Ensemble semi-supervised learning in facial expression recognition

Purnawansyah [a,1], Adam Adnan [a,2,*], Herdianti Darwis [a,3], Aji Prasetya Wibawa [b,4], Triyanna Widyaningtyas [b,5], Haviluddin [c,6]

[a] Faculty of Computer Science, Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM 5, Makassar, 90231, Indonesia
[b] Universitas Negeri Malang, Jl. Semarang No. 5, Malang, 65145, Indonesia
[c] Universitas Mulawarman, Jl. Kuaro, Samarinda, 75119, Indonesia
[1] purnawansyah@umi.ac.id; [2] adamadnan.iclabs@umi.ac.id; [3] herdianti.darwis@umi.ac.id; [4] aji.prasetya@um.ac.id; [5] triyannaw.ft@um.ac.id; [6] haviluddin@unmul.ac.id
* corresponding author

ARTICLE INFO

Article history
Received December 09, 2024
Revised January 30, 2025
Accepted February 02, 2025
Available online February 03, 2025

Keywords
Deep learning
Ensemble learning
Facial expression recognition
Machine learning

ABSTRACT

Facial Expression Recognition (FER) plays a crucial role in human-computer interaction, yet improving its accuracy remains a significant challenge. This study aims to enhance the robustness and effectiveness of FER systems by integrating multiple machine learning techniques within a semi-supervised learning framework. The primary objective is to develop a more effective ensemble model that combines Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Support Vector Classifier (SVC), and Random Forest classifiers, utilizing both labeled and unlabeled data. The research implements data augmentation and feature extraction techniques, utilizing advanced architectures such as VGG19, ResNet50, and InceptionV3 to improve the quality and representation of facial expression data. Evaluations were conducted across three dataset scenarios: original, feature-extracted, and augmented, using various label-to-unlabeled ratios. The results indicate that the ensemble model achieved a notable accuracy improvement of 87% on the augmented dataset compared to individual classifiers and other ensemble methods, demonstrating superior performance in handling occlusions and diverse data conditions. However, several limitations exist. The study's reliance on the JAFFE dataset may restrict its generalizability, as it may not cover the full range of facial expressions encountered in real-world scenarios. Additionally, the effect of label-to-unlabeled ratios on the model's performance requires further exploration. Computational efficiency and training time were also not evaluated, which are critical considerations for practical implementation. For future research, it is recommended to employ cross-validation methods for more robust performance evaluation, explore additional data augmentation techniques, optimize ensemble configurations, and address the computational efficiency of the model to better advance FER technologies.

1. Introduction

Facial Expression Recognition (FER) has emerged as a critical technology in the realm of human-machine interaction and is witnessing rapid advancements within the fields of computer vision and affective computing [1]–[3] . In an era dominated by digital technology, the capacity for machines to interpret human emotions through the analysis of facial expressions is becoming increasingly vital. FER

https://doi.org/10.26555/ijain.v11i1.1880

encompasses a diverse array of practical applications, ranging from the enhancement of user experiences on smart devices to the improvement of interactions within virtual reality environments [4], [5]. For instance, in social media platforms, FER systems can facilitate content recommendations based on users' emotional responses [6], [7]. In the healthcare domain, FER can play a significant role in monitoring patients' emotional states and identifying psychological conditions such as depression or anxiety [8]. Additionally, in educational settings, this technology can be leveraged to create adaptive learning environments that respond to the emotional states of students, thereby fostering greater engagement and improving educational outcomes [9], [10].

Despite the significant potential of this technology, the development of FER systems continues to encounter various challenges [11]. Factors such as variability in facial expressions, differing lighting conditions, and diverse facial poses can adversely affect the accuracy of emotion recognition [12]–[14]. For instance, under low-light conditions, these systems often struggle to distinguish between similar expressions, which diminishes the reliability of emotion detection [15]. Furthermore, traditional supervised learning approaches frequently face limitations due to their requirement for large volumes of labeled data, which are not always accessible in many real-world application contexts [16]–[18].

To address these challenges, semi-supervised approaches have emerged as a promising solution [19], [20]. This methodology leverages a combination of labeled and unlabeled data to train models more efficiently, thereby reducing reliance on expensive and hard-to-obtain labeled datasets while maintaining or even enhancing model performance [21]–[24]. Previous research has also indicated that data augmentation techniques, which expand the training dataset with additional variations of the same images, can help mitigate overfitting issues and improve model generalization [25]–[27]. The integration of various machine learning models alongside ensemble learning techniques has demonstrated potential for optimizing both the accuracy and robustness of the models by capitalizing on the strengths of each model while minimizing their individual weaknesses [28], [29].

This study utilizes the JAFFE dataset, which comprises images of facial expressions from Japanese women categorized into seven distinct emotions. The dataset is processed into three different scenarios: the original dataset, an augmented dataset, and a dataset resulting from feature extraction. The research adopts a semi-supervised approach combined with ensemble learning to develop a more robust model by leveraging a variety of classifiers within this framework. Data partitioning is conducted to separate the training and testing datasets while maintaining a focus on the research objectives and the advantages offered by the proposed methods.

This study has several key contributions, namely:

- Development of a Comprehensive FER Framework: Integrating individual classification approaches and ensemble techniques within a semi-supervised paradigm to enhance the accuracy of facial expression recognition.
- Application of Data Augmentation and Feature Extraction Strategies: Improving data quality and feature representation through augmentation and extraction techniques.
- Comparative Analysis Across Various Scenarios: Conducting an in-depth investigation of model performance across three scenarios: the original dataset, the feature extraction dataset, and the augmented dataset, as well as various ratios of labeled and unlabeled data.
- Evaluation Using Various Metrics Methods: Employing comprehensive evaluation metrics to provide a thorough assessment of model performance and robustness.

The remainder of this article is structured as follows: Section 2 describes the research methodology, covering the dataset, data augmentation approaches, and feature extraction and classification methods, including ensemble techniques. Section 3 presents the experimental results and a detailed analysis of model performance. Finally, Section 4 summarizes the key findings of this research and offers recommendations for future work.

2. Method

This research employs a comprehensive approach to classify facial expressions using an ensemble model. Data from the JAFFE dataset, comprising images of Japanese women's facial expressions, is processed in three scenarios: original, augmentation, and feature extraction. The original scenario images undergo direct resizing and normalization, while the augmentation scenario involves prior image augmentation before resizing and normalization. Feature extraction is performed using three deep learning architectures: VGG19, ResNet50, and InceptionV3. The data is then divided into training and testing sets. Various models, including LSTM, CNN, SVC, and Random Forest, are trained using the training data and evaluated using metrics such as Accuracy, Kullback-Leibler Divergence, Intersection Jaccard, F1 Score, Cosine Similarity, and ROC-AUC. After the models generate predictions, these predictions are combined and used as input for an ensemble Random Forest Classifier. Finally, the performance of this ensemble model is evaluated. The methodological design and workflow are illustrated in the accompanying Fig. 1.

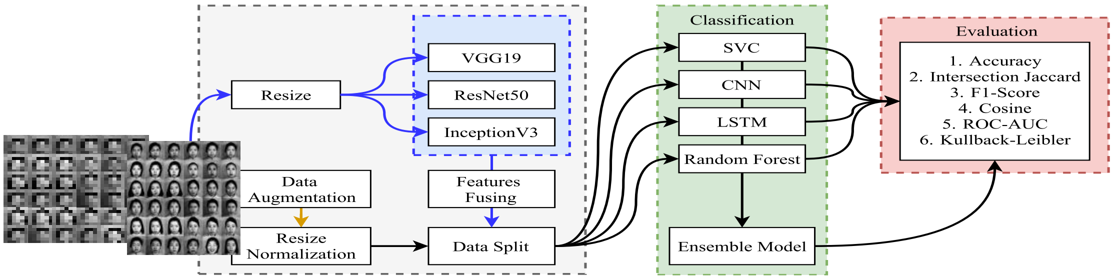


Fig. 1. Research Design

2.1. Dataset

The JAFFE dataset is well-established in the field of facial expression recognition research. It comprises 213 grayscale images, each measuring 256x256 pixels, depicting ten Japanese female subjects displaying one of seven distinct facial expressions: angry, disgusted, fearful, happy, neutral, sad, and surprised. These expressions were performed by professional models to ensure consistency and accuracy. The JAFFE dataset is extensively utilized in pattern recognition, computer vision, and machine learning investigations, as its precisely annotated images in .tiff or .png format are invaluable for developing and evaluating algorithms designed to accurately classify human facial expressions. Illustrative examples of these facial expressions from the JAFFE dataset can be found in Fig. 2.

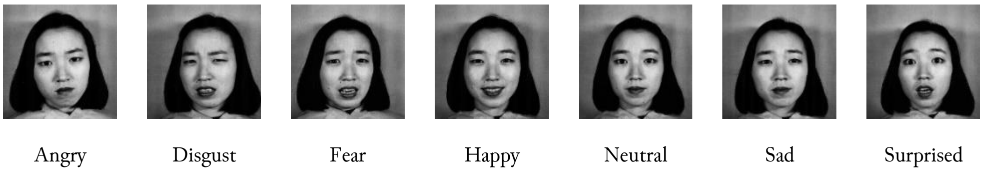


Fig. 2. Sample Images from JAFFE Dataset

2.2. Resize & Normalization

In this study, image preprocessing was conducted to ensure consistent inputs for the models. Specifically, the images were resized to 150x150 pixels to standardize their dimensions, which aligns with the model's input requirements. Furthermore, normalization was applied by dividing the pixel values by 255.0, transforming them into a range of 0 to 1. This normalization process enhances the training efficiency and stability by ensuring that the input values are on a comparable scale.

2.3. Data Augmentation

Occlusion, where objects or parts of objects to be recognized are partially obscured by other elements, poses a significant challenge in computer vision and pattern recognition [30]. To address this issue and enhance the robustness of facial expression recognition models, our research employed occlusion techniques by utilizing the Albumentations library with the coarse dropout method [27]. Coarse dropout is a data augmentation technique that randomly covers portions of an image with squares, mimicking real-world occlusion scenarios. By applying coarse dropout during the training process, our model learns to recognize crucial facial features even when certain parts are obscured. This approach enables the model to be more effective in handling diverse forms of occlusion, consequently improving the overall performance in facial expression recognition [31]. The application of this technique is exemplified in Fig. 3, which demonstrates the implementation of occlusion to simulate realistic conditions.

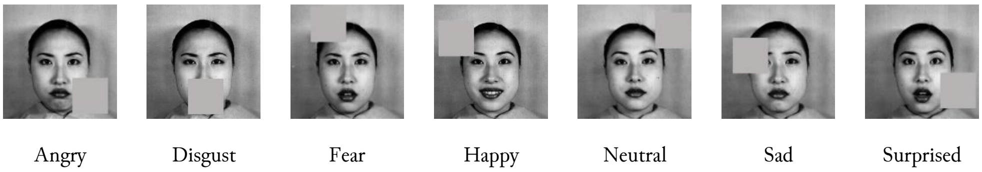


Fig. 3. Augmented JAFFE Dataset Samples Showing Occlusion Effects

2.4. Feature Extraction

This is a brief explanation of the three deep learning architectures used for feature extraction. The model architecture employed in this study is renowned for its simplicity and robust performance in image classification tasks. It consists of 19 layers, including 16 convolutional layers and 3 fully connected layers. Each convolutional layer utilizes a 3x3 kernel to preserve spatial dimensions, followed by 2x2 max pooling to reduce spatial dimensions. The input images are resized to 224x224 pixels to align with the architecture's input requirements. These convolutional layers are responsible for extracting crucial features, such as edges, textures, and shapes, which are then leveraged as data representations for subsequent classification models [32]. The architecture is illustrated in Fig. 4.

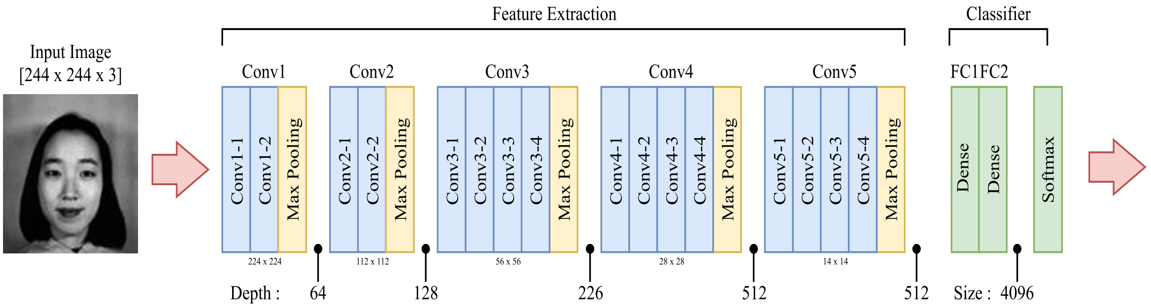


Fig. 4. Architecture of VGG-19

ResNet50, developed by Microsoft Research as part of the Residual Networks family, is a 50-layer deep neural network architecture that incorporates convolutional layers, batch normalization, and identity shortcuts within its residual blocks. The core concept of ResNet50 is residual learning, which directly passes the input identity to the next layer without modification, effectively mitigating the vanishing gradient issue in very deep networks [33]. The architecture is illustrated in Fig. 5.

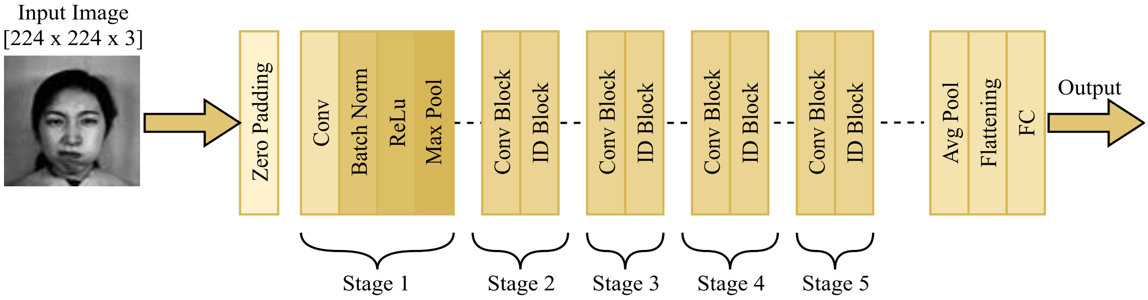


Fig. 5. Basic ResNet50 architecture

The InceptionV3 model, developed by Google and known as GoogLeNet, leverages Inception modules to efficiently extract features from images. This architecture integrates filters of diverse sizes within a single layer, enabling the model to effectively capture information at various scales. As shown in Fig. 6, Through the application of techniques such as batch normalization, RMSProp optimization, and factorized convolutions, InceptionV3 not only enhances performance but also improves computational efficiency. For optimal feature extraction, the InceptionV3 model requires input images to be at least 299x299 pixels in size [34].

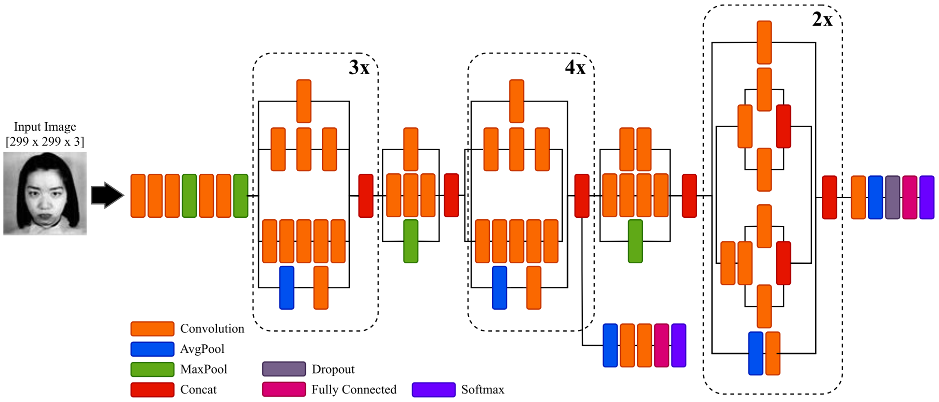


Fig. 6. Basic inception-V3 architecture

Following the feature extraction stage utilizing deep learning architectures, the resultant datasets are stored in individual dataframes corresponding to each architecture. The subsequent step involves consolidating these three dataframes into a single merged dataframe by applying the merge function. This process synthesizes the features represented by each architecture into a unified dataset. Each row in the merged dataframe corresponds to a single sample or image from the original dataset, with the columns representing the features extracted from the respective architectures. This combined dataframe serves as the input for further analysis or classification within the scope of this study.

2.5. Data Split

Data partitioning is a crucial step in developing machine learning models to ensure fair and representative training and testing [35]. In this study, the JAFFE dataset is divided into two main parts: 80% for training and 20% for testing. The purpose of this division is to provide enough data for the model to learn relevant patterns while reserving sufficient data for performance evaluation. Additionally, the training data is further split into labeled and unlabeled data with ratios of 20:80, 25:75, and 50:50, allowing for the application of semi-supervised learning. This partitioning helps us assess the effectiveness of the semi-supervised learning approach under various conditions. The data partitioning structure is summarized in Fig. 7, which illustrates the flow of data partitioning up to the stages of model training and testing.

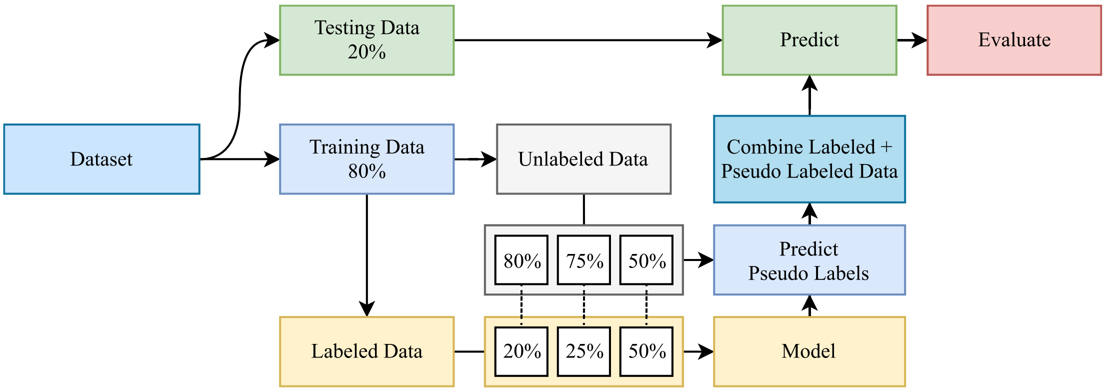


Fig. 7. Data Splitting Scenarios

2.6. Classification

This study employs a range of models and techniques to classify facial expressions, with the goal of enhancing the accuracy of the proposed system. As each model or method possesses unique strengths and limitations, the integration of these approaches into a single ensemble system is expected to mitigate the shortcomings of individual models, culminating in a more effective facial expression recognition system.

The Support Vector Classifier (SVC) is a supervised machine learning technique that determines the optimal separating hyperplane for classifying data, whether in linear or non-linear problems, by utilizing various kernel functions [36], [37]. To optimize SVC performance, several parameters were tested, including a linear kernel with C = 1, polynomial (poly) kernels with degrees of 2, 3, and 4 and C = 1, and a radial basis function (RBF) kernel with different gamma values: 0.001, 0.01, 0.1, 1, and 10. All configurations employed random_state=42 and probability=True to ensure stable results and enable probability predictions. Fig. 8 illustrates this hyperplane and margin concept.

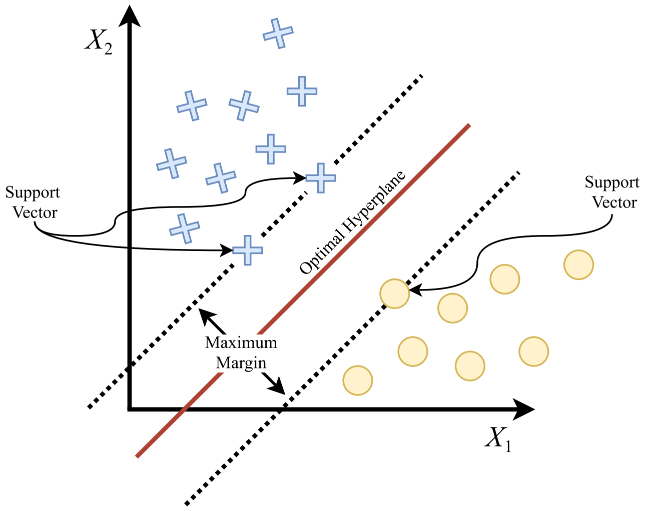


Fig. 8. SVC Algorithm

Long Short-Term Memory (LSTM) is a recurrent neural network architecture designed to process sequential data and address the vanishing gradient problem [38]. LSTM employs three key components input gate, forget gate, and output gate to regulate the flow of information, enabling the model to retain or discard information over the long term, as illustrated in Fig. 9. The LSTM model used in this study consists of three LSTM layers with 128, 64, and 32 units, followed by a dense layer with 64 neurons and a ReLU activation function, and an output dense layer with 7 neurons and a softmax activation function. The model was compiled using the 'adam' optimizer, 'sparse_categorical_crossentropy' loss, and 'accuracy' as the evaluation metric. With this configuration, the LSTM model is expected to yield optimal classification results and improve prediction accuracy.

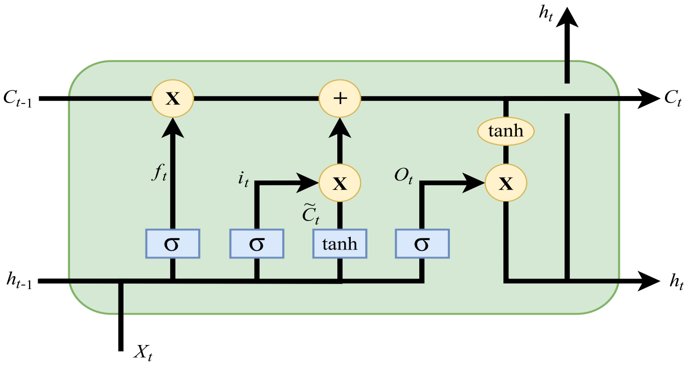


Fig. 9. LSTM Models

Convolutional Neural Networks (CNNs) are specialized network architectures designed to process grid-like data, such as images, using convolutional layers to extract relevant features and pooling layers to reduce dimensionality and computational complexity [39]. The final stage typically consists of fully connected layers that perform classification based on the extracted features [39]. Fig. 10 illustrates the CNN architecture. In this study, the CNN model was implemented with a Conv2D layer using 32 filters, a 3x3 kernel, ReLU activation, and an input shape of (150, 150, 3). For dimensionality reduction, MaxPooling2D with a 2x2 kernel was applied, followed by a flatten layer and a dense layer with 64 neurons and ReLU activation.

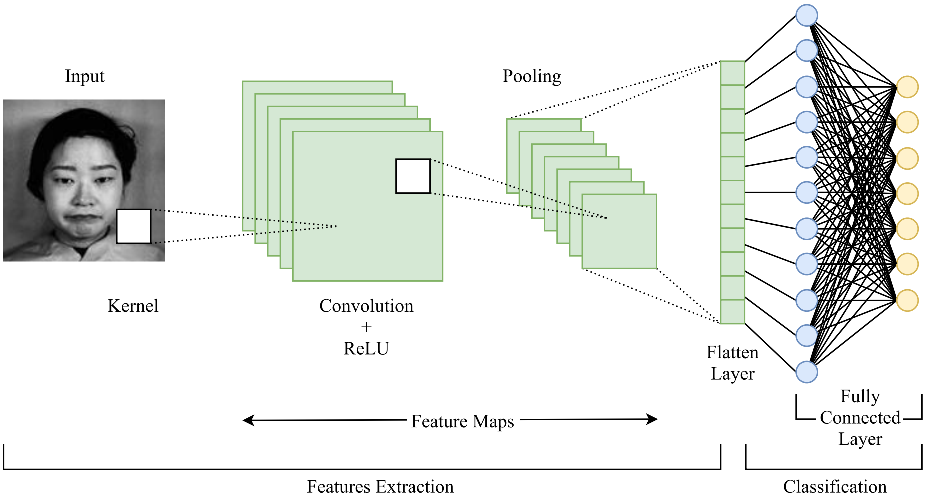


Fig. 10. CNN Architecture

The final layer consisted of a dense layer with 7 classes and a softmax activation function. The model was compiled using the 'adam' optimizer, 'sparse_categorical_crossentropy' loss, and 'accuracy' as the evaluation metric. This configuration is expected to enable the CNN model to deliver optimal performance in image classification tasks. Fig. 10 shows the CNN architecture.

Random Forest is an ensemble method that combines multiple decision trees to improve prediction accuracy and model robustness. The final prediction is obtained by averaging or majority voting from the individual trees [40], as shown in Fig.11. The parameters used include n_estimators = 100, which specifies the number of trees, and random_state = 42, which ensures consistent results.

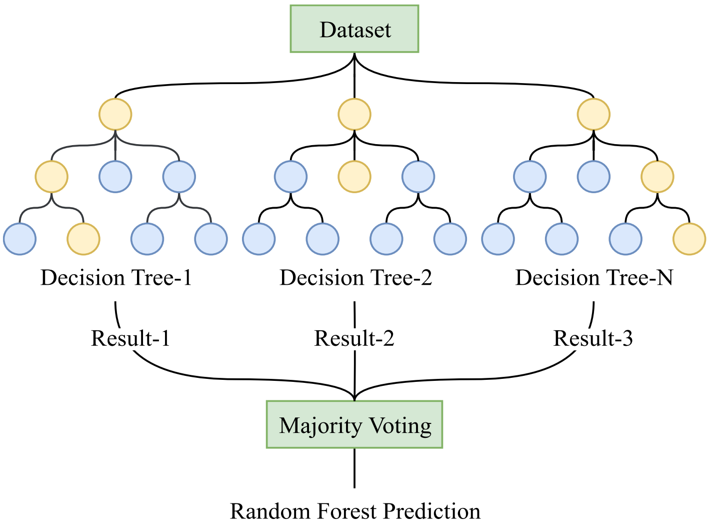


Fig. 11. Random Forest Workflow

The proposed ensemble approach combines predictions from several distinct models, including SVC, LSTM, CNN, and Random Forest, to improve accuracy and consistency in facial expression classification. Each model is independently trained on the available training data, and their individual predictions are then merged using a stacking technique. The stacking process involves treating the predictions from each model as additional features, which are then used as input to an ensemble Random Forest Classifier. This ensemble model utilizes both the stacked predictions and previously extracted features to generate the final classification outcome. By leveraging the complementary strengths of various model architectures, this ensemble learning strategy aims to optimize classification performance, address different data characteristics, and significantly improve the overall accuracy of the facial expression recognition system. The total number of model combinations formed from these 4 models is 15, with the final classification performed using Random Forest [41].

2.7. Evaluation Metrics

A variety of performance metrics are utilized to gain a comprehensive understanding of model capabilities. These include Accuracy, which quantifies the frequency of correct predictions; Kullback-Leibler Divergence, which assesses the discrepancy between predictions and ground truth [42]; Intersection Jaccard, which evaluates the similarity between predictions and reality [43]; F1-Score, which synthesizes precision and recall; Cosine Similarity, which measures the likeness between two predictions [44]; and ROC-AUC, which examines the balance of true and false predictions [45]. These metrics enable the analysis of individual model strengths and weaknesses, as well as the comparative performance of the ensemble model relative to standalone models. Detailed equations for these metrics can be found in (1) to (6).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+TN} \tag{1}$$

$$Intersection\ (Jaccard)\ =\ J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

$$F1 - Score\ = \frac{2 \cdot \frac{TP}{TP+FP} \cdot \frac{TP}{TP+FP}}{\left(\frac{TP}{TP+FP} + \frac{TP}{TP+FN}\right)} \tag{3}$$

$$\text{Cosine} = \frac{\text{J} \cdot \text{K}}{\|\text{J}\| \times \|\text{K}\|} \tag{4}$$

$$ROC - AUC\ =\ \int_0^1 \left(\frac{TP}{TP+FN}\right) d\ \left(\frac{FP}{FP+FN}\right) \tag{5}$$

$$Kullback - Leibler = D_{\text{K}\Lambda}(P \parallel Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \tag{6}$$

3. Results and Discussion

3.1. Experiments on Original JAFFE Dataset

Experiments on the original JAFFE dataset revealed variations in model performance based on the labeled and unlabeled data splits. In the 20:80 split, detailed in Table 1, the SVC model with a linear kernel achieved the highest accuracy of 0.38, an Intersection Jaccard (I-J) of 0.23, and an F1-Score of 0.36, while recording the lowest Kullback-Leibler Divergence of 2.27.

Table 1. Results for Original JAFFE (20:80)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.29	0.16	0.27	0.67	0.62	4.04
	LSTM				0.16	0.02	0.04	0.83	0.50	3.16
	Random Forest				0.29	0.18	0.28	0.79	0.54	4.95
	SVC	*Linear*			0.38	0.23	0.36	0.74	0.63	2.27
		Poly	*Degree*	2	0.29	0.17	0.28	0.75	0.62	2.36
				3	0.33	0.19	0.30	0.75	0.61	2.41
				4	0.33	0.19	0.31	0.74	0.61	2.42
		RBF	*Gamma*	0.001	0.24	0.10	0.18	0.68	0.54	2.58
				0.01	0.16	0.05	0.09	0.15	0.53	2.35
				0.1	0.13	0.01	0.03	0.83	0.48	2.39
				1	0.13	0.01	0.03	0.83	0.50	2.56
				10	0.13	0.01	0.03	0.83	0.50	3.19

[a.] RF : Random Forest; I-J : Intersection (Jaccard); K-L : Kullback Leibler

Both the SVC with an RBF kernel and LSTM attained the highest Cosine Similarity score of 0.83. In the 25:75 split, as shown in Table 2, Random Forest produced the best results with an accuracy of 0.38, an I-J of 0.28, and an F1-Score of 0.40, alongside the highest ROC-AUC of 0.65. The SVC with an RBF kernel achieved the lowest Kullback-Leibler Divergence of 2.12, and the same SVC configuration reached the highest Cosine Similarity score of 0.83.

Table 2. Results for Original JAFFE (25:75)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.27	0.14	0.24	0.72	0.63	2.84
	LSTM				0.18	0.06	0.10	0.70	0.49	2.46
	Random Forest				0.38	0.28	0.40	0.77	0.65	2.84
	SVC	*Linear*			0.38	0.25	0.39	0.76	0.64	2.38
		Poly	*Degree*	2	0.29	0.17	0.28	0.72	0.63	2.35
				3	0.33	0.22	0.34	0.73	0.64	2.31
				4	0.31	0.19	0.31	0.73	0.62	2.38
		RBF	*Gamma*	0.001	0.22	0.10	0.17	0.68	0.56	2.36
				0.01	0.16	0.02	0.04	0.83	0.57	2.21
				0.1	0.16	0.02	0.04	0.83	0.51	2.45
				1	0.13	0.01	0.03	0.83	0.50	2.12
				10	0.13	0.01	0.03	0.83	0.49	2.91

For the 50:50 split, detailed in Table 3, the SVC with a Polynomial kernel of Degree 3 stood out with an accuracy of 0.69, an I-J of 0.53, an F1-Score of 0.68, a Cosine Similarity of 0.94, and the lowest Kullback-Leibler Divergence of 1.09. The CNN achieved the highest ROC-AUC of 0.90, while the SVC with a Polynomial kernel of Degree 2 recorded the highest Cosine Similarity of 0.94

Table 3. Results for Original JAFFE (50:50)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.51	0.37	0.52	0.88	0.90	1.30
	LSTM				0.18	0.04	0.07	0.49	0.53	2.19
	Random Forest				0.62	0.44	0.60	0.87	0.84	1.32
	SVC	*Linear*			0.64	0.46	0.61	0.89	0.85	1.25
		Poly	*Degree*	2	0.64	0.49	0.64	0.94	0.86	1.22
				3	0.69	0.53	0.68	0.94	0.89	1.09
				4	0.67	0.52	0.66	0.93	0.89	1.14
		RBF	*Gamma*	0.001	0.47	0.33	0.47	0.90	0.77	1.53
				0.01	0.16	0.02	0.04	0.83	0.63	1.95
				0.1	0.16	0.02	0.04	0.83	0.49	2.26
				1	0.13	0.01	0.03	0.83	0.50	2.00
				10	0.13	0.01	0.03	0.83	0.49	2.38

3.2. Ensemble Classification Results for Original JAFFE Dataset

The evaluation of the ensemble model on the original JAFFE dataset demonstrated significant performance improvements across various labeled-to-unlabeled data partitions. For the 20:80 labeled-to-unlabeled ratio, detailed in Table 4 , the ensemble model combined with Random Forest (+ RF) chieved the highest accuracy of 0.69, an Intersection Jaccard (I-J) of 0.56, an F1-Score of 0.71, a Cosine Similarity of 0.90, a ROC-AUC of 0.96, and a Kullback-Leibler Divergence of 0.72.

Table 4. Results for Original JAFFE - Ensemble (20:80)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.38	0.20	0.31	0.78	0.81	1.41
	(LSTM) + RF	0.16	0.02	0.04	0.83	0.52	1.94
	(CNN) + RF	0.29	0.14	0.24	0.85	0.72	1.62
	(RF) + RF	0.38	0.21	0.32	0.83	0.77	1.52
	(SVC, LSTM) + RF	0.38	0.20	0.31	0.78	0.81	1.41
	(SVC, CNN) + RF	0.53	0.37	0.53	0.81	0.91	1.00
	(LSTM, CNN) + RF	0.29	0.14	0.24	0.85	0.72	1.62
	(SVC, RF) + RF	0.58	0.40	0.53	0.89	0.92	0.97
	(LSTM, RF) + RF	0.38	0.21	0.32	0.83	0.77	1.52
	(CNN, RF) + RF	0.47	0.28	0.42	0.83	0.89	1.09
	(SVC, LSTM, CNN) + RF	0.53	0.37	0.53	0.81	0.91	1.00
	SVC, LSTM, RF) + RF	0.58	0.40	0.53	0.89	0.92	0.97
	(SVC, CNN, RF) + RF	0.69	0.56	0.71	0.90	0.96	0.72
	(LSTM, CNN, RF) + RF	0.47	0.28	0.42	0.83	0.89	1.09
	(SVC, LSTM, CNN, RF) + RF	0.69	0.56	0.71	0.90	0.96	0.72

Similarly, in the 25:75 split, as shown in Table 5, the ensemble model combined with Random Forest (+ RF) showed optimal results, with an accuracy of 0.73, an I-J of 0.61, an F1-Score of 0.74, a Cosine Similarity of 0.91, a ROC-AUC of 0.97, and a Kullback-Leibler Divergence of 0.69.

Table 5. Results for Original JAFFE - Ensemble (25:75)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.33	0.19	0.31	0.70	0.78	1.49
	(LSTM) + RF	0.20	0.06	0.11	0.70	0.61	1.86
	(CNN) + RF	0.29	0.13	0.21	0.78	0.73	1.62
	(RF) + RF	0.40	0.28	0.40	0.79	0.82	1.35
	(SVC, LSTM) + RF	0.42	0.29	0.44	0.68	0.85	1.25
	(SVC, CNN) + RF	0.51	0.34	0.51	0.79	0.90	1.08
	(LSTM, CNN) + RF	0.42	0.27	0.41	0.75	0.86	1.24
	(SVC, RF) + RF	0.58	0.43	0.59	0.83	0.92	0.97
	(LSTM, RF) + RF	0.51	0.37	0.51	0.84	0.87	1.16
	(CNN, RF) + RF	0.62	0.47	0.64	0.85	0.94	0.85
	(SVC, LSTM, CNN) + RF	0.60	0.44	0.60	0.83	0.94	0.91
	SVC, LSTM, RF) + RF	0.62	0.48	0.63	0.87	0.93	0.91
	(SVC, CNN, RF) + RF	0.69	0.54	0.69	0.89	0.96	0.74
	(LSTM, CNN, RF) + RF	0.67	0.53	0.68	0.90	0.95	0.79
	(SVC, LSTM, CNN, RF) + RF	0.73	0.61	0.74	0.91	0.97	0.69

For the 50:50 split, detailed in Table 6, the ensemble model with Random Forest (+ RF) achieved the best overall outcomes, including an accuracy of 0.82, an I-J of 0.69, an F1-Score of 0.81, a Cosine Similarity of 0.95, a ROC-AUC of 0.99, and a Kullback-Leibler Divergence of 0.38. Across all scenarios, the ensemble models consistently demonstrated improved performance, with the lowest KullbackLeibler Divergence observed in the 50:50 split, indicating that the ensemble model more accurately approximates the underlying data distribution.

Table 6. Results for Original JAFFE - Ensemble (50:50)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.69	0.51	0.63	0.94	0.95	0.74
	(LSTM) + RF	0.18	0.04	0.07	0.49	0.58	1.89
	(CNN) + RF	0.53	0.35	0.49	0.83	0.90	1.05
	(RF) + RF	0.62	0.43	0.56	0.88	0.93	0.88
	(SVC, LSTM) + RF	0.71	0.56	0.70	0.94	0.96	0.70
	(SVC, CNN) + RF	0.73	0.57	0.71	0.94	0.97	0.51
	(LSTM, CNN) + RF	0.60	0.43	0.59	0.83	0.92	0.96
	(SVC, RF) + RF	0.76	0.61	0.74	0.94	0.97	0.51
	(LSTM, RF) + RF	0.67	0.49	0.64	0.88	0.94	0.82
	(CNN, RF) + RF	0.71	0.54	0.67	0.93	0.97	0.57
	(SVC, LSTM, CNN) + RF	0.78	0.63	0.77	0.95	0.98	0.48
	SVC, LSTM, RF) + RF	0.80	0.67	0.79	0.95	0.98	0.48
	(SVC, CNN, RF) + RF	0.78	0.62	0.75	0.95	0.98	0.43
	(LSTM, CNN, RF) + RF	0.76	0.59	0.73	0.93	0.98	0.53
	(SVC, LSTM, CNN, RF) + RF	0.82	0.69	0.81	0.95	0.99	0.38

3.3. Experiments on Augmented JAFFE Dataset

The experiments with the augmented JAFFE dataset revealed significant variations in results based on the labeled and unlabeled data splits. As detailed in Table 7, for the 20:80 split, the CNN model achieved an accuracy of 0.40, an Intersection Jaccard (I-J) of 0.26, an F1-Score of 0.38, a Cosine Similarity of 0.90, and a ROC-AUC of 0.68. In this split, the SVC with an RBF kernel exhibited the highest Kullback-Leibler Divergence of 2.33, indicating a less optimal fit with the original data distribution.

Table 7. Results for Augmented JAFFE (20:80)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.40	0.26	0.38	0.90	0.68	3.24
	LSTM				0.16	0.02	0.04	0.83	0.50	3.16
	Random Forest				0.22	0.11	0.20	0.67	0.56	3.99
	SVC	*Linear*			0.24	0.12	0.21	0.67	0.59	2.53
		Poly	*Degree*	2	0.24	0.13	0.22	0.72	0.56	2.66
				3	0.22	0.11	0.19	0.78	0.55	2.69
				4	0.27	0.14	0.24	0.79	0.56	2.65
		RBF	*Gamma*	0.001	0.20	0.07	0.13	0.67	0.57	2.50
				0.01	0.13	0.01	0.03	0.00	0.47	2.33
				0.1	0.13	0.01	0.03	0.83	0.49	2.33
				1	0.13	0.01	0.03	0.83	0.49	3.16
				10	0.13	0.01	0.03	0.83	0.50	3.19

For the 25:75 split, shown in Table 8, both CNN and Random Forest models showed similar accuracy at 0.31. However, Random Forest slightly outperformed with an I-J of 0.18 and an F1-Score of 0.31.

Table 8. Results for Augmented JAFFE (25:75)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.31	0.14	0.23	0.79	0.69	3.23
	LSTM				0.07	0.01	0.03	0.51	0.47	2.71
	Random Forest				0.31	0.18	0.31	0.75	0.56	3.55
	SVC	*Linear*			0.27	0.14	0.25	0.80	0.66	2.16
		Poly	*Degree*	2	0.22	0.12	0.20	0.76	0.63	2.26
				3	0.16	0.07	0.13	0.76	0.61	2.41
				4	0.13	0.07	0.12	0.78	0.58	2.40
		RBF	*Gamma*	0.001	0.16	0.05	0.10	0.72	0.57	2.37
				0.01	0.16	0.02	0.04	0.83	0.50	2.26
				0.1	0.16	0.02	0.04	0.83	0.51	2.14
				1	0.13	0.01	0.03	0.83	0.49	3.03
				10	0.13	0.01	0.03	0.83	0.49	2.91

The SVC with an RBF kernel recorded a Kullback-Leibler Divergence of 2.14, while the SVC with various gamma values achieved the highest Cosine Similarity of 0.83. In the 50:50 split, detailed in Table 9, Random Forest demonstrated the best performance with an accuracy of 0.56, an I-J of 0.37, an F1-Score of 0.51, and a Cosine Similarity of 0.86. Meanwhile, CNN achieved the highest ROC-AUC of 0.85 and a Kullback-Leibler Divergence of 1.45. Overall, data augmentation had a notable impact on model performance, with Random Forest showing better consistency across multiple evaluation metrics..

Table 9. Results for Augmented JAFFE (50:50)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.51	0.35	0.51	0.85	0.85	1.45
	LSTM				0.16	0.04	0.08	0.68	0.51	2.18
	Random Forest				0.56	0.37	0.51	0.86	0.79	1.55
	SVC	*Linear*			0.38	0.24	0.38	0.84	0.77	1.63
		Poly	*Degree*	2	0.38	0.25	0.39	0.84	0.77	1.63
				3	0.42	0.26	0.39	0.83	0.74	1.69
				4	0.36	0.21	0.32	0.82	0.70	1.76
		RBF	*Gamma*	0.001	0.22	0.09	0.14	0.84	0.66	1.84
				0.01	0.16	0.02	0.04	0.83	0.49	2.17
				0.1	0.16	0.02	0.04	0.83	0.52	2.05
				1	0.13	0.01	0.03	0.83	0.49	2.34
				10	0.13	0.01	0.03	0.83	0.49	2.38

3.4. Ensemble Classification Results for Augmented JAFFE Dataset

The experimental results using the augmented JAFFE dataset demonstrated that ensemble models outperformed individual models across various performance metrics. As detailed in Table 10, for the 20:80 data split, the ensemble model incorporating Random Forest achieved an accuracy of 0.73, an F1-Score of 0.72, a Cosine Similarity of 0.94, a ROC-AUC of 0.98, and a Kullback-Leibler Divergence of 0.60. Notably, the Random Forest model alone also recorded an Intersection Jaccard (I-J) of 0.58.

Table 10. Results for Augmented JAFFE - Ensemble (20:80)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.33	0.17	0.28	0.84	0.79	1.42
	(LSTM) + RF	0.16	0.02	0.04	0.83	0.52	1.94
	(CNN) + RF	0.42	0.24	0.35	0.89	0.83	1.32
	(RF) + RF	0.33	0.18	0.30	0.78	0.76	1.54
	(SVC, LSTM) + RF	0.33	0.17	0.28	0.84	0.79	1.42
	(SVC, CNN) + RF	0.56	0.39	0.56	0.92	0.93	0.87
	(LSTM, CNN) + RF	0.42	0.24	0.35	0.89	0.83	1.32
	(SVC, RF) + RF	0.62	0.45	0.61	0.90	0.94	0.86
	(LSTM, RF) + RF	0.33	0.18	0.30	0.78	0.76	1.54
	(CNN, RF) + RF	0.60	0.43	0.58	0.89	0.94	0.84
	(SVC, LSTM, CNN) + RF	0.56	0.39	0.56	0.92	0.93	0.87
	SVC, LSTM, RF) + RF	0.62	0.45	0.61	0.90	0.94	0.86
	(SVC, CNN, RF) + RF	0.73	0.57	0.72	0.94	0.98	0.60
	(LSTM, CNN, RF) + RF	0.60	0.43	0.58	0.89	0.94	0.84
	(SVC, LSTM, CNN, RF) + RF	0.73	0.58	0.72	0.94	0.98	0.60

In the 25:75 data split, shown in Table 11, the ensemble model including Random Forest exhibited the best performance, with an accuracy of 0.87, an I-J of 0.77, an F1-Score of 0.86, a Cosine Similarity of 0.95, a ROC-AUC of 0.99, and a Kullback-Leibler Divergence of 0.47.

Table 11. Results for Augmented JAFFE - Ensemble (25:75)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.33	0.16	0.27	0.85	0.80	1.44
	(LSTM) + RF	0.24	0.08	0.14	0.63	0.64	1.83
	(CNN) + RF	0.31	0.13	0.21	0.83	0.76	1.58
	(RF) + RF	0.33	0.17	0.27	0.85	0.78	1.50
	(SVC, LSTM) + RF	0.49	0.30	0.44	0.83	0.90	1.05
	(SVC, CNN) + RF	0.56	0.38	0.55	0.85	0.91	1.00
	(LSTM, CNN) + RF	0.38	0.22	0.34	0.73	0.84	1.23
	(SVC, RF) + RF	0.60	0.41	0.57	0.89	0.95	0.77
	(LSTM, RF) + RF	0.53	0.38	0.53	0.86	0.89	1.08
	(CNN, RF) + RF	0.47	0.30	0.44	0.86	0.90	1.05
	(SVC, LSTM, CNN) + RF	0.69	0.53	0.67	0.85	0.96	0.68
	SVC, LSTM, RF) + RF	0.78	0.64	0.78	0.91	0.98	0.55
	(SVC, CNN, RF) + RF	0.73	0.58	0.73	0.94	0.98	0.63
	(LSTM, CNN, RF) + RF	0.67	0.51	0.65	0.91	0.96	0.73
	(SVC, LSTM, CNN, RF) + RF	0.87	0.77	0.86	0.95	0.99	0.47

For the 50:50 data split, detailed in Table 12, the Random Forest model continued to lead, achieving an accuracy of 0.82, an I-J of 0.70, an F1-Score of 0.81, a ROC-AUC of 0.99, and a Kullback-Leibler Divergence of 0.48. Additionally, the ensemble models with Random Forest recorded the highest Cosine Similarity of 0.92. These findings indicate that ensemble models, particularly those combining Support Vector Classifier, Long Short-Term Memory, Convolutional Neural Network, and Random Forest, deliver superior performance in terms of accuracy, I-J, F1-Score, Cosine Similarity, and ROC-AUC compared to individual models. The lower Kullback-Leibler Divergence suggests a better fit with the original data distribution following augmentation.

Table 12. Results for Augmented JAFFE - Ensemble (50:50)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.36	0.21	0.33	0.81	0.79	1.43
	(LSTM) + RF	0.22	0.06	0.10	0.84	0.65	1.81
	(CNN) + RF	0.51	0.32	0.46	0.84	0.88	1.15
	(RF) + RF	0.58	0.38	0.52	0.89	0.90	1.01
	(SVC, LSTM) + RF	0.51	0.35	0.51	0.85	0.90	1.05
	(SVC, CNN) + RF	0.60	0.44	0.60	0.88	0.94	0.82
	(LSTM, CNN) + RF	0.60	0.46	0.61	0.89	0.93	0.88
	(SVC, RF) + RF	0.71	0.54	0.70	0.91	0.96	0.66
	(LSTM, RF) + RF	0.60	0.42	0.59	0.92	0.94	0.81
	(CNN, RF) + RF	0.67	0.50	0.66	0.89	0.96	0.67
	(SVC, LSTM, CNN) + RF	0.69	0.56	0.70	0.89	0.97	0.72
	SVC, LSTM, RF) + RF	0.73	0.58	0.73	0.90	0.98	0.58
	(SVC, CNN, RF) + RF	0.80	0.67	0.79	0.91	0.98	0.49
	(LSTM, CNN, RF) + RF	0.71	0.56	0.71	0.92	0.97	0.57
	(SVC, LSTM, CNN, RF) + RF	0.82	0.70	0.81	0.91	0.99	0.48

3.5. Experiments on Feature-Extracted JAFFE Dataset

The experiments conducted on the JAFFE dataset, utilizing deep learning-based feature extraction, revealed notable variations in performance across different data split ratios. As detailed in Table 13, for the 20:80 data split, the SVC model with a linear kernel achieved an accuracy of 0.37 and an Intersection Jaccard (I-J) of 0.22.

Table 13. Results for Feature Extraction JAFFE (20:80)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.23	0.10	0.17	0.78	0.63	3.47
	LSTM				0.14	0.01	0.03	0.83	0.50	3.13
	Random Forest				0.26	0.15	0.26	0.72	0.66	1.99
	SVC	*Linear*			0.37	0.22	0.36	0.78	0.75	1.91
		Poly	*Degree*	2	0.19	0.10	0.16	0.73	0.71	2.18
				3	0.16	0.09	0.15	0.73	0.70	2.25
				4	0.19	0.10	0.16	0.75	0.71	2.14
		RBF	*Gamma*	0.001	0.19	0.09	0.16	0.72	0.71	2.14
				0.01	0.14	0.01	0.03	0.83	0.57	2.37
				0.1	0.16	0.02	0.04	0.83	0.49	2.52
				1	0.14	0.01	0.03	0.83	0.50	2.15
				10	0.14	0.01	0.03	0.83	0.51	3.15

However, its performance in terms of Cosine Similarity and Kullback-Leibler (KL) Divergence was less notable compared to other models, such as LSTM and several configurations with RBF kernels, which reached the highest Cosine Similarity of 0.83. In the 25:75 data split, shown in Table 14, the SVC with a linear kernel showed improved results, with an accuracy of 0.60, an I-J of 0.47, the highest Cosine Similarity of 0.87, and a ROC-AUC of 0.81.

Table 14. Results for Feature Extraction JAFFE (25:75)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.33	0.18	0.26	0.70	0.70	2.66
	LSTM				0.14	0.01	0.03	0.00	0.49	3.01
	Random Forest				0.33	0.21	0.33	0.77	0.70	1.81
	SVC	*Linear*			0.60	0.47	0.60	0.87	0.81	1.54
		Poly	*Degree*	2	0.28	0.17	0.27	0.73	0.75	1.76
				3	0.28	0.17	0.28	0.73	0.76	1.78
				4	0.35	0.23	0.34	0.73	0.78	1.64
		RBF	*Gamma*	0.001	0.28	0.19	0.28	0.79	0.73	1.85
				0.01	0.19	0.07	0.12	0.65	0.61	1.97
				0.1	0.16	0.02	0.04	0.83	0.49	2.19
				1	0.14	0.01	0.03	0.83	0.50	1.97
				10	0.14	0.01	0.03	0.83	0.51	3.11

Nonetheless, the relatively high KL Divergence of 1.54 indicated some deviation from the original data distribution. For the 50:50 data split, detailed in Table 15, the SVC with a linear kernel achieved the highest accuracy of 0.65, an I-J of 0.54, an F1-Score of 0.65, a Cosine Similarity of 0.91, and a ROC-AUC of 0.86, with a KL Divergence of 1.21. These results demonstrate the consistent performance of the SVC with a linear kernel in terms of accuracy and other evaluation metrics when applied to the extracted features.

Table 15. Results for Feature Extraction JAFFE (50:50)

Dataset	Algorithm				Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	*CNN*				0.26	0.17	0.25	0.59	0.68	1.88
	LSTM				0.16	0.02	0.04	0.83	0.51	2.31
	Random Forest				0.28	0.19	0.28	0.70	0.77	1.64
	SVC	*Linear*			0.65	0.54	0.65	0.91	0.86	1.21
		Poly	*Degree*	2	0.21	0.11	0.16	0.84	0.78	1.65
				3	0.28	0.17	0.26	0.87	0.78	1.60
				4	0.37	0.25	0.36	0.78	0.80	1.53
		RBF	*Gamma*	0.001	0.33	0.23	0.33	0.79	0.78	1.61
				0.01	0.16	0.02	0.04	0.83	0.67	1.82
				0.1	0.16	0.02	0.04	0.83	0.50	2.37
				1	0.14	0.01	0.03	0.83	0.51	2.24
				10	0.14	0.01	0.03	0.83	0.51	2.38

3.6. Ensemble Classification Results for Feature-Extracted JAFFE Dataset

The feature extraction experiments on the JAFFE dataset revealed that ensemble models outperformed individual models across various performance metrics. As shown in Table 16, for the 20:80

data split, the combination of models with Random Forest (RF) achieved the highest accuracy of 0.51 and a ROC-AUC of 0.91, alongside the lowest Kullback-Leibler divergence of 1.03, indicating an effective approximation of the true data distribution.

Table 16. Results for Feature Extraction JAFFE - Ensemble (20:80)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.30	0.14	0.23	0.80	0.71	1.65
	(LSTM) + RF	0.16	0.02	0.04	0.83	0.51	1.94
	(CNN) + RF	0.23	0.09	0.15	0.81	0.65	1.79
	(RF) + RF	0.35	0.18	0.27	0.67	0.80	1.43
	(SVC, LSTM) + RF	0.30	0.14	0.23	0.80	0.71	1.65
	(SVC, CNN) + RF	0.35	0.20	0.30	0.82	0.79	1.45
	(LSTM, CNN) + RF	0.23	0.09	0.15	0.81	0.65	1.79
	(SVC, RF) + RF	0.44	0.26	0.39	0.79	0.86	1.21
	(LSTM, RF) + RF	0.35	0.18	0.27	0.67	0.80	1.43
	(CNN, RF) + RF	0.49	0.32	0.48	0.78	0.89	1.11
	(SVC, LSTM, CNN) + RF	0.35	0.20	0.30	0.82	0.79	1.45
	SVC, LSTM, RF) + RF	0.44	0.26	0.39	0.79	0.86	1.20
	(SVC, CNN, RF) + RF	0.51	0.35	0.50	0.80	0.91	1.03
	(LSTM, CNN, RF) + RF	0.49	0.32	0.48	0.78	0.89	1.11
	(SVC, LSTM, CNN, RF) + RF	0.51	0.34	0.48	0.81	0.91	1.03

For the 25:75 data split, detailed in Table 17, the ensemble of models and RF yielded the highest accuracy of 0.70, an Intersection Jaccard (I-J) of 0.55, an F1-Score of 0.69, and a ROC-AUC of 0.96, with the lowest Kullback-Leibler divergence of 0.66, suggesting a precise approximation of the original data distribution.

Table 17. Results for Feature Extraction JAFFE - Ensemble (25:75)

Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.44	0.26	0.37	0.81	0.85	1.24
	(LSTM) + RF	0.16	0.02	0.04	0.83	0.51	1.94
	(CNN) + RF	0.30	0.15	0.23	0.82	0.71	1.66
	(RF) + RF	0.40	0.21	0.32	0.77	0.82	1.36
	(SVC, LSTM) + RF	0.44	0.26	0.37	0.81	0.85	1.24
	(SVC, CNN) + RF	0.51	0.34	0.46	0.86	0.90	1.04
	(LSTM, CNN) + RF	0.30	0.15	0.23	0.82	0.71	1.66
	(SVC, RF) + RF	0.60	0.45	0.60	0.81	0.93	0.87
	(LSTM, RF) + RF	0.40	0.21	0.32	0.77	0.82	1.36
	(CNN, RF) + RF	0.53	0.37	0.53	0.86	0.91	1.00
	(SVC, LSTM, CNN) + RF	0.51	0.34	0.47	0.85	0.90	1.03
	SVC, LSTM, RF) + RF	0.60	0.45	0.60	0.81	0.93	0.87
	(SVC, CNN, RF) + RF	0.70	0.55	0.69	0.89	0.96	0.67
	(LSTM, CNN, RF) + RF	0.53	0.37	0.53	0.86	0.91	0.99
	(SVC, LSTM, CNN, RF) + RF	0.70	0.55	0.69	0.88	0.96	0.66

Similarly, for the 50:50 data split, presented in Table 18, the ensemble models with RF achieved the best accuracy of 0.79, an I-J of 0.66, a Cosine Similarity of 0.96, and a ROC-AUC of 0.98, with the lowest Kullback-Leibler divergence of 0.60, confirming optimal performance in terms of both accuracy and data distribution fit. These results highlight the effectiveness of ensemble models in improving accuracy and classification performance on the feature-extracted dataset, surpassing the capabilities of individual models.

Table 18. Results for Feature Extraction JAFFE - Ensemble (50:50)

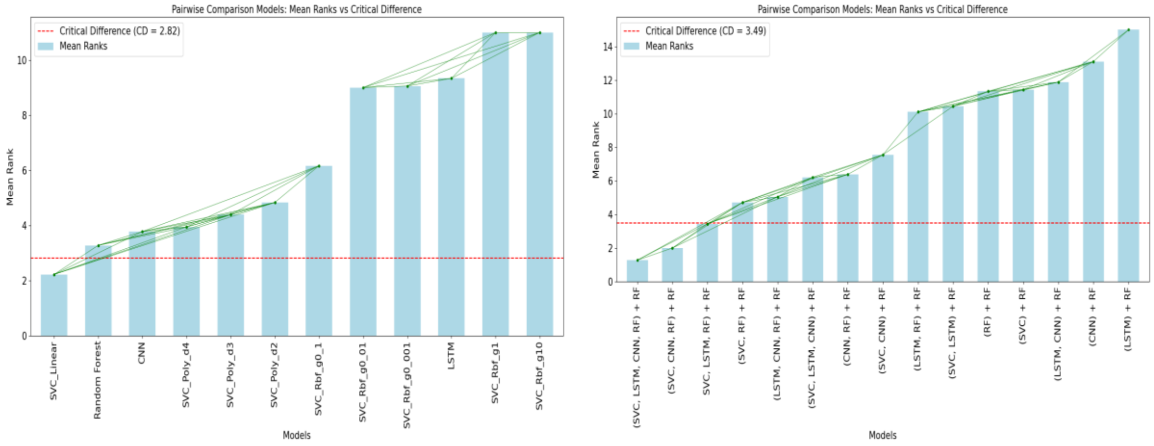
Dataset	Algorithm	Accuracy	I-J	F1-Score	Cosine	ROC-AUC	K-L
JAFFE	(SVC) + RF	0.47	0.30	0.43	0.82	0.84	1.29
	(LSTM) + RF	0.16	0.02	0.04	0.83	0.51	1.94
	(CNN) + RF	0.37	0.21	0.28	0.84	0.79	1.45
	(RF) + RF	0.44	0.28	0.40	0.88	0.84	1.25
	(SVC, LSTM) + RF	0.47	0.30	0.43	0.82	0.84	1.29
	(SVC, CNN) + RF	0.56	0.39	0.52	0.88	0.92	0.96
	(LSTM, CNN) + RF	0.37	0.21	0.28	0.84	0.79	1.45
	(SVC, RF) + RF	0.63	0.44	0.59	0.89	0.95	0.77
	(LSTM, RF) + RF	0.44	0.28	0.40	0.88	0.84	1.25
	(CNN, RF) + RF	0.60	0.44	0.59	0.92	0.94	0.87
	(SVC, LSTM, CNN) + RF	0.56	0.39	0.52	0.88	0.92	0.96
	SVC, LSTM, RF) + RF	0.63	0.44	0.59	0.89	0.95	0.77
	(SVC, CNN, RF) + RF	0.79	0.66	0.78	0.96	0.98	0.60
	(LSTM, CNN, RF) + RF	0.60	0.44	0.59	0.92	0.94	0.87
	(SVC, LSTM, CNN, RF) + RF	0.79	0.66	0.79	0.96	0.98	0.60

Table 19 presents the results of the Friedman test, which reveals significant performance differences between models in both semi-supervised learning (12 models) and ensemble semi-supervised learning (15 models). Following the identification of these differences, the post hoc Nemenyi test was applied to compare model pairs. In the semi-supervised case, 12 x 12 = 144 model pairs were evaluated, with 38 pairs showing significant differences. In the ensemble scenario, 15 x 15 = 225 model pairs were compared, resulting in 54 pairs with significant differences. These findings suggest that the ensemble approach is more effective at distinguishing model performance, leading to greater improvements in metrics such as Accuracy, ROC-AUC, and F1-Score.

Table 19. Friedman and Nemenyi Psot Hoc Test Result

Scenario	Metric	Statistic	p-Value	Significant Combination
Semi-Supervised Learning	Accuracy	79.47775551102211	1.862e-12	38
	I-J	83.84716157205246	2.649e-13	42
	F1-Score	83.35051546391755	3.308e-13	42
	ROC-AUC	79.40394477317558	1.924e-12	42
	K-L	47.37373343725649	1.846e-06	16
Ensemble Semi-Supervised Learning	Accuracy	114.5700483091788	7.235e-18	54
	I-J	114.16096096096094	8.692e-18	50
	F1-Score	113.70190571715145	1.068e-17	54
	Cosine	59.755609955120384	1.295e-07	24
	ROC-AUC	114.1059202577527	8.909e-18	57
	K-L	13.38181818181818	1.233e-17	56

Fig. 12 present the mean rank results obtained from the Friedman test, which compares multiple methods, followed by the Nemenyi test to identify pairs of methods with significant differences.



(a) Semi-Supervised (b) Ensemble Semi-Supervised

Fig. 12. Comparison of (a) and (b) based on Mean ranks vs Critical Difference

In addition to the Nemenyi post-hoc test, each model is ranked based on its performance in each dataset scenario, and these rankings are averaged across all dataset scenarios to obtain the average ranking of the model, which reflects the overall performance of the model across all dataset scenarios as show in Fig. 13.

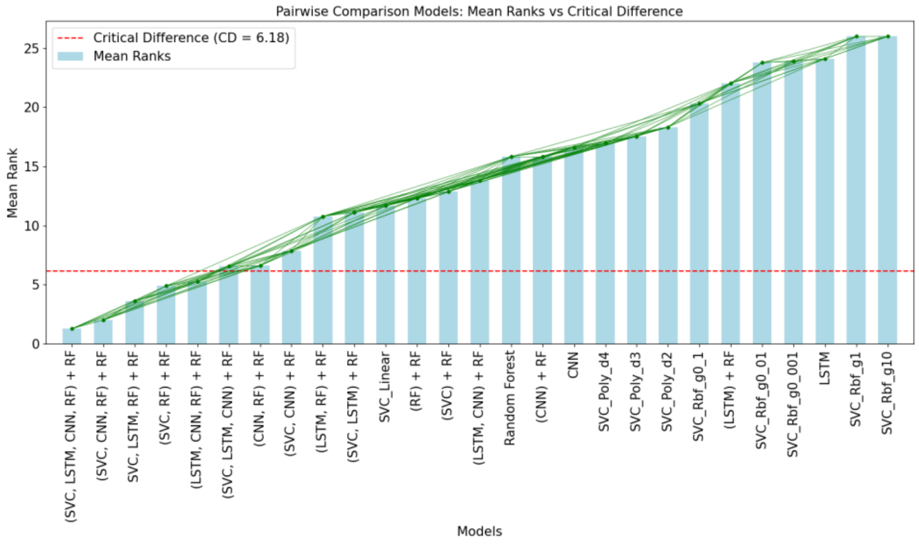


Fig. 13. Combined visualization of semi-supervised and ensemble semi-supervised based on mean ranks vs critical difference

The Critical Difference (CD) is determined using formula (7) to evaluate whether the performance differences between model pairs are statistically significant, considering the number of models ($k = 27$), the total number of dataset scenarios ($n = 9$), and the critical value ($q\alpha = 1.64899$) obtained from the Studentized Range distribution with degrees of freedom $df = n \times (k - 1) = 216$. If the difference in average ranks between two models exceeds the CD, the models are considered to have a significant performance difference, which helps identify which model is more effective.

$$CD = q_{\propto} \times \sqrt{\frac{k(k+1)}{6n}} \tag{7}$$

Fig. 14 shows the pairwise comparison results of the 27 models, consisting of 12 semi-supervised models and 15 semi-supervised ensemble models, by combining the average ranking results of both model groups. Models located below the Critical Difference (CD) line at a value of 6.18, such as (SVC, LSTM, CNN, RF) + RF, (SVC, CNN, RF) + RF, (SVC, LSTM, RF) + RF, (SVC, RF) + RF, and (LSTM, CNN, RF) + RF, indicate that these combined models have lower average ranks and, therefore, perform better overall compared to the other models. In contrast, models located above the CD line, such as (SVC, RF) + RF, CNN, LSTM, (SVC, LSTM) + RF, and several other models, do not show significant performance differences from the other models despite having higher ranks, which suggests that models incorporating Random Forest (RF) provide more stable and superior performance.

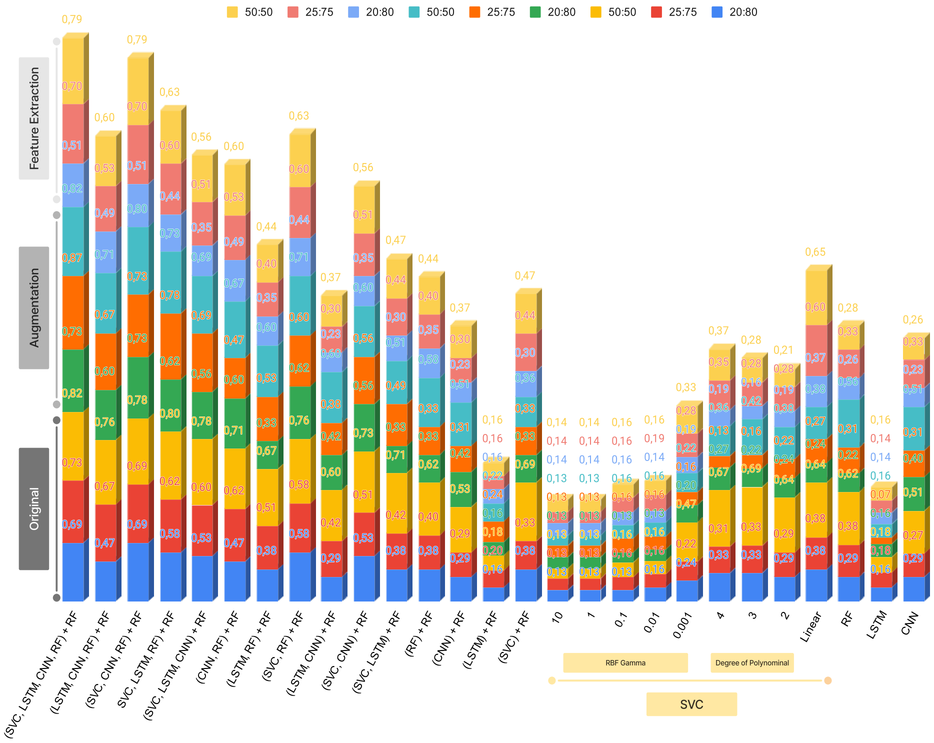


Fig. 14. Accuracy Comparison Across Different Methods and Data Scenarios

As shown in Table 20, the semi-supervised approach proposed in this study, which combines the SVC, LSTM, CNN, and Random Forest models with Random Forest as the final model, achieved an accuracy of 87%. This method has advantages in its flexibility to handle both labeled and unlabeled data, better computational efficiency compared to other deep learning-based approaches, and the ability to leverage the strengths of various models through ensemble techniques. However, the accuracy achieved is still lower compared to the DBN-GSA method by Alenazy et al. [46], which reached 96%, due to DBN-GSA's ability to optimize data with a limited number of labels. On the other hand, the Weighted Sparse Coding (WSC) method by Jiafa et al. [47], which also employs a semi-supervised approach, only achieved an accuracy of 83.1%, indicating that the proposed approach in this study performs better in utilizing unlabeled data. Nevertheless, the complex training process of the ensemble model presents a challenge compared to supervised methods such as CNN-SVM by Jabbooree et al., which is simpler but still managed to achieve an accuracy of 89.23%. Therefore, this method provides a competitive and adaptive alternative for facial expression recognition, particularly on the JAFFE dataset.

Table 20. Comparison of Facial Expression Recognition Algorithm Accuracies on the JAFFE Dataset

Researcher	Algorithm	Approach		Accuracy
		Semi-supervised	*Supervised*	
Jabbooree et al. [48]	CNN-SVM	×	√	89.23%
Hu et al. [49]	(Gabor, CS-LSMP, OMFMs) + SVM	×	√	82.86%
Yuan et al. [50]	EEPP	×	√	85.79%
Alenazy et al. [46]	DBN-GSA	√	×	96%
Jiafa et al. [47]	WSC	√	×	83.1%
Our Proposed	(SVC, LSTM, CNN, RF) + RF	√	×	87%

4. Conclusion

This study investigates the efficacy of ensemble learning approaches in facial expression recognition utilizing the JAFFE dataset, with an emphasis on diverse data scenarios and label-to-unlabeled ratios. The findings demonstrate that ensemble methods, particularly those that incorporate Random Forest, consistently outperform individual models across a range of evaluation metrics, including accuracy, Intersection Jaccard, F1-Score, Cosine Similarity, ROC-AUC, and Kullback-Leibler divergence. These results substantiate the hypothesis that ensemble learning enhances the robustness and performance of models, especially when integrated with data augmentation and feature extraction techniques. Nonetheless, the study acknowledges several limitations, notably the restricted and potentially unrepresentative size of the dataset, which may hinder the model's generalizability to real-world scenarios. Furthermore, the analysis did not address the computational efficiency and training time of the ensemble model in comparison to individual models, a consideration critical for practical applications. Recommendations for future research include the utilization of larger and more diverse datasets to yield more representative findings, an exploration of computational efficiency to encourage optimization of methodologies, and the implementation of k-fold cross-validation to facilitate a more stable performance evaluation and mitigate the risk of bias arising from specific training/testing data partitions.

Acknowledgment

We would like to thank the Faculty of Computer Science at Universitas Muslim Indonesia for their invaluable guidance, expertise, and financial support.

References

[1] S. Saurav, R. Saini, and S. Singh, “Facial expression recognition using dynamic local ternary patterns with kernel extreme learning machine classifier,” IEEE Access, vol. 9, pp. 120844–120868, 2021, doi: 10.1109/ACCESS.2021.3108029.

[2] Z. Ullah et al., “Emotion recognition from occluded facial images using deep ensemble model,” Comput. Mater. Contin., vol. 73, no. 3, pp. 4465–4487, 2022, doi: 10.32604/cmc.2022.029101.

[3] R. Zhi, C. Zhou, T. Li, S. Liu, and Y. Jin, “Action unit analysis enhanced facial expression recognition by deep neural network evolution,” Neurocomputing, vol. 425, no. xxxx, pp. 135–148, Feb. 2021, doi: 10.1016/j.neucom.2020.03.036.

[4] H. S. Cha and C. H. Im, “Performance enhancement of facial electromyogram-based facial-expression recognition for social virtual reality applications using linear discriminant analysis adaptation,” Virtual Real., vol. 26, no. 1, pp. 385–398, 2022, doi: 10.1007/s10055-021-00575-6.

[5] R. Singh, S. Saurav, T. Kumar, R. Saini, A. Vohra, and S. Singh, "Facial expression recognition in videos using hybrid CNN & ConvLSTM," Int. J. Inf. Technol., vol. 15, no. 4, pp. 1819–1830, 2023, doi: 10.1007/s41870-023-01183-0.

[6] A. S. Qazi, M. S. Farooq, F. Rustam, M. G. Villar, C. L. Rodríguez, and I. Ashraf, "Emotion Detection Using Facial Expression Involving Occlusions and Tilt," Appl. Sci., vol. 12, no. 22, p. 11797, Nov. 2022, doi: 10.3390/app122211797.

[7] S. M. Gowri, A. Rafeeq, and S. Devipriya, "Detection of real-time facial emotions via deep convolution neural network," Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021, no. Iciccs, pp. 1033–1037, 2021, doi: 10.1109/ICICCS51141.2021.9432242.

[8] B. Li and D. Lima, "Facial expression recognition via ResNet-50," Int. J. Cogn. Comput. Eng., vol. 2, pp. 57–64, Jun. 2021, doi: 10.1016/j.ijcce.2021.02.002.

[9] H. Sikkandar and R. Thiyagarajan, "Deep learning based facial expression recognition using improved cat swarm optimization," J. Ambient Intell. Humaniz. Comput., vol. 12, no. 2, pp. 3037–3053, 2021, doi: 10.1007/s12652-020-02463-4.

[10] S. Begaj, A. O. Topal, and M. Ali, "Emotion recognition based on facial expressions using convolutional neural network (CNN)," in 2020 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), Dec. 2020, pp. 58–63, doi: 10.1109/CoNTESA50436.2020.9302866.

[11] L. Zhao, "A Facial Expression Recognition Method Using Two-Stream Convolutional Networks in Natural Scenes," J. Inf. Process. Syst., vol. 17, no. 2, pp. 399–410, 2021. [Online]. Available at: https://s3.ap-northeast-2.amazonaws.com/journal-home/journal/jips/fullText/582/15.pdf.

[12] A. R. Khan, "Facial emotion recognition using conventional machine learning and deep learning methods: current achievements, analysis and remaining challenges," Information, vol. 13, no. 6, p. 268, May 2022, doi: 10.3390/info13060268.

[13] S. Umer, R. K. Rout, C. Pero, and M. Nappi, "Facial expression recognition with trade-offs between data augmentation and deep learning features," J. Ambient Intell. Humaniz. Comput., vol. 13, no. 2019, pp. 721–735, 2022, doi: 10.1007/s12652-020-02845-8.

[14] A. Fnaiech, H. Sahli, M. Sayadi, and P. Gorce, "Fear facial emotion recognition based on angular deviation," Electron., vol. 10, no. 3, pp. 1–16, 2021, doi: 10.3390/electronics10030358.

[15] E. G. Dada, D. O. Oyewola, S. B. Joseph, O. Emebo, and O. O. Oluwagbemi, "Facial Emotion Recognition and Classification Using the Convolutional Neural Network-10 (CNN-10)," Appl. Comput. Intell. Soft Comput., vol. 2023, pp. 1–19, Oct. 2023, doi: 10.1155/2023/2457898.

[16] M. K. Chowdary, T. N. Nguyen, and D. J. Hemanth, "Deep learning-based facial emotion recognition for human–computer interaction applications," Neural Comput. Appl., vol. 35, no. 32, pp. 23311–23328, 2023, doi: 10.1007/s00521-021-06012-8.

[17] T. Han, C. Liu, R. Wu, and D. Jiang, "Deep transfer learning with limited data for machinery fault diagnosis," Appl. Soft Comput. J., vol. 103, p. 107150, 2021, doi: 10.1016/j.asoc.2021.107150.

[18] T. Nguyen and R. Raich, "Incomplete label multiple instance multiple label learning," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 3, pp. 1320–1337, 2022, doi: 10.1109/TPAMI.2020.3017456.

[19] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Epps, and B. W. Schuller, "Multi-task semi-supervised adversarial autoencoding for speech emotion recognition," IEEE Trans. Affect. Comput., vol. 13, no. 2, pp. 992–1004, 2022, doi: 10.1109/TAFFC.2020.2983669.

[20] C. Choi et al., "Semi-supervised target classification in multi-frequency echosounder data," ICES J. Mar. Sci., vol. 78, no. 7, pp. 2615–2627, 2021, doi: 10.1093/icesjms/fsab140.

[21] K. Zhang, H. Wang, W. Liu, M. Li, J. Lu, and Z. Liu, "An efficient semi-supervised manifold embedding for crowd counting," Appl. Soft Comput., vol. 96, p. 106634, Nov. 2020, doi: 10.1016/j.asoc.2020.106634.

[22] H. Seyed Alinezhad, J. Shang, and T. Chen, "Early classification of industrial alarm floods based on semisupervised learning," IEEE Trans. Ind. Informatics, vol. 18, no. 3, pp. 1845–1853, Mar. 2022, doi: 10.1109/TII.2021.3081417.

[23] M. Srinivas, S. Saurav, A. Nayak, and A. P. Murukessan, “Facial expression recognition using fusion of deep learning and multiple features,” Mach. Learn. Algorithms Appl., pp. 229–246, 2021, doi: 10.1002/9781119769262.ch13.

[24] R. Guo, Y. Peng, W. Kong, and F. Li, “A semi-supervised label distribution learning model with label correlations and data manifold exploration,” J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 10, pp. 10094–10108, 2022, doi: 10.1016/j.jksuci.2022.10.008.

[25] E. Lashgari, D. Liang, and U. Maoz, “Data augmentation for deep-learning-based electroencephalography,” J. Neurosci. Methods, vol. 346, no. February, p. 108885, Dec. 2020, doi: 10.1016/j.jneumeth.2020.108885.

[26] Y. Li, X. Yan, Z. Wang, B. Zhang, and Z. Jia, “Clear the fog of negative emotions: A new challenge for intervention towards drug users,” J. Affect. Disord., vol. 294, no. July, pp. 305–313, 2021, doi: 10.1016/j.jad.2021.07.029.

[27] J. Kim and D. Lee, “Facial Expression Recognition Robust to Occlusion and to Intra-Similarity Problem Using Relevant Subsampling,” Sensors, vol. 23, no. 5, p. 2619, Feb. 2023, doi: 10.3390/s23052619.

[28] A. Lumini, L. Nanni, and G. Maguolo, “Deep learning for plankton and coral classification,” Appl. Comput. Informatics, vol. 19, no. 3–4, pp. 265–283, 2023, doi: 10.1016/j.aci.2019.11.004.

[29] H. Saleh, S. Mostafa, A. Alharbi, S. El-Sappagh, and T. Alkhalifah, “Heterogeneous ensemble deep learning model for enhanced arabic sentiment analysis,” Sensors, vol. 22, no. 10, pp. 1–28, 2022, doi: 10.3390/s22103707.

[30] A. Kortylewski, Q. Liu, A. Wang, Y. Sun, and A. Yuille, “Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion,” Int. J. Comput. Vis., vol. 129, no. 3, pp. 736–760, Mar. 2021, doi: 10.1007/s11263-020-01401-3.

[31] J. Park, W. Kang, H. Il Koo, and N. I. Cho, “Face Swapping for Low-Resolution and Occluded Images In-the-Wild,” IEEE Access, vol. 12, pp. 91383–91395, 2024, doi: 10.1109/ACCESS.2024.3421528.

[32] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, “Transfer learning for image classification using VGG19: Caltech-101 image data set,” J. Ambient Intell. Humaniz. Comput., vol. 14, no. 4, pp. 3609–3620, 2023, doi: 10.1007/s12652-021-03488-z.

[33] S. Sethi, M. Kathuria, and T. Kaushik, “Face mask detection using deep learning : An approach to reduce risk of coronavirus spread,” J. Biomed. Inform., vol. 120, no. September 2020, p. 103848, 2021, doi: 10.1016/j.jbi.2021.103848.

[34] K. Liu, S. Yu, and S. Liu, “An improved inceptionV3 network for obscured ship classification in remote sensing images,” IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 13, pp. 4738–4747, 2020, doi: 10.1109/JSTARS.2020.3017676.

[35] V. R. Joseph, “Optimal ratio for data splitting,” Stat. Anal. Data Min. ASA Data Sci. J., vol. 15, no. 4, pp. 531–538, 2022, doi: 10.1002/sam.11583.

[36] G. Lantzanakis, Z. Mitraka, and N. Chrysoulakis, “X-SVM: An extension of C-SVM algorithm for classification of high-resolution satellite imagery,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 5, pp. 3805–3815, May 2021, doi: 10.1109/TGRS.2020.3017937.

[37] H. Darwis, Z. Ali, Y. Salim, and P. L. L. Belluano, “Max feature map cnn with support vector guided softmax for face recognition,” Int. J. Informatics Vis., vol. 7, no. 3, pp. 959–966, 2023, doi: 10.30630/joiv.7.3.1751.

[38] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” Artif. Intell. Rev., vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[39] H. Yu, L. T. Yang, Q. Zhang, D. Armstrong, and M. J. Deen, “Neurocomputing convolutional neural networks for medical image analysis : State-of-the- art , comparisons , improvement and perspectives,” Neurocomputing, vol. 444, pp. 92–110, 2021, doi: 10.1016/j.neucom.2020.04.157.

[40] M. Mafarja, T. Thaher, M. A. Al-betar, I. A. Doush, and H. Turabieh, “Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning,” pp. 18715–18757, 2023, doi: 10.1007/s10489-022-04427-x.

[41] W. Wattanapornprom, C. Thammarongtham, A. Hongsthong, and S. Lertampaiporn, “Ensemble of multiple classifiers for multilabel classification of plant protein subcellular localization,” Life, vol. 11, no. 4, p. 293, Mar. 2021, doi: 10.3390/life11040293.

[42] L. Zou, H. Wu, R. Liu, C. Yi, J. He, and Y. Li, “A new method for LDPC blind recognition over a candidate set using Kullback-Leibler divergence,” IEEE Commun. Lett., vol. 28, no. 5, pp. 964–968, May 2024, doi: 10.1109/LCOMM.2024.3373074.

[43] P.-J. Chao et al., “Using deep learning models to analyze the cerebral edema complication caused by radiotherapy in patients with intracranial tumor,” Sci. Rep., vol. 12, no. 1, p. 1555, 2022, doi: 10.1038/s41598-022-05455-w.

[44] W. Hu, L. Wu, M. Jian, Y. Chen, and H. Yu, “Cosine metric supervised deep hashing with balanced similarity,” Neurocomputing, vol. 448, pp. 94–105, 2021, doi: 10.1016/j.neucom.2021.03.093.

[45] Z. E. Huma et al., “A hybrid deep random neural network for cyberattack detection in the industrial internet of ihings,” IEEE Access, vol. 9, pp. 55595–55605, 2021, doi: 10.1109/ACCESS.2021.3071766.

[46] 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, 2021, doi: 10.1007/s12652-020-02235-0.

[47] J. Mao, Y. Hu, and R. Wan, “Face Expression Recognition Method Based on Weak Annotation and Conditional Generative Adversarial Neural Networks,” The Lancent Pschch. pp. 1–35, 2024, doi: 10.2139/ssrn.4820797.

[48] A. I. Jabbooree, L. M. Khanli, P. Salehpour, and S. Pourbahrami, “Geometrical Facial Expression Recognition Approach Based on Fusion CNN-SVM,” Int. J. Intell. Eng. Syst., vol. 17, no. 1, pp. 457–468, 2024, doi: 10.22266/ijies2024.0229.40.

[49] M. Hu, C. Yang, Y. Zheng, X. Wang, L. He, and F. Ren, “Facial Expression Recognition Based on Fusion Features of Center-Symmetric Local Signal Magnitude Pattern,” IEEE Access, vol. 7, pp. 118435–118445, 2019, doi: 10.1109/ACCESS.2019.2936976.

[50] S. Yuan and X. Mao, “Exponential elastic preserving projections for facial expression recognition,” Neurocomputing, vol. 275, pp. 711–724, 2018, doi: 10.1016/j.neucom.2017.08.067.