# Multi-Class Emotion Classification through Visual Content Analysis Using Deep Learning Techniques

Florin Octavian ROBU
*Faculty of Automation, Computers, Electrical Engineering and Electronics*
*Dunarea de Jos University of Galati*
Galati, Romania
fr145@student.ugal.ro

Simona MOLDOVANU*
*Department of Computer Science and Information Technology*
*Faculty of Automation, Computers, Electrical Engineering and Electronics*
*Dunarea de Jos University of Galati*
Galati, Romania
simona.moldovanu@ugal.ro

Ioana-Diana MOLDOVANU
*Department of Psychology*
*Danubius International University*
Galati, Romania
m.ioana31@yahoo.com

***Abstract*— Facial emotion recognition plays an important role for human-computer interaction, including therapeutic applications, security and behavioural analysis of a person.**

**Despite several strategies for facial emotion recognition hybrid deep learning models, particularly Convolutional Neural Network (CNN) and Machine Learning (ML), brought significant potential robustness in automatic feature extraction capabilities and computational efficiency. In this study, we attain the highest single-network classification accuracy on the Face Expression Recognition Dataset (FERD) with the anger, disgust, fear, happiness, neutrality, sadness, and surprise facial expression. We utilize a custom-built CNN and Pycaret AutoML, carefully optimize its hyperparameters, and explore diverse optimization techniques and pre-processing methods. To the best model attains an accuracy of 67.26% on FERD with Ensemble Stacking Classifier (ESC) on proposed CNN architecture.**

***Keywords—Deep Learning, Emotion classification, Convolutional Neural Networks CNNs, Face Expression Recognition Dataset (FERD)***

## I. INTRODUCTION

In the contemporary digital era, images serve as an essential source of information, and the capacity for automatic analysis and interpretation has grown essential across various domains. Image content analysis facilitates significant advancements in intelligent surveillance systems, computer-aided medical diagnosis, advanced human-machine interaction, and sentiment analysis of visual content.

In this context, deep learning, an effective subset of artificial intelligence, has transformed the domain of image evaluation [1]. Convolutional neural networks (CNNs) enable computers to hierarchically acquire intricate characteristics directly from unprocessed visual data, surpassing conventional image processing techniques [2].

This research concentrates on a pertinent application of image content analysis utilizing Deep Learning: the classification of face expressions.

The automated recognition of human emotions expressed through facial expressions holds considerable promise across multiple applications, including enhancing human-computer interfaces by responding to the user's emotional condition, offering resources for psychological research in emotion comprehension, and functioning in security systems to identify stress or suspicion.

The project intends to categorize facial images into seven specific emotional classifications: anger, disgust, fear, happiness, neutrality, sadness, and surprise, utilizing the "Face Expression Recognition Dataset" from Kaggle repository [3].

In this context, we developed and executed a singular customized convolutional neural network aimed at face expression classification. This CNN was created and trained using a customized architectural model designed for analysing grey scale images. We prioritized images processing, concentrating on human face detection, face cropping, and face scaling to augment pertinent details.

This research aims to deliver an in-depth comprehension of the facial expression classification procedure utilizing deep learning, proposing the main contributions:

1. Selecting a dataset with the anger, disgust, fear, happiness, neutrality, sadness, and surprise facial expression.
2. A rigorous preprocessing process was implemented to crop and resize human faces, which resulted in a significant increase in model accuracy, demonstrating the importance of data cleaning. Additional data balancing techniques, such as SMOTE (Synthetic Minority Oversampling Technique) [4], were also tested to compensate for class imbalance.
3. We explored optimization techniques, including CBAM (Convolutional Block Attention Module) [5] and differential augmentation, which did not improve results, reflecting a scientific approach through hypothesis testing and documentation of both successes and failures.
4. We performed a multi-class classification with seven facial expressions;
5. We implemented a hybrid model CNN and PyCaret AutoML;
6. We evaluate the classifications with accuracy, F1-score, and accuracy quality metrics.

## II. RELATED WORK

The Face Expression Recognition Dataset has been utilized in numerous studies and initiatives for face emotion recognition, frequently acting as a foundational resource for machine learning research. Current contributions establish a robust basis, emphasizing fundamental data processing and implementations utilizing basic models. For instance, Ohéix described the application of the dataset in code illustrations

**Cite as:** F. O. Robu, S. Moldovanu, and I. D. Moldovanu, "Multi-class emotion classification through visual content analysis using deep learning techniques", *Syst. Theor. Control Comput. J. ,* vol. 5, no. 1, pp. 14–19, Dec. 2025.
**DOI:** 10.52846/stccj.2025.5.1.65

[3], emphasizing critical setup tasks, including data loading and normalization. Teja adopted a comparable methodology [6], employing a simple neural network to directly analyze the raw data for emotion classification. The outcomes achieved with these fundamental methods were often moderate, with accuracy seldom exceeding 50–55%, attributable to the simplicity of the design.

Kharwal [7] explored a convolutional neural network (CNN) model utilizing the same dataset, emphasizing the basic configuration and direct image training without employing advanced changes.

The study proposed by Li et al. involves six facial expressions [8]. This research classified RGB images and highlights the importance of utilizing diverse datasets to improve the accuracy and robustness of facial expression recognition systems.

These publications are relevant to the current research as they demonstrate the application of the dataset in fundamental machine learning contexts, so establishing a robust foundation for the more advanced methodologies described in this paper.

## III. EXPERIMENT SETUP AND AI TOOLS

The study was developed in Python environment, utilizing several standard libraries for artificial intelligence. The primary frameworks employed for developing, training, and assessing the CNN model were TensorFlow (version 2.19.0) [9] and Keras (version 3.9.2) [10]. During the stage of pre-processing, the OpenCV (version 4.11.0.86) [11] and Dlib (version 20.0.0) [12] libraries were used, enabling functions such as reading, resizing, and notably, face detection in images. The chosen dataset, the "Face Expression Recognition Dataset" from the Kaggle platform was downloaded [3], it has a large number of facial images categorized into seven emotional classifications. A significant problem noted was the disproportionate distribution of classes, with the quantity of images for emotions like "disgust" being markedly smaller than that for "happiness". A class weighting approach was implemented during training to rectify this imbalance, imposing greater penalties for classification errors in underrepresented classes, hence enhancing the model's learning efficiency from these classes.

### *a) Face expression recognition dataset*

The dataset is structured in two main directories: train and test. All images in the dataset are in grayscale and have a uniform size of 48 x 48 pixels (size 1521 Byte).

The train directory is intended for training deep learning (DL) models, containing 28,821 images distributed as follows:

- angry: 3993 images
- disgust: 436 images
- fear: 4103 images
- happy: 7164 images
- neutral: 4982 images
- sad: 4938 images
- surprise: 3205 images

The test directory, which contains 7,066 images, has the following distribution:

- angry: 960 images
- disgust: 111 images
- fear: 1018 images
- happy: 1825 images
- neutral: 1216 images
- sad: 1139 images
- surprise: 797 images

There is a significant imbalance in the class distribution, which could influence the training process, with models tending to perform better on classes with more examples. Therefore, it will be necessary to consider strategies to mitigate the effects of this imbalance

### *b) Hybrid approach: CNN and Machine Learning*

This paper presents a novel hybrid approach combining the feature extraction efficiency of a CNN with the classification efficacy of traditional machine learning models. The CNN was setup as a robust feature extractor rather than employing it for final predictions through a Softmax layer. The outputs of the penultimate dense layer were collected and stored as feature vectors in a .npz file, which served as the foundation for the ML modules. Furthermore, to address the class imbalance in the dataset, the Synthetic Minority Oversampling Technique (SMOTE) [4] was employed, which artificially generates new samples for neglected classes based on the extracted features, thereby enhancing the diversity and balance of the data utilized in subsequent machine learning modules.

TABLE I. FRAMEWORK OF THE CNN ARCHITECTURE

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_1 (Conv2D) | (None, 48, 48, 64) | 640 |
| batch_norm_1 (BatchNormalization) | (None, 48, 48, 64) | 256 |
| max_pool_1 (MaxPooling2D) | (None, 24, 24, 64) | 0 |
| dropout_1 (Dropout) | (None, 24, 24, 64) | 0.25 |
| conv2d_2 (Conv2D) | (None, 24, 24, 128) | 73,856 |
| batch_norm_2 (BatchNormalization) | (None, 24, 24, 128) | 512 |
| max_pool_2 (MaxPooling2D) | (None, 12, 12, 128) | 0 |
| dropout_2 (Dropout) | (None, 12, 12, 128) | 0.3 |
| conv2d_3 (Conv2D) | (None, 12, 12, 256) | 295,168 |
| batch_norm_3 (BatchNormalization) | (None, 12, 12, 256) | 1,024 |
| max_pool_3 (MaxPooling2D) | (None, 6, 6, 256) | 0 |
| dropout_3 (Dropout) | (None, 6, 6, 256) | 0.35 |
| conv2d_4 (Conv2D) | (None, 6, 6, 512) | 1,180,160 |
| batch_norm_4 (BatchNormalization) | (None, 6, 6, 512) | 2,048 |
| max_pool_4 (MaxPooling2D) | (None, 3, 3, 512) | 0 |
| dropout_4 (Dropout) | (None, 3, 3, 512) | 0.4 |
| flatten_layer (Flatten) | (None, 4608) | 0 |
| dense_1 (Dense) | (None, 512) | 2,359,808 |
| batch_norm_5 (BatchNormalization) | (None, 512) | 2,048 |
| dropout_5 (Dropout) | (None, 512) | 0.6 |
| dense_2 (Dense) | (None, 256) | 131,328 |
| batch_norm_6 (BatchNormalization) | (None, 256) | 1,024 |
| dropout_6 (Dropout) | (None, 256) | 0.5 |
| output_layer (Dense) | (None, 7) | 1,799 |

The CNN architecture was carefully designed to optimize the extraction of relevant features. The model has four sequential convolutional blocks, each with a Conv2D layer for feature extraction, succeeded by BatchNormalization for training stability, MaxPooling2D for dimensionality reduction, and Dropout for regularization. This profound architecture enables the network to acquire an advanced hierarchy of features. To prevent overfitting, Dropout rates were carefully implemented, and EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint call-backs were

employed to guarantee stable, efficient training and enhanced model generalization. The complex framework of the CNN architecture is depicted in Table I.

This representation illustrates the data flow within the network, commencing from the input layer featuring 48x48 pixel grayscale images (one channel) to the output layer comprising seven units that represent the emotion classes. The data moves through four convolutional blocks incorporating batch normalization (BN), spatial down sampling via MaxPooling, and regularization through Dropout, subsequently followed by two dense layers with further normalization and regularization. Consequently, the suggested architecture is differentiated from a fundamental convolutional network by being more robust, efficient, and generalizable.

*c) Data processing stage*

Image processing is an important step in the development of an artificial intelligence model that classifies emotions through facial expressions. The primary objective is to convert unprocessed images into standardized and refined formats, optimized for effective processing by the learning algorithm. This phase is important for ensuring the model focuses on relevant facial traits, excluding irrelevant parts and minimizing data noise. The Python packages OpenCV [11] for image manipulation and Dlib [12] for advanced face detection were employed to accomplish this objective. The initial essential step was to separate the face from each image, as the model must learn only from facial characteristics. A high-performance approach was selected to attain superior results: the face detector from the Dlib package. Each original image was imported utilizing the cv2.imread() method in OpenCV [11], and the face detector was instantiated by calling dlib.get_frontal_face_detector(). This detector employs a pre-trained data file, shape_predictor_68_face_landmarks.dat [13], to precisely identify the facial inside the image, providing the coordinates of the surrounding region. These coordinates are essential for face cropping, a procedure that eliminates extraneous aspects from the original image and emphasizes the area of interest.

A distinct pre-processing method was employed based on these coordinates: face cropping, succeeded by scaling each crop—of varying dimensions—to a uniform size of 48×48 pixels, utilizing the cv2.resize() function in OpenCV. This technique ensures a homogeneous input for the neural network, amplifying the relevant aspects of facial expressions and enabling consistent data processing.

In this process, the multi-scale loop was implemented, which applies the Dlib detector at up sampling levels 0, 1, 2 and 3, and which represented an essential contribution to improving face detection and cropping performance. This strategy allowed for image processing at varying resolutions, resulting in the successful detection of 26,933 faces (of which 21,579 images are from the train category and 5,354 images are from testing) out of a total of 35,887 images, which equates to a detection rate of 75.05%.

Data quality was identified as a priority in this approach. During detection, two distinct scenarios were observed: success cases, where Dlib accurately detected and cropped a face, yielding high-quality images necessary for training (see Fig. 1), and failure cases, where the algorithm failed to identify any faces due to poor illumination, unfavourable angles, hidden faces, or other adverse factors (see Fig. 2). To develop an accurate model, it was determined to utilize solely precisely cropped faces, disregarding unsuccessful photographs. This methodology enhances the creation of a more resilient and precise model, optimizing the efficacy of the learning process.

Fig. 1. Examples of facial images detected and adjusted to 48×48 pixel size, with unimportant details removed for training.

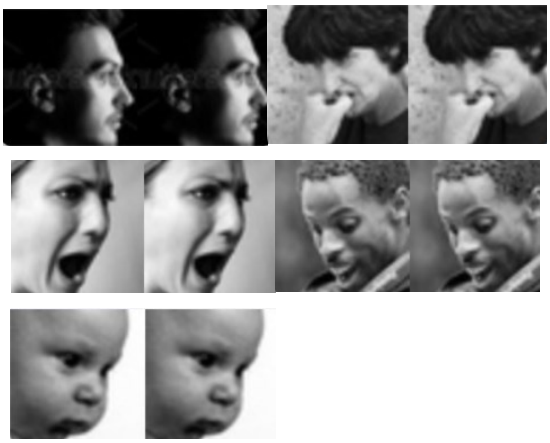

Fig. 2. Examples of images where face detection failed, ignored in the training process.

After pre-processing, the train directory containing 21,579 images distributed as follows:

- angry: 2981 images
- disgust: 352 images
- fear: 2798 images
- happy: 5917 images
- neutral: 3940 images
- sad: 3015 images
- surprise: 2576 images

The testing directory, which contains 5354 images, has the following distribution:

- angry: 714 images
- disgust: 90 images
- fear: 710 images
- happy: 1506 images
- neutral: 994 images
- sad: 714 images
- surprise: 626 images

To enhance the model's robustness and minimize overfitting, an approach of minor data augmentation was employed during training. This entails use the Keras ImageDataGenerator class, with minimal parameters like horizontal flipping and slight rotation, designed to prevent distortion of critical facial characteristics (e.g., eyes, mouth) while maintaining the integrity of the faces. A differential selective augmentation was initially attempted for underrepresented classes, enhancing these characteristics with a bespoke generator; however, this method was discontinued due to poor results seen.

As part of the optimization of the network architecture, an L2 regularization (with a coefficient of 0.0003) was integrated, which contributed to reducing the risk of overfitting and improving the generalization of the model, offering a robust alternative in limiting augmentation.

Alternative transformations, such horizontal and vertical shifts or scaling, were eschewed to preserve the integrity of the characteristics. The model architecture did not incorporate attention modules, such as the Convolutional Block Attention Module (CBAM), which synergizes channel attention and spatial attention to enhance the focus of convolutional neural networks on pertinent features. This decision was made due to the little size of the processed images, which were constrained to a resolution of 48x48 pixels, a scale considered inadequate for capturing the subtle details of face expressions. Initial assessments indicated that the use of CBAM in this context resulted in unsatisfactory performance, evidenced by reduced accuracy and an unwarranted rise in computing complexity.

Consequently, a streamlined design was favoured, relying just on convolutional and dense layers, to achieve an optimal equilibrium between performance and efficiency.

*d) PyCaret AutoML*

The features derived from the CNN model were converted into a Pandas DataFrame format, along by the emotional labels, so enhancing interoperability with PyCaret, an open-source low-code machine learning toolkit designed to expedite experimentation. The PyCaret environment was established by designating the target column ("target"), establishing a seed for reproducibility, and activating fix_imbalance=True to rectify class imbalance, prevalent in emotion recognition datasets (e.g., "happy" against "disgust").

A variety of machine learning algorithms were examined: Extra Trees Classifier (ET), Random Forest Classifier (RF), Light Gradient Boosting Machine (LightGBM), Ridge Classifier (ridge), Logistic Regression (LR), and Linear Discriminant Analysis (LDA), chosen for their enhanced efficacy relative to alternative models (e.g., Gradient Boosting, XGBoost, CatBoost, which presented computational constraints, or KNN, Naive Bayes, SVM, which exhibited inferior performance). Hyperparameters were fine-tuned to enhance accuracy, although PyCaret's default configurations offered a robust foundation.

The plan concluded with the implementation of ensemble methods to improve robustness. The Voting Classifier combines predictions through "hard" or "soft" voting, ensuring robust performance due to the diversity of the foundational models. Combining predictions from integrated ET, RF, LightGBM, LR, and LDA via a meta-model (Extra Trees) to rectify errors and enhance overall outcomes. Both methods utilized the complementarity of classifiers to achieve optimal performance.

## IV. RESULTS AND DISCUSSIONS

The assessment of the custom CNN model, trained on grayscale images from the "Face Expression Recognition Dataset," achieved an accuracy of 0.6718 on the testing set of 4975 images, showing the efficacy of the Softmax layer in classifying the seven emotional states: anger, disgust, fear, happiness, neutrality, sadness, and surprise. The classification report reveals performance variability, with the "happy" class attaining a precision of 0.87 and an F1 score of 0.87, based on 1432 images. In contrast, the "angry" class recorded an F1-score of 0.54, and the "sad" class an F1 score of 0.49, likely influenced by the complexity of facial features. The characteristics obtained from the penultimate dense layer were stored in a .npz file for subsequent processing. Figure 3 visually illustrates the model's performance through the confusion matrix, while Figure 4 showcases the achieved accuracy, emphasizing error distribution and classification efficacy.

TABLE II. THE QUALITY METRICS OBTAINED WITH MODEL CUSTOM-BUILT MODEL

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| angry | 0.59 | 0.53 | 0.55 | 714 |
| disgust | 0.47 | 0.78 | 0.59 | 90 |
| fear | 0.54 | 0.43 | 0.48 | 710 |
| happy | 0.88 | 0.86 | 0.87 | 1506 |
| neutral | 0.60 | 0.65 | 0.62 | 994 |
| sad | 0.49 | 0.52 | 0.50 | 714 |
| surprise | 0.75 | 0.82 | 0.78 | 626 |
| accuracy | - | - | 0.67 | 5354 |
| macro average | 0.62 | 0.65 | 0.63 | 5354 |
| weighted average | 0.67 | 0.67 | 0.67 | 5354 |
| Accuracy on the testing set: 0.6672 | | | | |

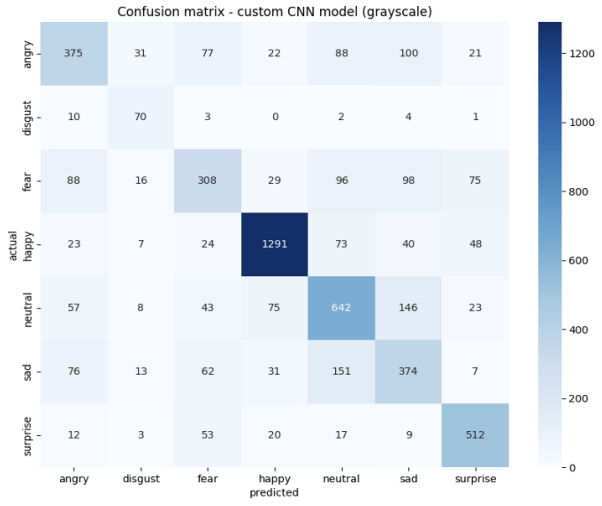


Fig. 3. Confusion matrix for custom-built CNN model.

The method resulted in the application of ensemble models to integrate the strengths of various classifiers and mitigate variation.

a) The Ensembling Voting Classifier amalgamates the predictions of the fundamental models via a voting mechanism. Voting can be classified as "hard" (majority) or "soft" (based on estimated probabilities). This is often simpler to execute and can yield solid and occasionally superior performance, particularly if the foundational models are adequately diversified.

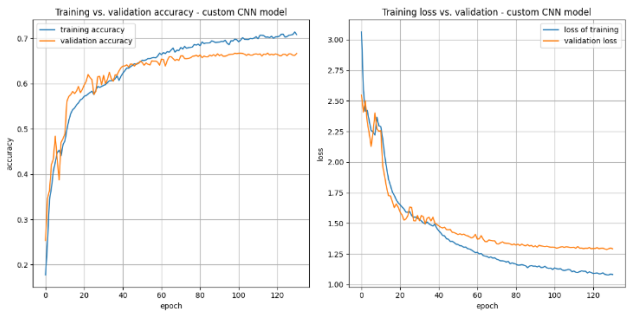


Fig. 4. Accuracy train vs. validation

The assessment of the Ensembling Voting Classifier, applied to the external testing set of 5,354 images, yielded an accuracy of 0.6713, indicating comparable efficacy to the original CNN model trained with Softmax. The classification report reveals diverse performance throughout the seven emotional categories (anger, disgust, fear, happy, neutral, sad, surprise), with the “happy” class attaining exceptional precision (0.88), recall (0.86), and F1-score (0.87), bolstered by a support of 1506 images. The "disgust" class had a high precision of 0.90, although a low recall of 0.52, indicating a propensity for underclassification. Conversely, the "angry" class achieved an F1-score of 0.57, and the "sad" class an F1-score of 0.50, reflecting limited efficacy, likely attributable to challenges in detecting nuanced characteristics. The macro average F1-score of 0.64 signifies a moderate class balance, while the weighted average of 0.67 demonstrates consistent overall performance, consistent with the data's uneven distribution. Subsequent aggregate analysis indicated an adjusted accuracy of 0.6713, an area under the ROC curve (AUC) of 0.9121, a recall of 0.6713, a precision of 0.6760, and an F1 score of 0.6723, thereby affirming the model's robustness. The Kappa (0.5985) and Matthews association Coefficient (MCC) (0.5987) metrics provide moderate agreement beyond chance levels and a consistent association between forecasts and actual labels, underscoring the capacity to manage class imbalance. Figure 5 visually depicts the efficacy of Ensembling Voting, displaying the confusion matrix that delineates the distribution of categorization errors among emotional categories.

b) Ensembling Stacking (Stacked Generalization) integrates the predictions of the basic models employed (ET, RF, LightGBM, LR, and LDA) via a meta-model.

The assessment of the Ensembling Stacking Classifier, utilized on the external testing set of 5,354 images, achieved an accuracy of 0.6726, indicating superior performance compared to previous models. The classification report indicates a diverse performance throughout the seven emotional categories (anger, disgust, fear, happy, neutral, sad, surprise), with the "happy" class achieving notable precision (0.87), recall (0.87), and F1-score (0.87), based on a sample of 1506 photos. The "disgust" class exhibited a precision of 0.90, although its recall was at 0.51, indicating a propensity for under classification. Conversely, the "angry" class achieved an F1-score of 0.56, and the "sad" class attained an F1-score of 0.50, reflecting constrained efficacy, likely attributable to challenges in detecting nuanced characteristics.

TABLE III. EVALUATION OF MODEL ON TESTING SET.

| Model | Accuracy | AUC | Recall | Precision | F1-score | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| Voting Classifier | 0.6713 | 0.9121 | 0.6713 | 0.6760 | 0.6723 | 0.5985 | 0.5987 |

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| angry | 0.55 | 0.59 | 0.57 | 714 |
| disgust | 0.90 | 0.52 | 0.66 | 90 |
| fear | 0.53 | 0.48 | 0.50 | 710 |
| happy | 0.88 | 0.86 | 0.87 | 1506 |
| neutral | 0.61 | 0.63 | 0.62 | 994 |
| sad | 0.48 | 0.53 | 0.50 | 714 |
| surprise | 0.80 | 0.78 | 0.79 | 626 |
| accuracy | - | - | 0.67 | 5354 |
| macro average | 0.68 | 0.63 | 0.64 | 5354 |
| weighted average | 0.68 | 0.67 | 0.67 | 5354 |

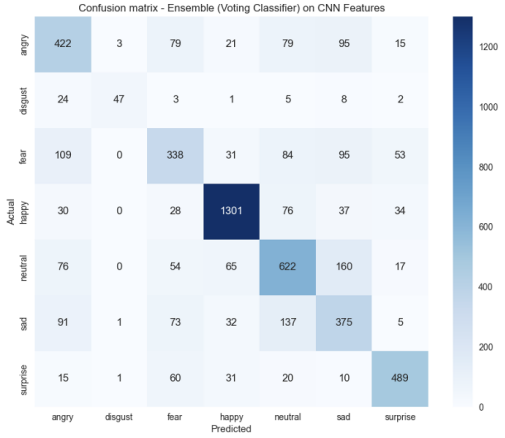


Fig. 5. Confusion matrix for model on testing set

The macro average F1-score of 0.64 signifies a reasonable equilibrium between the classes, while the weighted average of 0.67 demonstrates consistent overall performance, consistent with the data's unequal distribution. Subsequent aggregate analysis demonstrated an area under the ROC curve (AUC) of 0.899, signifying a strong capacity for class discrimination, alongside Kappa (0.599) and Matthews Correlation Coefficient (MCC) (0.600) metrics, which indicate moderate above-chance agreement and a consistent correlation between predictions and actual labels, underscoring the efficacy of ensembling stacking in addressing class imbalance.

Figure 6 visually depicts the efficacy of ensembling stacking through a confusion matrix, emphasizing the allocation of categorization failures among emotional categories.

The study's limitations encompass the lack of advanced hyperparameter optimization and intricate augmentation methods, in addition to the diminutive image size (48x48 pixels), which may hinder the detection of subtle expression details. The dependence on a singular dataset constrains the generalizability of the findings. Future directions encompass investigating more profound CNN architectures, enhancing data balancing methodologies on higher resolution image collections, and evaluating datasets such as RAF-DB (Real-world Affective Faces Database, comprising over 29,000 annotated real images [14]) and FER2013 (Facial Expression Recognition 2013, a benchmark featuring 35,000 grayscale images [15]) to ascertain the model's robustness.

TABLE IV. EVALUATION OF MODEL ON TESTING SET.

| Model | Accuracy | AUC | Recall | Precision | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| Stacking Classifier | 0.6726 | 0.8990 | 0.6726 | 0.6754 | 0.6725 | 0.5997 | 0.6000 |

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| angry | 0.54 | 0.59 | 0.56 | 714 |
| disgust | 0.90 | 0.51 | 0.65 | 90 |
| fear | 0.53 | 0.47 | 0.50 | 710 |
| happy | 0.87 | 0.87 | 0.87 | 1506 |
| neutral | 0.60 | 0.64 | 0.62 | 994 |
| sad | 0.50 | 0.51 | 0.50 | 714 |
| surprise | 0.80 | 0.78 | 0.79 | 626 |
| accuracy | - | - | 0.67 | 5354 |
| macro average | 0.68 | 0.62 | 0.64 | 5354 |
| weighted average | 0.68 | 0.67 | 0.67 | 5354 |

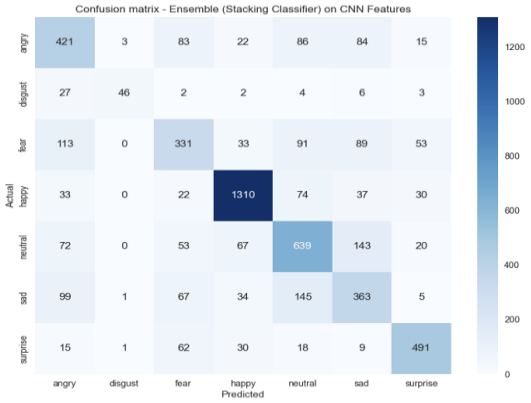


Fig. 6. Confusion matrix for model on testing set.

This study enhances the current literature by illustrating the viability of a hybrid method and introduces new avenues for the advancement of emotion recognition systems. This study's results demonstrate the efficacy of the suggested hybrid method, which integrates a proprietary CNN with ensembling voting and ensembling stacking classifier approaches, for categorizing face emotions in the "Face Expression Recognition Dataset." The preliminary CNN model attained a testing set accuracy of 0.6672, indicating robust performance of the softmax layer, whereas ensembling voting (0.6713) and stacking (0.6726) exhibited similar stability, corroborated by AUC values (0.9121 for voting and 0.8990 for stacking). An in-depth examination of the classification reports indicated exceptional performance for the "happy" class, supported by substantial support (1506 images); however, the "angry" and "sad" classes exhibited low F1-score (below 0.56), implying challenges in discerning nuanced features, likely affected by data complexity and class imbalance.

A notable element of the study was the effect of data pre-processing, evidenced by an accuracy enhancement exceeding 3% in training with cropped images relative to a network that processed uncropped images (which achieved an accuracy of approximately 64%, based on preliminary tests conducted by the author), underscoring the vital role of face isolation and resizing. This enhancement underscores the significance of pre-processing techniques in maximizing model efficacy. The combined contributions of the proprietary CNN architecture, balancing approaches like SMOTE, ensembling strategies (voting and stacking), and thorough pre-processing were crucial for the achieved results, demonstrating a concerted effort to optimize classification efficiency.

## V. CONCLUSIONS

The hybrid approach, which extracts features from CNN and incorporates them into machine learning modules, provides enhanced flexibility, enabling additional modifications through techniques like SMOTE and ensembling. The subpar performance of the "disgust" category (accuracy 0.92-0.94, recall 0.51-0.54), along with the insufficient support (90 images), underscores the necessity for enhanced data balancing methodologies. The Kappa (0.5935 for voting, 0.5915 for stacking) and MCC (0.5935 for voting, 0.5916 for stacking) metrics demonstrate moderate agreement exceeding chance, confirming the efficacy of ensembling voting and ensembling stacking in addressing imbalance; however, the slight enhancements imply a limitation on current performance.

## REFERENCES

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[2] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[3] Ohéix, J. (n.d.). Face Expression Recognition Dataset. Kaggle. Retrieved September 01, 2025, from https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset/data.

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

[5] Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), 3-19. https://doi.org/10.1007/978-3-030-01234-2_1

[6] Teja, R. (2023). Emotion Classification on Face Expression Dataset. Kaggle. Retrieved September 01, 2025, from https://www.kaggle.com/code/ravitejakemidi/emotion-classification-on-face-expression-dataset.

[7] Kharwal, A. (2023). Facial Emotion Recognition with Kaggle Dataset. Kaggle. Retrieved September 01, 2025, from https://www.kaggle.com/code/amanjharwal/facial-emotion-recognition-with-kaggle-dataset.

[8] Li, J.; Jin, K.; Zhou, D.; Kubota, N.; Ju, Z. Attention mechanism-based CNN for facial expression recognition. Neurocomputing 2020, 411, 340–350.

[9] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv preprint arXiv:1603.04467. https://arxiv.org/abs/1603.04467.

[10] Chollet, F. (2015). Keras: Deep Learning library for Theano and TensorFlow. https://keras.io.

[11] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools, 120; 122-125.

[12] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research, 10, 1755-1758. https://dl.acm.org/doi/10.5555/1577069.1755843.

[13] Dlib. (n.d.). shape_predictor_68_face_landmarks.dat. Retrieved September 01, 2025, from http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2.

[14] Li, S., Deng, W., & Du, J. (2017). Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Goodfellow, I., et al. (2013). Challenges in Representation Learning: A Report on Three Machine Learning Contests. arXiv.1307.0414