Emotion Recognition using Deep Learning Techniques

Jakob Tormalm
HKUST
Clear Water Bay
hjtormalm@connect.ust.hk

Abstract

This project explores facial emotion recognition (FER) using deep learning techniques. A ResNet18 model is trained on the FER2013 dataset to classify facial expressions into seven emotions: angry, disgust, fear, happy, sad, surprise, and neutral. The system is designed for efficient, real-time performance on Apple M1 hardware. Preliminary experiments show the model reaching around 68% accuracy on the test set after 10 epochs, with strong results for the happy and surprise classes. A live webcam demo using OpenCV has also been implemented, capable of detecting faces and predicting emotions in real time. Future work will focus on improving class balance, tuning hyperparameters, and enhancing robustness under different lighting and pose conditions.

1. Abstract

Facial Emotion Recognition (FER) aims to classify human affective states from facial expressions using computer vision and deep learning. This project develops a ResNet18-based FER pipeline trained on the FER2013 dataset and deployed in a real-time webcam system. FER2013's low-resolution grayscale images make the task challenging, especially for subtle and visually overlapping emotions such as *fear*, *sad*, and *angry*. To address these difficulties, the project incorporates preprocessing, augmentation, fine-tuning, ablation studies, and a full deployment pipeline using OpenCV. Experiments show that the model provides competitive accuracy while maintaining real-time inference speed on a MacBook M1. The final system demonstrates how to translate deep-learning methodology into a practical interactive application.

2. Introduction

FER is an established task in computer vision with applications in assistive technology, affective computing, education platforms, safety monitoring, and interactive entertainment. Despite advances in deep learning, FER remains challenging due to occlusions, identity differences, pose variation, cultural diversity in expression, and subtle differences between certain emotion classes. Low-resolution datasets such as FER2013 intensify these difficulties.

This project builds a complete FER system focusing on efficient architectures suitable for real-time settings. It investigates the performance trade-offs of a compact ResNet18 backbone, explores augmentation strategies for noisy data, examines common misclassification patterns, and deploys a functional, responsive webcam system. The emphasis is not only on model accuracy but also on use-case practicality and interpretability.

Suggested Figures:

- High-level system pipeline diagram.
- Example FER2013 images illustrating difficulty and variability.

3. Related Work

Traditional FER solutions employed handcrafted descriptors such as Gabor filters, HOG, or LBP combined with SVM-based classification. These methods struggled with illumination changes and intra-class variability. The transition to deep learning improved robustness due to end-to-end feature learning. CNN-based models such as VGG, Inception, and ResNet became standard. Variants incorporating attention modules, multi-branch fusion, spatial transformers, and facial landmark guidance have improved fine-grained emotion discrimination.

Recent work explores transformer architectures, temporal modeling via LSTMs or 3D CNNs, and cross-domain generalization through large-scale datasets like AffectNet and RAF-DB. However, many high-performing systems rely on heavy architectures unsuitable for real-time inference on consumer hardware. This project focuses on the practical end of the spectrum, emphasizing efficient inference while retaining strong performance.

Suggested Figures:

- Comparison of FER pipelines: handcrafted vs. CNN-based.

- Diagram highlighting typical FER challenges (pose, lighting, occlusion).

4. Data

FER2013 provides 35k labeled images of size 48×48, each assigned to one of seven emotion categories. Images are grayscale, low-resolution, and captured “in the wild,” making the dataset representative but difficult. The training set is imbalanced, with *disgust* significantly underrepresented, affecting stability during training.

Preprocessing includes:

- Normalization to zero mean and unit variance.
- Data augmentation: horizontal flips, small rotations, random crops.
- Optional resizing to 224×224 when using pretrained CNN backbones.

The augmentation is crucial for FER2013 because many images contain noise, poor cropping, or ambiguous expressions.

Suggested Figures:

- Class distribution histogram.
- Visualization grid of augmented samples.
- Examples of noisy/ambiguous labels.

5. Methods

5.1. Model Architecture

The project uses ResNet18, pretrained on ImageNet. Only the final fully connected layer is modified for seven output classes. This architecture balances depth, feature richness, and speed. Earlier layers capture edges and textures, while deeper layers specialize in higher-level emotional cues. Fine-tuning allows the model to adapt to facial-expression patterns despite the low resolution of FER2013.

Suggested Figures:

- Architecture diagram showing residual blocks.
- Comparison table of ResNet18 vs. MobileNet, VGG, EfficientNet in FLOPs/params.

5.2. Training Protocol

The model is trained for 10 epochs using:

- Adam optimizer (lr $= 0.001$)
- Weight decay $= 1 \times 10^{-4}$
- Batch size 32
- MPS acceleration on MacBook M1

A ReduceLROnPlateau scheduler decreases the learning rate when validation accuracy stagnates. Augmentation reduces overfitting and helps the model generalize to real webcam inputs.

5.3. Evaluation Metrics

Evaluation uses accuracy, macro-F1, weighted-F1, and confusion matrices. Macro metrics are emphasized due to class imbalance. Qualitative analyses examine common misclassifications and model attention through heatmaps.

5.4. Deployment Pipeline

The live system uses OpenCV:

1. Capture frames from webcam.
2. Use Haar Cascade to detect face bounding boxes.
3. Crop, resize, normalize, and feed to the ResNet18 model.
4. Display predicted emotion on-screen.

Suggested Figures:

- Screenshot of real-time predictions.
- Visualization of face detection and preprocessing crops.

6. Experiments

6.1. Quantitative Performance

Initial training yields:

- Training accuracy: 78%
- Test accuracy: 68.3%

The confusion matrix shows strong performance on *happy* and *surprise*, while *fear*, *sad*, and *angry* often overlap. The *disgust* class suffers due to small sample size.

Suggested Figures:

- Large annotated confusion matrix.
- Precision/recall/F1 bar plots per class.

6.2. Ablation Studies

Three ablations were performed:

(1) With vs. without augmentation: Removing augmentation reduces test accuracy by 3–4

(2) Haar Cascade vs. DNN face detector: The DNN detector improves cropping quality but reduces real-time FPS. Haar Cascade remains optimal for interactive use.

(3) Input resolution: Feeding 224×224 resized images improves accuracy slightly but slows down inference.

Suggested Figures:

- Accuracy curves comparing augmented vs. non-augmented training.
- Side-by-side face detection examples for Haar vs. DNN.

6.3. Failure Case Analysis

Misclassifications often arise from:

- Occlusions (hands, hair, masks)
- Side-profile faces
- Dim lighting
- Ambiguous expressions

Grad-CAM reveals that the model often fixates on the mouth region, which may explain confusion between similar classes when mouth shape lacks detail.

Suggested Figures:

- Misclassified examples with predicted vs. true labels.
- Grad-CAM heatmaps.

7. Conclusion

This project demonstrates that a compact ResNet18 architecture can achieve respectable FER performance on FER2013 while supporting real-time deployment on consumer hardware. The model handles expressive emotions well but struggles with subtle or low-resolution categories. Improvements could include more robust face detectors, larger and cleaner datasets, attention mechanisms, or transformer-based architectures. Temporal modeling using video sequences could further enhance performance in realistic applications.

Suggested Figures:

- Final system overview diagram summarizing the training and deployment pipeline.