

Face Emotion Recognition System

Pradeepa S¹, Anamika S S², Mariammal Alias Madhu M³, Oviya S⁴

B.Tech. In Artificial Intelligence And Data Science, Sengunthar Engineering College, Tamil Nadu, India

Abstract- Facial expressions provide useful visual cues for human-computer interaction, but automatic interpretation remains challenging because expressions vary across individuals, lighting conditions, head pose, and image quality. This paper presents a face emotion recognition system that combines computer vision with a convolutional neural network (CNN) to classify facial expressions from images and live video. The system detects a face using OpenCV's Haar Cascade face detector, extracts and preprocesses the facial region, and passes a 48×48 grayscale image to a CNN trained using the FER-2013 dataset. The network uses convolutional blocks, batch normalization, max pooling, dropout, and a softmax output layer to classify seven expression categories: angry, disgust, fear, happy, neutral, sad, and surprise. The implementation also integrates separate pretrained age-group and gender models, allowing the application to display multiple facial attributes in a unified interface. Emotion confidence scores and bounding boxes are overlaid on the detected face, while temporal averaging across recent frames is used to reduce rapid changes in predictions. The project demonstrates an end-to-end facial analysis pipeline using Python, TensorFlow/Keras, and OpenCV. Quantitative performance should be reported from the final held-out test run; the project report supplied for this paper does not provide a verified final test-results table, so no unverified performance claim is made here.

Index Terms- Facial emotion recognition, convolutional neural network, deep learning, computer vision, FER-2013, OpenCV, real-time facial analysis.

I INTRODUCTION

Human communication conveys information through facial expressions as well as spoken and written language. Recognizing facial expressions automatically is therefore an active problem in computer vision and human-computer interaction. A facial emotion recognition system can support interfaces that respond to visible expressions and can be explored in settings such as learning applications, interactive systems, and user-experience research. However, facial appearance changes with illumination, pose, occlusion, image resolution, and individual differences, making reliable classification difficult.

Traditional approaches often rely on handcrafted descriptors such as Local Binary Patterns or Histogram of Oriented Gradients and classifiers such as support vector machines. These methods require feature design and may not capture all expression patterns. Convolutional neural networks learn hierarchical image features directly from labelled examples and are widely used for image classification.

This paper describes the design and implementation of a face emotion recognition system. Its main contribution is an integrated application pipeline: face localization, facial-image preprocessing, CNN-based emotion classification, optional age and gender estimation, and visual display of predictions. The system supports static image input and live webcam frames. The paper describes the architecture and implementation while distinguishing implemented design choices from performance values that require further measurement.

II. RELATED WORK

Facial expression recognition has been studied using both conventional machine-learning methods and deep neural networks. Handcrafted approaches extract texture, edge, or geometric features before classification. Their performance can be affected by changes in lighting, pose, and facial appearance. CNN-based approaches instead learn features through stacked convolutional layers and can model increasingly complex visual patterns.

The FER-2013 dataset is commonly used for facial-expression classification and contains grayscale facial images assigned to seven emotion labels. Its low-resolution images and class imbalance make it a useful but challenging benchmark. Data augmentation, regularization, class weighting, and careful validation are common strategies for improving generalization. The present project uses augmentation and class weighting in its CNN training pipeline and uses dropout and batch normalization to support training stability. In contrast to systems that only classify an already-cropped image, the proposed application also includes face detection and a real-time display pipeline.

III. SYSTEM ARCHITECTURE

The system is organized into five main stages: (1) image acquisition, (2) face detection and preprocessing, (3) emotion classification, (4) optional age and gender estimation, and (5) result visualization.

- **Image Acquisition:** The input is obtained from a webcam or a user-provided image. For live operation, frames are read continuously from the camera.
- **Face Detection:** OpenCV's Haar Cascade frontal-face detector identifies candidate face regions in each frame. The detector returns bounding-box coordinates that are used to crop the facial region. The project report also identifies deep-learning-based face detection as a possible alternative when improved robustness is required.
- **Preprocessing:** The cropped face is resized to 48×48 pixels and converted to grayscale for emotion classification. Pixel values are normalized to the interval $[0, 1]$. The age and gender networks use colour input prepared as a 227×227 blob with mean subtraction, consistent with the pretrained Caffe models used by the implementation.
- **Emotion Classification:** A CNN trained on FER-2013 estimates the probabilities of seven emotion classes. The class with the largest output probability is selected as the predicted expression. The system displays the selected class and its confidence score.
- **Age and Gender Estimation:** Separate pretrained Caffe-based models estimate a gender label and an approximate age range. The age model outputs one of eight age bands: 0–2, 4–6, 8–12, 15–20, 25–32, 38–43, 48–53, and 60–100. These outputs are independent predictions and should be treated as estimates rather than definitive personal attributes.
- **Result Visualization:** The application draws a bounding box around each detected face and displays emotion, confidence, gender, and age-group labels. Emotion probabilities can be shown as horizontal bars. To reduce frame-to-frame flicker, the implementation averages predictions from the most recent three frames.

Cnn Model and Training

The emotion model accepts a 48×48 grayscale image. The documented V3 architecture begins with pixel rescaling and data augmentation, including horizontal flipping, small rotations, zooming, and translations. The CNN then uses four convolutional blocks with increasing channel depth. Convolutional layers use ReLU activation, and batch-normalization layers are used to stabilize intermediate activations. Max-pooling layers reduce spatial dimensions, while dropout layers reduce overfitting.

The classifier uses a fully connected layer and a final softmax layer with one output per emotion class.

The model is compiled using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss. The documented training configuration uses a batch size of 64 and a maximum of 30 epochs. Class weights are

calculated from training-set class counts using capped square-root inverse-frequency weighting to give minority classes additional influence without excessively amplifying them.

ModelCheckpoint retains the best validation-accuracy model, EarlyStopping monitors validation loss, and ReduceLROnPlateau reduces the learning rate when validation loss stops improving.

The dataset is organized into training and validation directories, with class labels inferred from directory names. The report identifies FER-2013 for emotion training and UTKFace/Adience as possible datasets for age and gender model training. The age and gender inference code uses pretrained Caffe model files.

IV. IMPLEMENTATION

The application is implemented in Python. TensorFlow/Keras is used to define, train, save, and load the emotion CNN. OpenCV is used for webcam access, face detection, image resizing, colour conversion, and loading the pretrained Caffe age and gender networks. NumPy supports array manipulation and probability processing. The implementation loads the trained emotion model and the age/gender network configuration and weights before entering the video-processing loop.

For each frame, the program detects faces and processes each detected region independently. The emotion model receives the normalized grayscale crop; the age and gender networks receive their prepared colour blob. The predicted labels and confidence are then rendered on the displayed frame. The loop continues until the user exits, with the ESC key documented as the termination control. The same general processing path can be adapted to uploaded images.

The modular design allows the emotion CNN to be retrained without replacing the face detector or demographic-estimation networks. It also supports future changes to the interface or inference models. Any deployment that stores prediction history should minimize retained personal data, define access controls, and make users aware of how facial analysis is used.

V. EVALUATION METHODOLOGY

A defensible evaluation should use a held-out test set that is not used for model fitting or model selection. The test dataset should contain the same seven emotion categories and should be evaluated with the selected final model. The implementation includes code to calculate overall test accuracy, a classification report, a confusion matrix, and per-

class accuracy. Precision, recall, and F1-score should be reported for each class as well as macro- and weighted-average values when available.

For real-time testing, the hardware, camera resolution, lighting conditions, number of faces, and software versions should be recorded. Useful operational measures include average inference time per frame, frames processed per second, and the frequency of missed or incorrect face detections. Results should be repeated under controlled conditions. Age and gender estimates require separate evaluation datasets and appropriate metrics; they should not be inferred from emotion accuracy.

The project report supplied for preparing this manuscript describes the evaluation code and intended metrics but does not include a verified final test-results table. Therefore, exact accuracy, precision, recall, F1-score, latency, and demographic-model performance are not asserted in this paper. These values should be populated from the actual final run before submission.

VI. DISCUSSION AND LIMITATIONS

The primary strength of the system is its integration of face detection, emotion classification, optional demographic estimates, and a unified visual display. The CNN learns image features rather than depending exclusively on handcrafted descriptors. Data augmentation, dropout, class weighting, and validation-based callbacks are intended to improve generalization and training stability. Averaging recent frame predictions can make the displayed label less volatile during webcam use.

Several limitations remain. Facial expressions are not a definitive measurement of a person's internal emotional state, and the output should be interpreted as a visual-expression category rather than a psychological diagnosis. Performance can be affected by lighting, occlusion, camera angle, image quality, and dataset imbalance. Age and gender estimates may also be inaccurate for individuals whose appearance differs from the training data. Evaluation on a held-out test set and on varied real-world conditions is required before making claims about robustness. Privacy-preserving processing, informed consent, bias analysis, and limits on consequential use are important considerations for any deployment.

VII. CONCLUSION AND FUTURE WORK

This paper presented a face emotion recognition system that combines OpenCV face detection with a CNN trained on FER-2013 and integrates the resulting emotion prediction into a live visual interface. The documented model processes

48×48 grayscale face images and classifies seven expressions. Separate pretrained networks provide approximate age-group and gender outputs, and the interface displays predictions with a bounding box and confidence information.

Future work includes publishing complete test metrics, improving class-wise performance, expanding and balancing training data, evaluating alternative CNN or attention-based architectures, testing under varied lighting and pose, and measuring real-time throughput on defined hardware. Further improvements may include more robust face detection, model compression for edge devices, and stronger privacy controls. Quantitative claims should be added only after repeatable experiments have been completed.

Acknowledgment

The authors acknowledge the open-source maintainers of Python, TensorFlow/Keras, OpenCV, and the FER-2013 dataset resources used in the development process. Any institutional acknowledgement required by the department may be added before submission.

REFERENCES

1. I. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov,
2. J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergamo, P. Xie, L. Romaszko, B. Xu, Z. Chrapski, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests," in **Neural Information Processing**, 2013, pp. 117–124.
3. I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergamo, P. Xie, L. Romaszko, B. Xu, Z. Chrapski, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests," in **Neural Information Processing**, 2013. [Online]. Available: <https://arxiv.org/abs/1307.0414>
4. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in **Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 2001, vol. 1, pp. I-511–I-518.
5. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in **Advances in Neural Information Processing Systems**, vol. 25, 2012, pp. 1097–1105.
6. TensorFlow, "Keras documentation." [Online]. Available: <https://keras.io/>

7. OpenCV, “OpenCV documentation.” [Online]. Available: <https://docs.opencv.org/>
8. Kaggle, “FER-2013 dataset.” [Online]. Available: <https://www.kaggle.com/datasets/msambare/fer2013>