International Journal of Autonomous Robotics and Intelligent Systems
Vol. 1, No. 1, June 2025 [37-56]

Performance Evaluation of Human Facial Expression using Various Classification Methods

Loh, W.H.[1], Ismail, A.H.[1*] and Harun, H.R.[2]

[1]Dept. of Mechatronics, Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis (UniMAP), Pauh Putra Campus, 02600 Arau, Perlis, MALAYSIA.
[2]Dept. of Electrical and Electronics Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), 43400 Serdang, Selangor, MALAYSIA.

Received 24 June 2025, Revised 2 July 2025, Accepted 18 July 2025

ABSTRACT

This study assesses and contrasts the efficacy of raw picture pixels and image vectors as features in face expression classification. The CKPLUS dataset is utilized, and the issue of class imbalance is tackled by data augmentation. The dataset is partitioned into a 70% training set and 30% validation set. The training set consists of 175 images for each class, while the validation set consists of 75 images. The features are displayed using Matplotlib for raw pixels and t-SNE for vector features, then categorized using Random Forest and CNN classifiers. The performance is evaluated by utilizing confusion matrices, accuracy, precision, recall, and F1-score. The findings indicate that the Random Forest algorithm, when combined with vector features, obtains the maximum level of accuracy (99.6190%). Additionally, CNNs using raw pixel features also demonstrate strong performance. The precision, recall, and F1-scores exhibit similarity among the different approaches, with Random Forest (vector feature) and 2D CNN (raw pixels) showing somewhat better performance compared to other methods. These findings suggest that vector features have superior performance when used in conjunction with Random Forest, whereas raw pixel features are more successful when utilized with CNN.

Keywords: Facial Expression Recognition, Confusion Matrix, Convolution Neural Network (CNN), Random Forest Classifier, Vector Feature

1. INTRODUCTION

The global Emotion Detection and Recognition Market is estimated to be worth $23.5 billion in 2022 with projection of significant growth and reach $42.9 billion by 2027, and a Compound Annual Growth Rate (CAGR) of 12.8% [1]. This growth is primarily attributed to the increasing integration of Artificial Intelligence (AI) and Machine Learning (ML) technologies in the fields of biometrics, security, and surveillance. The growth has been expedited due to the impact of the Covid-19 pandemic. This has resulted in heightened government efforts and the implementation of touchless identity verification systems. The study conducted by Cowen et al. has identified a total of 16 facial expressions that can be easily understood and interpreted [2]. These expressions include amusement, anger, and sadness.

David mentioned emotional face expressions are essential for interpersonal communication and relationships, providing the best insight into a person's personality, feelings, goals, and intentions [3]. Extracted features will strongly influence machine learning and AI training. Grayscale pixel values, mean channel pixel values, and edge features are the most common and beginner-friendly feature extraction methods. However, the use of vector feature extraction in

*ihalim@unimap.edu.my

face expressions is relatively restricted, hence there are few comparisons between raw pixel features and vector features in facial expression categorization.

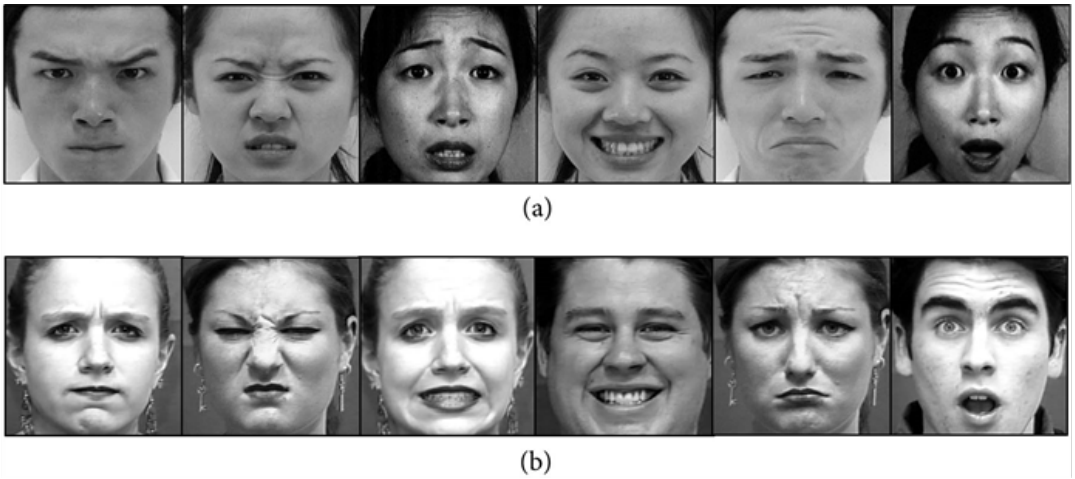


Figure 1. Example of facial expression exerted from [4]: (a) East-Asian samples and (b) Western Caucasian samples

The are vast methods for facial expression recognition and classifiers. This hinders classifiers using different extracted features from being compared as too many feature-classifier combinations. Thus, this kind of study is arduous and time-consuming. Wang et al. used CNN to extract features and random forest to categorize photos on different datasets [5]. This proved that CNN and random forest classifier are both strong classifiers. Some image examples are shown in Figure 1, which based on two difference world's regions.

There is little evidence on individual classification by separate traits, underlining the need for greater research. After classification, face expression recognition accuracy and confusion matrices from different feature extraction methods and classifiers must be compared. Vandana & Nikhil M. analyse the accuracy of multiple CNN algorithms for facial emotion identification [6].

1.1 Related Works

The field of facial expression recognition (FER) has seen significant advancements with the integration of deep learning and machine learning techniques. Wang et al. [5] proposed a hybrid model combining Random Forest with Convolutional Neural Networks (CNN), leveraging the robust feature extraction capabilities of CNNs alongside the classification power of Random Forests. Their approach demonstrated improved accuracy over standalone classifiers, indicating that ensemble methods can enhance the performance of FER systems, especially in handling complex facial features. Similarly, Ilyas et al. [7] incorporated Discrete Wavelet Transform (DWT) to preprocess facial features before feeding them into a deep CNN. This preprocessing step improved the model's ability to capture subtle expression changes by focusing on multi-resolution features in the frequency domain.

Other studies further explored the impact of deep learning architectures on FER accuracy and real-time performance. He et al. [8] developed a deep learning algorithm tailored for facial expression classification, emphasizing the value of optimized CNN structures for learning discriminative facial features. Yolcu et al. [9] applied FER in a biomedical context, utilizing deep learning to monitor neurological disorders, thus highlighting the growing interdisciplinary relevance of FER technologies. Additionally, Dino and Abdulrazzaq [10] evaluated traditional classifiers such as SVM, KNN, and MLP, showing that while deep learning models generally outperform classical methods, the latter still hold merit in lightweight or resource-constrained

applications. Together, these works illustrate a comprehensive landscape of FER strategies—ranging from hybrid models and feature engineering to purely data-driven deep architectures—each contributing to more accurate, efficient, and versatile emotion recognition systems. Table 1 summarizes the related works being the references in this paper.

Table 1 Summary of related works

Researcher	Dataset	Feature Extraction Method	Classification Method	Evaluation Method
Y. Wang et. al. [5]	JAFEE, CKPLUS, FER2013, The Real-world Affective Faces Database (RAF-DB)	HOG, CNN	C4.5 classifier, improved C4.5 classifier, CNN, one decision tree, random forest, new random forest	Accuracy, run time
B.R. Ilyas et. al. [7]	Extended Cohn-Kanade Dataset (CKPLUS), Japanese Female Facial Expression Database (JAFFE)	Histogram Equalization (HE), Discrete Wavelet Transforms (DWT) and Deep CNN	Deep CNN	Confusion Matrix and Recognition Rate
B. He [8]	Fer2013	AlexNet, VGG, ResNet.	AlexNet, VGG, ResNet.	Accuracy
G. Yolcu et al [9]	Radboud Face Database (RaFD)	CNN	Cascaded CNN	Accuracy
Dino and Abdulrazzaq [10]	CKPLUS	Histogram of Oriented Gradients (HOG)	SVM, KNN, MLPNN	10-fold validation, Comparing Accuracy
Y. Chen et. al [11]	Subset of CK image dataset	Center Symmetric Local Binary Pattern (CSLBP) algorithm, Rotation Invariant Local Phase Quantization (RILPQ) algorithm	SVM	Accuracy
K. Shan et. al. [12]	JAFFE, CKPLUS	CNN	CNN	Accuracy
N. Arora et. al. [13]	Faces94 dataset	CNN	CNN	Accuracy
K.C. Liu et. al. [14]	FER2013	CNN	CNN	Average Weighting method
S. M. Gowri et. al. [15]	FER2013	Dlib facial landmark detector.	CNN	Accuracy
M.I.N.P Munasinghe [16]	CKPLUS	Calculate distance between facial landmark pairs.	Random Forest Classifier	Accuracy
S. Bhogan et. al. [17]	FER2013	Pre-trained CNN model	CNN, KNN, Random Forest	Accuracy, Loss and confusion matrix

2. METHODOLOGY

The process of developing a facial expression recognition model using AI encompasses various essential steps. The project commences by conducting research on human facial expressions and comprehending the significance of AI in Facial Expression Recognition (FER). Subsequently, a suitable dataset is chosen and subjected to preprocessing techniques to optimize the training process. The dataset is divided into training and validation batches to facilitate feature extraction, training, and validation processes. The evaluation and comparison of the results include metrics such as accuracy, confusion matrix, precision, recall, and F1-score. The final phase involves the collection, analysis, and comparison of results, along with in-depth

discussions on the training process and comparative analysis. The flowchart provides a visual representation of the entire project process and the steps involved in making decisions are shown in Figure 2.

2.1 Dataset Selection

The facial expression recognition research considered multiple datasets, namely the Facial Expression Recognition 2013 (FER 2013) [18], Extended Cohn-Kanade (CKPLUS) [19], Facial Emotion Detection Dataset (FEDD) [20], and Static Facial Expressions in the Wild (SFEW) [21]. Every dataset possesses distinct advantages and disadvantages and compared in Table 2.

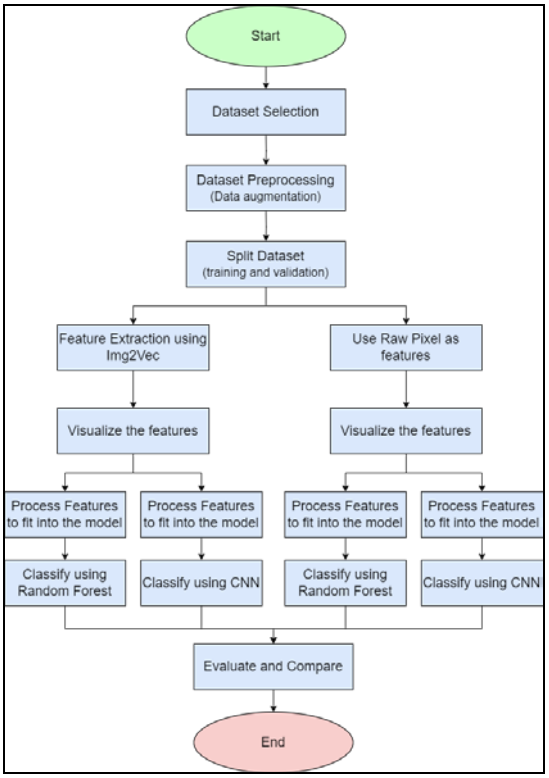


Figure 2. Project development flowchart

Table 2 Dataset comparison

	FER2013 [18]	**CKPLUS [19]**	**FEDD [20]**	**SFEW [21]**
File type	CSV	PNG	JPG	PNG
Total data	34034	981	637	1,158
Facial expressions	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Anger, Contempt, Disgust, Fear, Happy, Sadness, Surprise	Happy, Sad	Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise
Popularity	Highly popular	Popular	Not popular	Not popular
Feature extraction	Not available	Available	Available	Available

The CKPLUS [19] dataset was selected for this research due to its strong reputation in the field of facial expression recognition and its suitability for deep learning applications. Its widespread use in previous studies makes it an ideal benchmark for training and evaluating models, allowing for fair performance comparisons across different methodologies. One of its key advantages is the use of PNG format, which preserves image quality without compression artifacts, ensuring that crucial facial features remain intact for accurate feature extraction. Additionally, the dataset provides images with consistent and appropriate resolution, reducing the need for extensive preprocessing and making it well-suited for input into convolutional neural networks (CNNs).

Another reason for choosing CKPLUS is its comprehensive set of labelled facial expressions, which includes anger, contempt, disgust, fear, happiness, sadness, and surprise. These expressions represent both the six basic human emotions and a more complex emotion (contempt), offering a diverse and challenging dataset for training models to recognize subtle facial variations. The dataset also includes multiple subjects and expression intensities, enhancing the robustness and generalizability of trained models. For this study, CKPLUS was employed during both the training and comparison phases, ensuring that the model was evaluated on a well-established and emotion-rich dataset. Figure 3 illustrates examples of images used from the CKPLUS dataset, highlighting the quality and expressiveness of the facial data utilized in this research.

Figure 3. CKPLUS dataset from Kaggle [19]

2.2 Dataset Preprocessing

The selected dataset for this project exhibited an imbalanced distribution of data in each facial expression, which required the implementation of preprocessing techniques and data augmentation methods to achieve balanced dataset. The number of images in each facial expression folder was adjusted to 250 by duplicating random images as necessary. The augmented dataset was subsequently divided into two parts: 70% (175 images per expression, totalling 1225 images) for training and 30% (75 images per expression, totalling 525 images) for validation.

This process ensured that each expression was equally represented in both the training and validation sets. The datasets were shuffled randomly in order to minimize bias and enhance randomness.

2.3 Feature Extraction

2.3.1 Raw Image Pixel Feature Extraction

Machines utilize pixels to store images, and these pixels are represented as matrix numbers, which are determined by the dimensions of the image as stated by Singh [22]. The dataset utilized in this study consists of grayscale images, where the pixel values represent the intensity or brightness.

The utilization of raw grayscale pixel values as features for machine learning is a direct and uncomplicated approach. In this method, the number of feature vectors is equivalent to the number of pixels present in the image. An example of this is a 48 × 48 image, which consists of 2,304 features that are arranged in a one-dimensional array. The raw pixel features can be visualized by plotting them using Python libraries such as `Matplotlib` [23]. The following method is used to map pixel values to colours to display the distribution of pixel intensities. The pixel values range from 0 (representing black) to 255 (representing white).

2.3.2 Image Vector Feature Extraction using `Img2Vec`

C. Safka published a project that utilized `PyTorch` to transform images into vectors using `Img2Vec` [24]. The picture must be in RGB format and loaded using the Python Imaging Library (PIL). However, the datasets employed in this study are in grayscale format. Consequently, processing of the image must be conducted to verify their alignment with the input conditions.

Subsequently, the image undergoes processing using a pre-trained ResNet-18 model [25]. This processing involves undergoing transformations into feature maps at different layers. The output features from the fully connected layer are transformed into a 512-channel feature vector. This feature vector is then ready to be use as input for both the random forest and CNN classifiers.

The high-dimensional feature vector is visualized using t-SNE (t-distributed Stochastic Neighbour Embedding), which is a technique for reducing dimensionality [26]. The algorithm utilizes a Gaussian kernel to compute similarities in a high-dimensional space. These similarities are then mapped to lower dimensions, while ensuring that the original similarities are preserved. Figure 4 shows the pre-trained Resnet-18 model used for extracting the feature vector [25].

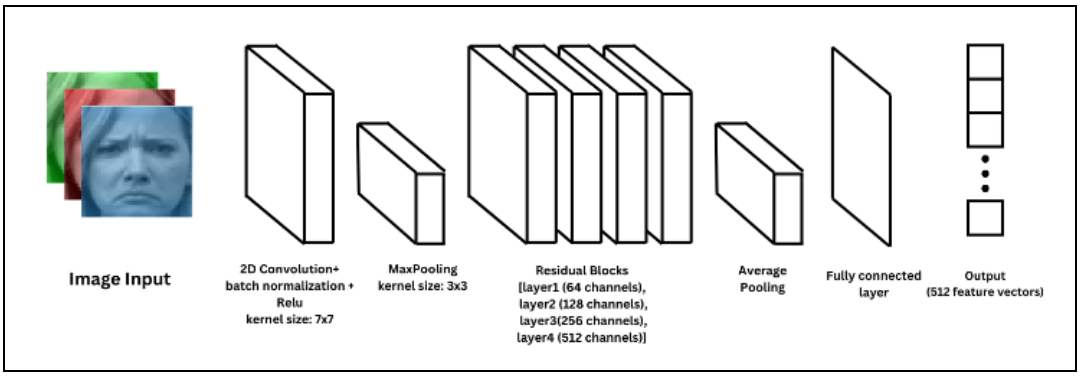


Figure 4. Pre-trained Resnet-18 model used for extracting the feature vector [25]

2.4 Classifier Development

2.4.1 Random Forest Classifier

The Random Forest (RF) algorithm is a supervised machine learning technique that is commonly used for both classification and regression tasks. It involves combining multiple classifiers to improve performance. The algorithm employs a technique called bagging to generate diverse feature subsets and optimize for information gain. This is achieved by creating multiple decision trees during the training process. The prediction is determined by aggregating the votes from all trees and selecting the majority class as the final prediction [27].

The Random Forest classifier is utilized in this project, implemented using the scikit-learn library. It is applied to both raw pixel features and vector features for the purpose of training and validation. Figure 5 shows the typical RF architecture.

2.4.2 Convolutional Neural Network (CNN)

In this project, 2D CNN and 1D CNN are used for raw pixel features and vector features as input respectively. These features are passed to a Convolution Neural Network model which is built using TensorFlow [28] library for training and validating.

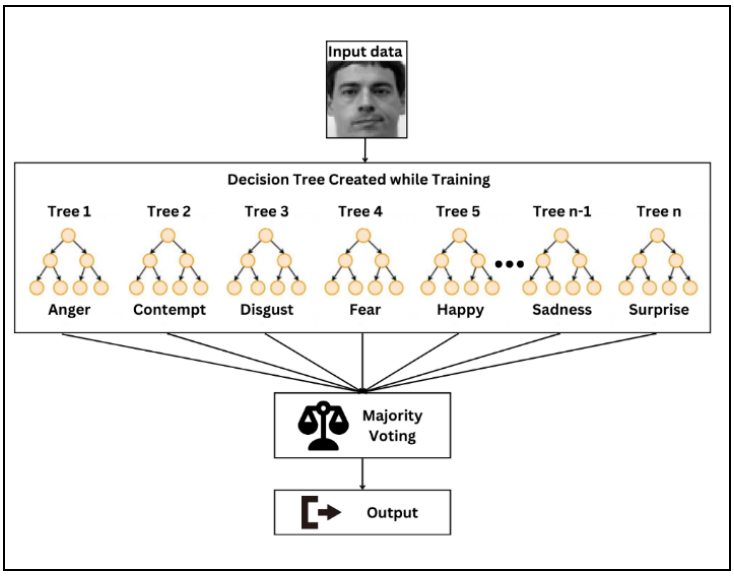
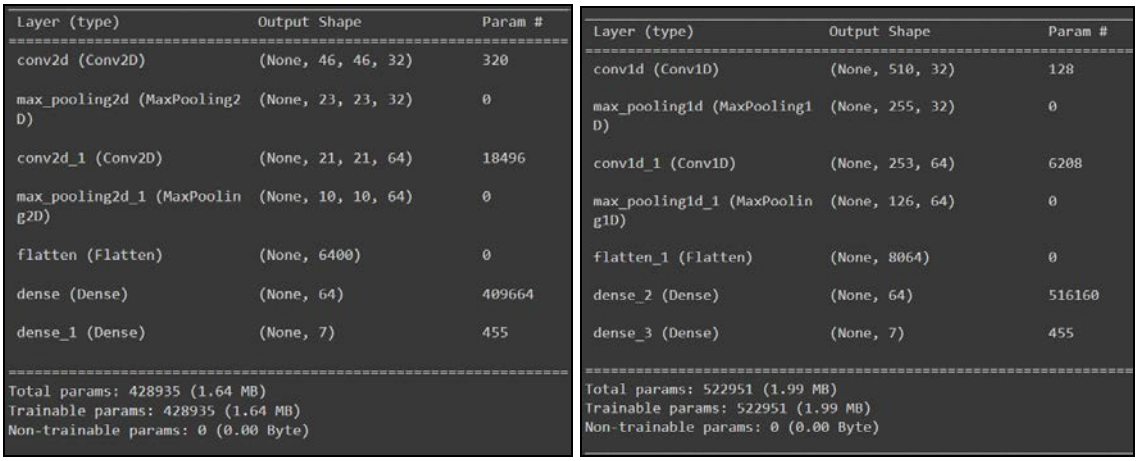


Figure 5. A typical architecture of a Random Forest classifier

The 2D convolutional layer consist of 32 filters and a kernel size of 3x3. The ReLU activation function is employed to collect spatial data from grayscale pictures of size 48x48. The architecture incorporates a max-pooling layer to decrease the spatial dimensions, succeeded by an additional convolutional layer consisting of 64 filters and a kernel size of 3x3. Next, the feature maps are transformed into a one-dimensional vector and sent through a fully linked dense layer consisting of 64 neurons, which are activated using the ReLU function. The last layer consists of 7 neurons that utilize a SoftMax activation function to produce probability for classifying different categories.

The 1D convolutional layer is composed of 32 filters with a kernel size of 3x3. It utilizes the ReLU activation function to extract local features from input sequences that have a size of 512. A max-pooling layer decreases the dimensionality. Subsequently, there is another convolutional layer comprising of 64 filters, followed by an extra max-pooling layer. The resulting output is

converted into a one-dimensional vector and then fed into a dense layer consisting of 64 neurons. The activation function used in this layer is ReLU, which helps to extract additional features from the data. The last layer consists of 7 neurons that utilize a softmax activation function to provide classification probabilities for each of the 7 classes. Figure 6 shows the summary of 1D and 2D CNN model taken in this project.



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 46, 46, 32)	320
max_pooling2d (MaxPooling2D)	(None, 23, 23, 32)	0
conv2d_1 (Conv2D)	(None, 21, 21, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 64)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 64)	409664
dense_1 (Dense)	(None, 7)	455

Total params: 428935 (1.64 MB)
Trainable params: 428935 (1.64 MB)
Non-trainable params: 0 (0.00 Byte)

(a)

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 510, 32)	128
max_pooling1d (MaxPooling1D)	(None, 255, 32)	0
conv1d_1 (Conv1D)	(None, 253, 64)	6208
max_pooling1d_1 (MaxPooling1D)	(None, 126, 64)	0
flatten_1 (Flatten)	(None, 8064)	0
dense_2 (Dense)	(None, 64)	516160
dense_3 (Dense)	(None, 7)	455

Total params: 522951 (1.99 MB)
Trainable params: 522951 (1.99 MB)
Non-trainable params: 0 (0.00 Byte)

(b)

Figure 6. Summary of CNN model taken in this project: (a) 2D CNN and (b) 1D CNN

2.5 Analysis Method

2.5.1 Confusion Matrix

A confusion matrix assesses the classification accuracy of a machine learning model by measuring the number of true positives, true negatives, false positives, and false negatives. True positives accurately detect positive cases, whereas true negatives accurately detect negative cases. False positives arise when negative instances are incorrectly categorized as positive, and false negatives occur when positive instances are incorrectly classed as negative. The confusion matrix offers important measures for analysis of accuracy, precision, recall, F1-score, and support. A general confusion matrix is shown in Table 3.

Table 3 Generalized confusion matrix

		Predicted Value	
		Positive	***Negative***
Actual Value	***Positive***	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

2.5.2 Accuracy and Precision

Accuracy shows the overall correctness of the model's predictions, generally a high accuracy indicates a good overall performance. However, not in every case work in this way. When the dataset is unbalanced (the number of samples in one class is much larger than the number of samples in the other classes), accuracy cannot be considered a reliable measure anymore, because it provides an overoptimistic estimation of the classifier ability on the majority class [29].

Precision on the other hand is a measure that shows the accuracy of the model that was predicted turned out to be true. This is how the reliability of the model is determined. Precision is a beneficial evaluation for scenarios where minimizing false positives is more important than avoiding false negatives.

2.5.3 Recall and F1-Score

Recall, also known as sensitivity or true positive rate, is the percentage of correctly categorized positive samples among all real positive samples. It evaluates the algorithm's ability to appropriately recognize affirmative situations.

The F1-score is calculated as the harmonic means of accuracy and recall. The assessment provides a comprehensive evaluation of the algorithm's performance, considering both its precision and recall. It is particularly helpful when dealing with classes that are imbalanced.

3. RESULTS AND DISCUSSION

3.1 Original Dataset

The original CKPLUS [19] posted by A. Shawon. consists of 981 images which are separated into 7 directories with label anger (135 images), contempt (54 images), disgust (177 images), fear (75 images), happy (207 images), sadness (84 images), and surprise (249 images). Each of these directories consist of a different number of 48x48 gray scaled Portable Network Graphic (PNG) images. Table 4 and Figure 7 shows the count of each facial expression consist in the dataset against the facial expression label.

Table 4 Facial expression count of original CKPLUS dataset

Facial Expression	Count
Anger	135
Contempt	54
Disgust	77
Fear	75
Happy	207
Sadness	84
Surprise	249

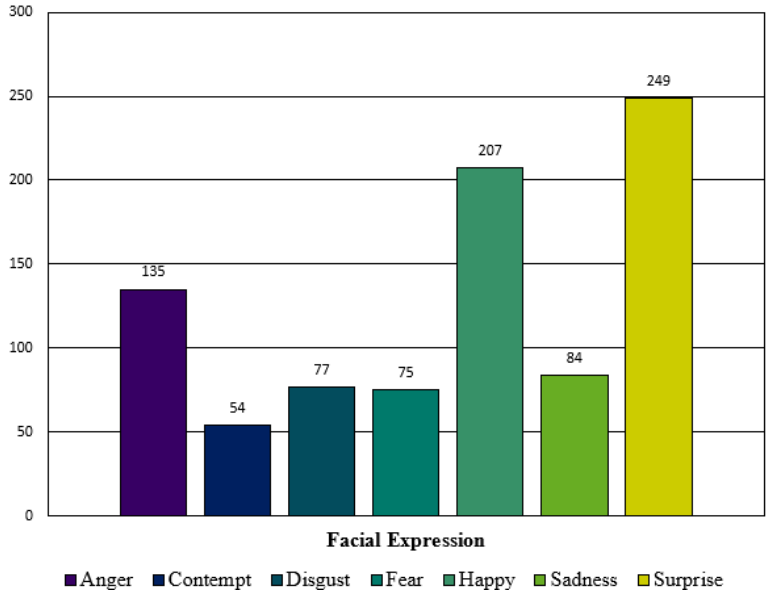


Figure 7. Original CKPLUS dataset distribution

3.2 Data Preprocessing

To address the issue of imbalanced class distribution within the dataset, data augmentation techniques were employed to artificially increase the number of images in underrepresented facial expression categories. This step was crucial to prevent model bias toward majority classes and to ensure that each emotion had equal representation during training. Through augmentation methods such as rotation, flipping, zooming, and slight translations, the dataset was balanced so that each facial expression category contained exactly 250 images. Figure 8 visually illustrates the balanced distribution across the seven facial expression categories. This uniformity across classes enhances the model's ability to learn features representative of all emotions rather than overfitting to the most frequent ones.

Following augmentation and balancing, the dataset was strategically split into training and validation subsets to evaluate the model's performance effectively. Each class was divided such that 70% of the images (175 per category) were allocated for training, while the remaining 30% (75 per category) were reserved for validation, as shown in Figure 9. The division was performed randomly to introduce variability and prevent the model from memorizing specific patterns from a fixed image sequence. This randomization ensured that the model encountered a diverse set of facial representations during both training and evaluation, thereby promoting better generalization and robustness. By maintaining a consistent image count and an unbiased distribution, the training process became more stable, and the validation phase provided a more meaningful assessment of the model's accuracy across all expression types.

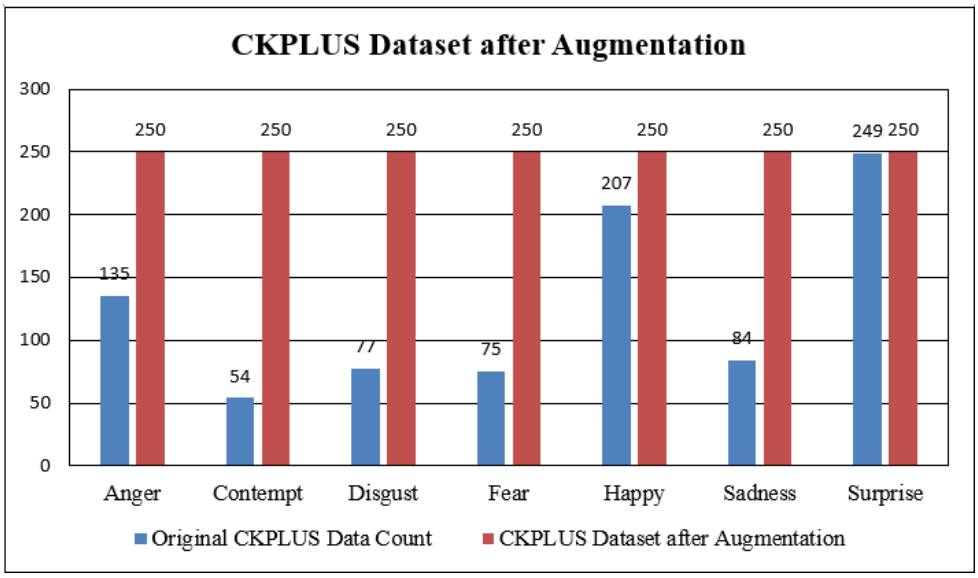


Figure 8. CKPLUS dataset distribution before and after augmentation

3.3 Feature Extraction and Visualization

3.3.1 Raw Pixel Extraction and Visualization

The raw grayscale images of the dataset, which are transformed into feature maps capturing essential information for distinguishing facial expressions are shown in Figure 9. Each feature map represents distinct facial features by accentuating differences in pixel brightness. These maps transform image intensities into arrays, which provide the basis for machine learning algorithms to classify facial emotions. Through the comparison of these arrays across several feature maps, a machine learning algorithm can detect patterns and effectively classify face emotions. This highlights the significance of proper pixel intensity mapping in the process.

The original grayscale images which have dimensions of 48x48 pixels, must be transformed into a 4D format in order to be used as input for a CNN. The necessary 4D dimensions are defined as a matric vector in the form of [`batch_size, height, width, channels`], with the values set to [-1, 48, 48, 1]. In this context, -1 represents the default batch size, 48 refers to the height and width, and 1 signifies grayscale images. The method of reshaping involves reorganizing the data matrix while preserving the original pixel information, allowing for the use of unprocessed pixel data in training and validation. The reshaped 4D pixel features are displayed in Figure 10.

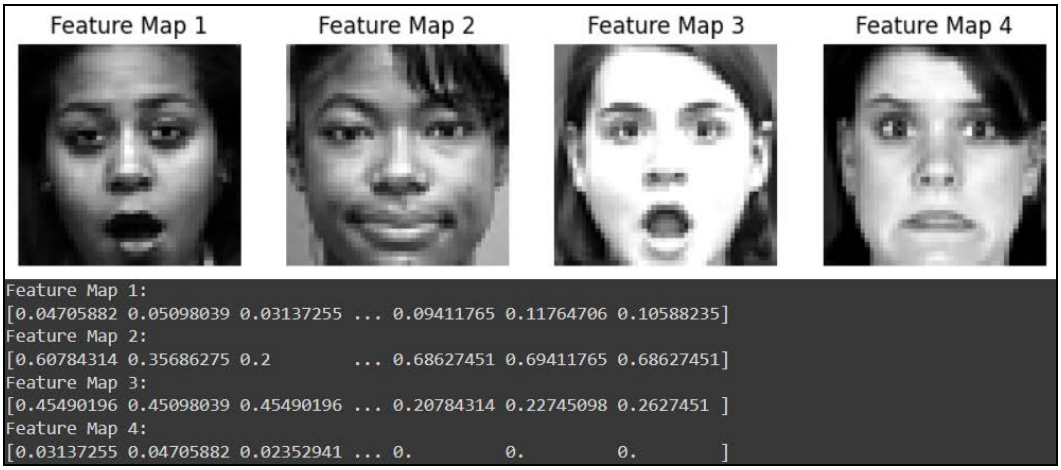


Figure 9. Raw pixel feature visualization

Figure 10. 4D reshaped raw pixel features visualization

3.3.2 Image Vector Feature Extraction and Visualization

The 512-dimensional vector features extracted using the pre-trained ResNet-18 model through the `Img2Vec` pipeline are high-level representations that encapsulate important facial attributes. However, due to the complexity and non-linearity of facial expression data, especially in datasets like CKPLUS, visualizing these features in their original dimensional space is not practical. To facilitate interpretation and gain insights into the distribution of expressions, the t-Distributed Stochastic Neighbour Embedding (t-SNE) technique is applied. This dimensionality reduction method projects the data into a 2D or 3D space while preserving the local structure of the original high-dimensional space. Figure 11 illustrates that the features do not form distinct clusters corresponding to different emotional classes. This lack of separation indicates that the model's learned representations may not be sufficiently discriminative for classification tasks using basic methods.

The observed overlap and scattering of data points suggest that the facial features extracted are highly entangled, making it challenging for linear classifiers or nearest neighbor algorithms to distinguish between emotion classes accurately. One plausible explanation for this ambiguity is the potential similarity or subtlety of facial expressions within the dataset, where emotional cues may be too nuanced for simple models to decode. Additionally, the compilation process of

the CKPLUS dataset, which may have emphasized uniform lighting and frontal poses, could have inadvertently led to homogeneous features across different expressions. As a result, more advanced techniques such as deep neural networks with non-linear decision boundaries, ensemble models, or additional domain-specific pre-processing are necessary to achieve reliable emotion recognition performance, which has been the discussion in this study.

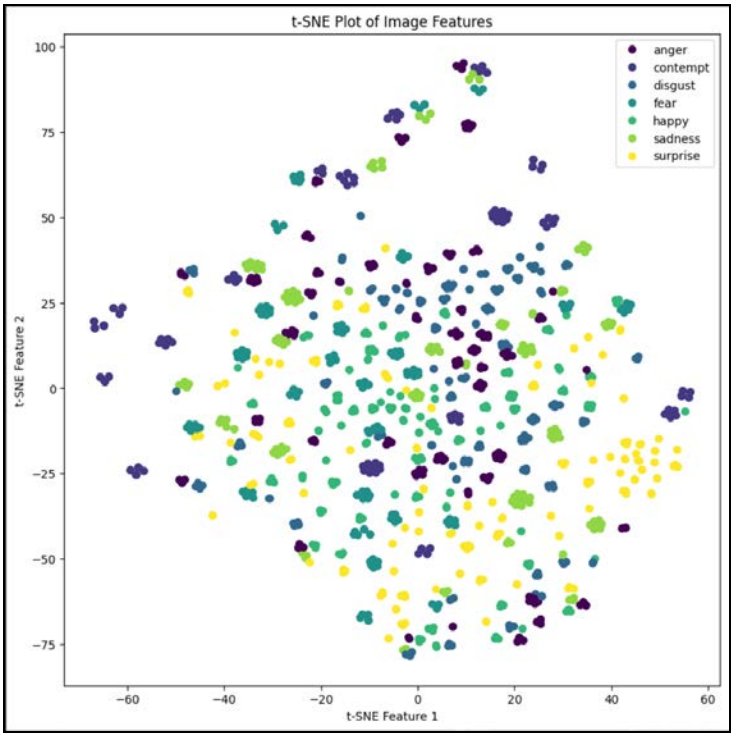
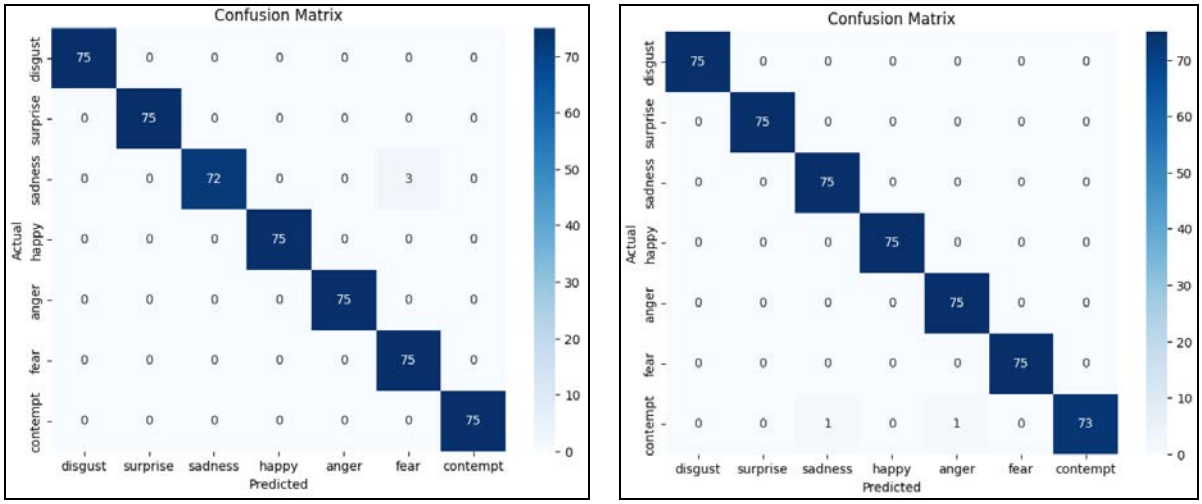


Figure 11. Vector features visualization using t-SNE dimensional reduction

3.4 Classification Comparison and Analysis

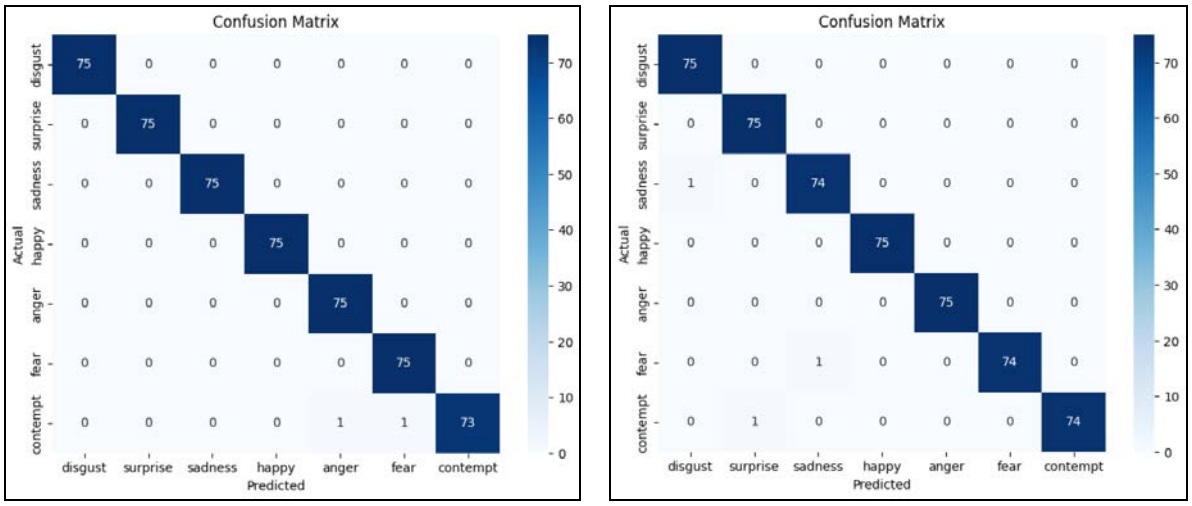
3.4.1 Confusion Matrix

The performance of classification models applied to facial expression recognition was evaluated using confusion matrices for different feature extraction methods and classifiers. Two main types of features were considered as discussed beforehand, which are raw pixel values and extracted feature vectors. The Random Forest classifier was applied to both types, as shown in Figure 12, while convolutional neural networks (CNNs) were used for classification in Figure 13. These matrices allow a direct comparison of how well each model performs under different input representations.



(a) (b)

Figure 12. Confusion matrix of Random Forest Classifier on: (a) raw pixel feature and (b) feature vector



(a) (b)

Figure 13. Confusion matrix of: (a) 2D CNN classifier on raw pixel feature and (b) 1D CNN classifier on vector feature

For the Random Forest classifier, the model performed moderately when using raw pixel features, with noticeable misclassifications across various expression categories. This result is expected since raw pixel data is high-dimensional and does not inherently represent meaningful patterns. However, when feature vectors were used instead, classification accuracy improved considerably. This improvement suggests that feature engineering or dimensionality reduction helps the Random Forest classifier to focus on more discriminative information, resulting in fewer classification errors.

The CNN-based models significantly outperformed the Random Forest approach. The 2D CNN, applied directly to raw pixel images, captured spatial features effectively and demonstrated strong classification performance with minimal misclassifications, as evidenced by the clearer diagonal patterns in the confusion matrix. This supports the advantage of deep learning architectures in processing visual data, as 2D CNNs are designed to exploit spatial hierarchies and local dependencies in images.

Meanwhile, the 1D CNN classifier, which was applied to the extracted feature vectors, also showed good performance, although it may slightly trail behind the 2D CNN if spatial relationships in the original data are essential. Nonetheless, both CNNs outperformed the traditional Random Forest classifier, indicating that deep learning methods are more effective for facial expression recognition tasks. Overall, the results highlight the importance of using appropriate feature representations and model architectures tailored to the data type for optimal classification performance.

3.4.2 Accuracy and Precision

Table 5 and Table 6 tabulated the accuracy and precision comparison of all four classifiers discussed in this study. Generally, all four classifiers achieved very high accuracy and precision, indicating strong overall performance in facial expression recognition. The accuracy values for all models range between 99.43% and 99.62%, showing only slight differences between them. Specifically, both the Random Forest with Vector Features and the 2D CNN with Raw Pixel Features achieved the highest accuracy at 99.6190%, demonstrating that both hand-crafted feature extraction and deep learning can be effective under the right conditions.

In terms of precision, which reflects how well the classifier avoids false positives for each emotion class, the results are also remarkably high. The Random Forest (Vector Feature) and 2D CNN (Raw Pixel Feature) each reached an average precision of 99.7143%, closely followed by the 1D CNN (Vector Feature) at 99.5714%. The Random Forest using Raw Pixel Features, while still strong, had the lowest average precision at 99.4286%, with the only slight dip observed in the "Fear" category (96%).

From a class-wise perspective, all classifiers performed perfectly or near-perfectly across most emotion categories, with only minor variations in precision. The most consistent performances across all seven classes were achieved by CNN-based models, suggesting their robustness in handling visual data. Random Forest benefited significantly from the use of feature vectors, improving precision over using raw pixels alone.

Overall, while the differences in performance metrics are small, the 2D CNN using raw pixel input and Random Forest using feature vectors stand out as the top performers. This confirms that both deep learning and classical machine learning can yield excellent results when paired with appropriate feature representations, but CNNs generally offer more consistent class-wise precision for visual classification tasks.

Table 5 Accuracy comparison between all four classifiers

Classification Method	Accuracy (%)
Random Forest (Raw Pixel Feature)	99.4286
Random Forest (Vector Feature)	99.6190
2D CNN (Raw Pixel Feature)	99.6190
1D CNN (Vector Feature)	99.4286

Table 6 Precision comparison between all four classifiers

Classification Method	Precision (%)							
	Disgust	**Surprise**	**Sadness**	**Happy**	**Anger**	**Fear**	**Contempt**	**Average**
Random Forest (Raw Pixel Feature)	100	100	100	100	100	96	100	99.4286
Random Forest (Vector Feature)	100	100	99	100	99	100	100	99.7143
2D CNN (Raw Pixel Feature)	100	100	100	100	99	99	100	99.7143
1D CNN (Vector Feature)	99	99	99	100	100	100	100	99.5714

3.4.3 Recall and F1-Scores

Table 7 and Table 8 tabulated the performance of the classifiers in terms of recall and F1-Scores. All four classifiers demonstrate that all four classification methods achieve high performance in emotion recognition tasks. The recall scores are consistently above 99% across all emotion categories, with the 1D CNN using vector features slightly outperforming the others with an average recall of 99.71%. This suggests that the 1D CNN is particularly effective at correctly identifying true positive cases across diverse emotional expressions.

In terms of F1-score, which balances precision and recall, the 2D CNN with raw pixel features and the Random Forest with vector features both lead with an average of 99.57%. This indicates that these models not only identify emotions accurately but also maintain a low rate of false positives. The strong performance of CNNs, especially the 2D variant, highlights their ability to capture spatial patterns in image data, making them well-suited for tasks involving facial emotion recognition.

Interestingly, the use of vector features appears to enhance the performance of Random Forest models, suggesting that feature engineering or dimensionality reduction can improve traditional machine learning approaches. Meanwhile, CNNs show robustness regardless of feature type, though 2D CNNs slightly edge out in F1-score, possibly due to their deeper architecture and spatial awareness.

Overall, while all models are highly effective, with 2D CNN offer a slight advantage in terms of balanced performance, and the 1D CNN excels in recall. These findings support the use of deep learning models for emotion classification, especially when high sensitivity and precision are required.

Table 7 Recall comparison

Classification Method	Recall (%)							
	Disgust	**Surprise**	**Sadness**	**Happy**	**Anger**	**Fear**	**Contempt**	**Average**
Random Forest (Raw Pixel Feature)	100	100	96	100	100	100	100	99.4286
Random Forest (Vector Feature)	100	100	100	100	100	100	97	99.5714
2D CNN (Raw Pixel Feature)	100	100	100	100	100	100	97	99.5714
1D CNN (Vector Feature)	100	100	99	100	100	99	99	99.7143

Table 8 F1-Scores comparison

Classification Method	F1-score (%)							
	Disgust	**Surprise**	**Sadness**	**Happy**	**Anger**	**Fear**	**Contempt**	**Average**
Random Forest (Raw Pixel Feature)	100	100	98	100	100	98	100	99.4286
Random Forest (Vector Feature)	100	100	99	100	99	100	99	99.5714
2D CNN (Raw Pixel Feature)	100	100	100	100	99	99	99	99.5714
1D CNN (Vector Feature)	99	99	99	100	100	99	99	99.2857

4. CONCLUSION

The CKPLUS Dataset is utilized in this research, and the issue of uneven picture count is addressed by employing data augmentation techniques. The data is divided into a training set comprising 70% of the data and a testing set comprising the remaining 30%. Feature extraction techniques encompass vector feature extracted from `Img2Vec` and raw pixel features, which are respectively displayed using t-SNE and `Matplotlib`.

The classification of these features is performed using Random Forest and CNN classifiers. The evaluation is done using a confusion matrix, accuracy, precision, recall, and F1-score. The result of Random Forest (raw pixel feature) obtained an accuracy of 99.4286%. This accuracy improved to 99.6190% when using vector features. The 2D CNN (raw pixel features) acquired an accuracy of 99.6190%, whilst the 1D CNN employing vector features attained an accuracy of 99.4286%.

The Random Forest model (vector feature) and the 2D CNN model with raw pixel features had the greatest precision scores, both reaching an accuracy of 99.7143%. The recall scores indicated that the 1D CNN with vector feature achieved a score of 99.71%, while the Random Forest with vector feature and the 2D CNN with raw pixel achieved scores of 99.57%. The Random Forest model using vector feature and the 2D CNN model with raw pixel had the greatest F1-scores, both reaching an accuracy of 99.57%. The findings suggest that among all the approaches, vector features exhibit the highest performance when combined with Random Forest, whereas raw pixel features improve the performance of CNN.

Within the topic of Facial Expression Recognition (FER), most existing datasets predominantly consist of Western faces. However, datasets from Southeast Asia, such as JAFEE, need authorization for access. Potential future endeavours may entail the development of a comprehensive dataset consisting of facial expressions specific to the Malaysian population. Feature extraction is crucial for optimizing machine learning models, and several techniques have yet to be investigated owing to time limitations. Subsequent investigation should prioritize these techniques to improve the performance of the model. Furthermore, it is advisable to investigate sophisticated categorization techniques, such as Kolmogorov-Arnold Networks (KAN). The implementation of KAN was impeded by the limited RAM capacity in Google Colab, underscoring the necessity of addressing computational limits for future research.

ACKNOWLEDGEMENTS

The authors would like to express gratitude to all personnel whom directly or indirectly involved in the studies. May Allah SWT blessed our everyday life and ease our way to Jannahtulfirdauws.

REFERENCES

[1] Markets and Markets, "Emotion Detection and Recognition (EDR) Market," *Emotion Detection and Recognition Market Size & Forecast*, Mar. 2023. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/emotion-detection-recognition-market-23376176.html

[2] A. S. Cowen, D. Keltner, F. Schroff *et al.*, "Sixteen facial expressions occur in similar contexts worldwide," *Nature*, vol. 589, pp. 251–257, 2021.

[3] [D. Matsumoto, "The Many Benefits of Reading Facial Expressions of Emotion," *humintell.com*, Apr. 6, 2021. [Online]. Available: https://www.humintell.com/2021/04/benefits-of-reading-facial-expressions-of-emotion/

[4] G. Benitez-Garcia, T. Nakamura, and M. Kaneko, "Methodical Analysis of Western-Caucasian and East-Asian Basic Facial Expressions of Emotions Based on Specific Facial Regions," *Journal of Signal and Information Processing*, vol. 8, pp. 78–98, 2017.

[5] Y. Wang, Y. Li, Y. Song, and X. Rong, "Facial expression recognition based on random forest and Convolutional Neural Network," *Information*, vol. 10, no. 12, 2019.

[6] Vandana and N. Marriwala, "Facial expression recognition using convolutional neural network," in *Advances in Intelligent Systems and Computing*, Springer, 2022.

[7] B. R. Ilyas, B. Mohammed, M. Khaled, A. T. Ahmed, and A. Ihsen, "Facial Expression Recognition Based on DWT Feature for Deep CNN," in *Proc. 6th Int. Conf. Control, Decision and Information Technologies (CoDIT)*, Paris, France, 2019, pp. 344–348.

[8] B. He, "Deep learning algorithm for Facial Expression Classification," in *Proc. CVIDL & ICCEA*, Changchun, China, 2022, pp. 386–391.

[9] G. Yolcu *et al.*, "Deep learning-based facial expression recognition for monitoring neurological disorders," in *Proc. IEEE Int. Conf. Bioinformatics and Biomedicine (BIBM)*, Kansas City, MO, USA, 2017, pp. 1652–1657.
[10] H. I. Dino and M. B. Abdulrazzaq, "Facial Expression Classification Based on SVM, KNN and MLP Classifiers," in *Proc. Int. Conf. Advanced Science and Engineering (ICOASE)*, Zakho - Duhok, Iraq, 2019, pp. 70–75.
[11] H. Du, Y. Chen, and Z. Shu, "Facial expression recognition algorithm based on local feature extraction," in *Proc. IEEE 4th Int. Conf. Power, Electronics and Computer Applications (ICPECA)*, Jan. 2024.
[12] K. Shan, J. Guo, W. You, D. Lu, and R. Bie, "Automatic facial expression recognition based on a deep convolutional-neural-network structure," in *Proc. IEEE 15th Int. Conf. Software Engineering Research, Management and Applications (SERA)*, London, UK, 2017.
[13] N. Arora, P. Yadav, K. Tripathi, and S. Sharma, "Performance of CNN for different facial expression images with varying input dataset sizes," in *Proc. Int. Conf. Device Intelligence, Computing and Communication Technologies (DICCT)*, Dehradun, India, 2023, pp. 351–355.
[14] K. C. Liu, C. C. Hsu, W. Y. Wang, and H. H. Chiang, "Real-Time Facial Expression Recognition Based on CNN," in *Proc. Int. Conf. System Science and Engineering (ICSSE)*, Dong Hoi, Vietnam, 2019, pp. 120–123.
[15] S. M. Gowri, A. Rafeeq, and S. Devipriya, "Detection of real-time Facial Emotions via Deep Convolution Neural Network," in *Proc. 5th Int. Conf. Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2021, pp. 1033–1037.
[16] M. I. N. P. Munasinghe, "Facial Expression Recognition Using Facial Landmarks and Random Forest Classifier," in *Proc. IEEE/ACIS 17th Int. Conf. Computer and Information Science (ICIS)*, Singapore, 2018, pp. 423–427.
[17] S. Bhogan, K. Sawant, N. Gondalekar, R. Carvalho, V. Kalangutkar, and A. Mathew, "Facial Emotion Detection using Machine Learning and Deep Learning Algorithms," in *Proc. 2nd Int. Conf. Edge Computing and Applications (ICECAA)*, Namakkal, India, 2023, pp. 1134–1139.
[18] H. Verma, "fer2013," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/datasets/deadskull7/fer2013
[19] A. Shawon, "CKPLUS Dataset," Kaggle, 2019. [Online]. Available: https://www.kaggle.com/datasets/shawon10/ckplus/data
[20] A. Alameer, "Facial Emotion Detection Dataset," University of Salford, 2023. [Online]. Available: https://doi.org/10.17866/rd.salford.22495669.v1
[21] A. Villacorta, "DataSetsSFEW," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/airanvillacorta/datasetssfew
[22] A. Singh, "3 Beginner-Friendly Techniques to Extract Features from Image Data using Python," *Analytics Vidhya*. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/
[23] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
[24] C. Safka, "Extract a feature vector for any image with PyTorch," *Becoming Human: Artificial Intelligence Magazine*, *Medium*. [Online]. Available: https://becominghuman.ai/extract-a-feature-vector-for-any-image-with-pytorch-9717561d1d4c
[25] C. Safka, "img2vec-pytorch: Use pre-trained models in PyTorch to extract vector embeddings for any image," *PyPI*. [Online]. Available: https://pypi.org/project/img2vec-pytorch/
[26] A. A. Awan, "Introduction to t-SNE," *DataCamp*. [Online]. Available: https://www.datacamp.com/tutorial/introduction-t-sne
[27] Amandp13, "Random Forest classifier using Scikit-learn," *GeeksforGeeks*. [Online]. Available: https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/

[28] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Nov. 2015. [Online]. Available: https://www.tensorflow.org
[29] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*.