EasyChair Preprint

№ 10394

Optical-Flow Based Symmetric Feature Extraction for Facial Expression Recognition

Mohammad Ali Zeraatkar, Javad Hassannataj Joloudari, Kandala N V P S Rajesh, Silvia Gaftandzhieva and Sadiq Hussain

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

June 14, 2023

Optical-Flow Based Symmetric Feature Extraction for Facial Expression Recognition

Mohammad Ali Zeraatkar[1], Javad Hassannataj Joloudari[2,*], Kandala N V P S Rajesh[3], Silvia Gaftandzhieva[4,*], Sadiq Hussain[5]

[1]Department of Computer Engineering, Islamic Azad University, Tehran, Iran
[2]Department of Computer Engineering, Faculty of Engineering, University of Birjand, Birjand, Iran
[3]School of Electronics Engineering, VIT-AP University, Vijayawada, India
[4]Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", Plovdiv, Bulgaria
[5]Examination Branch, Dibrugarh University, Dibrugarh 786004, Assam, India

Corresponding authors[*]: javad.hassannataj@birjand.ac.ira and sissiy88@uni-plovdiv.bg

Abstract

Facial expression analysis is one of the most important 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 a challenge in this field. To address this issue this research presents an innovative and novel method to extract key features from a face during a facial expression fully automatically. These features can be used by various machine learning models to analyze emotions. We used the optical flow algorithm to extract motion vectors, which were then divided into sections on the subject's face. Finally, each section and its symmetric section were used to calculate a new vector. 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, and with an SVM classifier, we were able to keep the accuracy over 98 % by only looking at 10% 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 [1], incl. in the field of education to identifies the students' emotions during online learning sessions and help teachers change their teaching strategies in virtual learning environments and engage students[2], [3], [4], [5]. Although FER is very easy and intrinsic for humans, and some pieces of evidence show humans can easily recognize emotion even from different cultures [6], 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 is to design machine learning models which can 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 types of research in the field were focused on analyzing the importance of designing elegant and rich features; for example, Roshan Zamir et al. [7] investigated the areas of interest in the face used by classifiers such as C5.0, CRT, QUEST, CHAID, Deep Learning, and Discriminant algorithms, and showed the

eyes and mouth are the most influential parts. Nguyen et al. [8] explore the patterns of emotional regulation in collaborative learning, use Artificial Intelligence to examine learners' associated emotions and emotional synchrony in regulatory activities and propose 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 [6], [9]. So, in such systems, a camera captures the human face. The captured video [10], [11] or image [12] is analyzed by various techniques such as Gabor filters, local binary patterns (LBP), convolutional neural networks (CNN), and histogram of oriented gradients (HOGs) to interpret extreme and subtle [12], [13] emotions.

Researchers have proposed many solutions thus far, but maintaining high accuracy while being quick and spontaneous remains a challenge in this domain. 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:

a. Proposed a new feature extraction method based on the symmetricity of the face to extract the underlying facial information for face expression detection.
b. Obtained 98 % accuracy on the CK+ dataset [14] only by utilizing 10% of the face region to extract the features.
c. Novel feature selection and optical flow algorithm to utilize sequence data of facial expression results in robust performance.
d. High generalization because of innovative face segmentation method and proposed symmetric merge of vectors.
e. Two different approaches followed for the data preparation for time-series and non-time-series models
f. Our proposed approach outperforms the state-of-the-art.

This paper is organized as follows: Related works on the proposed objective are discussed in Section 2. The methodology of the proposed approach is presented in Section 3. Section 4 presents the results and discussion of our work. The future scope of the work and conclusions are given in Section 5.

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., 2011 [15] proposed a pose deduction by nose position in the picture, which was more a preprocessing approach than feature extraction, but the results were comparable. Ji and Idrissi, 2012 [16], 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 [17]. In 2008, Ahmad R et al. [18] 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, still, the optical flow has been wildly used along with many models. Recently, the work in [19] 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% accuracy for CASME2 datasets, respectively. Another method [7] used data mining to analyze facial expression by computing a new feature set named motion vector based on optical flow. The other feature that was exploited more is the local binary pattern (LBP) [20] is a texture descriptor for images. The work in [21] 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 recent works used it and proved its efficacy in FER. Happy and Routray

[20] explored the LBP along with PHOG and obtained 94% accuracy for the JAFFE dataset. Recently, Saurav et al [22] investigated LBP features with a bunch of other texture descriptors. Besides, there are many other features also explored by the researcher for FER. Vasanth P.C et al [23] used a Gabor filter along with the LBP feature and tried to classify using SVM. HOG is another texture descriptor made by a gradient filter on the edges. Xu et al. [24] gained 92% accuracy on the CK dataset using these Gabor filters and HOG features.

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.

2.2.1 Machine Learning Models

As facial expression is a classification problem, many classifiers have been used in the literature. Muzammil et al. [26] explored various parameters of the SVM, and the simulations are validated on the JAFFE dataset and obtained an accuracy of 87 % and 77 % on the MUFE dataset. Muid et al [25] developed a fuzzy logic-based method for FER. The approach achieved an accuracy of 81.22 %. Another fuzzy-based-FER is proposed by Liliana et al. The proposed method got 90% accuracy on Cohn–Kanade dataset. Rahul et al. [26] presented a FER using a probabilistic machine learning model, namely, hidden Markov models along 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. [27] 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. (2007) [28] used a simple ID3 classification tree algorithm on the JAFFE dataset and showed an accuracy of 75 %. Salmam et al. 2016 [29] used a simple Classification and Regression Tree (CART) model and reported an accuracy of 89.9% on the JAFFE dataset.

2.2.2 Deep Learning Models

The major limitation of the usage of 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 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 in for searching the best features manually. Qin et al. [30] 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 gain a lot of attention. Mahersia, H., 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, [32] used nested LSTM models by exploiting convolutional layers for FER. The recent popular DL methods are attention-based networks. Fernandez et al [33] proposed a FER method based on the attention model along with Gaussian space representation to learn multi-level features and got 90.3 % accuracy on the CK+ dataset. Minaee et al, [34] also employed this attention mechanism for FER. Alenazy et al [35] 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, like, overfitting and generalization.

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, which has been designed to meet these criteria, is visually represented in the Figure 1.

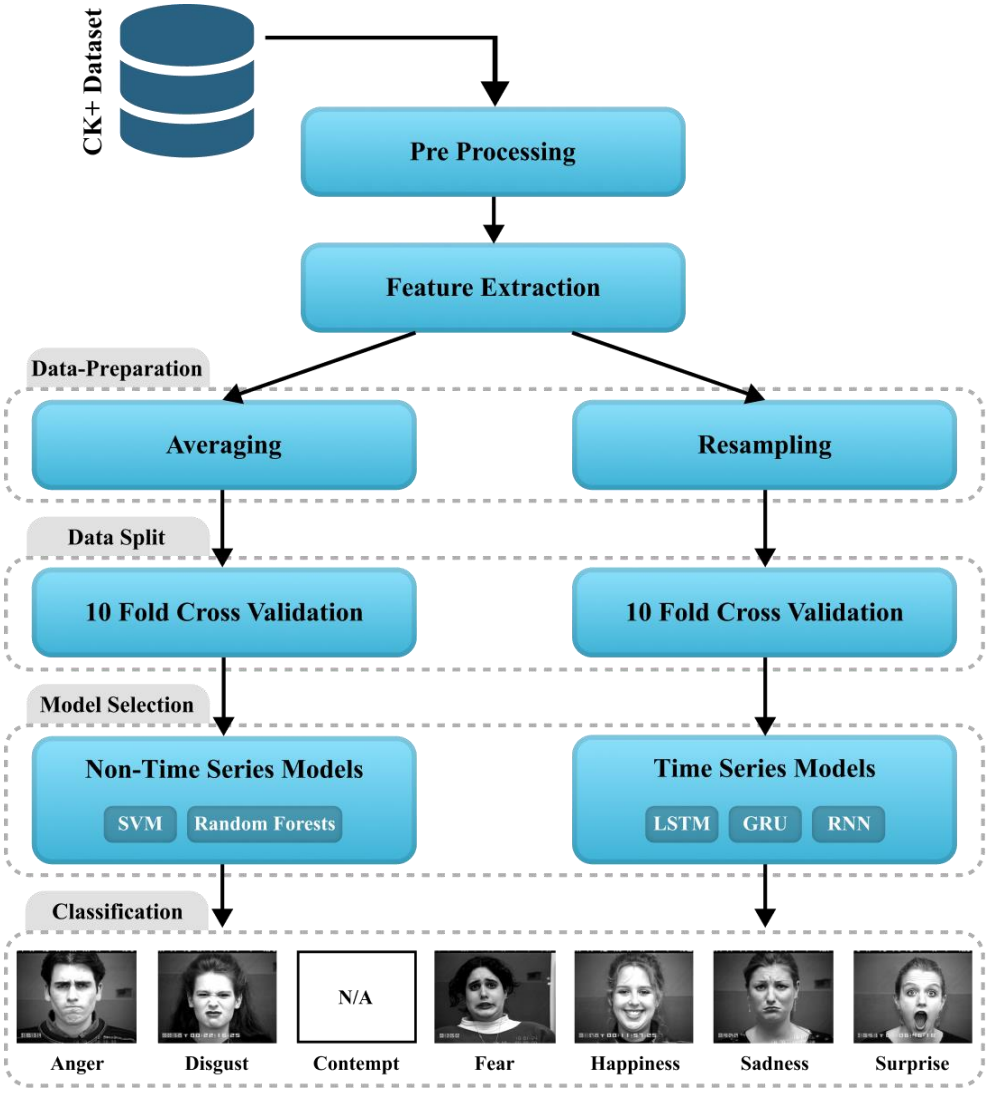

Figure 1. Schematic representation of the overall methodology used in our work.

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

In this section, we outline the pre-processing steps we took to improve the quality of our data before applying our feature extraction techniques. Specifically, we employed two pre-processing methods: high pass addition and face boundary & nose tip detection. High pass addition was applied to the facial expression image sequence to emphasize skin texture, enabling the optical flow algorithm to better track deformations in the 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.

3.1.1. High Pass Addition

In order 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 [36], 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 was done to enhance 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, the high-pass filter was used to isolate the high-frequency components of each frame, which primarily correspond to skin texture. We then combined the results of the filtering with the original frames to produce a new sequence that better highlights the skin texture. An illustration of this effect is provided in Figure 2, 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.

3.1.2. Face boundary & 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 in order to define the segments on the subject face. As there are many powerful methods for automatic facial boundary detection and it is not the focus of our work, In our experiments, we employed a pre-trained boundary detector model for this purpose and the nose tip position was extracted by the provided landmarks in the dataset.

3.2. Feature Extraction

Figure 3 depicts the steps for the feature extraction method. Each block in the diagram will be described in more depth below.

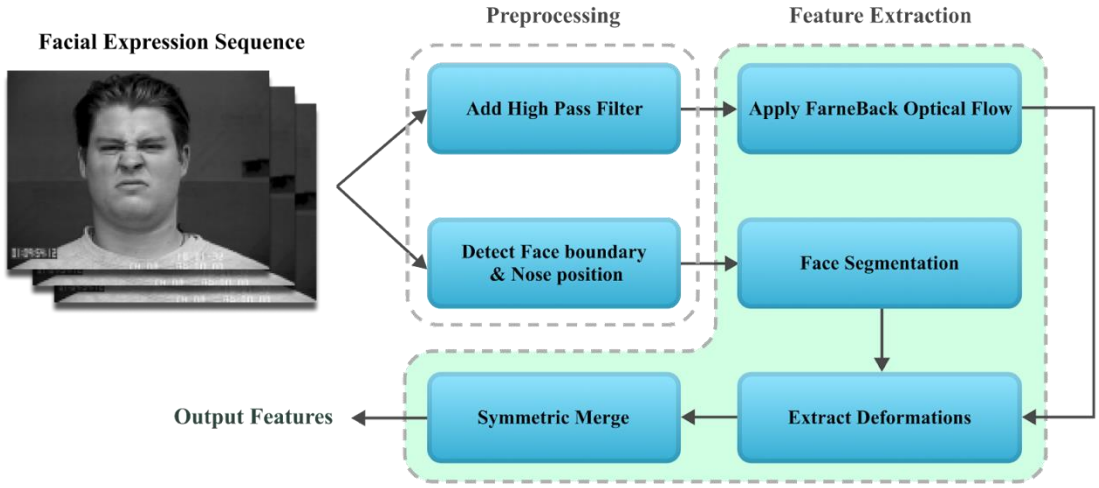

Figure 3. A diagram presenting an overview of the different feature extraction stages utilized in our work.

3.2.1. 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 James J. Gibson [37] in the 1950s and later developed by Berthold K.P. Horn and Brian G. Schunck in the 1980s [38]. 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 particularly useful in various applications, such as video compression, motion analysis, and facial expression recognition.

The Farneback method, proposed by Gunnar Farneback in 2003 [39], is an efficient algorithm for estimating optical flow. It is 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 x, y are spatial coordinates and t is the time, the optical flow equation can be written as:

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

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:

$$(2)\ \ IxVx + IyVy = -It$$

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 first 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 estimation of the optical flow. 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 vectors sequence, as demonstrated in Figure 4.

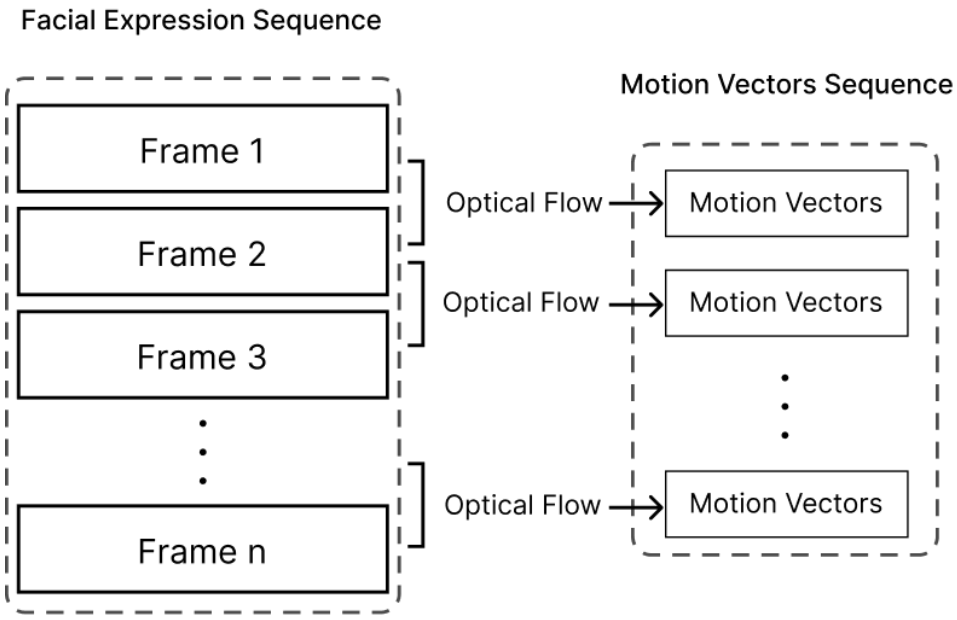

Figure 4. A diagram showing how optical flow is applied on facial expression sequence frames to build a motion vectors sequence.

The optical flow algorithm is applied to the entire frame border in order to simplify the proposal of our solution. However, in order 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.

3.2.2. Face Segmentation

In order to improve generalization and minimize computational complexity we need to take an average of motion vectors on the face, so an innovative facial segmentation method is performed to define areas on the face which will be used to extract a mean vector by averaging motion vectors in them.

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

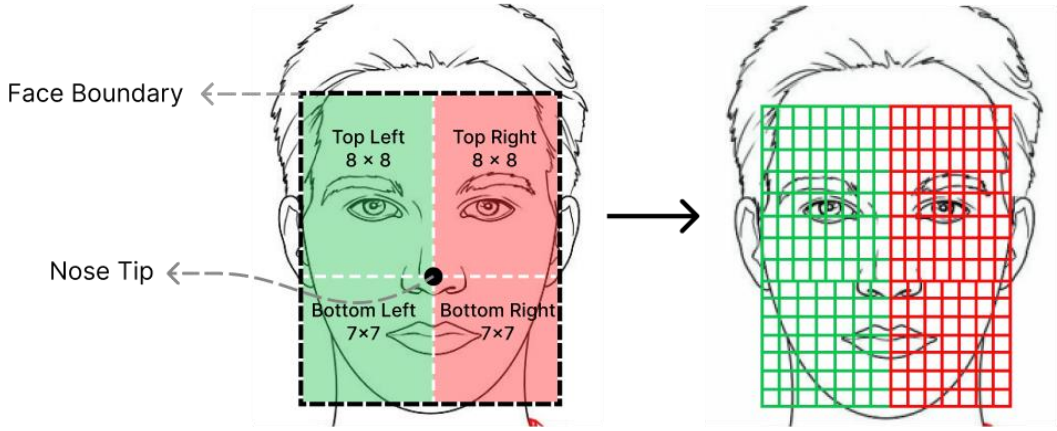

Figure 5. An example of Facial Area Segmentation and Grid Configuration

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

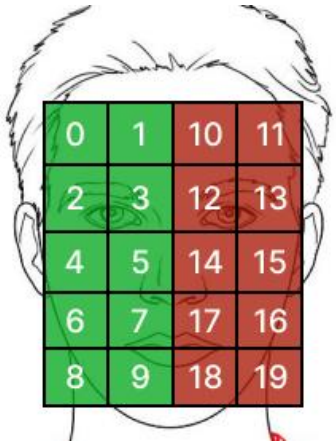

Figure 6. An example of Facial Area Segmentation and Grid Configuration

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.

The equation below shows how we mapped every segment in the left part to its symmetric section in the right

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

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.

3.2.3. Extract Deformations & Symmetric merge

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

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

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

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

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 canceling each other out.

3.3. Data Preparation

In this section, we outline the steps we took to prepare our data for use in our models. Specifically, we provide two distinct methods for preparing data, one for time-series models and the other for non-time-series models. For time-series models, we employed resampling techniques 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.

3.3.1. Resampling

Before going to train models by the features which were extracted from dataset samples, we need to perform some normalization techniques.
Since facial expression samples in the CK+ dataset, are not of the same length, we need to perform a resampling technique in order 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:

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

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

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.

3.3.2. Averaging

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

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

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) [50] 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 it is 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.

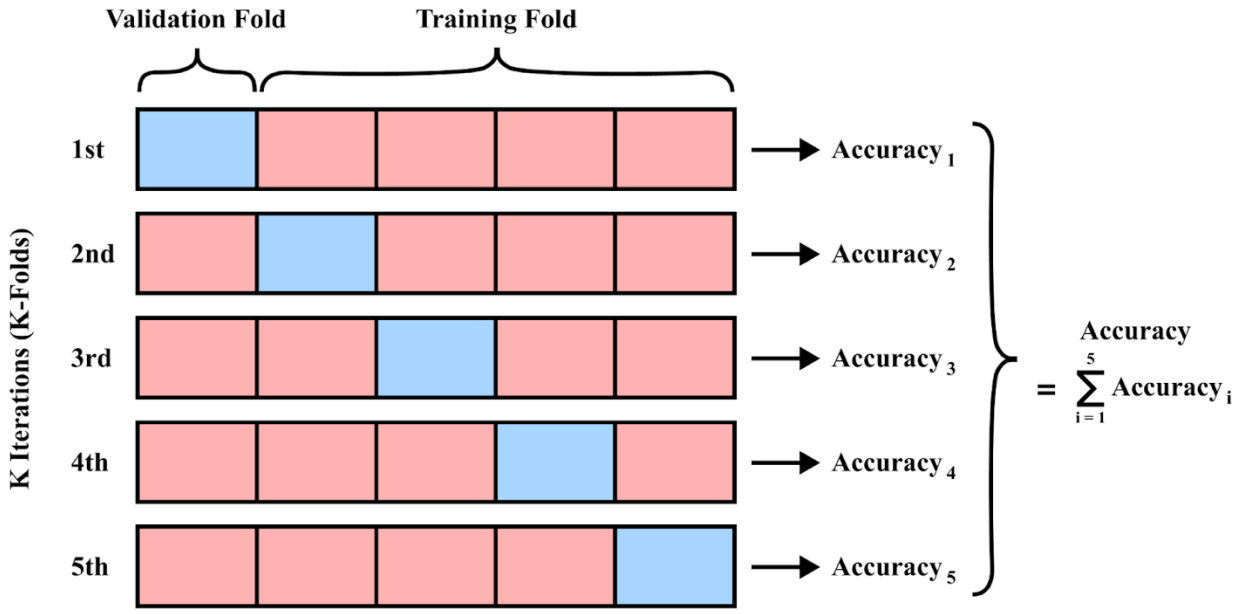

Figure 7. Illustration of k-fold cross-validation for evaluating model performance

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

3.5. Model Selection

Different types of models were used to evaluate our work which will be explained in depth below.
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 was used for time-series data.

3.5.1. Support Vector Machines

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

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

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.[41]

3.5.2. Random Forest

Random forest (RF) is a powerful and versatile ensemble learning method that can be used for both classification and regression tasks. It was first introduced by Leo Breiman in 2001 [42] 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 to reduce the correlation between individual trees, which in turn reduces the overall variance of the model [43].
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 [44].
Formally, the Gini impurity for a node can be calculated as:

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

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

3.5.3. XGBoost

XGBoost [45], 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.

3.5.4. Ensemble learning

Ensemble learning is a machine learning technique where multiple models are combined to improve predictive performance. It helps to reduce overfitting, increase robustness and has been successful in various applications. A popular example of ensemble learning is the Random Forest algorithm, which combines multiple decision trees to create a more accurate predictor. [46]

3.5.5. 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 forward 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.

3.5.6. 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 selectively 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 various benchmarks [47]. The configuration that we used for this model is provided in Appendix B.

3.6. Reducing Feature Dimensionality

In order to maximize the computational efficiency of our work we decided to select the most important features.
To do that, we used a measure of feature importance calculated during the training of the random forest model. Random forest are a tree-based model commonly used 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 important features for our analysis.

Both the results obtained with all features and the selected features are provided in the results section. 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 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 [48] and employed the Adam optimizer [49]. In contrast, the other models in this study were implemented using the scikit-learn library [50]. This experimental environment provided the necessary computational resources and tools to effectively test and evaluate our proposed feature extraction technique for facial expression classification.

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 performance of the classifier. 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.

4.1.1. Accuracy

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

$$(11)\ Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ Number\ of\ predictions}$$

Table 1 provides a comparison of 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 the use of selected features in FNN yields competitive accuracies, with only a marginal decrease of 0.32% compared to using all features. This suggests that we can 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	SVM	XGBoost	Random Forest	Bagging (SVM)	FNN	LSTM
Accuracy All Features	97.53%	93.26%	94.49%	97.24%	**98.17%**	94.50%
Accuracy Selected Features	96.94%	91.38%	94.49%	97.25%	**97.85%**	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 be used to inform the development of more accurate emotion recognition models, as it highlights the specific facial features that are most important for recognizing certain emotions. The significance of different facial regions for emotion recognition using a Bagging SVM classifier is shown in Figure 8.

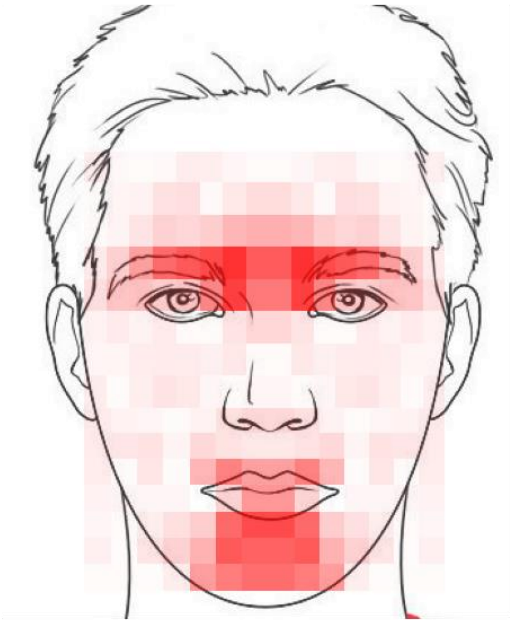

Figure 8. Heatmap illustrating the significance of different facial regions for emotion recognition using a Bagging SVM classifier.

4.3. 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 [51]. Automated facial expression analysis has been conducted in numerous studies, especially in the field of driver fatigue surveillance, medical treatment, sociable robots, and many human-computer interaction approaches. Based on a cross-cultural study, Ekman and Friesen [52] denoted six basic emotions as surprise, sadness, happiness, fear, disgust, and anger regardless of culture. Subsequently incorporated emotion was contempt [53]. 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 [54]. Since then, optical flow-based automated facial expression detection gained a lot of interest [55].

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 [51]. 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 that occurred 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 advantages: 1. less calculation or memory space; 2. simple and easy recognition processing and 3. 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 disadvantages of processing huge amounts of computation while it has the advantages 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 that are 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 [54] applied optical flow to track the movement units. Optical flow focuses on facial deformation. The method is easy to be 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 and enhanced chip processing abilities and well-designed network architectures, many studies shifted to deep learning [13]. Deep learning techniques achieved state-of-the-art recognition accuracy. Deep learning approaches have some limitations. 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.

Table 2. Comparison of accuracy of our work with the state-of-the-art FER.

Reference No and Year	Dataset (Number of Images)	Cross-Validation Scheme	Method (Features + Classifier)	Accuracy (%)
[10] 2011	Cohn-Kanade (CK) database 1184)	10-Fold	3D Gabor features + SVM	94.48
[11] 2012	e Cohn–Kanade AU-Coded Facial Expression Database (348)	10-Fold	Local binary patterns, vertical time backward (VTB) and face moments + SVM	97
[14] 2017	Extended Cohen-Kanade (CK+) facial expression dataset (410)	10-Fold	Optical flow maps+ LIBSVM	93.17
[21] 2021	JAFFE dataset	10-Fold	Gabor Filter+ICA+ HMM	88
[29] 2021	Extended Cohen-Kanade (CK+) facial expression dataset (593)	70% for training and 30 % for testing	Deep Learning (CNN)	98
[3] 2021	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

From the above table, the following things can be understood.

(i) Most of the works are evaluated on the Cohen-Kanade/ Extended Cohen-Kanade (CK+) facial expression datasets.

(ii) Almost, everybody has utilized the k-fold cross-validation scheme for validation and testing the model.

(iii) From the results, it is evident that our approach provided superior results compared to the other state-of-the approaches.

(iv) Though the work in [34] 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. Also, we don't 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 98% accuracy in CK+ dataset by analyzing 10% 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, a huge amount of data is uploaded by users on numerous platforms, such as video, audio, text, and image. 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 the development of an effective ecosystem for digital education.

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] M. Magdin, M. Turcani, and L. Hudec, "Evaluating the Emotional State of a User Using a Webcam," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 1, p. 61, 2016, doi: 10.9781/ijimai.2016.4112.

[2] U. Ayvaz, H. Gürüler, and M. Devrim, "Use of facial emotion recognition in e-learning systems". *ITLT*, vol. 60, no. 4, 2017, 95 – 104

[3] Z. Zhang, Z. Li, H. Liu, T. Cao, and S. Liu, "Data-driven online learning engagement detection via facial expression and mouse behavior recognition technology". *Journal of Educational Computing Research*, 58(1), 2020, 63-86.

[4] C. Sumalakshmi and P. Vasuki, "Fused deep learning based Facial Expression Recognition of students in online learning mode". *Concurrency and Computation: Practice and Experience*, 34(21), 2022, e7137.

[5] W. Maqableh, F. Alzyoud, and J. Zraqou, "The use of facial expressions in measuring students' interaction with distance learning environments during the COVID-19 crisis". *Visual informatics*, 7(1), 2023, 1-17.

[6] A. Tcherkassof and D. Dupré, "The emotion–facial expression link: evidence from human and automatic expression recognition," *Psychol. Res.*, vol. 85, no. 8, pp. 2954–2969, Nov. 2021, doi: 10.1007/s00426-020-01448-4.

[7] M. Roshanzamir *et al.*, "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.

[8] A. Nguyen, Y. Hong, and P. Nguyen, Emotional Regulation in Synchronous Online Collaborative Learning: A Facial Expression Recognition Study, *ICIS 2022 Proceedings*, 2022, 12.

[9] A. Krason, R. Fenton, R. Varley, and G. Vigliocco, "The role of iconic gestures and mouth movements in face-to-face communication," *Psychon. Bull. Rev.*, vol. 29, no. 2, pp. 600–612, Apr. 2022, doi: 10.3758/s13423-021-02009-5.

[10] M. Rashid, S. A. R. Abu-Bakar, and M. Mokji, "Human emotion recognition from videos using spatio-temporal and audio features," *Vis. Comput.*, vol. 29, no. 12, pp. 1269–1275, Dec. 2013, doi: 10.1007/s00371-012-0768-y.

[11] J. Arunnehru and M. Kalaiselvi Geetha, "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, vol. 660. Cham: Springer International Publishing, 2017, pp. 321–342. doi: 10.1007/978-3-319-44790-2_15.

[12] N. Rawal and R. M. Stock-Homburg, "Facial Emotion Expressions in Human–Robot Interaction: A Survey," *Int. J. Soc. Robot.*, vol. 14, no. 7, pp. 1583–1604, Sep. 2022, doi: 10.1007/s12369-022-00867-0.

[13] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1195–1215, Jul. 2022, doi: 10.1109/TAFFC.2020.2981446.

[14] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, San Francisco, CA, USA: IEEE, Jun. 2010, pp. 94–101. doi: 10.1109/CVPRW.2010.5543262.

[15]L. Zhang and D. Tjondronegoro, “Facial expression recognition using facial movement features,” *IEEE Trans. Affect. Comput.*, vol. 2, no. 4, pp. 219–229, 2011.
[16]Y. Ji and K. Idrissi, “Automatic facial expression recognition based on spatiotemporal descriptors,” *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1373–1380, Jul. 2012, doi: 10.1016/j.patrec.2012.03.006.
[17]U. Demir, E. Ghaleb, and H. K. Ekenel, “A Face Recognition Based Multiplayer Mobile Game Application,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochano and E. Hancock, Eds., in Lecture Notes in Computer Science, vol. 8827. Cham: Springer International Publishing, 2014, pp. 214–223. doi: 10.1007/978-3-662-44654-6_21.
[18]A. R. Naghsh-Nilchi and M. Roshanzamir, “An Efficient Algorithm For Motion Detection Based Facial Expression Recognition Using Optical Flow,” Aug. 2008, doi: 10.5281/ZENODO.1063056.
[19]B. Allaert, I. M. Bilasco, and C. Djeraba, “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*, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 235–242. doi: 10.5220/0006127402350242.
[20]S. L. Happy and A. Routray, “Robust facial expression classification using shape and appearance features,” in *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, Kolkata, India: IEEE, Jan. 2015, pp. 1–5. doi: 10.1109/ICAPR.2015.7050661.
[21]M. Abdulrahman and A. Eleyan, “Facial expression recognition using Support Vector Machines,” in *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, Malatya, Turkey: IEEE, May 2015, pp. 276–279. doi: 10.1109/SIU.2015.7129813.
[22]S. Saurav, S. Singh, R. Saini, and M. Yadav, “Facial Expression Recognition Using Improved Adaptive Local Ternary Pattern,” in *Proceedings of 3rd International Conference on Computer Vision and Image Processing*, B. B. Chaudhuri, M. Nakagawa, P. Khanna, and S. Kumar, Eds., in Advances in Intelligent Systems and Computing, vol. 1024. Singapore: Springer Singapore, 2020, pp. 39–52. doi: 10.1007/978-981-32-9291-8_4.
[23]V. P.C. and N. K.R., “Facial Expression Recognition Using SVM Classifier,” *Indones. J. Electr. Eng. Inform. IJEEI*, vol. 3, no. 1, pp. 16–20, Mar. 2015, doi: 10.11591/ijeei.v3i1.126.
[24]X. Xu, C. Quan, and F. Ren, “Facial expression recognition based on Gabor Wavelet transform and Histogram of Oriented Gradients,” in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, China: IEEE, Aug. 2015, pp. 2117–2122. doi: 10.1109/ICMA.2015.7237813.
[25]M. Mufti and A. Khanam, “Fuzzy Rule Based Facial Expression Recognition,” in *2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA’06)*, Sydney, Australia: IEEE, 2006, pp. 57–57. doi: 10.1109/CIMCA.2006.109.
[26]M. Rahul, R. Shukla, P. K. Goyal, Z. A. Siddiqui, and V. Yadav, “Gabor Filter and ICA-Based Facial Expression Recognition Using Two-Layered Hidden Markov Model,” in *Advances in Computational Intelligence and Communication Technology*, X.-Z. Gao, S. Tiwari, M. C. Trivedi, and K. K. Mishra, Eds., in Advances in Intelligent Systems and Computing, vol. 1086. Singapore: Springer Singapore, 2021, pp. 511–518. doi: 10.1007/978-981-15-1275-9_42.
[27]S. Bashyal and G. K. Venayagamoorthy, “Recognition of facial expressions using Gabor wavelets and learning vector quantization,” *Eng. Appl. Artif. Intell.*, vol. 21, no. 7, pp. 1056–1064, Oct. 2008, doi: 10.1016/j.engappai.2007.11.010.
[28]S. Noh, H. Park, Y. Jin, and J.-I. Park, “Feature-Adaptive Motion Energy Analysis for Facial Expression Recognition,” in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, N. Paragios, S.-M. Tanveer, T. Ju, Z. Liu, S. Coquillart, C. Cruz-Neira, T. Müller, and T. Malzbender, Eds., in Lecture Notes in Computer Science, vol. 4841. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 452–463. doi: 10.1007/978-3-540-76858-6_45.
[29]F. Z. Salmam, A. Madani, and M. Kissi, “Facial Expression Recognition Using Decision Trees,” in *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, Beni Mellal, Morocco: IEEE, Mar. 2016, pp. 125–130. doi: 10.1109/CGiV.2016.33.
[30]S. Qin, Z. Zhu, Y. Zou, and X. Wang, “Facial expression recognition based on Gabor wavelet transform and 2-channel CNN,” *Int. J. Wavelets Multiresolution Inf. Process.*, vol. 18, no. 02, p. 2050003, Mar. 2020, doi: 10.1142/S0219691320500034.
[31]H. Mahersia and K. Hamrouni, “Using multiple steerable filters and Bayesian regularization for facial expression recognition,” *Eng. Appl. Artif. Intell.*, vol. 38, pp. 190–202, Feb. 2015, doi: 10.1016/j.engappai.2014.11.002.
[32]Z. Yu, G. Liu, Q. Liu, and J. Deng, “Spatio-temporal convolutional features with nested LSTM for facial expression recognition,” *Neurocomputing*, vol. 317, pp. 50–57, Nov. 2018, doi: 10.1016/j.neucom.2018.07.028.
[33]P. D. M. Fernandez, F. A. G. Pena, T. I. Ren, and A. Cunha, “FERAtt: Facial Expression Recognition With Attention Net,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 837–846. doi: 10.1109/CVPRW.2019.00112.

[34]S. Minaee, M. Minaei, and A. Abdolrashidi, “Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network,” *Sensors*, vol. 21, no. 9, p. 3046, Apr. 2021, doi: 10.3390/s21093046.
[35]W. M. Alenazy and A. S. Alqahtani, “Gravitational search algorithm based optimized deep learning model with diverse set of features for facial expression recognition,” *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 2, pp. 1631–1646, Feb. 2021, doi: 10.1007/s12652-020-02235-0.
[36]O. Susladkar *et al.*, “ClarifyNet: A high-pass and low-pass filtering based CNN for single image dehazing,” *J. Syst. Archit.*, vol. 132, p. 102736, Nov. 2022, doi: 10.1016/j.sysarc.2022.102736.
[37]N. Malcolm and J. J. Gibson, “The Perception of the Visual World.,” *Philos. Rev.*, vol. 60, no. 4, p. 594, Oct. 1951, doi: 10.2307/2181436.
[38]B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, Aug. 1981, doi: 10.1016/0004-3702(81)90024-2.
[39]G. Farnebäck, “Two-Frame Motion Estimation Based on Polynomial Expansion,” in *Image Analysis*, J. Bigun and T. Gustavsson, Eds., in Lecture Notes in Computer Science, vol. 2749. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370. doi: 10.1007/3-540-45103-X_50.
[40]B. Efron, “Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation,” *J. Am. Stat. Assoc.*, vol. 78, no. 382, pp. 316–331, Jun. 1983, doi: 10.1080/01621459.1983.10477973.
[41]C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
[42]L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[43]A. Liaw, M. Wiener, and others, “Classification and regression by randomForest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[44]T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning*. in Springer Series in Statistics. New York, NY: Springer, 2001. doi: 10.1007/978-0-387-21606-5.
[45]T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
[46]T. G. Dietterich, “Ensemble Methods in Machine Learning,” in *Multiple Classifier Systems*, in Lecture Notes in Computer Science, vol. 1857. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15. doi: 10.1007/3-540-45014-9_1.
[47]S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[48]M. Abadi *et al.*, “Tensorflow: a system for large-scale machine learning.,” in *Osdi*, Savannah, GA, USA, 2016, pp. 265–283.
[49]D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. doi: 10.48550/arXiv.1412.6980.
[50]F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[51]T. Wu, S. Fu, and G. Yang, “Survey of the Facial Expression Recognition Research.,” in *BICS*, Springer, 2012, pp. 392–402.
[52]P. Ekman and W. V. Friesen, “Constants across cultures in the face and emotion.,” *J. Pers. Soc. Psychol.*, vol. 17, no. 2, pp. 124–129, 1971, doi: 10.1037/h0030377.
[53]D. Matsumoto, “More evidence for the universality of a contempt expression,” *Motiv. Emot.*, vol. 16, no. 4, pp. 363–368, Dec. 1992, doi: 10.1007/BF00992972.
[54]K. Mase, “Recognition of Facial Expression from Optical Flow,” *IEICE Trans. Inf. Syst.*, vol. E74-D, no. 10, pp. 3474–3483, Oct. 1991.
[55]H. Liu, “Research Progress of Facial Expression Recognition Based on Deep Learning,” in *Proceedings of the 2020 Conference on Artificial Intelligence and Healthcare*, Taiyuan China: ACM, Oct. 2020, pp. 1–4. doi: 10.1145/3433996.3434361.

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 LSTM model is as below:

LSTM					
Layer Num	**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 FNN model is as below:

FNN					
Layer Num	**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, Disgust, Contempt, Fear, Happiness, Sadness, and Surprise. The matrix allows for an assessment of the model's performance for each individual class and provides a detailed breakdown of the number of true positives, false positives, true negatives, and false negatives predicted by the model.

Confusion Matrix for SVM on Selected Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	96	0	4	0	0	0	0
Disgust	0	95	0	0	0	5	0
Contempt	0	0	100	0	0	0	0
Fear	0	3.33	0	88.33	8.33	0	0
Happiness	0	0	0	0	100	0	0
Sadness	10	0	0	0	0	90	0
Surprise	0	1.11	0	0	0	0	98.89

In the table depicted above, 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.
Confusion Matrix for XGBoost on Selected Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	93	0	7	0	0	0	0
Disgust	0	65	0	0	25	5	5
Contempt	5	0	93.33	0	1.67	0	0
Fear	0	0	0	71.67	20	8.33	0
Happiness	0	1.43	0	0	98.57	0	0
Sadness	15	0	0	3.33	0	78.33	3.33
Surprise	0	0	0	0	1.25	0	98.75

In the table depicted above, 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.

Confusion Matrix for Random Forest on Selected Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	91.5	0	4	2	0	2.5	0
Disgust	0	90	0	0	5	5	0
Contempt	3.33	0	95	1.67	0	0	0
Fear	0	0	0	85	8.33	6.67	0
Happiness	0	0	0	0	100	0	0
Sadness	13.33	0	0	0	0	86.67	0
Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, 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.

Confusion Matrix for Bagging on Selected Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	96	0	4	0	0	0	0
Disgust	0	90	0	0	0	10	0
Contempt	0	0	100	0	0	0	0
Fear	3.33	0	0	90	6.67	0	0
Happiness	0	0	0	0	100	0	0
Sadness	6.67	0	0	0	0	93.33	0
Surprise	0	1.11	0	0	0	0	98.89

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

Confusion Matrix for FNN on Selected Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	95.5	0	4.5	0	0	0	0
Disgust	0	95	0	0	0	5	0
Contempt	0	0	100	0	0	0	0
Fear	0	0	0	96.67	3.33	0	0
Happiness	0	0	0	0	100	0	0
Sadness	6.67	0	0	0	0	93.33	0
Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, 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.

Confusion Matrix for LSTM on Selected Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	97.5	0	2.5	0	0	0	0
Disgust	0	95	0	0	0	5	0
Contempt	3.33	0	95	0	1.67	0	0
Fear	0	3.33	0	78.33	10	3.33	5
Happiness	0	0	0	0	100	0	0
Sadness	3.33	6.67	0	6.67	0	83.33	0
Surprise	0	1.25	0	0	1.11	0	97.64

In the table depicted above, 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.

Confusion Matrix for SVM On All Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	95.5	2.5	2	0	0	0	0
Disgust	0	95	0	0	0	5	0
Contempt	2	0	98	0	0	0	0
Fear	0	0	0	100	0	0	0
Happiness	0	0	0	0	100	0	0
Sadness	11.67	0	0	0	0	88.33	0
Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, 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.

Confusion Matrix for Bagging on All Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	95.5	2	2.5	0	0	0	0
Disgust	0	95	0	0	0	5	0
Contempt	1.67	0	98.33	0	0	0	0
Fear	0	0	0	96.67	0	0	3.33
Happiness	0	0	0	0	100	0	0
Sadness	10	0	0	0	0	90	0

Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, 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.

Confusion Matrix for XGBoost on All Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	93	0	5	0	0	2	0
Disgust	5	75	0	0	15	5	0
Contempt	3.33	0	95	1.67	0	0	0
Fear	0	0	0	68.33	8.33	10	13.33
Happiness	0	0	0	0	100	0	0
Sadness	15	0	0	0	0	85	0
Surprise	0	0	0	0	1.25	0	98.75

In the table depicted above, 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.

Confusion Matrix for Random Forest on All Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	93	2.5	4.5	0	0	0	0
Disgust	0	80	0	0	10	10	0
Contempt	3.67	0	94.67	1.67	0	0	0
Fear	0	0	0	85	3.33	6.67	5
Happiness	0	0	0	0	100	0	0
Sadness	11.67	0	0	0	0	88.33	0
Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, 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.

Confusion Matrix for FNN on All Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	97.5	0	2.5	0	0	0	0
Disgust	5	90	0	0	0	5	0
Contempt	0	0	100	0	0	0	0
Fear	0	0	0	100	0	0	0

Happiness	0	0	0	0	100	0	0
Sadness	8.33	0	0	0	0	91.67	0
Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, 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.

Confusion Matrix for LSTM on All Features:

	Anger	Disgust	Contempt	Fear	Happiness	Sadness	Surprise
Anger	90	2.5	5	0	0	2.5	0
Disgust	5	90	0	0	0	5	0
Contempt	5	0	91.67	0	3.33	0	0
Fear	0	0	0	86.67	10	3.33	0
Happiness	0	0	0	0	100	0	0
Sadness	8.33	0	0	0	0	91.67	0
Surprise	0	1.25	0	0	0	0	98.75

In the table depicted above, the most correctly classified emotions are surprise and happiness while the most incorrectly classified emotion is anger and disgust with fthe LSTM classifier having all the features. Sadness is misclassified as anger while fear is misclassified as happiness and sadness.