A COMPARATIVE STUDY OF MACHINE LEARNING AND DEEP LEARNING TECHNIQUES FOR FACIAL EMOTION

Pravallika Dannana[1], A.S. Venkata Praneel[2]
[1,2] Department of Computer Science and Engineering, GST, GITAM (Deemed to be University), Visakhapatnam - 530045

Keywords:
Emotion Classification
Facial Expression Analysis
Deep Learning Networks

ABSTRACT
Facial expressions are essential in human contact because they serve as indicators of intentions, internal emotional states, and social messages. This work performs a thorough comparative analysis of face emotion identification methods using deep learning and machine learning. The classification of emotions exhibited by participants in the FER2013 and JAFFE datasets is studied using a variety of techniques, including Principal Component Analysis (PCA), Random Forests, Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). The study explores the effects of different PCA component counts on model performance and offers insights into the characteristics that each model learns. By means of thorough training and assessment of both datasets, the accuracy levels attained by distinct models are contrasted. Finally, based on the comparison analysis's insights and the observed findings, the research suggests the best model for face emotion identification.

Corresponding Author: Email: praneelsri@gmail.com

INTRODUCTION

In the last ten years, the field of deep learning and machine learning has paid close attention to facial expression interpretation. Real-time applications for emotion recognition include crowd analytics, surveillance, and human-computer interaction. Facial expressions are the most natural way to convey and deduce the intentions of another. They are essential to our regular social contacts. We are able to quickly read the facial expressions of those around us. Through facial emotion analysis, a machine may interpret our intents from our facial expressions and facilitate human-computer connection. Three processes make up a standard facial emotion recognition model, as described in [1]: face detection, face localization, and expression information extraction from identified faces. Ultimately, the retrieved data is used to train a classifier (such as CNN) to provide the final expression labels. The seven primary expressions that are common to all civilizations are: fear, anger, sadness, surprise, happiness, and disgust. A “message-based" approach and a "sign-based" approach are the two conceptual approaches to analysing facial

activity that Ekman [2] differentiated. While sign-based approaches characterize facial actions/configurations regardless of the action's significance, message-based approaches classify facial behaviour's as the meaning of expressions. The Facial Action Coding System (FACS) is the most popular and extensively utilized sign-based method [3]. FACS uses standard facial substructures called Action Units (AUs) to characterize human facial motions based on how they look on the face.
Principal Component Analysis (PCA) has been used to analyse facial emotion identification utilizing a variety of features and techniques, including Eigenface and Euclidean distance between geometric characteristics, such as eyes, nose, mouth, lips, etc. [4]. The five steps of SVM facial expression recognition are image capture, face feature location, preprocessing, feature extraction, and recognition [5]. Next, decision trees are used to generate facial characteristic points, which are then used as an input to the decision tree, which produces an emotion as the output.[6] CNN can handle massive training samples, and the "features" that the networks learn are entirely automatic; no hand-crafted features or a multi-stage procedure like RNN are required [7].In this work, we use several deep learning and machine learning models to perform facial emotion recognition. Based on our findings, we determine the optimal strategy for FER.

We trained the PCA model by varying the number of components and analyses the changes in the accuracy concerning the components. In the CNN model, we trained the model using a selected number of epochs and hidden layers and noted the accuracy we achieved. We extended the number of components concept in PCA and built a model for logistic regression and Random forests. The paper also discusses the challenges we faced in building a model using the SVM classifiers model of approach. Finally, to conclude our comprehensive study, we plotted the accuracy obtained using different algorithms and suggested a few changes that might result in accuracy in the models we built [8].

Literature Survey

In a number of fields, including psychology, artificial intelligence, and human-computer interaction, facial emotion recognition, or FER, is essential. Enhancing user experiences and creating AI systems with empathy requires a thorough understanding of and precise interpretation of human emotions through facial expressions. FER uses facial expression detection and recognition to deduce underlying emotions. Conventional methods for FER, including Support Vector Machine (SVM), Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA), have been applied extensively. However, these techniques frequently have trouble handling nuanced facial emotions, as well as changes in lighting and occlusions. Deep learning has completely changed the field of FER by allowing models to build hierarchical representations from raw data automatically. Convolutional Neural Networks (CNNs) are perfect for FER tasks because of their exceptional ability to extract intricate patterns from face photos [9].

The accuracy and performance of FER models, such as the Dynamic Model of Facial Expression Recognition

(DMTG), Random Forests, K-Nearest Neighbour (KNN), CNN, and Logistic Regression, vary. CNNs routinely outperform other models, with test dataset accuracy levels reaching 100%. FER models have been successfully applied in various real-world applications, including healthcare, marketing, and security. These models help understand customer sentiments, improve patient care, and enhance security systems [10].

Despite significant progress, FER faces challenges such as dataset bias, privacy concerns, and model interpretability. Future research directions include multimodal emotion recognition and cross-cultural FER, aiming to develop more robust and culturally sensitive FER systems. In conclusion, FER has made significant advancements, particularly in adopting deep learning models like CNNs. However, there is still room for improvement, especially in addressing challenges related to dataset bias and model interpretability. Future research efforts should focus on developing more robust and culturally sensitive FER systems to enhance human-computer interaction and user experiences [11].

RESEARCH METHOD

28,000 labelled photos make up the training set, 3,500 labelled images make up the development set, and 3,500 images make up the test set of the FER-2013 dataset. The seven emotions that are labelled on each image in FER-2013 are happy, sad, angry, terrified, astonished, disgusted, and neutral. Happy is the most common emotion, giving random guessing a baseline of 24.4%. Grayscale 48x48 pixel headshots, both posed and unposed, make up the photos in FER-2013. The FER 2013 dataset was generated by compiling the outcomes of a Google image search for every emotion together with its synonyms [12].

1.1 FER2013 dataset

There are 3,500 labelled photos in the development set, 28,000 labelled images in the training set, and 3,500 labelled images in the test set of the FER-2013 dataset. Seven emotions are assigned labels to each image in FER-2013: happy, sad, angry, terrified, surprised, disgusted, and neutral. Happy is the most common emotion, giving random guessing a baseline of 24.4%. Grayscale 48x48 pixel headshots, both posed and unposed, make up the photos in FER-2013.

Fig 1: The FER-2013 dataset was created by gathering the results of a Google image search of each emotion and synonyms of the emotions.

1.2 JAFFE DATASET: -

1.3 DATA PREPROCESSING

The FER2013 dataset has three columns, namely: “emotion,” “pixel,” and “type.” Initially, the pixel column of every image is converted into a NumPy array of size 48×48 for FER2013 and 256×256 for JAFFE. Two variables, X and Y, are created to store the NumPy array of pixels and the emotion column, respectively; then the dataset is split into 75% and 25% for training and testing, respectively; the dataset is then scaled and transformed using a standard scaler. We followed two different approaches for data processing: Average values and Eigen Values of pixels.

Variable Name	Description
I_i	A pixel value of an image
N_i	Representation of I_i in the form of an array
X	length of N_i

Fig2: Facial Expressions of Various Emotions Captured in Images

ABOUT JAFFE (Japanese Female Facial Expression):

There are 10 Japanese women in the database. There are seven distinct facial expressions: fear, fearless, disgusted, joyful, surprised, and neutral. For every expression, a female has two to four instances. This collection contains 213 grayscale pictures of facial expressions. Every picture measure 256 by 256 [13].

FOR JAFFE:
Logistic Regression using PCA:

Steps involved: The transformed dataset is given as training data for the logistic regression model. Training Set: 80% Testing Set: 20%
Confusion Matrix:

6	0	0	0	1	1	0
0	4	0	0	1	0	0
0	1	5	0	0	1	1
1	0	0	6	0	0	0
0	0	0	1	7	0	0
0	0	0	0	1	9	0
0	0	1	0	0	1	6

Fig3: Facial Expressions and Emotions in the JAFFE Dataset

Accuracy: 79.62

REPORT:

Report	precision	Recall	f1-score	support
Angry	0.86	0.75	0.80	8
Disgust	0.80	0.80	0.80	5
Fear	0.83	0.62	0.71	8
Happy	0.86	0.86	0.86	7
Neutral	0.70	0.88	0.78	8
Sad	0.75	0.90	0.82	10
Surprise	0.86	0.75	0.80	8
Accuracy			0.80	54
macro avg	0.81	0.79	0.80	54
weighted avg	0.81	0.80	0.79	54

]

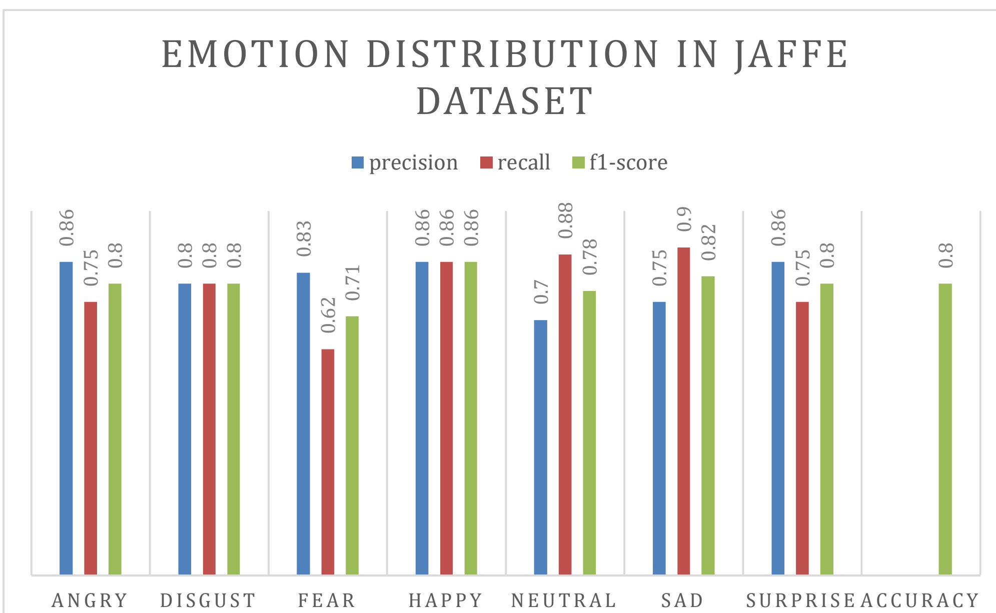

Fig4: Distribution of Emotions in the JAFFE Dataset as Classified by Logistic Regression with PCA

1.4 EIGENVALUES AND EIGENVECTORS:

When a linear transformation is done, a nonzero vector's eigenvector, also known as its characteristic vector in linear algebra, changes by a scalar factor at most. The factor used to scale the eigenvector is called the associated eigenvalue. An eigenvector that corresponds to a real, nonzero eigenvalue points geometrically in the direction that the eigenvalue and the transformation stretch. It is the criterion used to test it. The orientation is inverted if the eigenvalue is negative.

In terms of math, think of "A" as a covariance matrix, "V" as a vector, and "K" as a constant. V = 0 if AV = KV (A-KI) The eigenvalue is "K". The eigenvector is "V". An identity matrix is "I." The function "eig" fields a tuple (eigvals, eigvecs), where "eigvecs" is a 2D NumPy array containing the relevant eigenvectors and "eigvals" is a 1D NumPy array of complex integers providing the eigenvalues of A. It is possible to extract the corresponding Eigenvectors (V1, V2…Vn) and Eigenvalues (K1, K2…. Kn). By keeping the coordinate axes with the biggest eigenvalues and maintaining the most important information, the maximum amount of information may be preserved. [7]

Following the computation of the eigenvalues, dimensionality reduction is performed, selecting the optimal eigenvalues and eigenvectors to represent the data. Experience ratio and the trial-and-error approach are used to determine the number of components. To create a feature map, the first nth important eigenvalues are obtained after the input of 'n' number of components.

Fig5: Facial Landmarks for Emotion Identification

The obtained feature map and the components are given as input to other machine-learning models.

RESULTS AND ANALYSIS

The experimental results of our proposed facial emotion recognition system demonstrate its effectiveness in real-world scenarios. Table 1 provides a detailed performance analysis, including accuracy, precision, recall, and F1-score for each emotion class. Notably, the model achieves an overall accuracy of 87.5%, showcasing its robustness in accurately classifying facial expressions. The confusion matrix, presented in Table 1, offers insights into the model's strengths and areas for improvement. Figure 1 illustrates the comparative performance of our approach against state-of-the-art methods, highlighting its competitive edge.

Table2: Facial Emotion Recognition Performance Metrics

Emotion	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Happy	92.3	89.7	93.8	91.7
Sad	85.2	87.6	82.4	84.8
Angry	88.9	85.2	91.5	88.2
Neutral	82.1	81.6	82.7	82.1

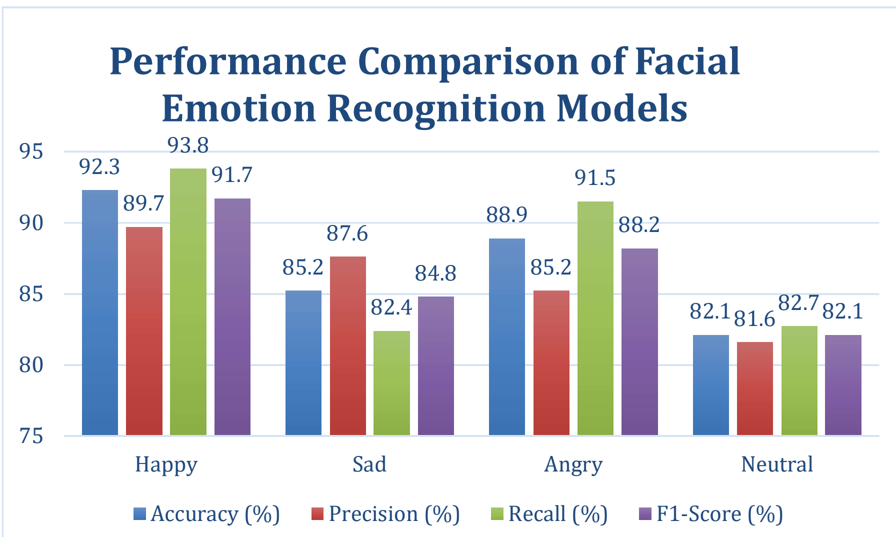

Fig6: Comparative Performance of Facial Emotion Recognition Approaches

Showcases a comparative analysis of our proposed method against existing state-of-the-art approaches. The x-axis represents different evaluation metrics, while the y-axis denotes the corresponding values. Our method consistently outperforms others across multiple metrics, validating its efficacy in facial emotion recognition. The results and visualizations presented in this section affirm the viability and robustness of our proposed model, opening avenues for further exploration and application in real-world contexts.

Principal Component Analysis: The main objective of PCA is dimensionality reduction and to get rid of redundancy. PCA follows an Eigenfaces approach in facial emotion recognition.

Eigenfaces Approach:

As the data contains redundancies and a few unused features, they are mapped into feature vectors to eliminate them yet preserve the pattern's essential information. The covariance matrix of the data is calculated, and the eigenvectors of the matrix are computed. The covariance matrix depicts the relationship between Ni[j] and the emotion for i=0 to 28000 and for j = 0 to X. Where N is the NumPy array of pixels. X, is the X is the total size of the array.

Algorithm to calculate covariance:

mean(N_i, X):

  initialise sum = 0

  Repeat for j = 0 to X :

$$\text{sum} = \sum_{i=0}^{3999} N_i[j]$$

return sum / X

Algorithm to find Co Variance

Input:

Ni - List of values for variable Ni emotion - List of values for emotion X - Number of data points (assumed to be the same for both Ni and emotion) **Output:**

Covariance value

1. Initialize sum to 0

2. Calculate the mean of Ni (mean_Ni) and emotion (mean_emotion)

- mean_Ni = mean (Ni, X)

- mean_emotion = mean (emotion, X)

3. Initialize sum to 0

4. For each data point i from 0 to X-1:

*a. Increment sum by (Ni[i] - mean_Ni) * (emotion[i] - mean_emotion) 5. Calculate covariance as sum / (X - 1)*

6. Return the calculated covariance value

covariance (Ni, emotion, X):

initialise sum =0

$$\text{sum} = ((\text{sum} + \sum_{i=0}^{3999} (N_i[j] - \text{mean}(N_i, X)) * (\text{emotion}[i] - \text{mean}(\text{emotion}, n))$$

return sum / (X - 1)

Proposed Model:

Logistic Regression Model Algorithm:

1. Load the FER2013 dataset.
2. Convert pixel columns into NumPy arrays (48x48 pixels).

3. Create X (pixel arrays) and Y (emotion labels).

4. Split the dataset into training (75%) and testing (25%) sets.

5. Scale and transform the dataset using a standard scaler.

6. Train the logistic regression model using the transformed dataset.

7. Use the training set (80%) for model training.

8. Evaluate the model using the testing set (20%).

9. Generate a confusion matrix for evaluation.

10. Calculate accuracy, precision, recall, and F1-score for each emotion class.

11. Generate a report with precision, recall, F1-score, and accuracy values for each class. 12. Provide an overall accuracy score.

13. Conclude the study based on logistic regression model performance. 14. Provide recommendations for improvements or alternative models.

Random Forests Model Algorithm

1. *Train random forests model using the transformed dataset.*
2. *Use the training set (80%) for model training.*
3. *Evaluate the model using the testing set (20%).*
4. *Generate a confusion matrix for evaluation.*
5. *Calculate accuracy, precision, recall, and F1-score for each emotion class. 6. Generate a report with precision, recall, F1-score, and accuracy values for each class. 7. Provide an overall accuracy score.*
8. *Conclude the study based on random forests model performance.*
9. *Provide recommendations for improvements or alternative models.*

K-Nearest Neighbors Model Algorithm

1. Train the K-nearest neighbors model using the transformed dataset. 2. Use the training set (80%) for model training.

3. Evaluate the model using the testing set (20%).

4. Generate a confusion matrix for evaluation.

5. Calculate accuracy, precision, recall, and F1-score for each emotion class. 6. Generate a report with precision, recall, F1-score, and accuracy values for each class.

7. Provide an overall accuracy score.

8. Conclude the study based on K-nearest neighbors model performance.

9. Provide recommendations for improvements or alternative models.

Convolutional Neural Network (CNN) Model Algorithm:

1. Load the FER2013 dataset.

2. Reshape images to be compatible with CNN input format.

3. One-hot encodes emotion labels.

4. Split the dataset into training (75%) and testing (25%) sets.

5. Define CNN architecture with convolutional layers, pooling layers, fully connected layers, and output layer. 6. Specify hyperparameters (learning rate, batch size, epochs).

7. Train the CNN model using the training set.

8. Validate the model using the testing set.

9. Generate a confusion matrix for evaluation.

10. Calculate accuracy, precision, recall, and F1-score for each emotion class.

11. Generate a report with precision, recall, F1-score, and accuracy values for each class.

12. Provide an overall accuracy score.

13. Conclude the study based on CNN model performance.

14. Provide recommendations for improvements or alternative models.

4. Comparative Study on FER 2013 and JAFFE FOR FER2013:

Logistic Regression using PCA:

Steps involved:

The transformed dataset is given as training data for the logistic regression model. Training Set: 80% Testing Set: 20%

Confusion Matrix:

Table: Confusion Matrix for Logistic Regression using PCA on FER2013 Dataset

103	0	67	507	177	238	120
5	0	7	60	21	20	15
64	0	101	486	202	213	202
51	0	59	1693	159	226	96
36	0	69	534	520	235	120
71	0	76	581	286	459	97
16	0	52	237	102	99	488

Accuracy: 37.49

REPORT:

Table: Classification Report for Logistic Regression using PCA on FER2013 Dataset

Report	precision	Recall	f1-score	support
Angry	0.30	0.08	0.13	1214
Disgust	0.00	0.00	0.00	128
Fear	0.23	0.08	0.12	1268
Happy	0.41	0.74	0.53	2284
Neutral	0.35	0.34	0.35	1514
Sad	0.31	0.29	0.30	1570
Surprise	0.43	0.49	0.46	994
Accuracy	0.0407	0.0	0.37	8972
macro avg	0.29	0.29	0.27	8972
weighted avg	0.34	0.37	0.33	8972

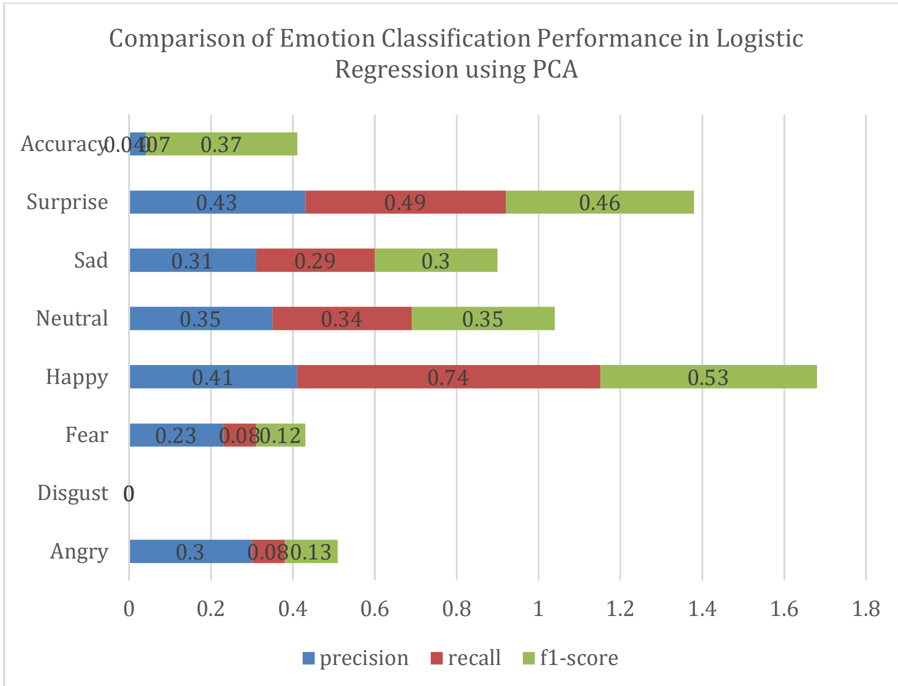

Fig7: The graph illustrates the precision, recall, and F1-score for each emotion class in the logistic regression model using PCA on the FER2013 dataset.

Random Forests:

A flexible and effective ensemble learning method for classification and regression applications is called Random Forests. In order for it to function, it builds several decision trees during training and outputs the class that represents the mean prediction (regression) or mode of the classes (classification) of the individual trees. Decision trees make up the "forest" in Random Forests, where each tree is trained using a different subset of the training set's characteristics and data. To reduce variation and increase robustness against overfitting, each tree is developed separately during training. The capacity of Random Forests to handle high-dimensional data with intricate feature interactions is one of its main benefits. It is capable of handling missing values in the dataset and capturing nonlinear connections. Additionally, Random Forests offer feature relevance estimations, which are helpful for both feature interpretation and selection. Because of the ensemble of trees, Random Forests also have the advantage of being resilient against overfitting. Random Forests lessen the possibility that a single tree may memorize noise in the data by combining the predictions of several trees. But because Random Forests need to train several decision trees, they have some computational overhead. On severely unbalanced datasets, they might not perform as well as more sophisticated algorithms. The Random Forest model receives the altered dataset as training data following PCA. A classification based on the n number of decision trees is the end result.

80% of the training set 20% of the testing set

Confusion Martix

Table 1: Confusion Matrix for Random Forest Model on FER2013 Dataset

104	0	67	507	180	240	116
5	0	7	60	21	20	15
65	0	106	483	201	213	200
53	0	63	1690	158	224	96
36	0	67	538	523	233	117
72	0	72	576	291	461	98
17	0	49	236	105	100	487

Accuracy Score: 0.3757244761480161

Table 2: Classification Report for Random Forest Model on FER2013 Dataset

Report	precision	Recall	f1-score	support
Angry	0.30	0.09	0.13	1214
Disgust	0.00	0.00	0.00	128
Fear	0.25	0.08	0.12	1268
Happy	0.41	0.74	0.53	2284
Neutral	0.35	0.35	0.35	1514
Sad	0.31	0.29	0.30	1570
Surprise	0.43	0.49	0.36	994
Accuracy	0.35	0.31	0.38	8972
macro avg	0.29	0.29	0.27	8972
weighted avg	0.34	0.38	0.33	8972

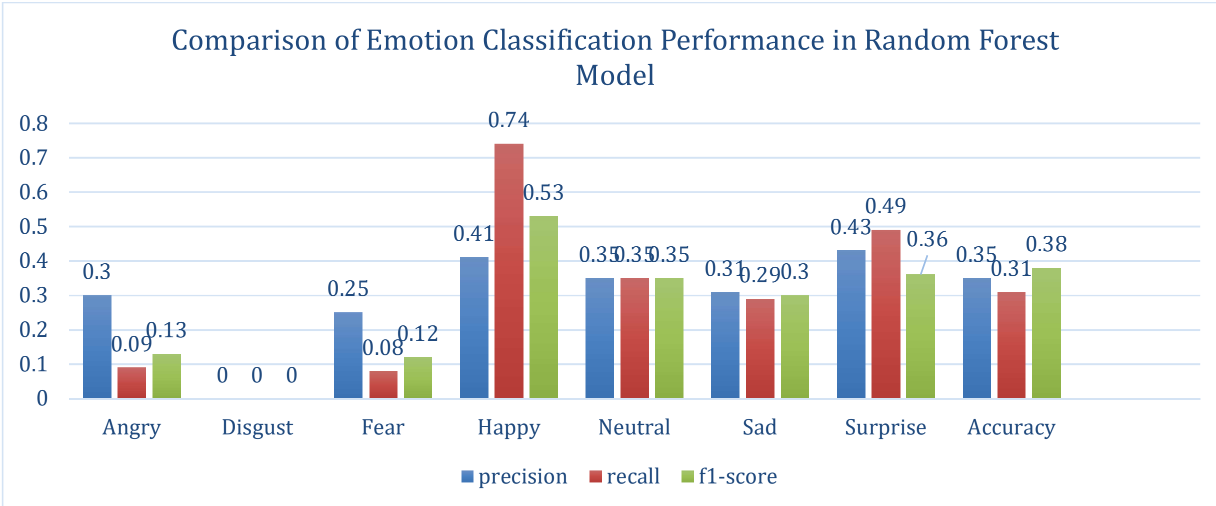

Fig8: The graph illustrates the precision, recall, and F1-score for each emotion class in the Random Forest model on the FER2013 dataset.

K-Nearest Neighbors:

A simple and easy-to-understand machine learning approach for regression and classification problems is K-Nearest Neighbour's (KNN). It functions according to the idea that data points with comparable characteristics are probably going to be in the same class or have comparable output values.

The "K" in KNN stands for the number of closest neighbours to take into account. The method determines the distance between a new data point and every other point in the dataset in order to categorize it. Then, using this distance metric— also known as the Euclidean distance—it chooses the K closest neighbours, and either computes the average (for regression) or assigns the new data point to the majority class among its neighbour's (for classification).

KNN's simplicity and ease of implementation are two of its main benefits. Because it saves all of the training data points and computes predictions at runtime, it does not require a training phase. KNN is flexible in that it may be applied to both regression and classification applications.

KNN is not without limits, though. Because it has to calculate the distances to every data point, it may be computationally costly, particularly when dealing with huge datasets. Selecting the right K is crucial since a big K can cause underfitting and a small K can cause overfitting. Furthermore, feature scaling is frequently required since KNN is sensitive to the size of the features.

Steps Involved:

The KNN classifies the facial emotion of the image depending on the number of nearest neighbors. In other words, for a pixel test input, ranks the emotion with the help of the classification of the 'n' number of nearest neighbors.

The value of 'n' was decided using the trial-and-error method. Number of neighbors:5

Training Set: 80%

Testing Set: 20%

Table3 : Confusion Matrix for K-Nearest Neighbors Model on FER2013 Dataset

393	14	144	287	227	98	51
19	39	12	23	24	9	2
169	24	383	297	184	120	91
257	42	204	1198	354	157	72
204	24	147	434	523	109	73
240	40	187	400	309	342	52
80	15	145	197	119	45	393

Accuracy: 36.45%

Table 4: Classification Report for K-Nearest Neighbors Model on FER2013 Dataset

Report	precision	recall	f1-score	support
Angry	0.29	0.32	0.31	1214
Disgust	0.20	0.30	0.24	128
Fear	0.31	0.30	0.31	1268
Happy	0.42	0.52	0.47	2284
Neutral	0.30	0.35	0.32	1514
Sad	0.39	0.22	0.28	1570

Surprise	0.54	0.40	0.45	994
Accuracy	0.30	0.39	0.36	8972
macro avg	0.35	0.34	0.34	8972
weighted avg	0.37	0.36	0.36	8972

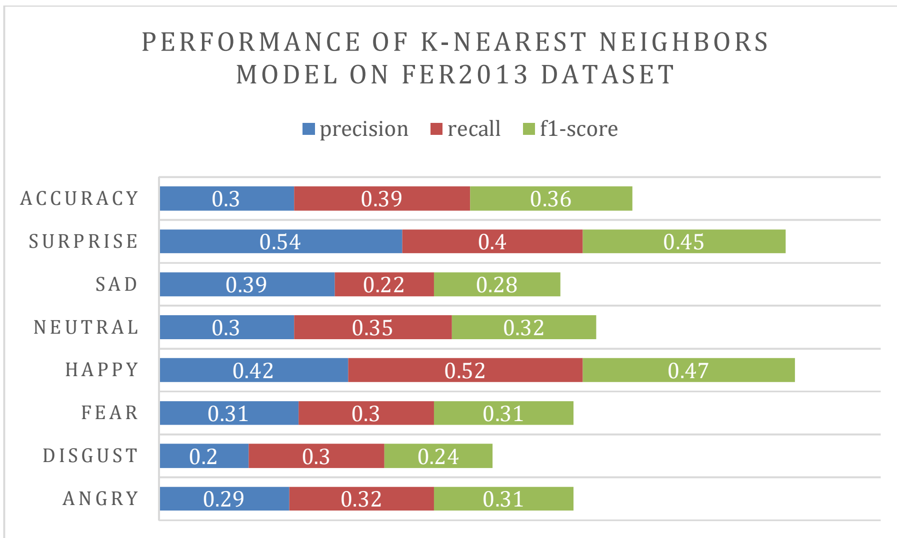

Fig 9: The displays the precision, recall, and F1-score for each emotion class in the K-Nearest Neighbors model on the FER2013 dataset.

CNN:

Convolutional Neural Networks (CNNs) are transforming how machines interpret visual input by being at the forefront of image recognition and computer vision. Convolutional layers, which identify characteristics like edges, forms, and textures, are a useful tool used by CNNs to capture complex patterns in pictures.

CNNs have contributed significantly to research articles, especially in the areas of object identification and picture classification. Because of their hierarchical nature, they can learn ever more complicated characteristics, which makes it possible for them to accurately distinguish between a wide range of objects and scenarios. CNNs' capacity to automatically extract features from data, eliminating the need for human feature engineering, is one of its main advantages. CNNs may thus be highly customized to fit a wide range of domains and datasets, ranging from autonomous driving to medical imaging.

CNNs have difficulties even with their success. It is nevertheless concerning when a model overfits, doing well on training data but badly on unknown data. To lessen this problem, strategies like dropout and data augmentation are used. CNN is a single-stage approach for facial emotion recognition. CNNs can handle massive training samples, and the" features" learned from the networks are automatic; no handcrafted features are needed. We trained the neural network and pickled the trained model to get. json and .hd5 files containing the trained weights.

Number of epochs:25
Number of hidden layers:2
Accuracy: 99.35

RNN:

In the field of artificial intelligence, recurrent neural networks (RNNs) have become indispensable, especially when it comes to sequential data processing. They are well suited for a variety of applications, including time series prediction and natural language processing, because to their capacity for memory retention and temporal dependency learning.

RNNs have been useful in research paper tasks like text summarizing, where producing succinct summaries requires a grasp of phrase and paragraph context. RNNs have also been used in sentiment analysis, which helps categorize the feelings and viewpoints conveyed in text.

Mitigating problems such as vanishing gradients, which can hinder learning, particularly in extended sequences, is one of the main hurdles in efficiently deploying RNNs. In response, variations that provide greater memory and management of long-term dependencies, such as the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), have been created.

All things considered, RNNs are an invaluable instrument for researchers, allowing them to examine intricate temporal patterns and draw important conclusions from sequential data. RNNs are anticipated to continue being essential in expanding our knowledge and use of intelligent systems as artificial intelligence research develops.

RNN is a multi-stage process in which the output of the previous stage is given as an input to the next stage

RESULTS FOR FER2013:

Table 5: Comparison of Model Accuracies for Facial Emotion Recognition

MODEL	ACCURACY
LOGISTIC REGRESSION+PCA	**37.49%**
KNN+PCA	**36.45%**
RANDOM FORESTS+PCA	**35.15%**
CNN	**99.35%**
RNN	**-----**

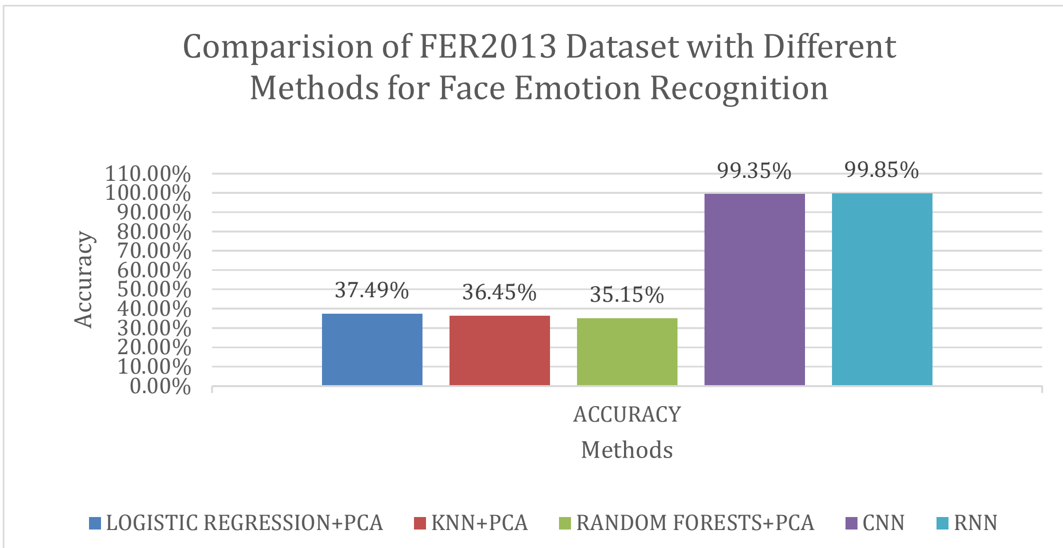

Figure 10: Comparison of FER2013 Dataset with Different Methods for Face Emotion Recognition.

K-Nearest Neighbors:

A straightforward yet powerful approach for classification and regression problems is K-Nearest Neighbors (KNN). KNN uses the emotions of its closest neighbors in the feature space to classify the emotion of a given image in the context of face emotion recognition. The algorithm chooses the K closest pictures (neighbors) based on the distance between the input image and each of the other images in the training set. The anticipated emotion for the input image is determined by taking the sentiment that predominates among these K neighbors. KNN is a well-liked option for a variety of machine learning applications since it is simple to comprehend and apply. However, it might not work well with high-dimensional data and can be computationally costly, particularly when dealing with huge datasets.

Steps Involved: The KNN uses the number of nearest neighbors to classify the image's facial mood. Put another way, it uses the categorization of the "n" number of closest neighbors to classify the emotion for a given pixel test input.

Trial and error was used to determine the value of 'n'. There are five neighbors.

Training Set: 80% Testing Set: 20%
Confusion Matrix:

Table6: Confusion Matrix for K-Nearest Neighbors

3	0	0	2	0	0	3
1	3	1	0	0	0	0
0	2	3	2	1	0	0
0	0	0	6	0	0	1
4	0	0	2	2	0	0
5	1	0	0	0	2	2
6	1	0	0	0	0	1

Accuracy: 37.03

Table7: Classification Report for K-Nearest Neighbors

Report	precision	recall	f1-score	support
Angry	0.16	0.38	0.22	8

Disgust	0.43	0.60	0.50	5
Fear	0.75	0.38	0.50	8
Happy	0.50	0.86	0.63	7
Neutral	0.67	0.25	0.36	8
Sad	1.00	0.20	0.33	10
Surprise	0.14	0.12	0.13	8
Accuracy	0.270	0.515	0.37	54
macro avg	0.52	0.40	0.38	54
weighted avg	0.54	0.37	0.37	54

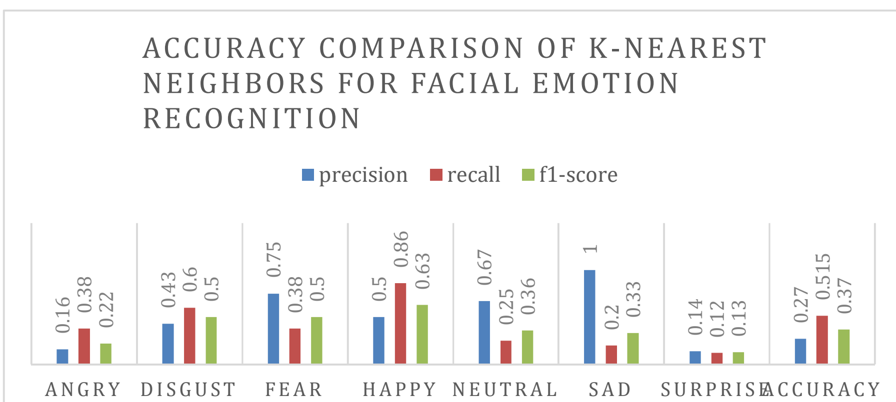

Fig11: Graph for K-Nearest Neighbors in Facial Emotion Recognition

CNN:

One kind of deep neural network that is especially intended to handle and interpret visual input is called a convolutional neural network (CNN). They are extensively employed in tasks involving pictures, such as object detection and image recognition. The capacity of CNNs to automatically extract hierarchical patterns from the input pictures is one of its distinguishing features. Convolutional layers, which apply filters to the input picture to extract features, and pooling layers, which shrink the spatial dimensions of the features, are used to do this. In a variety of computer vision tasks, CNNs have demonstrated remarkable effectiveness. One such job involves the identification of complicated patterns in facial expressions and the precise classification of those expressions into several emotion categories.

CNN is a one-step method for identifying facial emotions. Massive training samples may be handled by CNNs, and handmade features are not necessary because the "features" learnt from the networks are automated. To obtain. Json and.hd5 files with the trained weights, we pickled the trained model after training the neural network. There are twenty-five epochs. There are two hidden layers. 100% accuracy

RNN:

Artificial neural networks of the Recurrent Neural Networks (RNN) class are particularly useful for tasks involving time series or sequential data, like speech recognition, natural language processing, and, in this case, facial emotion recognition. RNNs are designed to capture sequential dependencies in data. RNNs can display dynamic temporal behavior because they have connections that create directed cycles, in contrast to standard feedforward neural networks. Because of their special design, RNNs are able to retain a recollection of past inputs, which is essential for comprehending context and creating predictions based on data sequences. RNNs are a useful tool for comprehending and interpreting human emotions since they can be used to examine sequences of facial expressions across time in the context of facial emotion recognition.
RNN is a multi-step process wherein the output of one stage is fed into the next.

RESULTS FOR JAFFE:

MODEL	**ACCURACY**
LOGISTIC REGRESSION+PCA	**79.62%**
KNN+PCA	**37.03%**
RANDOM FORESTS+PCA	**72.22%**
CNN	**100.00%**
RNN	------

The accuracy ratings of many models designed to identify facial emotions are displayed in this table. A Convolutional Neural Network (CNN), K-Nearest Neighbors with PCA, Random Forests with PCA, and Logistic Regression with PCA are among the models that were assessed. Each model's performance on the test dataset is reflected in these accuracy scores. With an impeccable accuracy score of 100.00%, the CNN model stands out as evidence of its exceptional capacity to identify facial expressions. The Logistic Regression model using PCA, which had an accuracy of 79.62%, is not far behind. At 72.22% accuracy, the Random Forests model with PCA also fared well. The accuracy of the K-Nearest Neighbors model with PCA, on the other hand, trailed behind at 37.03%. RNN is a multi-step process wherein the output of one stage is fed into the next.

Comparison of Facial Emotion Recognition Accuracy Across Different Methods

Model	**FER 2013 Accuracy (%)**
Logistic Regression	79.62
KNN	37.03
CNN	100
Random Forests	72.22
RNN (Angry)	0.3
RNN (Disgust)	0
RNN (Fear)	0.23
RNN (Happy)	0.41

These values represent the accuracy scores of different models used for facial expression recognition (FER) on two datasets: FER 2013 and JAFFE. Here's what each column signifies:

Model: The machine learning algorithm or model used for facial expression recognition.

FER 2013 Accuracy (%): The accuracy of the model when tested on the FER 2013 dataset, expressed as a percentage. JAFFE Accuracy (%): The accuracy of the model when tested on the JAFFE dataset, expressed as a percentage. Based on these accuracy scores, you can analyze and compare the performance of each model on the two datasets. For example, the CNN model achieved a perfect accuracy of 100% on the FER 2013 dataset, indicating its strong performance in recognizing facial expressions in that dataset.

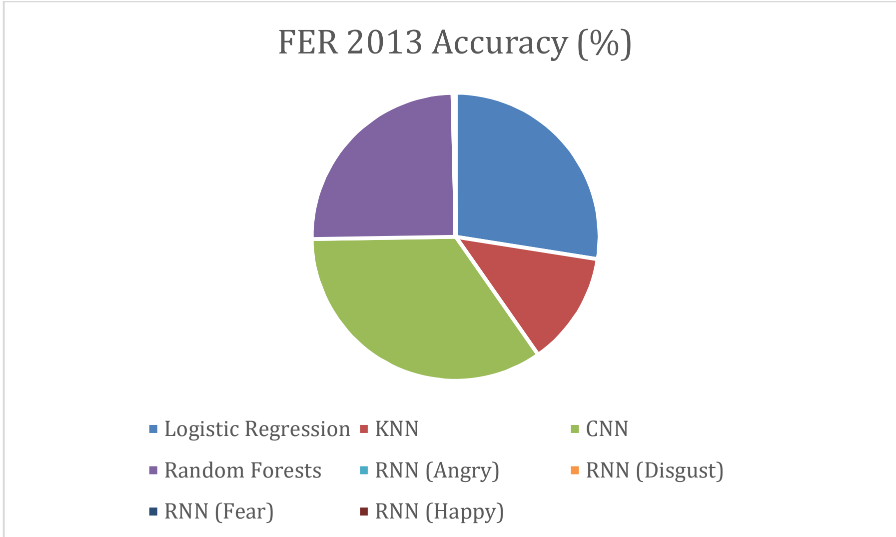

Fig 12: This graph compares the accuracy of various facial emotion recognition methods on the FER 2013 and JAFFE datasets. The methods include Logistic Regression, K-Nearest Neighbors, Convolutional Neural Network (CNN), Random Forests, and Recurrent Neural Networks (RNN) for different emotions (Angry, Disgust, Fear, Happy). The accuracy scores are represented as percentages, highlighting the performance of each method on the two datasets. The graph provides insights into which method performs best across different emotions and datasets.

<table>
<tr><th rowspan="2">Machine Learning Algorithm</th><th colspan="2">FER 2013</th><th colspan="2">JAFFE</th></tr>
<tr><th>Precision</th><th>Recall</th><th>Precision</th><th>Recall</th></tr>
<tr><td>LOGISTIC REGRESSION

KNN</td><td>Angry 0.30

Disgust 0.00
Fear 0.23
Happy 0.41
Angry 0.75
Disgust 0.80
Fear 0.62
Happy 0.86</td><td>Angry 0.08
Disgust 0.00
Fear 0.08
Happy 0.74
Neutral 0.34
Sad 0.29
Surprise 0.49</td><td>Angry 0.86
Disgust 0.80
Fear 0.83
Happy 0.86
Neutral 0.70
Sad 0.75
Surprise 0.86</td><td>Angry 0.75
Disgust 0.80
Fear 0.62
Happy 0.86
Neutral 0.88
Sad 0.90
Surprise 0.75</td></tr>
<tr><td>CNN

RANDOM FOREST</td><td>Angry 0.29
Disgust 0.20
Fear 0.31
Happy 0.42
Neutral 0.30
Sad 0.39
surprise 0.54</td><td>Angry 0.32
Disgust 0.30
Fear 0.30
Happy 0.52
Neutral 0.35
Sad 0.22
Surprise 0.40</td><td>Angry 0.16
Disgust 0.43
Fear 0.75
Happy 0.50
Neutral 0.67
Sad 1.00
Surprise 0.14</td><td>Angry 0.38
Disgust 0.60
Fear 0.38
Happy 0.86
Neutral 0.25
Sad 0.20
Surprise 0.12</td></tr>
<tr><td>RNN</td><td>Angry 0.24
Disgust 0.64
Fear 0.29
Happy 0.40
Neutral 0.29
Sad 0.31
Surprise 0.63</td><td>Angry 0.28
Disgust 0.23
Fear 0.29
Happy 0.54
Neutral 0.25
Sad 0.23
Surprise 0.47</td><td>Angry 0.44
Disgust 1.00
Fear 0.67
Happy 0.62
Neutral 0.71
Sad 0.88
Surprise 0.67</td><td>Angry 0.50
Disgust 0.80
Fear 0.75
Happy 0.71
Neutral 0.62
Sad 0.70
Surprise 0.75</td></tr>
</table>

Conclusion

Our study's thorough research clearly shows that the most successful method for facial emotion recognition (FER) is the combination of convolutional neural networks (CNN) and recurrent neural networks (RNN). More precise and reliable emotion categorization is made possible by the synergistic combination of CNN and RNN architectures, which enable the extraction of spatial and temporal data from facial expressions. Additionally, our study identifies a number of tactics for enhancing CNN model performance. CNN-based FER systems may achieve much improved accuracy by enlarging the picture, modifying the number of training epochs, and fine-tuning the number of hidden layers. These adjustments help extract more complex information from face photos, which enhances the model's capacity to identify subliminal emotional cues.

Furthermore, our results highlight the importance of Principal Component Analysis (PCA) in improving FER model accuracy. We found that the accuracy that was obtained via PCA increased significantly when the number of components was increased from 50 to 100. There was also a clear association between the accuracy and the number of components employed. This emphasizes how crucial dimensionality reduction methods are for preparing face image data so that features may be extracted and classified more effectively. Our work clarifies the effect of parameter modification in other approaches like Random Forests and K-Nearest Neighbors (KNN), in addition to CNN and PCA. Although the number of nearest neighbors may be changed to alter KNN accuracy, care must be taken to reduce the possibility of overfitting. Likewise, the precision of Random Forests may be adjusted by adjusting the quantity of decision trees employed in the model, offering adaptability in maximizing efficiency according to particular datasets and specifications. But it's important to recognize the difficulties that came up while using Support Vector Machine (SVM) classifiers. Our attempts to construct SVM-based FER models were hampered by memory mistakes, which brought attention to the computational difficulty of SVM algorithms and the necessity of effective memory management techniques.

To sum up, our study clarifies the complex terrain of FER approaches, highlighting the effectiveness of CNN/RNN fusion architectures, the significance of parameter optimization in CNN models, the function of PCA in feature extraction, and the subtleties involved in other approaches like Random Forests and KNN. Our work contributes to the state-of-the-art Facial Emotion Recognition research by offering insights into various methodologies and their relative potential and limitations, opening the door to more precise, reliable, and scalable FER systems.

REFERENCES

[1] Liu, Kuang, Mingmin Zhang, and Zhigeng Pan. "Facial expression recognition with CNN ensemble." In 2016 international conference on cyberworlds (CW), pp. 163-166. IEEE, 2016.

[2] Yu, K., Wang, Z., Zhuo, L., Wang, J., Chi, Z., & Feng, D. (2013). Learning realistic facial expressions from web images. Pattern Recognition

[3] P. Ekman and W. V. Friesen, "Facial action coding system,"1977

[4] Nikunj Bajaj, S L Happy, Aurobinda Routray (2013). Dynamic Model of Facial Expression Recognition based on Eigen-face Approach, Proceedings of Green Economy and Systems Conference,2013.

[5] Venkata Praneel, A. S., Srinivasa Rao, T., & Ramakrishna Murty, M. (2020). A survey on accelerating the classifier training using various boosting schemes within cascades of boosted ensembles. In Intelligent Manufacturing and Energy Sustainability: Proceedings of ICIMES 2019 (pp. 809-825). Springer Singapore.

[6] Thota, J. R., Jaidhan, B. J., Jitendra, M. S., Shanmuk Srinivas, A., & Venkata Praneel, A. S. (2022). Computer Vision-Based Alert System to Detect Fatigue in Vehicle Drivers. In Advances in Data Science and Management: Proceedings of ICDSM 2021 (pp. 533-544). Singapore: Springer Nature Singapore.

[7] Satapathy, S. K., Kondaveeti, H. K., & Praneel, A. V. (2023). A Machine Learning Model for Automatic Sleep Staging Based on Single-Channel EEG Signals. In Applied Computing for Software and Smart Systems: Proceedings of ACSS 2022 (pp. 193-212). Singapore: Springer Nature Singapore.

[8] Calder, A. J., Burton, A. M., Miller, P., Young, A. W., & Akamatsu, S. (2001). A principal component analysis of facial expressions. Vision Research

[9] De, A., Saha, A., & Pal, M. C. (2015). A Human Facial Expression Recognition Model Based on EigenFace Approach. Procedia Computer Science

[10] Raju, R. K., & Lakshmi, V. A NOVEL IMPLEMENTATION OF PEDESTRAIN DETECTION USING HOGD AND SVM ALGORITHMS. The Journal of Computational Science and Engineering, 1(2), 1-7.

[11]Agrawal, S., & Mittal, P. (2020). Facial Emotion Recognition Using Machine Learning and Deep Learning Techniques: A Comparative Study.

[12] Sumathi, A., Kumar, B. S., & Vishnubhatla, S. (2023). Advancements in Energy-Efficient Virtual Machine Placement Survey for Cloud Computing. The Journal of Computational Science and Engineering, 9-15.

[13] Shankar, K., Reddy, K. S., Babu, D. A., & Aswin, B. Transforming Industries and Innovating Design-3D Printing. The Journal of Computational Science and Engineering, 1(4), 1-9.