REAL-TIME FACIAL EXPRESSION RECOGNITION

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Abstract

This study explores the development and evaluation of a real-time Facial Expression Recognition (FER) system using computer vision and deep learning techniques. While traditional sentiment analysis has focused primarily on text, facial expressions offer a more intuitive and immediate channel for emotion recognition. Leveraging Convolutional Neural Networks (CNN), DenseNet, and Random Forest classifiers, this project compares model performance on benchmark datasets and real-time webcam input. Comprehensive preprocessing techniques—such as normalization, data augmentation, and class imbalance correction—were implemented to improve model robustness.

Experimental results indicate that CNN achieved the highest classification accuracy, while DenseNet provided superior generalization and computational efficiency in real-time conditions. Key challenges addressed include pose variation, occlusion, lighting variability, and latency. The project emphasizes the need for ethically aware and computationally efficient FER systems.

By integrating deep learning models with real-time data processing, this research contributes to the advancement of emotionally intelligent systems and highlights future directions such as lightweight architectures, explainable AI, and privacy-preserving techniques.

I. Introduction

1.1. State the Problem

In recent years, artificial intelligence (AI) has made significant progress in enabling machines to perceive, interpret, and respond to human behavior. Among various subfields, emotion and sentiment analysis has emerged as a pivotal area in human-computer interaction, aiming to recognize and interpret human affective states. Traditional sentiment analysis methods have primarily relied on textual input; however, emotions are often conveyed more vividly through non-verbal cues such as facial expressions, body posture, and vocal tone. This shift has driven increased interest in incorporating computer vision (CV) techniques into affective computing to process visual emotional signals more accurately.

Facial Expression Recognition (FER), a specialized area within emotion analysis, has gained substantial attention due to its wide-ranging applications, including intelligent tutoring systems, psychological health monitoring, human-robot interaction, and customer behavior analysis. The integration of deep learning—particularly convolutional neural networks (CNNs)—with large-scale annotated datasets has significantly enhanced the performance of FER systems. These

models automatically learn hierarchical feature representations from raw facial images, outperforming traditional hand-crafted feature approaches.

Nonetheless, real-world FER systems still face considerable challenges such as variability in illumination, occlusion, head pose, and individual expression styles. Furthermore, many models trained on benchmark datasets demonstrate poor generalization in in-the-wild environments. The ethical implications of FER, including privacy concerns and algorithmic bias, also remain critical issues that must be addressed in practical deployments.

1.2. Objectives

This research aims to:

- Provide a comprehensive review of state-of-the-art emotion and sentiment recognition techniques using computer vision.
- Analyze and compare the performance of machine learning and deep learning models specifically Random Forest, CNN, and DenseNet in FER tasks.
- Implement and evaluate a real-time FER system using both benchmark datasets and webcam input.
- Address key limitations such as class imbalance, generalization, and ethical concerns.
- Propose recommendations to improve model accuracy, fairness, real-time efficiency, and deployment viability.

By pursuing these objectives, the study contributes to the advancement of emotionally intelligent AI systems that are accurate, real-time, and ethically responsible.

II. Literature Review

Emotion recognition is a cornerstone of affective computing, enabling AI systems to interpret human affect through data-driven analysis. Research in this field spans across modalities—text, speech, and vision—with facial expression recognition (FER) standing out as the most visually informative and universally applicable approach.

2.1 Foundations of Emotion Recognition in AI

Emotion recognition systems are generally based on either categorical models (e.g., Ekman's six basic emotions: happiness, sadness, anger, fear, surprise, and disgust) (Ekman, 1992) or dimensional models (e.g., valence-arousal scale). Affective computing, pioneered by Rosalind Picard (1997), laid the groundwork for building systems that simulate and respond to human emotional states. AI models are typically trained on labeled emotional datasets, learning to classify or estimate emotional states based on observed patterns.

Early sentiment analysis approaches relied on textual data and lexicon-based methods, which lacked the depth to interpret non-verbal cues. As a result, computer vision has emerged as a critical modality for visual emotion understanding, especially through facial analysis (Zeng et al., 2009).

2.2 Computer Vision-Based Facial Expression Recognition

Facial expressions serve as one of the most direct and universal indicators of human emotion. Traditional FER methods involved geometric and appearance-based techniques such as Local Binary Patterns (LBP), Gabor filters, and Histogram of Oriented Gradients (HOG) (Corneanu et al., 2016). Although interpretable and efficient, these hand-crafted features struggled under real-world variability like lighting changes or facial occlusions.

The adoption of computer vision techniques enabled automatic face detection, alignment, and normalization using tools like Dlib, OpenCV, and MediaPipe. These preprocessing steps improved consistency and accuracy in FER pipelines, particularly in uncontrolled environments (Ko, 2018).

2.3 Deep Learning and CNN-Based FER Models

The rise of deep learning, particularly Convolutional Neural Networks (CNNs), marked a significant leap in FER performance. CNN architectures such as VGGNet, ResNet, Inception, and DenseNet have demonstrated superior capabilities in learning multi-level spatial features directly from facial images. These networks are often trained on benchmark datasets like FER-2013, CK+, AffectNet, and JAFFE (Li and Deng, 2020).

Hybrid models combining CNNs with Long Short-Term Memory (LSTM) networks or attention mechanisms have also shown promise, especially in capturing temporal dynamics from video input. For instance, Hasani and Mahoor (2017) proposed a 3D-CNN + LSTM approach in their paper “Facial Expression Recognition Using Enhanced Deep 3D Convolutional Neural Networks,” which improves recognition of subtle or transient emotions. Despite performance gains, issues such as dataset bias, overfitting, and lack of robustness in real-world scenarios remain prevalent.

Studies such as Mollahosseini et al. (2017), in their work "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," report that while CNNs achieve accuracies ranging from 65% to 80% on controlled datasets, their generalizability to in-the-wild settings is limited. Furthermore, most FER systems still operate as black boxes, raising questions about explainability and user trust.

2.4 Current Gaps and Future Directions

Despite rapid advancements, several challenges persist in FER research:

- **Dataset Limitations**: Many datasets lack diversity in terms of age, ethnicity, and environmental conditions, leading to biased model behavior.

- **Real-Time Constraints**: Deep FER models are often computationally intensive, making them less suitable for deployment on edge devices.

- **Ethical and Privacy Concerns**: The use of facial data in FER poses significant risks related to surveillance, consent, and data misuse.

- **Lack of Explainability**: Most CNN-based models provide little insight into their decision-making processes, limiting interpretability.

To address these challenges, current research efforts are exploring:

- The development of larger, demographically diverse, and ethically sourced datasets (Zhao et al., 2021).

- Lightweight deep learning architectures (e.g., MobileNet, EfficientNet) optimized for real-time inference.

- The integration of Explainable AI (XAI) techniques, such as saliency maps or Grad-CAM, to enhance model transparency (Poria et al., 2017).

- Privacy-preserving approaches like federated learning and differential privacy for secure FER deployment.

In summary, while FER has benefited significantly from the integration of computer vision and deep learning, future research must prioritize fairness, robustness, explainability, and ethical integrity to ensure responsible adoption in real-world applications.

III. Data and Methodology

3.1 Dataset Description

The dataset used in this study is the Real-world Affective Faces Database (RAF-DB), a well-established benchmark for facial expression recognition tasks. RAF-DB contains 15,339 facial images sourced from the internet under a wide range of real-world conditions. These conditions include variations in illumination, occlusion, background clutter, facial pose, age, gender, and ethnicity, providing a realistic and diverse sample set for robust model training.

Each image in the dataset is manually labeled with one of seven basic emotion classes: *Angry, Disgusted, Fearful, Happy, Neutral, Sad*, and *Surprised*.

The dataset is available in two primary formats:

- CSV label files (train_labels.csv and test_labels.csv), which contain mappings between image filenames and their corresponding emotion labels. These files do not include raw pixel data.
- A directory of .jpg image files, structured into subfolders for training and testing. Each subfolder corresponds to a specific emotion class, allowing direct use in image-based deep learning pipelines.

In this project, only the image files in .jpg format were used for all modeling strategies. These images were organized into separate training and testing directories according to RAF-DB's predefined annotations.

3.2 Exploratory Data Analysis (EDA)

To develop a robust emotion recognition model, it is crucial to examine the dataset in terms of class distribution and visual characteristics prior to training. This exploratory data analysis (EDA) focuses on understanding the variability of facial expressions, the extent of class imbalance, and how the dataset is structured across training and testing subsets.

An analysis of class frequency in the training set reveals a notable imbalance, as shown in **Figure 1**. The *Happy* and *Neutral* categories dominate the dataset, while *Disgusted*, *Fearful*, and *Angry* are underrepresented. Such imbalance may bias models toward majority classes and reduce accuracy for minority emotions if not properly addressed. This observation justifies the later application of class balancing techniques in the training pipeline.

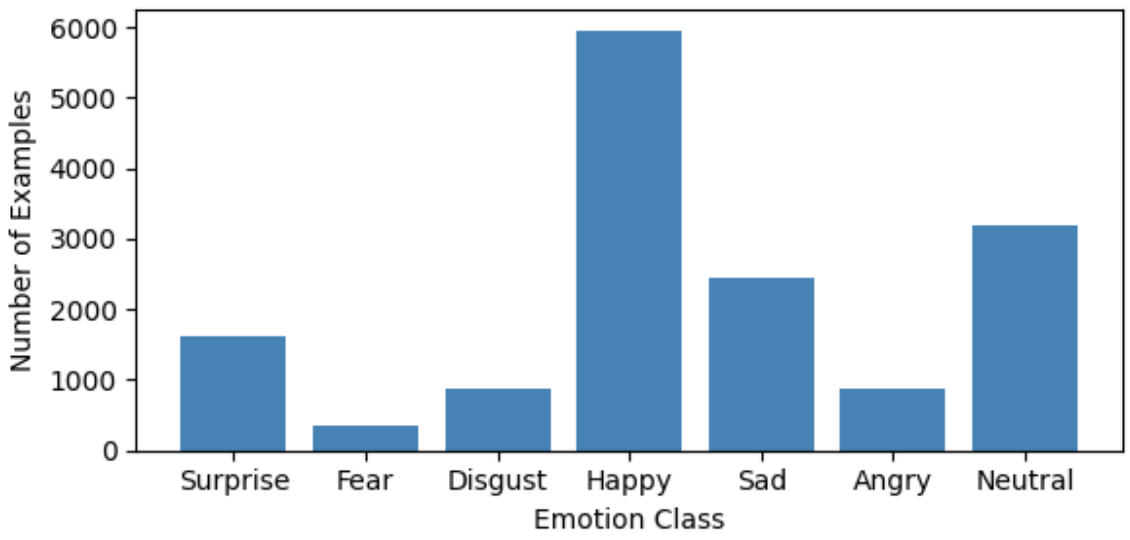


Figure 1: *Emotion label distribution in the RAF-DB training dataset*

To further assess the dataset structure, class distributions in both the training and test sets were compared. As illustrated in **Figure 2**, the imbalance persists across both subsets. The *Happy* class accounts for nearly 39% of training data and 18.6% of test data, while classes such as *Fearful* and *Disgusted* remain consistently low in both sets. This reinforces the notion that the dataset's inherent imbalance is not merely a byproduct of data splitting but a fundamental characteristic of RAF-DB.

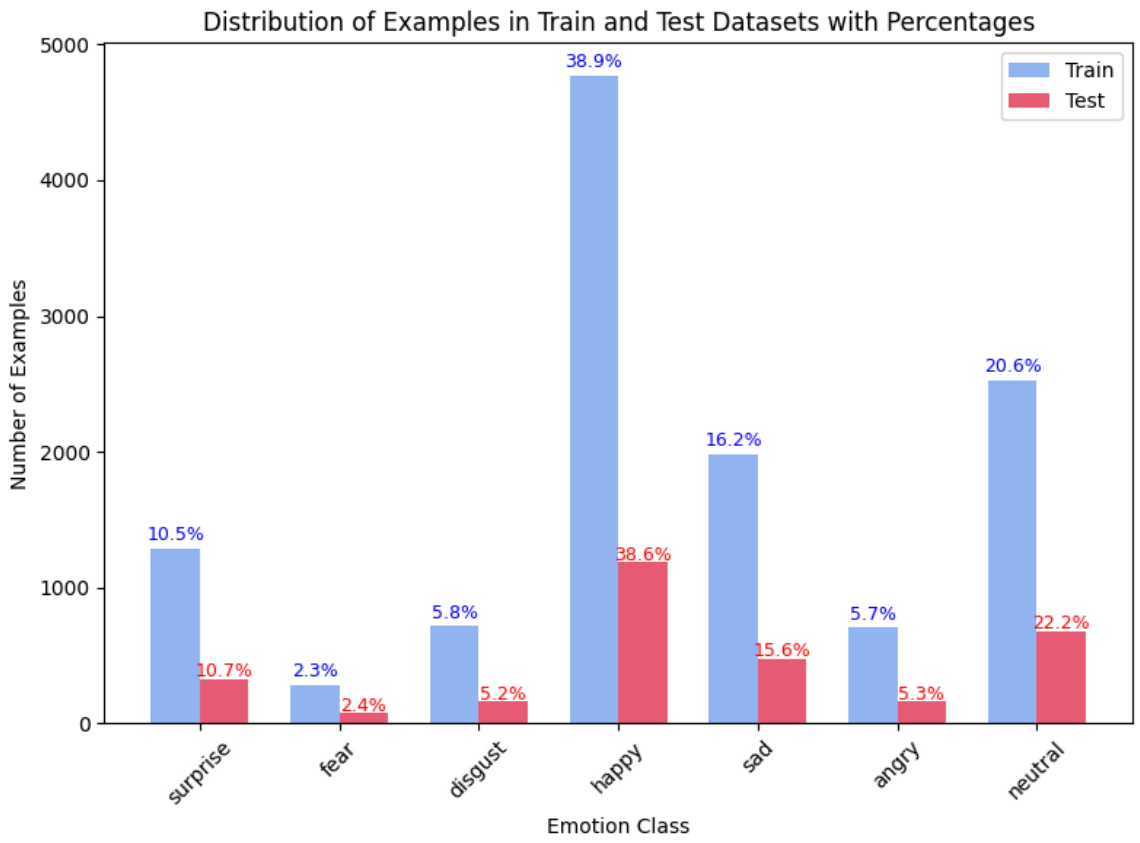


Figure 2: *Comparison of emotion class distributions in training and test datasets with percentages*

A visual inspection of representative training images is presented in **Figure 3**. These examples illustrate the diversity of facial expressions across different emotion categories, as well as variations in age, gender, lighting, and head pose. Certain emotions, such as *Surprise* or *Happy*, appear more pronounced and visually distinguishable, whereas others like *Neutral*, *Sad*, or *Fear* exhibit subtle differences. This diversity highlights the real-world complexity of facial expression recognition and presents a challenge for classification models.

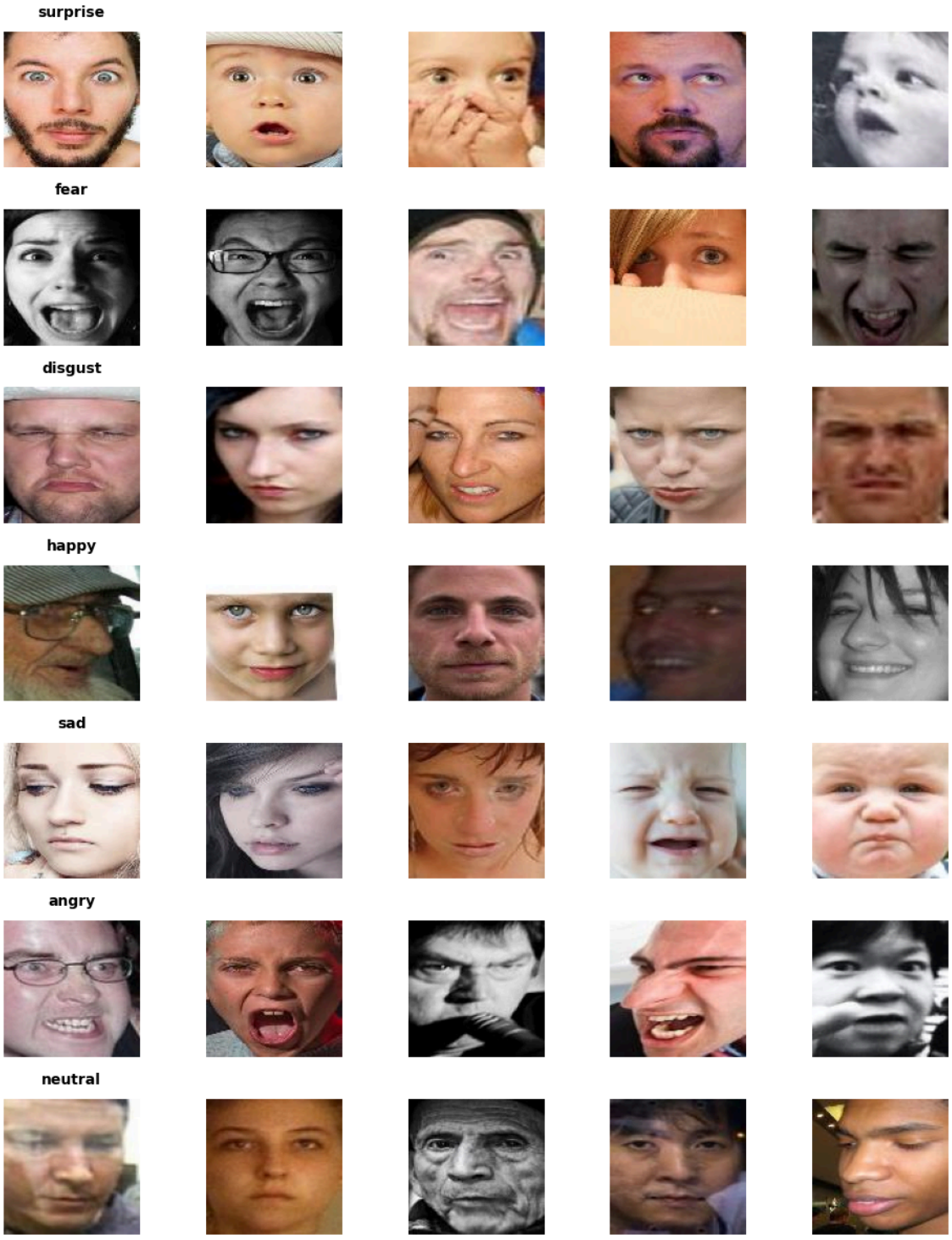


Figure 3: *Representative training images for each emotion class in RAF-DB*

Overall, this exploratory analysis highlights the real-world challenges posed by the RAF-DB dataset. The presence of class imbalance, intra-class variation underscores the importance of

robust preprocessing, augmentation, and model regularization strategies in subsequent stages of the pipeline.

IV. Implementation

4.1 . Handling Class Imbalance

A notable challenge of the RAF-DB dataset lies in its severe class imbalance. Emotions like Happy and Neutral dominate the distribution, while others such as Disgusted, Fearful, and Angry are significantly underrepresented. To mitigate this, a systematic three-step strategy was applied, as illustrated in Figure 6.

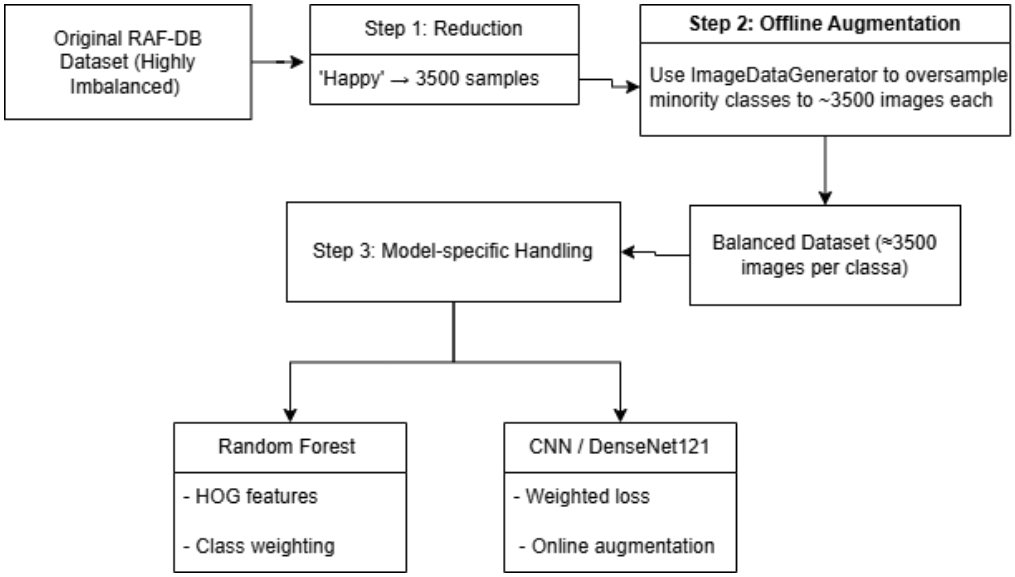


Figure 4: *Pipeline for class imbalance handling in RAF-DB*

The first step involved reducing the majority class. Specifically, the Happy class was downsampled to 3500 images using a custom function (reduce_class()), helping to curb the model's bias toward overly frequent classes. This reduction served as a foundation for creating a more balanced dataset.

In the second step, minority classes were synthetically expanded using targeted offline data augmentation. A class-aware ImageDataGenerator was employed to generate additional samples for underrepresented classes through transformations such as horizontal flips, rotations, zooming, and channel shifts. This augmentation continued until each class reached a uniform target size, resulting in a balanced and diverse training dataset with approximately 3500 images per class.

In the final step, this balanced dataset was used as a common training input for all models, with additional class imbalance handling tailored to each architecture. For the Random Forest model, HOG features were extracted from grayscale images and passed to a classifier trained using class_weight='balanced' to emphasize minority classes. In contrast, CNN and DenseNet121 were trained on normalized and reshaped images using a weighted categorical cross-entropy loss function, where class weights were calculated based on inverse class frequency. Furthermore, real-time online augmentation was applied during training to introduce further variation and improve generalization.

This combined approach beginning with majority class reduction, followed by minority class augmentation, and ending with model-specific techniques ensured all models were trained on a more equitable dataset distribution while leveraging methods best suited to their respective strengths.

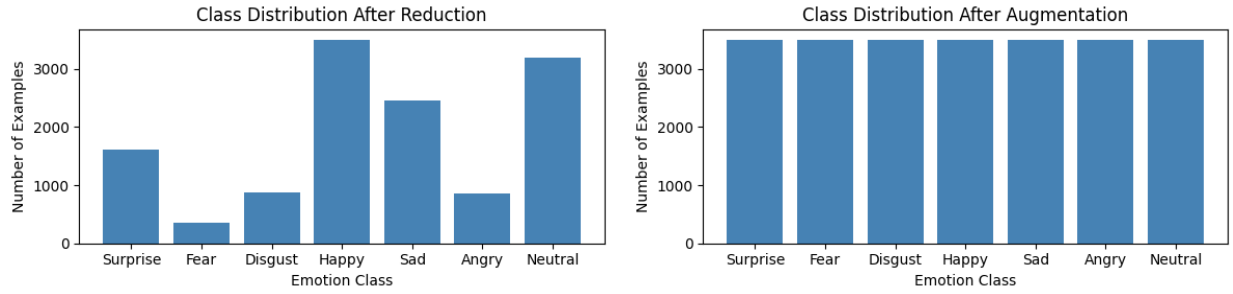


Figure 5. *Distribution after majority class reduction and balancing via augmentation*

4.2. Preprocessing and Normalization

Preprocessing plays a crucial role in preparing the RAF-DB dataset for effective training of machine learning and deep learning models. Given the dataset's dual format—CSV-based grayscale pixel values and image files in JPG format—distinct preprocessing strategies were applied to accommodate the requirements of different models.

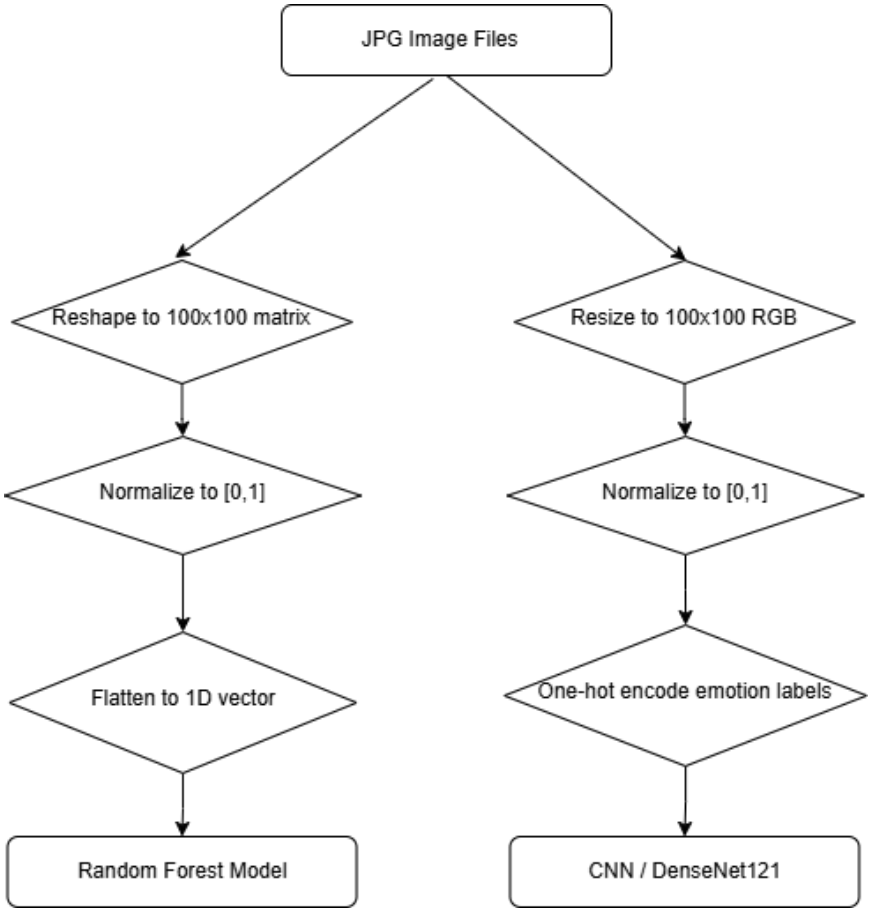


Figure 6. *Preprocessing workflows*

For traditional machine learning approaches such as Random Forest, the raw pixel data from the CSV file was first parsed and reshaped. Each image, originally stored as a space-separated string of pixel values, was converted into a 100×100 grayscale matrix. These matrices were then flattened into one-dimensional arrays to conform to the input format expected by scikit-learn classifiers. To ensure numerical stability and comparability across samples, pixel intensities were normalized to a [0, 1] range by dividing all values by 255.

For deep learning models such as Convolutional Neural Networks (CNNs) and DenseNet121, the preprocessing pipeline leveraged image files directly. Using Keras' ImageDataGenerator, all images were resized to a uniform shape of 100×100 pixels with three RGB channels. While the original grayscale information was preserved in the Random Forest pipeline, RGB conversion was used for CNN-based models to fully utilize pretrained weights from models like DenseNet121, which were originally trained on ImageNet. In this context, normalization was also applied by scaling pixel values to a [0, 1] range.

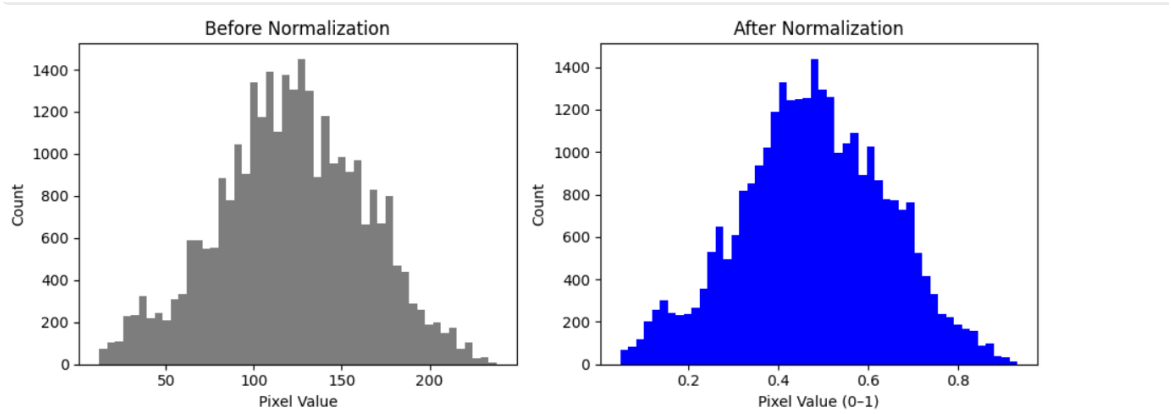


Figure 7: *Distribution of pixel intensities before and after normalization*

Histogram of pixel values before and after normalization. The left plot shows the distribution of original pixel intensities in the [0, 255] range, while the right plot shows the same image after rescaling to the [0, 1] range. Normalization ensures consistent input across samples and stabilizes training.

Furthermore, the dataset was split into training and testing sets following the original annotations provided by RAF-DB. Labels were converted from integer form to one-hot encoded vectors to serve as categorical targets in neural network models. This encoding ensured compatibility with the softmax activation used in the final classification layer.

The preprocessing strategy was carefully designed to preserve the semantic content of the images while ensuring that the data was in an appropriate format for each model type. These steps laid the foundation for subsequent stages such as class balancing and data augmentation, which further enhanced model robustness.

4.3. Data Augmentation

To further enhance model generalization and reduce overfitting, data augmentation techniques were applied during training. Given the relatively limited number of samples in minority emotion classes, especially under real-world variations in facial pose and lighting, augmentation helped introduce synthetic diversity without collecting new data.

For convolutional models such as CNN and DenseNet121, augmentation was implemented using Keras' ImageDataGenerator. This utility allowed for real-time generation of modified images during the training process, ensuring that the model was exposed to a more varied input distribution in each epoch.

The augmentation pipeline included the following transformations:

- Horizontal flipping, to account for left-right symmetry in facial expressions
- Random rotation up to 20 degrees, to simulate head tilts and non-frontal views
- Zooming (range of ±20%), to reflect slight distance variations from the camera
- Shear and width/height shifts, to emulate spatial distortion or misalignment

Figure 8. *Examples of augmented training images generated with horizontal flip, rotation, zoom, shift, and shear transformations using ImageDataGenerator.*

All augmented images retained their original label, preserving semantic consistency. Importantly, these transformations were applied only to the **training set**, not the validation or test sets, ensuring that evaluation remained fair and unbiased.

This strategy proved particularly beneficial in addressing two key challenges: the scarcity of minority class examples and the high intra-class variability present in RAF-DB. By exposing the model to a broader set of training conditions, augmentation encouraged the learning of more generalizable features and reduced the risk of memorizing training samples.

V. Model Architecture

To evaluate the effectiveness of different learning approaches in facial expression recognition, three distinct model architectures were implemented and compared: a traditional machine learning classifier (Random Forest), a custom-designed Convolutional Neural Network (CNN), and a pre-trained deep neural network (DenseNet121) using transfer learning.

5.1 Random Forest with HOG Features

The Random Forest classifier served as a baseline model to compare the performance of deep learning methods with traditional feature-based classification. Prior to training, Histogram of Oriented Gradients (HOG) features were extracted from the grayscale images reshaped from the CSV dataset. These features captured edge orientations and localized gradient structures, which are known to be effective in encoding facial geometry.

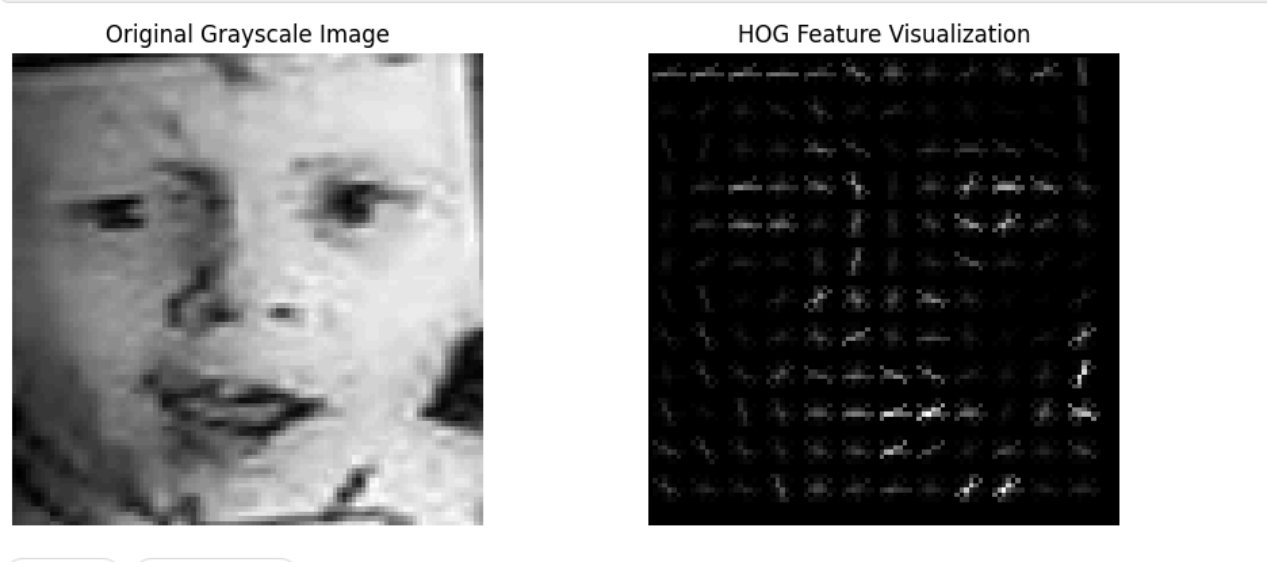


Figure 9. *Visualization of HOG (Histogram of Oriented Gradients) features.*

The left image shows the original grayscale facial image, while the right image illustrates the extracted HOG features that capture gradient orientation patterns commonly used in traditional facial expression recognition methods.

The extracted features were used to train a Random Forest model consisting of multiple decision trees, each trained on a random subset of the data. This ensemble approach offered robustness against overfitting and was computationally efficient. However, the performance of the model was constrained by its reliance on handcrafted features and limited representational power.

5.2 Convolutional Neural Network (CNN)

The second model was a custom-built CNN architecture tailored for image classification tasks. The network consisted of multiple convolutional layers followed by pooling and fully connected layers.

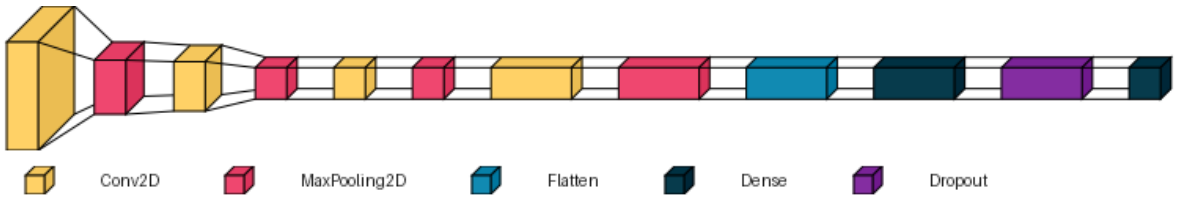


Figure 10 : *CNN architecture*

ReLU activations and dropout were employed to improve non-linearity and reduce overfitting, respectively.

The CNN was trained end-to-end on RGB image data resized to 100×100 pixels. Its ability to learn hierarchical spatial features directly from the raw image inputs allowed it to outperform the traditional Random Forest model. However, due to its limited depth and parameter scale, the CNN's capacity to generalize remained lower than that of more advanced architectures.

5.3 DenseNet121 (Transfer Learning)

The final and most powerful model in the experiment was DenseNet121, implemented using transfer learning. DenseNet121 is a deep convolutional neural network architecture pre-trained on the large-scale ImageNet dataset. In this study, the pre-trained weights were retained for the feature extraction layers, while the top classification layers were replaced with a custom output head tailored to the seven RAF-DB emotion classes.

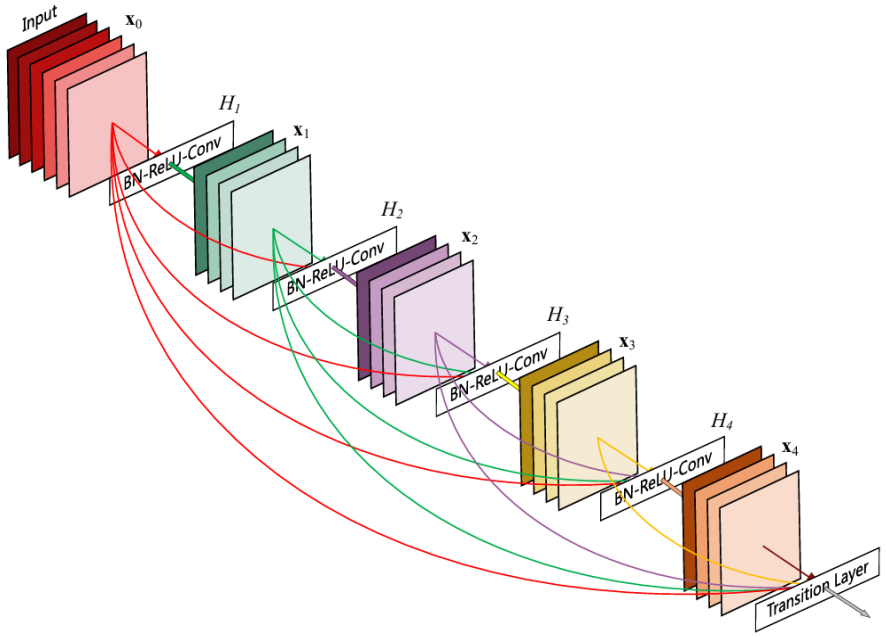


Figure 11: *An illustration of a 5-layer dense block with a growth rate of k = 4. Each layer receives input from all previous layers, exemplifying the core concept of dense connectivity in DenseNet*

The use of transfer learning enabled the model to leverage general visual representations learned from millions of natural images, significantly improving performance on a relatively small facial expression dataset. Only the final classification layers were fine-tuned on RAF-DB, allowing for efficient convergence and enhanced generalization even under limited data conditions.

By integrating both handcrafted and deep learning approaches, the study offered a comprehensive evaluation of model performance across different levels of complexity. The use of DenseNet121 as a transfer learning model highlighted the advantages of leveraging pre-trained knowledge for emotion recognition tasks in real-world scenarios.

VI. Evaluation Metrics

To assess the performance of the implemented models Random Forest, CNN, and DenseNet121 this study employed several standard evaluation metrics. These include accuracy, precision, recall, and F1-score, which together provide a more complete picture than accuracy alone, especially in the presence of class imbalance.

Accuracy reflects the overall proportion of correct predictions, but may overestimate performance on imbalanced datasets. Precision measures the correctness of positive predictions, while recall evaluates how well the model detects all actual positive cases. F1-score combines both into a single balanced metric.

The formulas and meanings of each metric are summarized below:

Table 1 : Summary of the performance evaluation metrics used for facial expression recognition.

Metric	**Formula**	**Meaning**
Accuracy	$Accuracy = \frac{TP + TN}{TP+TN+FP+FN}$	Proportion of correct predictions across all emotion classes. May be misleading with class imbalance.
Precision	$Precision = \frac{TP}{TP+FP}$	How many predicted positive labels are actually correct. Reduces false positives.

Recall	$Recall = \frac{TP}{TP+FN}$	Measures how many actual positives are correctly identified. Reduces false negatives.
F1-Score	$F1 - socre = 2\frac{Precision * Recall}{Precision + Recall}$	Harmonic mean of precision and recall. Useful when data is imbalanced.

Confusion Matrix: A visual representation of prediction results across emotion classes. It reveals misclassification patterns and helps interpret how well each class is distinguished.

ROC-AUC Curve (for CNN and DenseNet): Although originally designed for binary classification, one-vs-rest ROC curves were computed for each emotion class. This provides a probabilistic view of separability, especially valuable in multiclass tasks like emotion recognition.

VII. Model Deployment and Web Integration

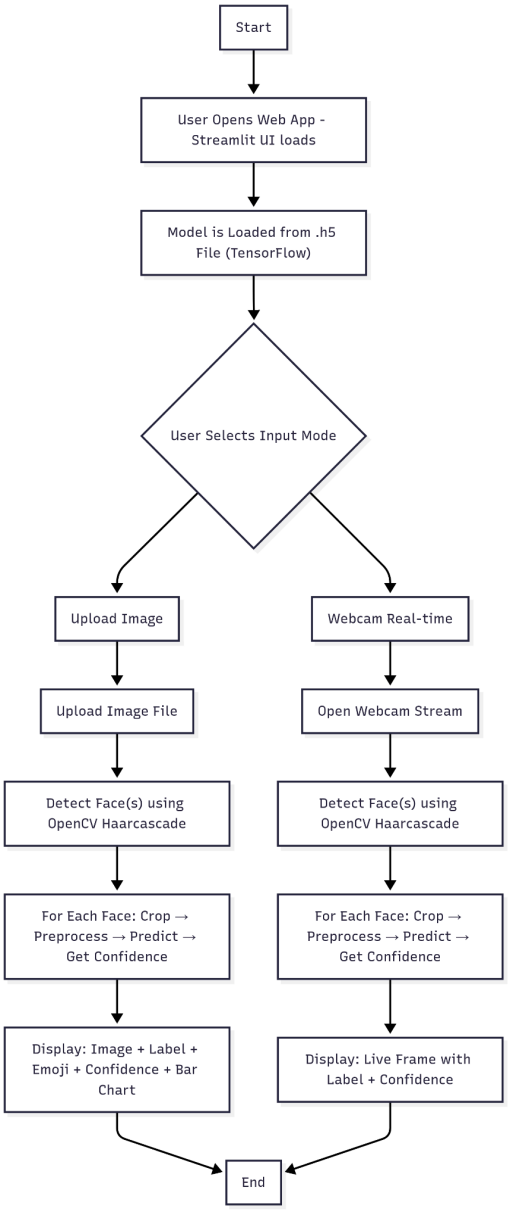


Figure 12: *Flow of web application execution.*

After completing the training and evaluation of the facial emotion recognition model, the system was deployed as an interactive web application using the Streamlit platform.The primary objective was to implement the deep learning model in a way that would enable end users to communicate with the system directly without needing to be highly knowledgeable about machine learning.

The application offers two main operating modes: uploading static images from the user's device or detecting emotions in real time via a webcam. The interface is designed to be intuitive and responsive, with full integration of prediction and visualization functionalities.

Upon launching the application, the trained deep learning model is loaded from an `.h5` file using TensorFlow. The loading process is optimized with `@st.cache_resource` to prevent unnecessary reloading across user sessions. Input data whether an uploaded image or a webcam frame is processed through OpenCV's Haar Cascade classifier to detect faces. Each detected face is converted to grayscale, resized to the target input size (100×100), normalized to the [0, 1] range, and channel-stacked to match the CNN's expected input format.

The model outputs a probability vector corresponding to seven emotion classes: surprise, fear, disgust, happy, sad, angry, and neutral. In the image upload mode, the system displays the detected face bounding box, the predicted emotion label, confidence score, corresponding emoji, and a bar chart showing the distribution of probabilities across all emotion classes. In webcam mode, real-time frames are displayed continuously with dynamic emotion predictions, showing bounding boxes and confidence scores without interrupting the video stream.

This solution demonstrates an effective integration of deep learning inference and web-based interactivity, resulting in a system that is both accessible and practical for real-time emotion recognition tasks. It supports both experimental evaluation and potential real-world deployment in applications such as education, healthcare, and human-computer interaction.

VIII. Experimental Results

8.1. Results

8.1.1 RAF-DB

-Accuracy:

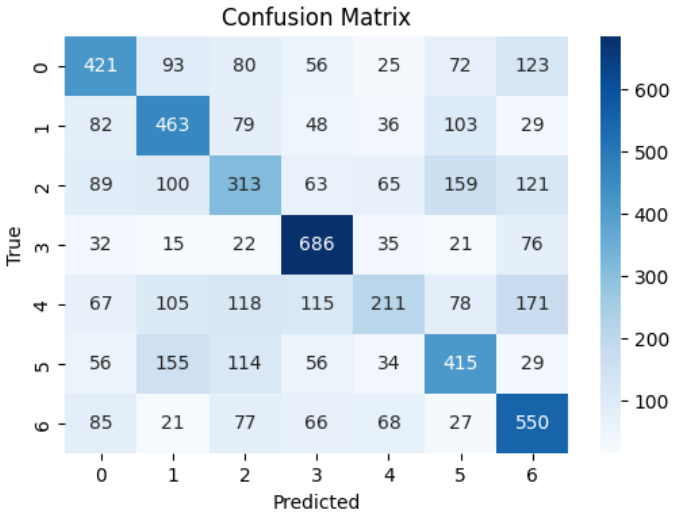
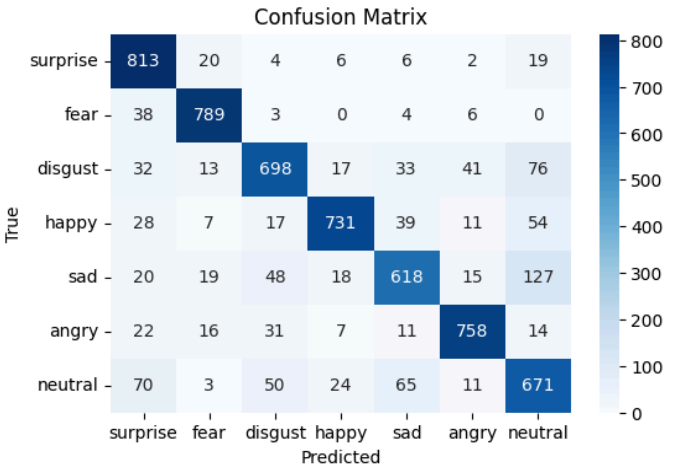
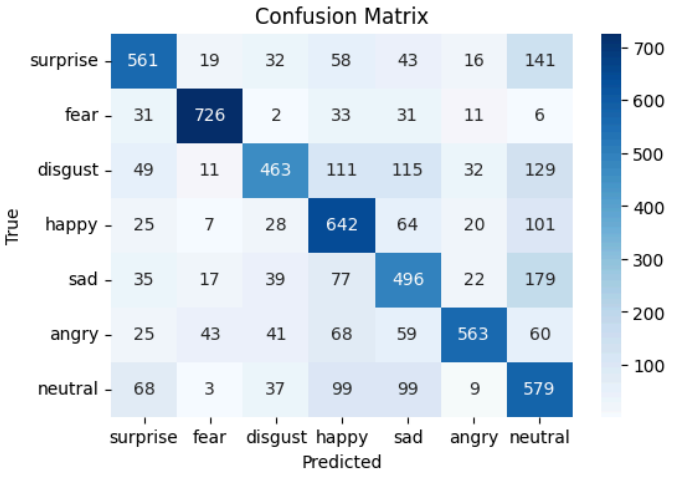
Model	Test Accuracy
Random Forest	49.94%
CNN	82.9%
DenseNet	65.8%

-Performance Metrics Evaluation:

Random Forest	Classification Report: precision recall f1-score support 1 0.51 0.48 0.49 870 2 0.49 0.55 0.52 840 3 0.39 0.34 0.37 910 4 0.63 0.77 0.69 887 5 0.45 0.24 0.32 865 6 0.47 0.48 0.48 859 7 0.50 0.62 0.55 894 accuracy 0.50 6125 macro avg 0.49 0.50 0.49 6125 weighted avg 0.49 0.50 0.49 6125
CNN	Classification Report: precision recall f1-score support surprise 0.79 0.93 0.86 870 fear 0.91 0.94 0.92 840 disgust 0.82 0.77 0.79 910 happy 0.91 0.82 0.87 887 sad 0.80 0.71 0.75 865 angry 0.90 0.88 0.89 859 neutral 0.70 0.75 0.72 894 accuracy 0.83 6125 macro avg 0.83 0.83 0.83 6125 weighted avg 0.83 0.83 0.83 6125
DenseNet	precision recall f1-score support surprise 0.71 0.64 0.67 870 fear 0.88 0.86 0.87 840 disgust 0.72 0.51 0.60 910 happy 0.59 0.72 0.65 887 sad 0.55 0.57 0.56 865 angry 0.84 0.66 0.73 859 neutral 0.48 0.65 0.55 894 accuracy 0.66 6125 macro avg 0.68 0.66 0.66 6125 weighted avg 0.68 0.66 0.66 6125

- Confusion Matrix:

Model	Confusion Matrix

Random Forest	
CNN	
DenseNet	

8.1.2 Fer-2013

-Accuracy:

Model	Test Accuracy
Random Forest	44.03%
CNN	61.47%
DenseNet	53.62%

-Performance Metrics Evaluation:

Random Forest

Classification Report:

	precision	recall	f1-score	support
1	0.39	0.27	0.32	1122
2	0.74	0.68	0.71	845
3	0.41	0.28	0.34	1198
4	0.53	0.39	0.45	852
5	0.31	0.50	0.38	1341
6	0.71	0.53	0.60	920
7	0.37	0.49	0.42	1396
accuracy			0.44	7674
macro avg	0.50	0.45	0.46	7674
weighted avg	0.47	0.44	0.44	7674

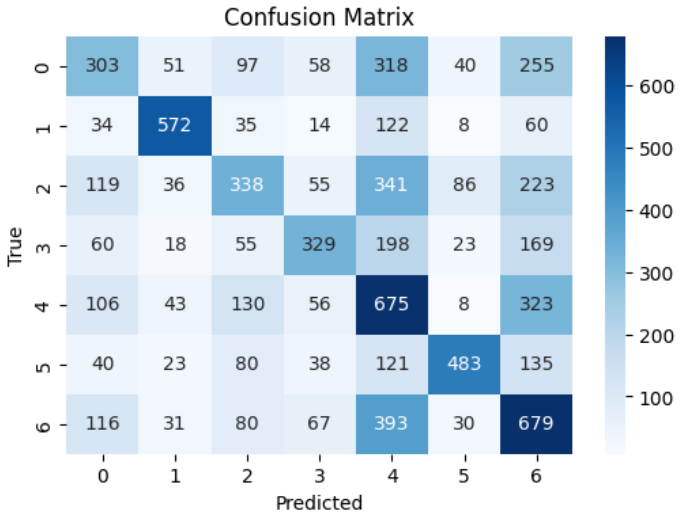
CNN

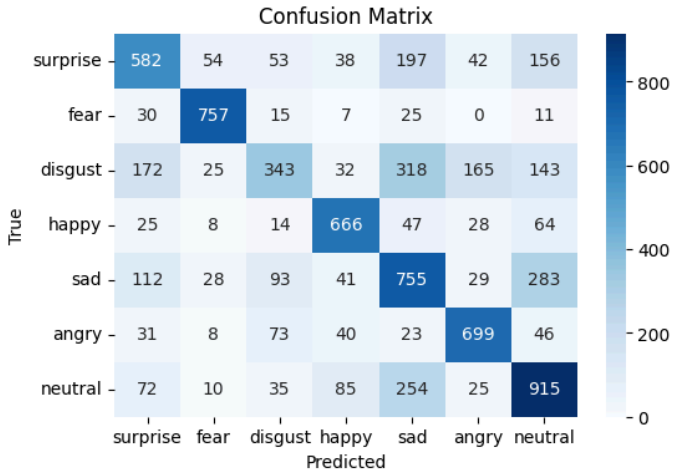
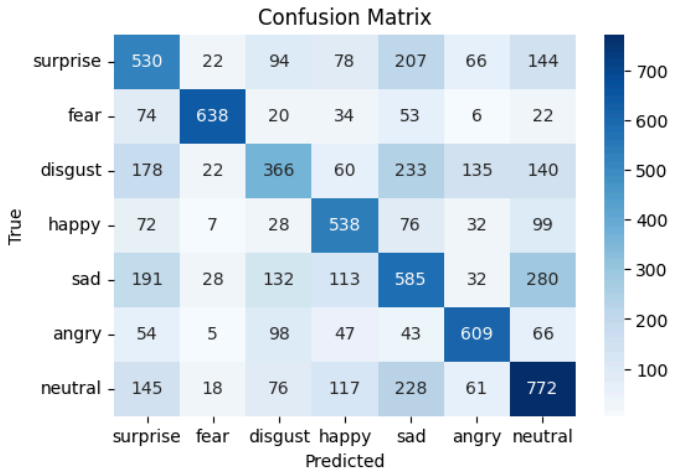
Classification Report:

	precision	recall	f1-score	support
surprise	0.57	0.52	0.54	1122
fear	0.85	0.90	0.87	845
disgust	0.55	0.29	0.38	1198
happy	0.73	0.78	0.76	852
sad	0.47	0.56	0.51	1341
angry	0.71	0.76	0.73	920
neutral	0.57	0.66	0.61	1396
accuracy			0.61	7674
macro avg	0.63	0.64	0.63	7674
weighted avg	0.61	0.61	0.61	7674

DenseNet	Classification Report: precision recall f1-score support surprise 0.43 0.46 0.44 1141 fear 0.86 0.75 0.80 847 disgust 0.45 0.32 0.38 1134 happy 0.55 0.63 0.59 852 sad 0.41 0.43 0.42 1361 angry 0.65 0.66 0.65 922 neutral 0.51 0.54 0.53 1417 accuracy 0.53 7674 macro avg 0.55 0.54 0.54 7674 weighted avg 0.53 0.53 0.53 7674

- Confusion Matrix:

Model	Confusion Matrix
Random Forest	

CNN	
DenseNet	

Experimental results indicate that the CNN model outperforms the other two models on both the FER-2013 and RAF-DB datasets. Notably, CNN achieved the best performance when trained on the RAF-DB dataset. Therefore, we selected this CNN model, saved its trained weights in an h5 file, and integrated it into the system for deployment in the web application.

8.2. User Integration

To evaluate the practical applicability of the proposed deep learning model, we developed a real-time facial emotion recognition system in the form of a web application. The system is implemented using **Streamlit**, an open-source Python framework designed to simplify the development of interactive user interfaces for machine learning models. The aim is to provide an intuitive, responsive, and deployable interface that bridges the gap between model development and end-user accessibility.

8.2.1. User Interface and System Architecture

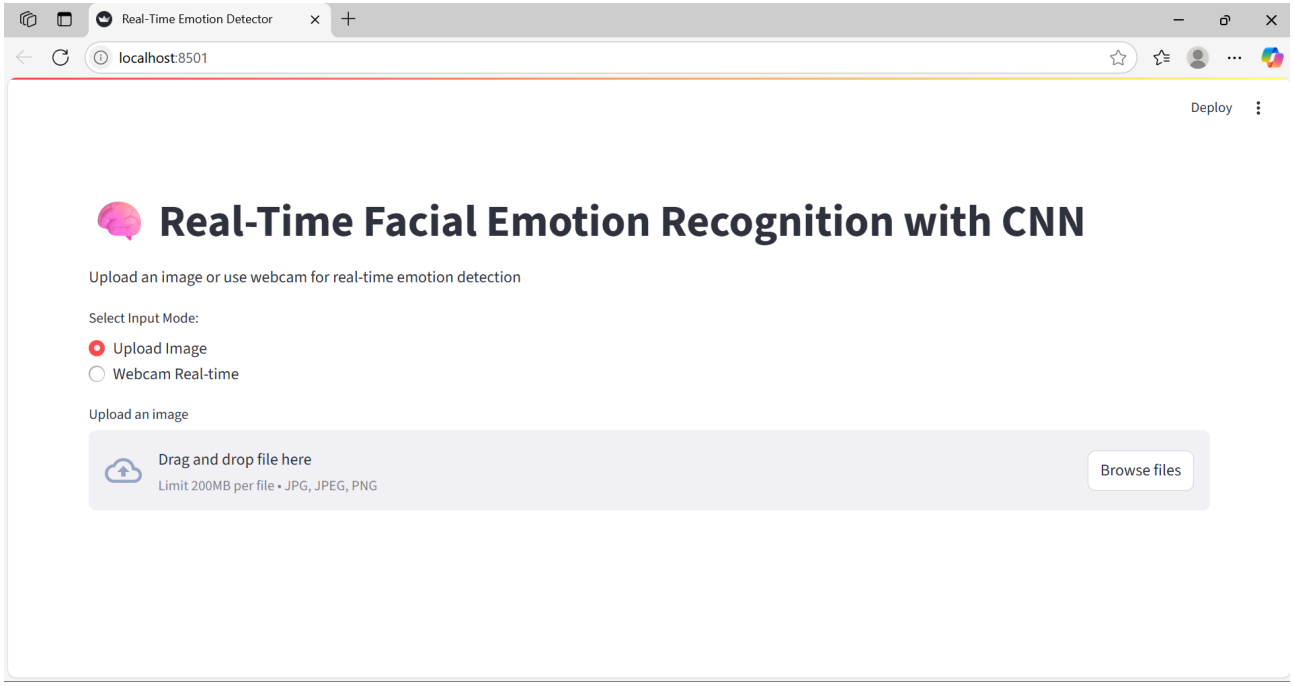
The user interface was designed to be minimalistic and user-friendly, allowing both technical and

non-technical users to interact with the system efficiently. The homepage displays the title *"Real-Time Facial Emotion Recognition with CNN"*, along with a brief description of the application's capabilities detecting facial emotions from either static images or real-time video streams.

Users are provided with two input modes:

- **Upload Image**: Allows users to upload a static image containing a human face for emotion analysis.
- **Webcam Real-time**: Enables the use of a device's webcam to detect and classify facial emotions in real time.

Once the input mode is selected, the system processes the input, detects faces, performs emotion classification, and displays the results directly on the web interface, including bounding boxes, emotion labels, emoji representations, and a bar chart of class-wise probabilities.



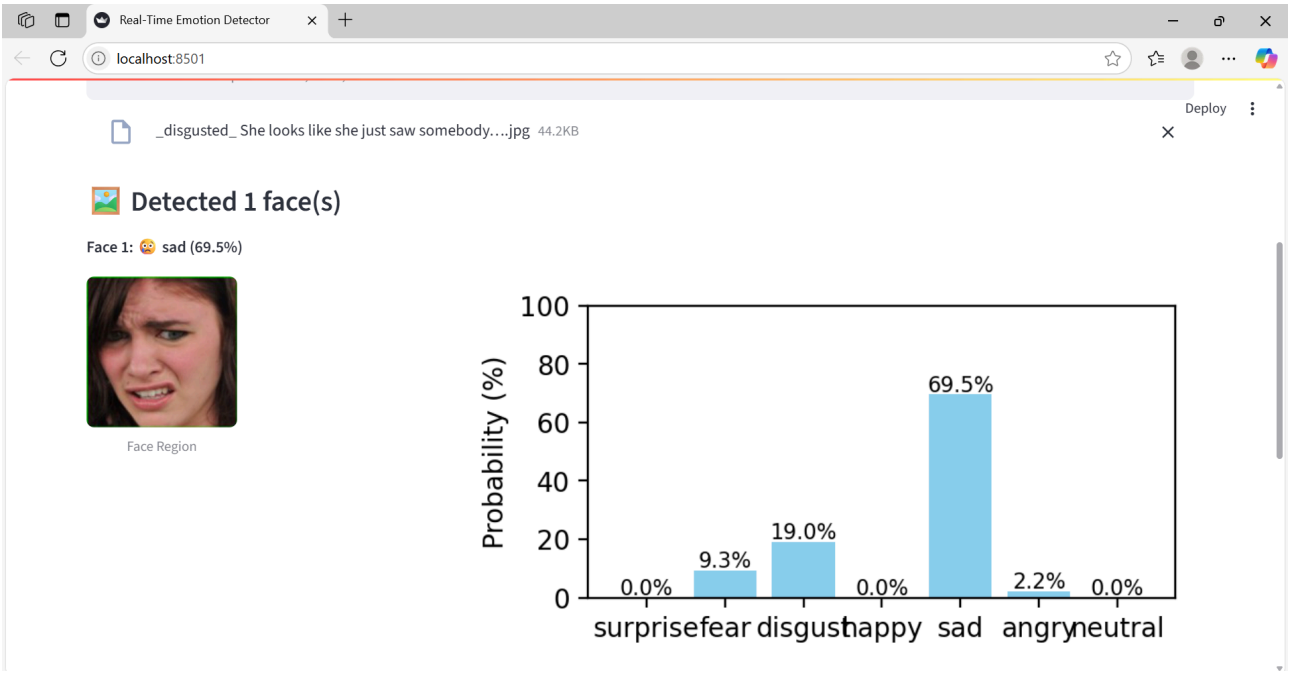
8.2.2. Emotion Recognition from Static Images

In the *Upload Image* mode, users can upload images in .jpg, .jpeg, or .png formats. The system performs the following steps:

- Detects all facial regions in the uploaded image;
- Extracts each face and applies the appropriate preprocessing pipeline;

- Uses the CNN model to predict the associated emotional state;
- Annotates the image with the predicted label, confidence score, and emoji overlay;
- Visualizes the prediction probabilities for each emotion class (e.g., *happy*, *sad*, *angry*, etc.) using a bar chart generated by matplotlib.

This visualization provides users with a clearer understanding of the model's confidence distribution and the degree of certainty associated with each prediction, which is especially useful for qualitative model evaluation.



8.2.3. Real-Time Emotion Recognition via Webcam

In the *Webcam Real-time* mode, the application accesses the device's webcam and processes incoming video frames in real time. The following procedures are executed continuously:

- Captures individual frames from the webcam feed;
- Detects facial regions in each frame;
- Applies preprocessing to the extracted face regions;
- Feeds the processed data into the CNN model for emotion prediction;
- Displays the detection results including bounding boxes, labels, emojis, and confidence scores directly on the live video stream.

The real-time output is rendered and updated continuously using Streamlit's st.image() function, enabling dynamic feedback with minimal latency. This interactive feature supports a variety of practical use cases, such as adaptive e-learning environments, psychological assessment interfaces, customer sentiment monitoring, and human-computer interaction systems.

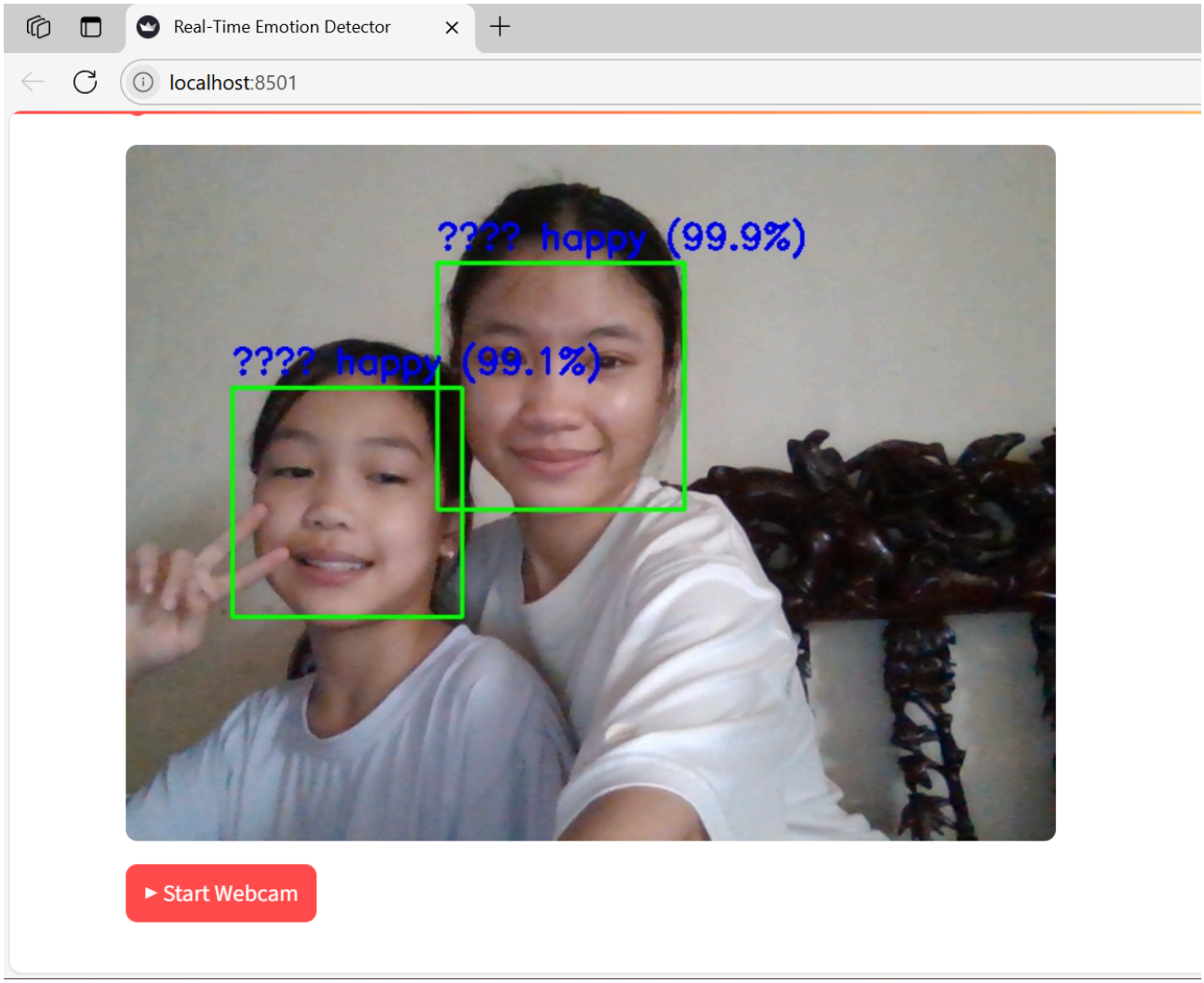


Figure 12: *Real-Time Emotion Recognition via Webcam*

IX. Conclusion and Recommendations

In this report, we designed and implemented a real-time facial expression recognition (FER) system leveraging both classical machine learning (Random Forest), modern deep learning architectures (CNN), Transfer Learning architectures (DenseNet). The study thoroughly addressed key stages including data preprocessing, augmentation to tackle class imbalance, model architecture design, and rigorous evaluation through both benchmark datasets and live webcam input.

Our findings reveal that:

- Deep learning models, particularly DenseNet and CNN, significantly outperform Random Forest in accuracy and robustness under variable conditions.

- Data augmentation, including geometric transformations and synthetic image creation, is essential to mitigate dataset bias and enhance model generalization.

- Real-time performance using webcam demonstrates the system's feasibility but also highlights critical challenges related to computation latency, lighting variation, and occlusion.

- Despite technological advancements, our system still faces practical limitations: susceptibility to changes in head pose, limited expression diversity in training data, and ethical considerations surrounding real-world deployment.

To elevate the reliability, fairness, and usability of FER systems, we recommend the following enhancements:

- Expand and diversify datasets: Collect facial expression images across different demographics (age, ethnicity, gender) and varied real-world environments, Annotate expressions with multiple labels to reduce annotation bias and improve model fairness.

- Adopt multimodal fusion: Complement facial expression data with other modalities such as speech or physiological signals (e.g., heart rate, skin conductivity) to reduce mispredictions from ambiguous facial cues.

- Optimize models for edge deployment: Implement model compression (quantization, pruning) and lightweight architectures (e.g., MobileNet, EfficientNet) to enable real-time performance on embedded devices and mobile platforms.

- Prioritize ethical design and interpretability: Integrate user consent mechanisms and ensure data is anonymized or stored securely, Apply Explainable AI (XAI) techniques—such as saliency maps to allow transparency in model predictions.

- Continuous benchmarking in realistic conditions: Perform ongoing evaluation with live webcam or CCTV data under diverse settings—indoors, outdoors, low-light, partial occlusion to uncover hidden failure modes and iteratively improve system resilience.

By following these recommendations, future FER systems will be more accurate, equitable, and viable for real-world applications from telemedicine and education to intelligent user interfaces and safety systems.

References

1. Corneanu, C.A., Simon, M.O., Cohn, J.F. and Guerrero, S.E. (2016) 'Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8), pp. 1548–1568. Available at: https://doi.org/10.1109/TPAMI.2016.2515606
2. Dhall, A., Goecke, R., Joshi, J., Wagner, M. and Gedeon, T. (2012) 'Emotion recognition in the wild challenge 2012: baseline, data and protocol', in *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI)*, Santa Monica, CA, pp. 552–556. Available at: https://doi.org/10.1145/2388676.2388776
3. Ekman, P. (1992) 'An argument for basic emotions', *Cognition and Emotion*, 6(3–4), pp.169–200. https://doi.org/10.1080/02699939208411068

4. Hasani, B. and Mahoor, M.H. (2017) 'Facial expression recognition using enhanced deep 3D convolutional neural networks', *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, pp. 2278–2283. Available at: https://doi.org/10.1109/CVPRW.2017.282
5. Ko, B.C. (2018) 'A brief review of facial emotion recognition based on visual information', *Sensors*, 18(2), p.401. https://doi.org/10.3390/s18020401

6. Li, S. and Deng, W. (2020) 'Deep facial expression recognition: A survey', *IEEE Transactions on Affective Computing*, 13(3), pp. 1195–1215. Available at: https://doi.org/10.1109/TAFFC.2020.2981446

7. Mollahosseini, A., Hasani, B. and Mahoor, M.H. (2017) 'AffectNet: A database for facial expression, valence, and arousal computing in the wild', *IEEE Transactions on Affective Computing*, 10(1), pp. 18–31. Available at: https://doi.org/10.1109/TAFFC.2017.2740923

8. Poria, S., Cambria, E., Bajpai, R. and Hussain, A. (2017) 'A review of affective computing: From unimodal analysis to multimodal fusion', *Information Fusion*, 37, pp. 98–125. Available at: https://doi.org/10.1016/j.inffus.2017.02.003

9. Zeng, Z., Pantic, M., Roisman, G.I. and Huang, T.S. (2009) 'A survey of affect recognition methods: Audio, visual, and spontaneous expressions', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1), pp. 39–58. Available at:

https://doi.org/10.1109/TPAMI.2008.52

10. Zhang, K., Zhang, Z., Li, Z. and Qiao, Y. (2016) ‘Joint face detection and alignment using multitask cascaded convolutional networks’, *IEEE Signal Processing Letters*, 23(10), pp. 1499–1503. Available at: https://doi.org/10.1109/LSP.2016.2603342