# A CNN-BASED APPROACH TO EMOTION RECOGNITION USING FER2013

Mariia Kalianova, Danylo Tovpyshko

La Sapienza University in Rome

## Abstract

This paper presents a convolutional neural network (CNN)-based method for facial emotion recognition using the FER2013 dataset. The goal is to classify facial expressions into one of seven basic emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The network is trained and tested using PyTorch, and deployed via a graphical interface built with Gradio and OpenCV. This research demonstrates the feasibility of deploying lightweight models for real-time emotion recognition in everyday applications.

## Index Terms

Emotion recognition, CNN, FER2013, PyTorch, facial expressions, Gradio, OpenCV.

## I. INTRODUCTION

Facial emotion recognition has gained attention in human-computer interaction and mental health assessment. Recognizing emotions from facial cues in real-time can enable adaptive interfaces, monitoring tools, and educational systems. In this work, we focus on creating a practical emotion recognition system using a self-built CNN trained on FER2013, a widely used public dataset for facial expressions.

## II. RELATED WORK

Prior studies employed deep learning models, especially CNNs, due to their strong performance in image-based tasks. Some works leveraged complex architectures like VGG or ResNet, while others opted for lightweight models suitable for real-time use. Our approach emphasizes simplicity and interpretability, targeting deployability rather than marginal accuracy gains.

## III. METHODOLOGY

## A. Dataset

The FER2013 dataset includes 35,887 grayscale 48x48 facial images categorized into seven emotion classes. Data is split into training, public test, and private test sets.

Dataset is preprocessed using certain transformations to reduce overfitting. Such techniques were used:

1. Random horizontal flip (so the model can identify both mirrored pictures and normal ones)
2. Random rotation (for robustness to small image rotation)
3. Tensor conversion (turns PIL image into tensor)
4. Normalization (scaling pictures)

## B. Model Architecture

We implemented a custom CNN named EmotionRecognitionCNN using PyTorch. It consists of multiple convolutional layers with ReLU activations, batch normalization, dropout, and max pooling, followed by fully connected layers for classification.

## C. Training

The model was trained using the cross-entropy loss and Adam optimizer. Number of training loops = 10 – optimal for training without risk of overfitting.

## IV. RESULTS

After training, the model achieved competitive accuracy 58.86% on the FER2013 test set, which is good result for this dataset. The final model was saved in emotion\_model3\_58.pth, demonstrating robust generalization and correct predictions on most emotions, with occasional confusion between sadness and neutral.

## V. DEPLOYMENT AND GUI

A real-time graphical user interface was developed using Gradio. It allows to open the web-site, upload videos, photos for recognition. The system is lightweight and suitable for educational or demonstration purposes. Also we implemented function that allows to detect emotion at real-time using web-camera.

## VI. CONCLUSION AND FUTURE WORK

This work shows that even relatively simple CNN architectures can achieve reliable facial emotion recognition when paired with proper data processing and deployment strategies. Future work will explore multi-modal input (e.g., audio), improved model interpretability, and more emotionally nuanced datasets.

## VII. CHALLENGES AND LIMITATIONS

## During the development of the emotion recognition system, several challenges were encountered:

## 1. Class Imbalance in FER2013:

## The FER2013 dataset is imbalanced, with emotions like “happy” and “neutral” being overrepresented compared to “disgust” or “fear”. This made it difficult for the model to generalize well across all categories, often leading to misclassifications of minority emotions.

## 2. Low-Resolution Grayscale Images:

## FER2013 images are only 48x48 pixels and in grayscale, limiting the amount of fine-grained facial detail the model can learn from. This constrained the depth and complexity of the CNN architecture that could be effectively utilized.

## 3. Real-time Constraints:

## Building a GUI capable of performing real-time inference on live webcam input required balancing model size and speed. More accurate models were often too slow for real-time use without GPU acceleration.

## 4. Emotion Ambiguity:

## Some emotions are inherently difficult to distinguish even for humans — for example, “sad” versus “neutral”. These ambiguities reflect in model performance, especially when evaluating on real-world faces not in the training set.

## 5. Limited Hardware Resources:

## Training and testing were conducted on a consumer-grade machine without a dedicated GPU. This slowed down experimentation and made hyperparameter tuning more limited.

## 6. Face Detection Failures:

## The GUI relies on OpenCV’s Haar Cascade for face detection, which can fail in poor lighting conditions or when the face is turned at an angle. This affected prediction reliability during live demos.

7. Failure of gradio webcam: