposed approach, delineating the steps encompassing preprocessing and feature extraction, training procedures, and classification strategies. In Section 4, we present a thorough analysis of our results, accompanied by insightful discussions, where we not only showcase classification outcomes but also unveil the underlying importance of features. Finally, Section 5 encapsulates our contributions, synthesizing key insights, and charting a course for future research endeavors, thereby enriching the discourse within the domain. Additionally, the appendices at the paper's denouement furnish supplementary materials including confusion matrices, detailed model configurations, and elucidating pseudo-codes.

2. Related Work

This section presents the detailed literature work carried out in the research domain. The methods followed by the researchers for facial expression recognition (FER) can be broadly classified into two ways. The first is computing new features to be used by classification models, and the second is designing and using new types of classification models (deep learning) to improve FER, which is discussed in the subsequent two subsections.

2.1. Different Feature Extraction Schemes

Most of the literature aimed to design new features from pictures of faces to exhale recognition by feature engineering. And they proved that these handcrafted features are good at improving the accuracy of FER. Zhang et al. [57] proposed a pose deduction by nose position in the picture, which was more of a preprocessing approach than feature extraction, but the results were comparable. Ji and Idrissi [22] proposed a new image normalization method to make images invariant to illumination and reduce noise to some degree. Another method was proposed based on histogram equalization to overcome illumination variations [12]. In 2008, Ahmad et al. [38] introduced a facial expression method based on optical flow that outperformed the methods until then. They used principal component analysis (PCA) on the optical flow of face shots. Their method is validated on the "Cohn-Kanade AU-Coded" dataset and achieved an accuracy of 94% for all frames of each face and 83% for using only the last frame of the face. Despite their better results, the method was time-consuming because of using optical flow on all frames and all parts of the face. However, optical flow is widely used along with many models. Recently, the work in [5] showed that leveraging optical flow to create consistent optical flow maps attained 93.17% and 95.34% accuracy for the CK+ dataset and 65.35% for CASME2 datasets, respectively. Another method [46] used data mining to analyze facial expression by computing a new feature set named motion vector based on optical flow. The other exploited feature is the local binary pattern (LBP) [18], which is a texture descriptor for images. The work in [2] used the LBP features to feed SVM and get 87% accuracy for the JAFFE dataset and 77% for the MUFE datasets, respectively. Although LBP is an old texture descriptor, many researchers used it and proved its efficacy in FER. Happy and Routray [18] explored the LBP along with PHOG and obtained 94% accuracy for the JAFFE dataset. Saurav et al. [48] investigated LBP features with a few other texture descriptors. Besides, there are many other explored features for FER. Vasanth et al. [52] used a Gabor filter with the LBP feature and tried to do classification using SVM. HOG is another texture descriptor made by a gradient filter on the edges. Xu et al. [55] gained 92% accuracy on the CK dataset using these Gabor filters and HOG features.

Some work incorporated feature extraction on facial area for other objectives such as facial beauty assessment [53, 3] and face recognition [25], the symmetry of the face was also primarily utilized in some of them [53].

2.2. Classification Models

The further important aspect of FER is choosing the appropriate classification algorithm. This section presents the literature that used various supervised machine-learning algorithms.

As facial expression is a classification problem, **many classifiers** have been used in the literature. Muid et al. [37] developed a fuzzy logic-based method for FER. The approach achieved an accuracy of 81.22 %. Liliana et al. proposed another fuzzy-based-FER, which got 90% accuracy on the Cohn–Kanade dataset. Rahul et al. [43] explored various parameters of the SVM, and the simulations were validated on the JAFFE dataset and obtained an accuracy of 87 % and 77 % on the MUFE dataset. Rahul et al. [43] presented an FER using a probabilistic machine learning model, namely, hidden Markov models with Gabor filter-based features and obtained 88% accuracy. Furthermore, a few methods also utilized clustering algorithms for FER. It is an unsupervised approach. Bashyal et al. [8] proposed a FER method based on learning vector quantization (LVQ), a clustering approach. Another popular set of models is from tree structures. Noh et al. [40] used a simple ID3 classification tree algorithm on the JAFFE dataset and showed an accuracy of 75 %. Salmam et al. [47] used a simple Classification and Regression Tree (CART) model and reported an accuracy of 89.9% on the JAFFE dataset.

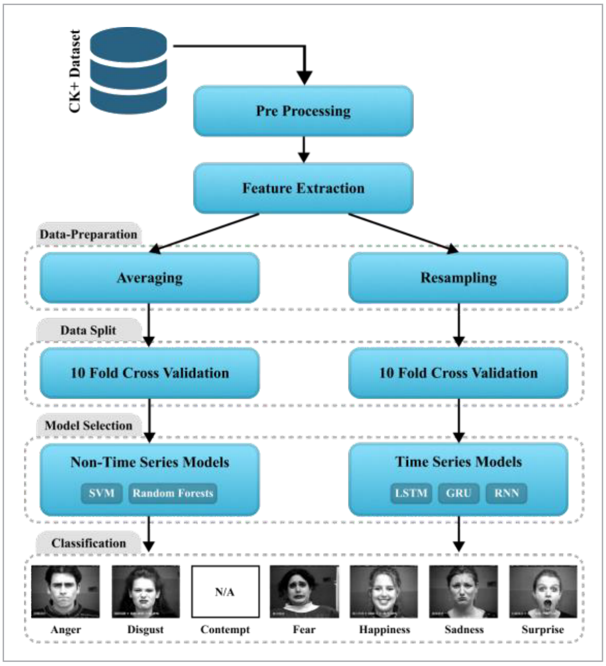
The major limitation of using formal machine learning (ML) models is the requirement for handcrafted features. An insignificant or large set of features can diminish the effectiveness of the ML model's performance. The best alternative to this problem is related to the utilization of **deep learning** (DL) models. They are showing their robustness in several fields. The main advantage of the DL models is their capacity to generate various feature maps (higher and lower levels) from the input without even knowing them. It will reduce the researchers' hard work for the best features manually searching. Qin et al. [42] proposed Gabor filters and wavelet transform with a 2-channel and Convolutional Neural Networks (CNNs) based method for FER and achieved 96.81% accuracy for the CK+ dataset. Recently, Pyramid-based DL models have gained a lot of attention. Mahersia and Hamrouni [31] implemented a FER method using multiple steerable filters and Bayesian regularization with Steerable pyramids. The method achieved an accuracy of 95.73 %. An LSTM and recurrent models have been one of the logical choices to process sequence data like facial expressions, which consist of sequences of images. Yu et al. [56] used nested LSTM models by exploiting convolutional layers for FER. The recent popular DL methods are attention-based networks. Fernandez et al. [17] proposed a FER method based on the attention model with Gaussian space representation to learn multi-level features and got 90.3 % accuracy on the CK+ dataset. Minaee et al. [36] also employed this attention mechanism for FER. Alenazy et al. [4] proposed a hybrid method using deep belief networks and GSA to optimize the DBN network to achieve a precise result of facial expression classification. Despite the advantages of the DL algorithms, they also suffer from a few shortcomings, such as overfitting and generalization.

Roshanzamir et al. [46] utilized data mining methods to automatically recognize facial expressions in the Cohen-Kanade facial expression dataset. The methods employed included deep learning, support vector machine, and C5.0. In addition, they examined the facial changes that occur during an expression, which are captured as motion vectors. These vectors are then utilized for expression analysis using the specified data mining methods. The findings indicate that deep learning outperforms other algorithms with an accuracy rate of 95.3%.

3. Proposed Method

The method used for feature extraction is crucial. As was already mentioned, there are various approaches. A good feature extraction method must have stability, accuracy, speed, and versatility, all of which have been difficult to achieve up to this point. The proposed approach for this work, designed to meet these criteria, is visually represented in Figure 1.

Figure 1
Schematic representation of the overall methodology used in our work



Based on Figure 1, the proposed methodology includes:

1. Preprocessing
2. Feature Extraction
3. Model-Specific Data Preparation
4. K-fold Data Split
5. Model Selection & Training
6. Classification

Stages 1 to 4 will be detailed in depth in the following sections, and the evaluation results will be provided in the results section.

3.1. Pre-Processing

This section outlines the pre-processing steps we took to improve data quality before applying our feature extraction techniques. Specifically, we employed two pre-processing methods: high pass addition and face boundary and nose tip detection. We used high pass addition to the facial expression image sequence to emphasize skin texture, enabling the optical flow algorithm for better tracking of deformations in facial muscles. Additionally, we utilized face boundary and nose tip detection to accurately locate the face boundary and nose tip positions, which were necessary for segmenting the facial area in the feature extraction phase.

High Pass Addition

To analyze deformations in the subject face area, we decided to use the optical flow algorithm and extract pixel-wise motion vectors in the facial expression sequence.

As the optical flow method is used to assess deformation on the face area, emphasizing the skin texture may considerably improve the algorithm's accuracy and resilience.

Based on some previous works [50], it has been shown that using the high-pass filter can emphasize textures in an image. Therefore, in our study, we used the high-pass filter addition on each frame of the expression sequence. This approach enhances the optical flow algorithm's ability to analyze deformations in the subject's face area and to improve the accuracy of pixel-wise motion vector extraction.

Specifically, we used the high-pass filter to isolate the high-frequency components of each frame, which primarily correspond to skin texture. We then combined the results from the filtering with the original frames to produce a new sequence that better highlights the skin texture. Figure 2 illustrates this effect, where a high-pass filter has been applied to an original image, resulting in an enhanced facial texture.

Figure 2
Example of the effect of high-pass filter addition on facial texture enhancement. The image on the right is generated by applying a high-pass filter to the original image on the left

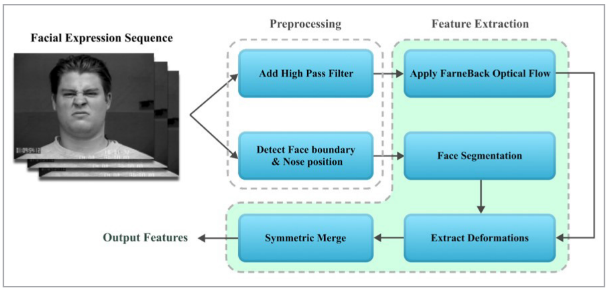
Face boundary and nose tip detection

Our technique is based on a novel facial area segmentation. The face boundary and position of the tip of the nose should be specified to define the segments on the subject face. Since there are many powerful methods for automatic facial boundary detection, it is not the focus of our work. To achieve this purpose in our experiments, we employed a pre-trained boundary detector model and extracted the nose tip position by the provided landmarks in the dataset.

3.2. Feature Extraction

Figure 3 depicts the steps for the feature extraction method.

Figure 3
An overview of the different feature extraction stages utilized in our work



Apply FarneBack Optical Flow

Optical flow is a computer vision technique that estimates the apparent motion of objects in a sequence of images or video frames. The concept was first introduced by Gibson [32] in the 1950s and later developed by Horn and Schunck in the 1980s [21]. It quantifies the displacement of pixels between consecutive frames, providing a dense motion field that represents the movement of the scene's objects. This information is useful, especially in various applications, such as video compression, motion analysis, and facial expression recognition.

The Farneback method, proposed by Farneback in 2003 [16], is an efficient algorithm for estimating optical flow based on the idea of approximating the neighborhood of each pixel in the image sequence by quadratic polynomials. By analyzing these polynomials, the Farneback method can compute the displacement fields that describe the motion between frames.

The core of optical flow estimation lies in solving the optical flow equation. Given an image sequence $I(x,y,t)$ where there are spatial coordinates and time, the optical flow equation can be written as:

$$I(x,y,t) = I(x+dx, y+dy, t+dt), \tag{1}$$

here, dx, dy represent the displacement of the pixel in the x and y directions, respectively, and dt is the time difference between frames.

The optical flow equation can be linearized using the Taylor series expansion and assuming small displacements, which leads to the following equation:

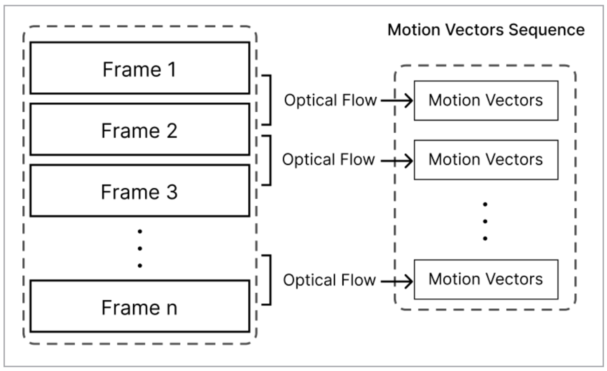
$$IxVx + IyVy = -It. \tag{2}$$

In this equation, Ix and Iy are the image gradients in the x and y directions, and Vx and Vy are the components of the optical flow vector (displacement) in the x and y directions, respectively. It represents the brightness constancy constraint, which assumes that the pixel intensity remains constant during motion.

The Farneback method solves the optical flow equation by constructing a set of quadratic polynomials that approximate the image sequence's intensity function. The polynomial expansion enables the algorithm to accurately represent motion on different scales, thus providing a robust optical flow estimation. Then, the method employs a hierarchical approach, computing the optical flow at multiple resolutions and iteratively refining the estimates at each level.

We applied the optical flow Farneback algorithm on a facial expression sequence to create a motion vector sequence, as demonstrated in Figure 4.

Figure 4
A diagram showing how optical flow is applied on facial expression sequence frames to build a motion vector sequence



The optical flow algorithm is applied to the entire frame border to simplify our solution. However, to enhance computing efficiency, it is feasible to use it simply on the detected face boundary or on some areas of the face, which we will define in the following section.

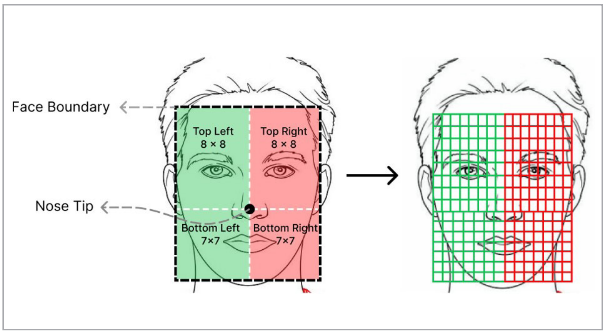
Face Segmentation

To improve generalization and minimize computational complexity, we need to take an average of motion vectors on the face to perform an innovative facial segmentation method to define areas on the face, which we will use to extract a mean vector by averaging motion vectors in them.

As illustrated in Figure 5, the face area is first divided into sections regarding the nose tip position. Then, the upper sections are converted to a grid of 8x8 and the bottom sections to a 7x7 grid. These are hyperparameters, and in our tests, they produced the best outcomes.

Figure 5

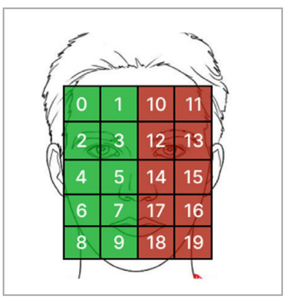
An example of Facial Area Segmentation and Grid Configuration



We assign indices to segments from Top left to bottom right in the left area first and then in the right area. Figure 6 provides an example of how we assigned indices to them (since it was hard to visualize numbers in 8x8 grids, in this example, we used 2x3 grids for the upper area, and 2x2 grids for the lower area):

Figure 6

An example of Facial Area Segmentation and Grid Configuration



Since we have the same grids on the left and right of the face area, every section in the left area has a symmetric segment in the right area.

Equation (3) shows how we mapped every segment in the left part to its symmetric section in the right

$$Sym(i) = -1 + N_{upper} + N_{lower} + \begin{cases} \left\lfloor \frac{i}{U_{cols}} \right\rfloor \times U_{cols} + U_{cols} - i \,\%\, U_{cols}, & i < N_{upper} \\ N_{upper} + \left\lfloor \frac{i - N_{upper}}{L_{cols}} \right\rfloor \times L_{cols} + L_{cols} - (i - N_{upper}) \,\%\, U_{cols}, & i \geq N_{upper} \end{cases} \quad (3)$$

The formula utilizes the input index i to map a section to a symmetric section. N_{upper} and N_{lower} are the total number of sections in the upper and lower half of the grid, respectively, and U_{cols} and L_{cols} are the number of columns in the upper and lower half of the grid, respectively. The $\%$ symbol is used to calculate the remainder after division.

The pseudo-code of the function is provided in Appendix A.

Extract Deformations and Symmetric merge

In this step, a mean vector in each segment will be calculated by averaging over all motion vectors extracted in that section area.

$$\overline{S(i)} = \frac{1}{N_i} \sum_{k=1}^{N_i} \overline{M_{i\,k}}, \quad (4)$$

where $\overline{S(i)}$ is the mean motion vector in segment i, N_i refers to the number of motion vectors in segment i and $M_{i\,k}$ is the motion vector with index k in segment i.

To standardize our features and eliminate any noise, we combined the final vectors of each symmetrical pair of facial sections. This approach gives the sense that facial structures are usually symmetrical, and most facial expressions occur in a horizontal symmetrical pattern. The formula we used for this process is as follows:

$$\overline{S_{merged}(i)} = \begin{bmatrix} S(i)_1 - S(Sym(i))_1 \\ S(i)_2 + S(Sym(i))_2 \end{bmatrix} \times \frac{1}{2}. \quad (5)$$

In this equation, we subtract the first component of the vectors, which corresponds to the X-axis in motion vectors, and add the second component, which represents the Y-axis. After that, we normalize these results by multiplying them by ½. We chose to subtract the X-axis components because the face's horizontal symmetry implies that the motion vectors in each section are likely to move in the opposite direction to their symmetrical counterparts. Subtracting these values helps to prevent them from cancelling each other out.

3.3. Data Preparation

This section outlines the steps we took to prepare our data for use in our models. Specifically, we provide two distinct methods for data preparation, one for time-series models and the other for non-time-series models. We employed resampling techniques for time-series models to ensure that our data was uniformly distributed across time. For non-time-series models, we utilized averaging techniques to summarize the features of our data.

Resampling

Before going to train models by the features extracted from dataset samples, we have to perform some normalization techniques.

Since facial expression samples in the CK+ dataset have different lengths, we need to perform a resampling technique to make them the same length and train time-series-based models.

In the resampling method we're using, we employ a technique known as linear interpolation. The complete formula for this resampling approach can be expressed as follows:

$$T[p] = R[\lfloor p \rfloor] + (R[\lceil p \rceil] - R[\lfloor p \rfloor]) \times (p - \lfloor p \rfloor) \quad (6)$$

$$p \in \{n \times \frac{R_n}{T_n} \mid n \in \{0,1,\dots,T_n - 1\}\}, \quad (7)$$

where T_n represents the desired number of samples, T is the resampled signal, R stands for the initial sequence of the signal, and R_n is the length of this original sequence.

Averaging

On the other hand, for non-time-series models, we used an averaging method as below:

$$R_{mean} = \frac{1}{R_n} \sum_{k=1}^{R_n} R[k], \quad (8)$$

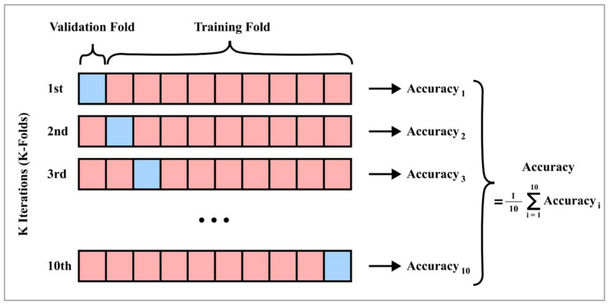
where R stands for the initial sequence of the signal, and R_n is the length of this original sequence.

3.4. Data Split

In the data split phase of our work, we employed a technique known as k-fold cross-validation (KCV) [14] to effectively split the dataset into multiple partitions for training and validation purposes. K-fold cross-validation is a widely used resampling method that aims to reduce the risk of overfitting and improve the accuracy of a model's generalization ability. As illustrated in Figure 7, it does so by dividing the dataset into k equally sized subsets or "folds" and then using each of these folds as a validation set while training the model on the remaining k-1 folds. This process is repeated k times, and the model's performance is evaluated using the average of the accuracy scores obtained from each iteration.

Figure 7

Illustration of k-fold cross-validation for evaluating model performance



In our work, we opted for a 5-fold cross-validation approach, which involved partitioning the dataset into five distinct folds. We trained the model with four from these folds and validated the remaining one during each iteration. The overall performance of our model was assessed by averaging the accuracy scores from each of the five iterations.

3.5. Model Selection

We used different model types to evaluate our work. Support Vector Machine (SVM), Random Forest (RF), XGBoost, Feed-Forward Neural Network (FNN), and an ensemble model with SVM estimators were used for non-time-series and Long Short-Term Memory (LSTM) network for time-series data. We assessed our approach using both traditional and deep learning models. While contemporary deep models strive for superior performance, their significantly higher computational complexity is acknowledged. By evaluating our method with classical models, we demonstrate its capability to deliver strong performance on resource-constrained hardware and real-time applications, emphasizing its practical utility across diverse computing environments.

Support Vector Machines

Support Vector Machines (SVMs) are supervised learning algorithms for classification, regression, and outlier detection. The optimization problem for linearly separable data can be formulated as:

$$\min_{w,b} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} \ \ subject\ to\ \ y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) \geq 1, \forall i = 1,2,\ldots,n \tag{9}$$

where $\boldsymbol{x_i}$ is the ith input vector, y_i is its associated binary label, $\boldsymbol{w}$ is the weight vector of the hyperplane, b is the bias term, and n is the number of training examples.

SVMs are a powerful and flexible machine learning algorithm that has been widely used in various applications [11].

Random Forest

Random forest (RF) is a powerful and versatile ensemble learning method for classification and regression tasks. It was first introduced by Breiman in 2001 [9] and has since gained widespread popularity due to its robustness, simplicity, and ability to handle large datasets with a high number of features.

The main idea behind random forest is to build a collection of decision trees and combine their predictions to produce a more accurate and stable output. Each decision tree is grown using a random subset of the training data, and at each node of the tree, a random subset of features is considered for splitting. This randomization strategy helps reduce the correlation between individual trees, which in turn reduces the overall variance of the model [27].

One of the key advantages of random forest is their ability to provide an estimate of feature importance. For each tree, the importance of a feature can be computed as the total decrease in impurity (e.g., Gini index or entropy) that results from all the splits on that feature, averaged across all trees in the forest [19].

Formally, the Gini impurity for a node can be calculated as:

$$Gini(p) = 1 - \sum_{i=1}^{C} (p_i)^2, \tag{10}$$

where p_i is the proportion of samples belonging to the class i in the node, and C is the total number of classes.

XGBoost

XGBoost [10], short for Extreme Gradient Boosting, is a powerful machine learning algorithm known for its exceptional performance in various predictive modeling tasks. It is an ensemble learning method that combines the predictions of multiple weak decision trees to create a strong predictive model. XGBoost utilizes a gradient boosting framework, which iteratively builds new decision trees to correct the mistakes of previous trees. It incorporates a range of advanced techniques, such as regularization, parallel processing, and tree pruning, to enhance its predictive accuracy and generalization capabilities. Due to its effectiveness and versatility, XGBoost has become a popular choice for various applications, including classification, regression, and ranking problems.

Ensemble Learning

Ensemble learning is a machine learning technique combining multiple models to improve predictive performance. It helps reduce overfitting, increase robustness, and has been successful in various applications. A famous example of ensemble learning is the Random Forest algorithm, which combines multiple decision trees to create a more accurate predictor [13].

Feed-Forward Neural Network

A Feed Forward Neural Network (FNN), also known as a multilayer perceptron, is a fundamental type of artificial neural network. It consists of an input layer, one or more hidden layers, and an output layer. In an FNN, information flows in a straight direction from the input layer through the hidden layers to the output layer without any loops or feedback connections. Each neuron in the network receives inputs, performs a weighted sum of those inputs, applies an activation function, and passes the result as output to the next layer. The hidden layers in an FNN allow for complex nonlinear transformations, enabling the network to learn and represent intricate relationships in the data. FNNs are extensively used in various machine learning tasks, including classification, regression, and pattern recognition, owing to their ability to model complex data relationships. The configuration that we used for this model is provided in Appendix B.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a recurrent neural network architecture designed to overcome the vanishing gradient problem and effectively handle long-term dependencies in sequence data. It introduces a memory cell and gating mechanisms to allow or prevent information flow through the cell. LSTMs have been successfully applied to various tasks involving sequence data. The original paper by Hochreiter and Schmidhuber (1997) provides a detailed description of LSTM architecture and its performance on different benchmarks [20]. The configuration that we used for this model is provided in Appendix B.

3.6. Reducing Feature Dimensionality

To maximize the computational efficiency of our work, we decided to select the most significant features.

To do that, we used a measure of feature importance calculated during the training of the random forest model. Random forests are a tree-based model used commonly for non-linear data regression and classification. During training, the model calculates feature importance scores based on a selected criterion, such as the 'gini' criteria. We selected the top 15% of features with the highest importance scores as the most significant features for our analysis.

The Results section provides the results obtained with all selected features. We found that the performance of models with the selected features was near optimal and comparable to the performance of models with all features. This finding suggests that the selected features contain most of the relevant information needed for accurate predictions while reducing the computational cost and complexity of the model.

3.7. Environment Setup

The hardware and software specifications for the experimental setup are as follows:

Memory: 13GB RAM, 16GB GPU

GPU: NVIDIA Tesla P100

CPU: Intel(R) Xeon(R) CPU @ 2.00GHz

For the implementation of deep learning models, we utilized the TensorFlow framework [1] and employed the Adam optimizer [23]. In contrast, the other models in this study were implemented using the scikit-learn library [41]. This experimental environment provided the necessary computational resources and tools to effectively test and evaluate our proposed feature extraction technique for facial expression classification.

3.7. CK+ Dataset

The Cohn-Kanade Extended (CK+) dataset is a widely recognized benchmark dataset in the field of facial expression analysis. It comprises a comprehensive collection of facial expressions captured in a controlled environment, featuring subjects sequentially displaying a range of emotions. The sequential nature of the images in CK+ allows for a more dynamic analysis of facial expressions over time, capturing the subtle changes and nuances in emotional expression. In this research, we specifically chose CK+ as our dataset of choice for its availability of sequential images and rich content, enabling a robust evaluation of our proposed optical-flow-based symmetric feature extraction method and allowing for meaningful comparisons with existing literature. It is important to note that the entire CK+ dataset was utilized in this work, ensuring comprehensive analysis and fair comparison with other works.

4. Results and Discussion

In this section, we present the results of our experiments and discuss the effectiveness of our proposed approach. Firstly, we report the classification accuracy achieved by our method and provide confusion matrices to evaluate the classifier performance. Furthermore, we conduct feature importance analysis using a heatmap to identify the most informative face sections for the classification task. Finally, in the "Discussion" subsection, we synthesize our results and provide a thorough evaluation of the proposed technique.

4.1. Classification Results

In this section, we present the classification results of our proposed feature extraction technique for the recognition of facial expressions. Firstly, we report the accuracy achieved by our method, which serves as a measure of the overall performance of the classifier. Additionally, we have provided confusion matrices in Appendix C to further evaluate the classification results by analyzing the distribution of correctly and incorrectly classified samples across different facial expressions. These results are essential in assessing the effectiveness of our proposed approach and comparing it to other existing methods.

Accuracy measures the percentage of correctly classified instances out of the total number of instances in the dataset. It is calculated as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ Number\ of\ predictions}. \tag{11}$$

Table 1 presents results from comparing the performance accuracy of different machine learning models utilized in our study, which include a mix of traditional algorithms and deep learning models. The accuracy metrics are provided for two distinct cases: one where all features are considered and the other where a subset of selected features is used. Remarkably, the Feedforward Neural Network (FNN) model stands out, achieving the highest accuracy of 98.17% among all tested models. Furthermore, it's worth noting that using selected features in FNN yields competitive accuracies, with only a marginal decrease of 0.32% compared to using all features. his conclusion allows us to leverage the selected features for more computationally efficient models while maintaining strong performance.

Table 1
Comparison of Model Accuracy on a Task with All Features and Subset of Features

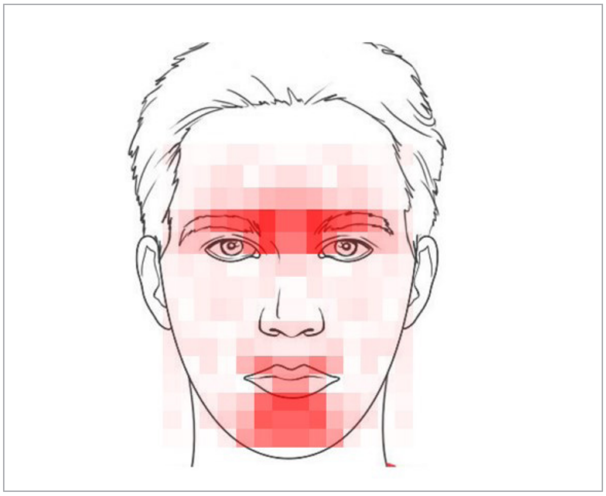
Model	Accuracy All Features	Accuracy Selected Features
SVM	97.53%	96.94%
XGBoost	93.26%	91.38%
Random Forest	94.49%	94.49%
Bagging (SVM)	97.24%	97.25%
FNN	98.17%	97.85%
LSTM	94.50%	94.48%

4.2. Features Importance: Heatmap Analysis of Face Sections

In this section of the results, we present a heatmap analysis of the most important square sections of the face as identified through feature selection using a bagging SVM classifier.

The heatmap analysis provides a visual representation of the importance of each square section of the face in emotion recognition. This analysis can inform the development of more accurate emotion recognition models, as it highlights the specific facial features, most important for recognizing certain emotions. Figure 8 shows the significance of different facial regions for emotion recognition using a Bagging SVM classifier.

Figure 8
Heatmap illustrating the significance of different facial regions for emotion recognition using a Bagging SVM classifier



4.3. Inference Time and Parallel Processing

Table 2 presents the inference time analysis for different machine learning models utilizing both all features and selected features for facial expression recognition. All these measurements are averages of 100 runs.

Parallel processing during training is feasible for models such as FNN, RF, and XGBoost, where training can be distributed across multiple processors or nodes. However, due to the sequential training nature of SVM and LSTM, parallel processing during train-

Table 2
Inference time analysis

Features	Method	Time per single-face	Time per batch of 64	Time per instance in batch
All Features	SVM	0.20ms	0.57ms	0.009ms
	RF	10.6ms	11.7ms	0.18ms
	XGBoost	4.2ms	2.6ms	0.041ms
	FNN	36ms	35ms	0.54ms
	LSTM	42ms	41ms	0.64ms
Selected Features	SVM	0.13ms	0.72ms	0.0013ms
	RF	10.6ms	11.9ms	0.18ms
	XGBoost	2.2ms	2.4ms	0.038ms
	FNN	36ms	32ms	0.5ms
	LSTM	47ms	40ms	0.62ms

ing is not typically feasible. Nevertheless, our results demonstrate that our method generalizes well even with a limited number of samples, enabling all models to be effectively trained on a single machine.

4.4. Discussion

Facial expression is one of the most universal, natural, and powerful signals for persons to convey their intentions and emotional states. FER has been a hot topic of research because of its life application value, practical value, and theoretical research value [54]. Automated facial expression analysis has been conducted in numerous studies, especially in driver fatigue surveillance, medical treatment, sociable robots, and many human-computer interaction approaches. Based on a cross-cultural study, Ekman and Friesen [15] denoted six basic emotions (surprise, sadness, happiness, fear, disgust, and anger) regardless of culture. Subsequently, the incorporated emotion was contempt [35]. Advanced research in psychology and neuroscience argued that six basic emotions are not universal but culture-specific. Mase and Pentland devised a novel theory utilizing optimal flow technique to recognize facial expressions [34]. Since then, optical flow-based automated facial expression detection gained a lot of interest [28].

Feature extraction plays a crucial role in FER. These feature extraction methods can be categorized as statistical feature extraction, motion feature extraction, and deformation feature extraction methods [54]. Statistical feature extraction technique exploits the characteristics of expression of images by statistics, such as moment invariant or histogram. This method requires more time for a large amount of computing. and it ignores precise information about local-subtle features.

The deformation feature extraction technique is mainly used to extract some facial deformation information, such as texture changes or geometric deformation. The former refers to the textures' disappearance or appearance and modifications occurring due to changing expressions. The latter refers to the modified relative distance between feature points that occurred due to a variety of expressions.

The recognition based on geometric features has the following three advantages: less calculation or memory space, simple and easy recognition processing, and the feature needs minimal information about the illumination difference. No local-subtle features and incomplete facial information are the disadvantages of the method. Texture feature extraction has the disadvantage of processing huge amounts of computation. It has the advantage of containing expression information efficiently and is insensitive to individual differences and light intensity.

The motion feature extraction method is applied to derive some feature areas and feature points' motion information from sequential expression images, such as the direction of feature points and movement distance.

The common techniques include model methods, optical flow methods, and feature point tracking. The feature point tracking method implies the movement of feature points selected in the face feature region and obtaining parameters to achieve face recognition. The method used minimal computation to derive only part of the feature points, but it misses some valuable features.

Mase [34] applied optical flow to track the movement units. Optical flow focuses on facial deformation. The method can be easily affected by non-rigid facial movement and uneven illumination. The majority of the traditional studies applied shallow learning or handcrafted features.

Due to sufficient training data, enhanced chip processing abilities, and well-designed network architectures, many studies shifted to deep learning [26]. Deep learning techniques achieved state-of-the-art recognition accuracy. Deep learning approaches have some limita-

Table 3
Comparison of accuracy of our work with the state-of-the-art FER

Ref.	Dataset (Number of Images)	Cross-Validation Scheme	Method (Features + Classifier)	Accuracy (%)
[57] 2011	Cohn-Kanade (CK) database 1184)	10-Fold	3D Gabor features + SVM	94.48
[22] 2012	e Cohn–Kanade AU-Coded Facial Expression Database (348)	10-Fold	Local binary patterns, vertical time backward (VTB), and face moments + SVM	97
[5] 2017	Extended Cohen-Kanade (CK+) facial expression dataset (410)	10-Fold	Optical flow maps+ LIBSVM	93.17
[2] 2021	JAFFE dataset	10-Fold	Gabor Filter+ICA+ HMM	88
[36] 2021	Extended Cohen-Kanade (CK+) facial expression dataset (593)	70% for training and 30 % for testing	Deep Learning (CNN)	98
[46] 2023	Extended Cohen-Kanade (CK+) facial expression dataset (593)	10-Fold	Motion Vector features Deep Learning	95.3
Proposed Method	Extended Cohen-Kanade (CK+) facial expression dataset (593)	5-Fold	Extraction of the symmetricity of the face features using Optical flow algorithm	98

tions. First, a small training dataset may lead to overfitting. Moreover, high inter-subject variations exist for various personal attributes such as level of expressiveness, ethnic background, gender, and age. Apart from subject identity bias, variations in occlusions, illumination, and pose are usual in unconstrained facial expression scenarios. In Table 2, we presented the state-of-the-art comparison of our work.

From the Table 2, we can conclude that:

1. Most works are evaluated on the Cohen-Kanade/Extended Cohen-Kanade (CK+) facial expression datasets.
2. Almost, everybody has utilized the k-fold cross-validation scheme for validation and testing the model.
3. Our approach provided superior results compared to the other state-of-the-art approaches.
4. Though the work in [36] reported an accuracy equal to our work, they utilized a deep learning model. The major disadvantage of employing deep learning models is their computational complexity. In addition, we do not know which features are responsible for the best results. Besides, we found significant features with fewer numbers that reduce the computational complexity.

5. Conclusion and Future Work

Mental state and emotional state can be represented by facial expressions. A person's mental ability and consciousness can be perceived by facial expression recognition (FER). Thus, FER has been protruding physiological biometrics for identity authentication in numerous applications, for instance, law enforcement, access controls of laptop computers and mobile phones, video surveillance, public security, healthcare, marketing, finance, marketing, and many more. Automation of facial change analysis from the frontal view in general is the key to designing human-machine interfaces and automatic FER. In this research, a novel feature extraction method is devised that can be utilized in different models and applications. The highlight of our study is that it yields over 97% accuracy in the CK+ dataset by analyzing 15% of the face area. It employs an optical flow algorithm to exploit sequence data of facial expressions. High generalization is achieved because of the innovative face segmentation method and proposed symmetric merge of vectors. Two different methods of data preparation are introduced for non-time-series and time-series models. Our method exhibits superior performance in comparison to the state-of-the-art methods in the domain.

With the advancement of user-generated content and social media, users upload a huge amount of data on numerous platforms, such as video, audio, text, and images. Hence, multimodal sentiment analysis will be one of our future works. Additionally, the fusion of modalities like physiological data, depth information from 3D face models, and infrared images is becoming a promising research domain, and we also aspire to work in that area. In the future, we plan to explore the application of the proposed method in higher education to improve the quality of educational services and contribute to developing an effective ecosystem for digital education.

Appendix A

The pseudo-code for the mapping function is provided below:

Inputs:

- top_half_grid: a list of two integers [rows, cols] defining the size of the upper half grid
- bottom_half_grid: a list of two integers [rows, cols] defining the size of the bottom half grid
- i: an integer representing the section to map to a symmetric section

Outputs:

- sym_section: an integer representing the mapped symmetric section

Algorithm:

1 Calculate the total number of sections in the upper half grid and the bottom half grid

N_Upper <- top_half_grid[0] * top_half_grid[1]

N_Lower <- bottom_half_grid[0] * bottom_half_grid[1]

2 If i is less than N_Upper:

a. Set i_in_grid to i

b. Set cols to top_half_grid[1]

c. Set upper_rows_sections to the largest multiple of cols that is less than or equal to i_in_grid

d. Set i_in_row to i_in_grid modulo cols

Else:

a. Set i_in_grid to i - N_Upper

b. Set cols to bottom_half_grid[1]

c. Set upper_rows_sections to N_Upper + the largest multiple of cols that is less than or equal to i_in_grid

d. Set i_in_row to i_in_grid modulo cols

3 Set sym_in_row to cols - (i_in_row + 1)

4 Set sym_section to upper_rows_sections + sym_in_row + N_Upper + N_Lower

5 Return sym_section

Appendix B

The configuration for the LSTM model is as provided in Table 1.

Table 1
The configuration for the LSTM model

Layer	Name	Units	Output Shape	Regularization	Activation
1	Input Layer	-	25x226	-	-
2	Bidirectional LSTM	100	100	l1=0.0004, l2=0.001	tanh
3	Dropout	20%	100	-	-
4	Dense	7	7	l1=1e-5, l2=1e-3	softmax

The configuration for the FNN model is as provided in Table 2.

Table 2
The configuration for the FNN model

Layer	Name	Units	Output Shape	Regularization	Activation
1	Input Layer	-	113x2	-	-
2	Flatten	-	226	-	-
3	Dense	80	80	-	-
4	Dense	80	80	-	-
5	Dense	7	7	-	softmax

Appendix C

Confusion Matrices

A multi-class confusion matrix is a table that summarizes the performance of a multi-class classification model by displaying the number of actual and predicted instances for each class. In this specific case, the matrix represents the classification of seven different emotion categories: Anger (A), Disgust (D), Contempt (C), Fear (F), Happiness (H), Sadness (SD), and Surprise (SR). The matrix allows for an assessment of the model's performance for each class and provides a detailed breakdown of the number of true positives, false positives, true negatives, and false negatives predicted by the model.

In Table 1, the most correctly classified emotions are contempt and happiness while the most incorrectly classified emotion is fear with SVM classifier on selected features. Fear is misclassified as disgust and happiness while sadness is misclassified as anger.

Table 1
Confusion Matrix for SVM on Selected Features

	A	D	C	F	H	SD	SR
A	96	0	4	0	0	0	0
D	0	95	0	0	0	5	0
C	0	0	100	0	0	0	0
F	0	3.33	0	88.33	8.33	0	0
H	0	0	0	0	100	0	0
SD	10	0	0	0	0	90	0
SR	0	1.11	0	0	0	0	98.89

In Table 2, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is disgust with the XGBoost classifier on selected features. Disgust is misclassified as happiness while sadness is misclassified as anger.

In Table 3, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is fear with Random Forest classifier on selected features. Fear is misclassified as happiness and sadness while sadness is misclassified as anger.

Table 2
Confusion Matrix for XGBoost on Selected Features

	A	D	C	F	H	SD	SR
A	93	0	7	0	0	0	0
D	0	65	0	0	25	5	5
C	5	0	93.33	0	1.67	0	0
F	0	0	0	71.67	20	8.33	0
H	0	1.43	0	0	98.57	0	0
SD	15	0	0	3.33	0	78.33	3.33
SR	0	0	0	0	1.25	0	98.75

Table 3
Confusion Matrix for Random Forest on Selected Features

	A	D	C	F	H	SD	SR
A	91.5	0	4	2	0	2.5	0
D	0	90	0	0	5	5	0
C	3.33	0	95	1.67	0	0	0
F	0	0	0	85	8.33	6.67	0
H	0	0	0	0	100	0	0
SD	13.33	0	0	0	0	86.67	0
SR	0	1.25	0	0	0	0	98.75

In Table 4, the most correctly classified emotions are contempt and happiness while the most incorrectly classified emotions are disgust and fear with Bagging classifier on selected features. Disgust is misclassified as sadness while sadness is misclassified as anger.

Table 4
Confusion Matrix for Bagging on Selected Features

	A	D	C	F	H	SD	SR
A	96	0	4	0	0	0	0
D	0	90	0	0	0	10	0
C	0	0	100	0	0	0	0
F	3.33	0	0	90	6.67	0	0
H	0	0	0	0	100	0	0
SD	6.67	0	0	0	0	93.33	0
SR	0	1.11	0	0	0	0	98.89

Table 5
Confusion Matrix for FNN on Selected Features.

	A	D	C	F	H	SD	SR
A	95.5	0	4.5	0	0	0	0
D	0	95	0	0	0	5	0
C	0	0	100	0	0	0	0
F	0	0	0	96.67	3.33	0	0
H	0	0	0	0	100	0	0
SD	6.67	0	0	0	0	93.33	0
SR	0	1.25	0	0	0	0	98.75

In Table 5, the most correctly classified emotions are contempt and happiness while the most incorrectly classified emotion is sadness with FFN classifier on selected features. Anger is misclassified as contempt and sadness while sadness is misclassified as anger.

In Table 6, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is fear with LSTM classifier on selected features. Sadness is misclassified as anger, disgust, and fear while fear is misclassified as surprise.

Table 6
Confusion Matrix for LSTM on Selected Features

	A	D	C	F	H	SD	SR
A	97.5	0	2.5	0	0	0	0
D	0	95	0	0	0	5	0
C	3.33	0	95	0	1.67	0	0
F	0	3.33	0	78.33	10	3.33	5
H	0	0	0	0	100	0	0
SD	3.33	6.67	0	6.67	0	83.33	0
SR	0	1.25	0	0	1.11	0	97.64

In Table 7, the most correctly classified emotions are fear and happiness while the most incorrectly classified emotion is sadness with the SVM classifier having all the features. Sadness is misclassified as anger while disgust is misclassified as sadness.

Table 7
Confusion Matrix for SVM on All Features

	A	D	C	F	H	SD	SR
A	95.5	2.5	2	0	0	0	0
D	0	95	0	0	0	5	0
C	2	0	98	0	0	0	0
F	0	0	0	100	0	0	0
H	0	0	0	0	100	0	0
SD	11.67	0	0	0	0	88.33	0
SR	0	1.25	0	0	0	0	98.75

In Table 8, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is sadness with the Bagging classifier having all the features. Sadness is misclassified as anger while disgust is misclassified as happiness.

Table 8
Confusion Matrix for Bagging on All Features

	A	D	C	F	H	SD	SR
A	95.5	2	2.5	0	0	0	0
D	0	95	0	0	0	5	0
C	1.67	0	98.33	0	0	0	0
F	0	0	0	96.67	0	0	3.33
H	0	0	0	0	100	0	0
SD	10	0	0	0	0	90	0
SR	0	1.25	0	0	0	0	98.75

Table 9
Confusion Matrix for XGBoost on All Features

	A	D	C	F	H	SD	SR
A	93	0	5	0	0	2	0
D	5	75	0	0	15	5	0
C	3.33	0	95	1.67	0	0	0
F	0	0	0	68.33	8.33	10	13.33
H	0	0	0	0	100	0	0
SD	15	0	0	0	0	85	0
SR	0	0	0	0	1.25	0	98.75

In Table 9, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is fear with the XGBoost classifier having all the features. Fear is misclassified as a surprise while disgust is misclassified as happiness, sadness, and disgust.

Table 10
Confusion Matrix for Random Forest on All Features

	A	D	C	F	H	SD	SR
A	93	2.5	4.5	0	0	0	0
D	0	80	0	0	10	10	0
C	3.67	0	94.67	1.67	0	0	0
F	0	0	0	85	3.33	6.67	5
H	0	0	0	0	100	0	0
SD	11.67	0	0	0	0	88.33	0
SR	0	1.25	0	0	0	0	98.75

In Table 10, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is sadness with the Random Forest classifier having all features. Sadness is misclassified as anger while disgust is misclassified as happiness and sadness.

In Table 11, the most correctly classified emotions are contempt, fear, and happiness while the most incorrectly classified emotion is disgust with the FNN classifier having all the features. Sadness is misclassified as anger while disgust is misclassified as anger and sadness.

In Table 12, the most correctly classified emotions are surprise and happiness. In contrast, the most incorrectly classified emotions are anger and disgust with the LSTM classifier having all the features. Sadness is misclassified as anger while fear is misclassified as happiness and sadness.

Table 11
Confusion Matrix for FNN on All Features

	A	D	C	F	H	SD	SR
A	97.5	0	2.5	0	0	0	0
D	5	90	0	0	0	5	0
C	0	0	100	0	0	0	0
F	0	0	0	100	0	0	0
H	0	0	0	0	100	0	0
SD	8.33	0	0	0	0	91.67	0
SR	0	1.25	0	0	0	0	98.75

Table 12
Confusion Matrix for LSTM on All Features

	A	D	C	F	H	SD	SR
A	90	2.5	5	0	0	2.5	0
D	5	90	0	0	0	5	0
C	5	0	91.67	0	3.33	0	0
F	0	0	0	86.67	10	3.33	0
H	0	0	0	0	100	0	0
SD	8.33	0	0	0	0	91.67	0
SR	0	1.25	0	0	0	0	98.75

Acknowledgement

This paper is partly supported by the European Union-NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project № BG-RRP-2.004-0001-C01. The paper reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M. TensorFlow: A System for Large-Scale Machine Learning. In OSDI, Savannah, 2016, 265-283.
2. Abdulrahman, M., Eleyan, A. Facial Expression Recognition Using Support Vector Machines. In 2015 23rd Signal Processing and Communications Applications Conference (SIU), 2015, 276-279. https://doi.org/10.1109/SIU.2015.7129813
3. Akingbesote, D., Zhan, Y., Maskeliūnas, R., Damaševičius, R. Improving Accuracy of Face Recognition in the Era of Mask-Wearing: An Evaluation of a Pareto-Op-

timized FaceNet Model With Data Preprocessing Techniques. Algorithms, 2023, 16(6), 292. https://doi.org/10.3390/a16060292

4. Alenazy, W., Alqahtani, A. Gravitational Search Algorithm-Based Optimized Deep Learning Model With Diverse Set of Features for Facial Expression Recognition. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(2), 1631-1646. https://doi.org/10.1007/s12652-020-02235-0
5. Allaert, B., Bilasco, I., Djeraba, C. Consistent Optical Flow Maps for Full and Micro Facial Expression Recognition. In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2017, 235-242. https://doi.org/10.5220/0006127402350242
6. Arunnehru, J., Geetha, M. Automatic Human Emotion Recognition in Surveillance Video. In Intelligent Techniques in Signal Processing for Multimedia Security, N. Dey and V. Santhi, Eds., in Studies in Computational Intelligence, 2017, 660, 321-342. https://doi.org/10.1007/978-3-319-44790-2_15
7. Ayvaz, U., Gürüler, H., Devrim, M. Use of Facial Emotion Recognition in E-Learning Systems. ITLT, 2017, 60(4), 95-104. https://doi.org/10.33407/itlt.v60i4.1743
8. Bashyal, S., Venayagamoorthy, G. Recognition of Facial Expressions Using Gabor Wavelets and Learning Vector Quantization. Engineering Applications of Artificial Intelligence, 2008, 21(7), 1056-1064. https://doi.org/10.1016/j.engappai.2007.11.010
9. Breiman, L. Random Forests. Machine Learning, 2001, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
10. Chen, T., Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 785-794. https://doi.org/10.1145/2939672.2939785
11. Cortes, C., Vapnik, V. Support-Vector Networks. Mach. Learn., 1995, 20(3), 273-297. https://doi.org/10.1007/BF00994018
12. Demir, U., Ghaleb, E., Ekenel, H. A Face Recognition-Based Multiplayer Mobile Game Application. Lecture Notes in Computer Science, 2014, 8827, 214-223. https://doi.org/10.1007/978-3-662-44654-6_21
13. Dietterich, T. Ensemble Methods in Machine Learning. Lecture Notes in Computer Science, 2000, 1857, 1-15. https://doi.org/10.1007/3-540-45014-9_1
14. Efron, B. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association, 1983, 78(382), 316-331. https://doi.org/10.1080/01621459.1983.10477973
15. Ekman, P., Friesen, W. Constants Across Cultures in the Face and Emotion. Journal of Personality and Social Psychology, 1971, 17(2), 124-129. https://doi.org/10.1037/h0030377
16. Farnebäck, G. Two-Frame Motion Estimation Based on Polynomial Expansion. Lecture Notes in Computer Science, 2003, 2749, 363-370. https://doi.org/10.1007/3-540-45103-X_50
17. Fernandez, P., Pena, F., Ren, T., Cunha, A. FERAtt: Facial Expression Recognition With Attention Net. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, 837-846. https://doi.org/10.1109/CVPRW.2019.00112
18. Happy, S., Routray, A. Robust Facial Expression Classification Using Shape and Appearance Features. In 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), 2015, 1-5. https://doi.org/10.1109/ICAPR.2015.7050661
19. Hastie, T., Friedman, J., Tibshirani, R. The Elements of Statistical Learning. In Springer Series in Statistics, 2001. https://doi.org/10.1007/978-0-387-21606-5
20. Hochreiter, S., Schmidhuber, J. Long Short-Term Memory. Neural Computing, 1997, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
21. Horn, B., Schunck, B. Determining Optical Flow. Artificial Intelligence, 1981, 17(1-3), 185-203. https://doi.org/10.1016/0004-3702(81)90024-2
22. Ji, Y., Idrissi, K. Automatic Facial Expression Recognition Based on Spatiotemporal Descriptors. Pattern Recognition Letters, 2012, 33(10), 1373-1380. https://doi.org/10.1016/j.patrec.2012.03.006
23. Kingma, D., Ba, J. Adam: A Method for Stochastic Optimization. arXiv, 2017. DOI: 10.48550/arXiv.1412.6980.
24. Krason, A., Fenton, R., Varley, R., Vigliocco, G. The Role of Iconic Gestures and Mouth Movements in Face-to-Face Communication. Psychonomic Bulletin and Review, 2022, 29(2), 600-612. https://doi.org/10.3758/s13423-021-02009-5
25. Lebedeva, I., Ying, F., Guo, Y. Personalized Facial Beauty Assessment: A Meta-Learning Approach. The Visual Computer, 2023, 39(3), 1095-1107. https://doi.org/10.1007/s00371-021-02387-w
26. Li, S., Deng, W. Deep Facial Expression Recognition: A Survey. IEEE Trans. Affect. Comput., 2022, 13(3), 1195-1215. https://doi.org/10.1109/TAFFC.2020.2981446

27. Liaw, A., Wiener, M. Classification and Regression by Random Forest. R News, 2022, 2(3), 18-22, 2002.

28. Liu, H. Research Progress of Facial Expression Recognition Based on Deep Learning. In Proceedings of the 2020 Conference on Artificial Intelligence and Healthcare, 2020, 1-4. https://doi.org/10.1145/3433996.3434361

29. Lucey, P., Cohn, J., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I. 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, 2010, 94-101. https://doi.org/10.1109/CVPRW.2010.5543262

30. Magdin, M., Turcani, M., Hudec, L. Evaluating the Emotional State of a User Using a Webcam. International Journal of Interactive Multimedia and Artificial Intelligence, 2016, 4(1), 61. https://doi.org/10.9781/ijimai.2016.4112

31. Mahersia, H., Hamrouni, K. Using Multiple Steerable Filters and Bayesian Regularization for Facial Expression Recognition. Engineering Applications of Artificial Intelligence, 2015, 38, 190-202. https://doi.org/10.1016/j.engappai.2014.11.002

32. Malcolm, N., Gibson, J. The Perception of the Visual World. Philosophical Review, 1951, 60(4), 594. https://doi.org/10.2307/2181436

33. Maqableh, W., Alzyoud, F., Zraqou, J. The Use of Facial Expressions in Measuring Students' Interaction With Distance Learning Environments During the COVID-19 Crisis. Visual Informatics, 2023, 7(1), 1-17. https://doi.org/10.1016/j.visinf.2022.10.001

34. Mase, K. Recognition of Facial Expression From Optical Flow. IEICE Transactions on Information and Systems, 1991, E74-D(10), 3474-3483.

35. Matsumoto, D. More Evidence for the Universality of a Contempt Expression. Motivation and Emotion, 1992, 16(4), 363-368. https://doi.org/10.1007/BF00992972

36. Minaee, S., Minaei, M., Abdolrashidi, A. Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network. Sensors, 2021, 21(9), 3046. https://doi.org/10.3390/s21093046

37. Mufti, M., Khanam, A. Fuzzy Rule-Based Facial Expression Recognition. In 2006 International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), 2006, 57-57. https://doi.org/10.1109/CIMCA.2006.109

38. Naghsh-Nilchi, A., Roshanzamir, M. An Efficient Algorithm for Motion Detection-Based Facial Expression Recognition Using Optical Flow, 2008. DOI: 10.5281/ZENODO.1063056.

39. Nguyen, A., Hong, Y., Nguyen, P. Emotional Regulation in Synchronous Online Collaborative Learning: A Facial Expression Recognition Study. ICIS 2022 Proceedings, 2022, 12.

40. Noh, S., Park, H., Jin, Y., Park, J. Feature-Adaptive Motion Energy Analysis for Facial Expression Recognition. Lecture Notes in Computer Science, 2007, 4841, 452-463. https://doi.org/10.1007/978-3-540-76858-6_45

41. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res., 2011, 12, 2825-2830.

42. Qin, S., Zhu, Z., Zou, Y., Wang, X. Facial Expression Recognition Based on Gabor Wavelet Transform and 2-Channel CNN. International Journal of Wavelets, Multiresolution and Information Processing, 2020, 18(02), 2050003. https://doi.org/10.1142/S0219691320500034

43. Rahul, M., Shukla, R., Goyal, P., Siddiqui, Z., Yadav, V. Gabor Filter and ICA-Based Facial Expression Recognition Using Two-Layered Hidden Markov Model. Advances in Intelligent Systems and Computing, 2021, 1086, 511-518. https://doi.org/10.1007/978-981-15-1275-9_42

44. Rashid, M., Abu-Bakar, S., Mokji, M. Human Emotion Recognition From Videos Using Spatio-Temporal and Audio Features. The Visual Computer, 2013, 29(12), 1269-1275. https://doi.org/10.1007/s00371-012-0768-y

45. Rawal, N., Stock-Homburg, R. Facial Emotion Expressions in Human-Robot Interaction: A Survey. International Journal of Social Robotics, 2022, 14(7), 1583-1604. https://doi.org/10.1007/s12369-022-00867-0

46. Roshanzamir, M., Jafari, M., Alizadehsani, R., Roshanzamir, M., Shoeibi, A., Gorriz, J. M., Khosravi, A., Nahavandi, S., Acharya, U. R. What Happens in Face During a Facial Expression? Using Data Mining Techniques to Analyze Facial Expression Motion Vectors, 2021. DOI: 10.48550/ARXIV.2109.05457.

47. Salmam, F., Madani, A., Kissi, M. Facial Expression Recognition Using Decision Trees. In 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), 2016, 125-130. https://doi.org/10.1109/CGiV.2016.33

48. Saurav, S., Singh, S., Saini, R., Yadav, M. Facial Expression Recognition Using Improved Adaptive Local Ternary Pattern. Advances in Intelligent Systems and Computing, 2020, 1024, 39-52. https://doi.org/10.1007/978-981-32-9291-8_4

49. Sumalakshmi, C., Vasuki, P. Fused Deep Learning-Based Facial Expression Recognition of Students in Online Learning Mode. Concurrency and Computation: Practice and Experience, 2022, 34(21), e7137. https://doi.org/10.1002/cpe.7137

50. Susladkar, O., Deshmukh, G., Nag, S., Mantravadi, A., Makwana, D., Ravichandran, S., Chavhan, G. H., Mohan, C. K., Mittal, S. ClarifyNet: A High-Pass and Low-Pass Filtering-Based CNN for Single Image Dehazing. Journal of Systems Architecture, 2022, 132, 102736. https://doi.org/10.1016/j.sysarc.2022.102736

51. Tcherkassof, A., Dupré, D. The Emotion-Facial Expression Link: Evidence From Human and Automatic Expression Recognition. Psychological Research, 2021, 85(8), 2954-2969. https://doi.org/10.1007/s00426-020-01448-4

52. Vasanth, P. C., Nataraj, K. R. Facial Expression Recognition Using SVM Classifier. Indones. International Journal on Electrical Engineering and Informatics, IJEEI, 2015, 3(1), 16-20. https://doi.org/10.11591/ijeei.v3i1.126

53. Wei, W., Ho, E. S., McCay, K. D., Damaševičius, R., Maskeliūnas, R., Esposito, A. Assessing Facial Symmetry and Attractiveness Using Augmented Reality. Pattern Analysis and Applications, 2022, 1-17.

54. Wu, T., Fu, S., Yang, G. Survey of the Facial Expression Recognition Research. In BICS, Springer, 2012, 392-402. https://doi.org/10.1007/978-3-642-31561-9_44

55. Xu, X., Quan, C., Ren, F. Facial Expression Recognition Based on Gabor Wavelet Transform and Histogram of Oriented Gradients. In 2015 IEEE International Conference on Mechatronics and Automation (ICMA), 2015, 2117-2122. https://doi.org/10.1109/ICMA.2015.7237813

56. Yu, Z., Liu, G., Liu, Q., Deng, J. Spatio-Temporal Convolutional Features With Nested LSTM for Facial Expression Recognition. Neurocomputing, 2018, 317, 50-57. https://doi.org/10.1016/j.neucom.2018.07.028

57. Zhang, L., Tjondronegoro, D. Facial Expression Recognition Using Facial Movement Features. IEEE Transactions on Affective Computing, 2011, 2(4), 219-229. https://doi.org/10.1109/T-AFFC.2011.13

58. Zhang, Z., Li, Z., Liu, H., Cao, T., Liu, S. Data-Driven Online Learning Engagement Detection via Facial Expression and Mouse Behavior Recognition Technology. Journal of Educational Computing Research, 2020, 58(1), 63-86. https://doi.org/10.1177/0735633119825575