Facial Emotion Recognition with FER-2013 and EDFER

Nour Hassan 8357
Yomna Gharib 8175
Basem Hesham 8000

September 2025

Abstract

This report documents a Facial Emotion Recognition (FER) project using FER-2013 and an extended EDFER dataset. It preserves the original structure and figures from the submitted Word document and summarizes dataset preparation, preprocessing, model architecture, training strategy, and results. Replace or expand paragraphs below with the exact text from your source as needed.

1 Introduction

Facial Emotion Recognition (FER) aims to automatically identify human emotions from face images. This project uses FER-2013 as the primary dataset and complements it with EDFER (extended FER) versions. The goals include data cleaning, label normalization, augmentation, building a robust preprocessing pipeline, and evaluating on validation and test sets.

2 Dataset Preparation

We found that FER-2013 contained many mislabeled samples and that some classes were under-represented. Therefore we added the EDFER dataset.

- **EDFER:** used entirely for training after label normalization.
- **FER-2013:** used for both training/validation splitting and for the final test set.

2.1 Split policy

- **TRAIN:** all EDFER + 75% of FER-2013 training (per-class, deterministic shuffle).
- **VAL:** remaining 25% of FER-2013 training.
- **TEST:** FER-2013 test split (PrivateTest).

2.2 Final counts

```
=== Summary ===
Output: /kaggle/content/fer2013_by_usage_1
Totals -> train: 50240  val: 7178  test: 7178

Per-class (train/val/test):
     angry   6991 /   999 /   958
   disgust    763 /   109 /   111
      fear   7170 /  1024 /  1024
     happy  12626 /  1804 /  1774
   neutral   8689 /  1241 /  1233
       sad   8452 /  1208 /  1247
  surprise   5549 /   793 /   831
```

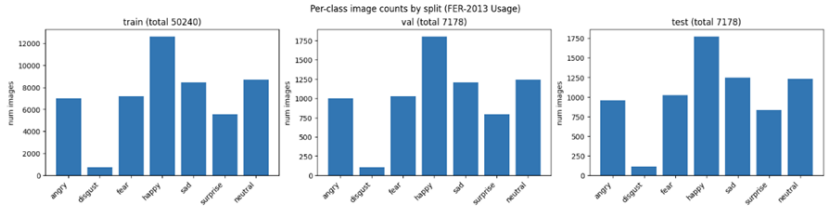

Figure 1: Training / validation/test split summary .

2.3 Label normalization

EDFER labels were mapped to the FER-2013 canonical labels: `disgusted` → `disgust`, `fearful` → `fear`, `surprised` → `surprise`. Other labels were lowercased and canonicalized.

3 Class rebalancing

3.1 Targeted offline augmentation (Albumentations)

Goal: push most classes toward a 7,000 target; set the `disgust` target to 1,200 (class initially had 763 training samples).

- **Marked for augmentation:** angry (+9), disgust (+437), surprise (+1,451).
- **Transform stack (probabilistic, seed-controlled):**
  - Resize to 48 (short side), HorizontalFlip
  - Affine: translate/scale ±0.03, rotate ±6°, mild shear
  - Local/global contrast: CLAHE / RandomBrightnessContrast
  - RandomGamma, subtle color jitter (HSV / RGB shift)
  - Sharpening (UnsharpMask / Sharpen), light brightness/contrast

3.2 Undersampling

A separate undersampled training view was created with a maximum of 9,000 samples per class, named `train_undersampled`, to avoid over-dominance of the `happy` class.

```
Found 48511 files belonging to 7 classes.
Found 7178 files belonging to 7 classes.
Classes: ['angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise']
Num classes: 7
Train counts per class : [7000 1200 7170 9000 8689 8452 7000]
```

Figure 2: Example of undersampling.

4 Preprocessing & input pipeline

We used ResNet-50 as the backbone, so the data was preprocessed accordingly.

- **Framework:** TensorFlow / Keras.
- **Image size:** 224×224, RGB (ResNet preprocessing).
- **Online augmentation (training only):** RandomFlip, RandomRotation (± 0.08), RandomTranslation (0.06, 0.06), RandomZoom (± 0.04), RandomContrast(0.06).
- **Batch size:** 64; use `AUTOTUNE` prefetching.
- **Reproducibility:** Python / NumPy / TF seeds set.

5 Model architecture

- **Backbone:** ResNet-50 (ImageNet weights, `include_top=False`, pooling='avg').
- **Head:** Dense(32, ReLU) $\rightarrow$ BatchNorm $\rightarrow$ Dropout(0.4) $\rightarrow$ Dense(7, Softmax) with ℓ_2 regularization $= 5 \times 10^{-4}$ on the final dense.
- **BatchNorm:** BN layers were kept non-trainable during fine-tuning to stabilize feature statistics.

6 Training strategy

6.1 Loss, optimizer, weights

- **Loss:** `SparseCategoricalCrossentropy`.
- **Class weights:** inverse-frequency weights (normalized from disk counts). Auto-computed unless a weights file is provided.
- **Training phases:**
  1. **Phase 1 — frozen backbone (10 epochs).**
     - Optimizer: Adam, learning rate 5×10^{-4}.
     - Train only the classification head for quick alignment.
  2. **Phase 2 — fine-tuning (up to epoch 30).**
     - Unfreeze the last 50 layers of ResNet-50 (keep batch-normalization layers frozen).
     - Optimizer: AdamW (when available) with `weight_decay` $= 1 \times 10^{-4}$; starting learning rate 8×10^{-5}.
     - LR scheduler: `ReduceLROnPlateau` (factor $= 0.5$, patience $= 2$, `min_lr` $= 1 \times 10^{-7}$).

- Early stopping on validation loss (patience = 6, restore best weights).
- Optional callback: when the learning rate drops, gently increase weight decay (cap at 1×10^{-3}) to improve generalization.

6.2 Metrics (macro, robust to imbalance)

We report:

- Accuracy
- Macro precision / Macro recall / Macro F1

7 Results

7.1 Validation

- Val Accuracy: 0.8859
- Val Loss: 0.4294
- Macro metrics improved steadily with fine-tuning; best epoch restored by EarlyStopping.

7.2 Test (strict held-out FER-2013 test)

- Test Accuracy (raw): 0.9439 on 7,178 images.
- Model.evaluate: loss = 0.1831, accuracy = 0.9439.

Per-class (precision / recall / F1) highlights from the classification report:

- **happy:** strong (F1 $\approx$ 0.967)
- **surprise:** strong (F1 $\approx$ 0.970)
- **neutral:** high recall ($\approx$ 0.961), balanced F1 ($\approx$ 0.936)
- **fear:** comparatively lower recall ($\approx$ 0.900), F1 ($\approx$ 0.917) — the most confusable class
- **disgust:** excellent (F1 $\approx$ 0.996) despite lower sample size thanks to targeted augmentation

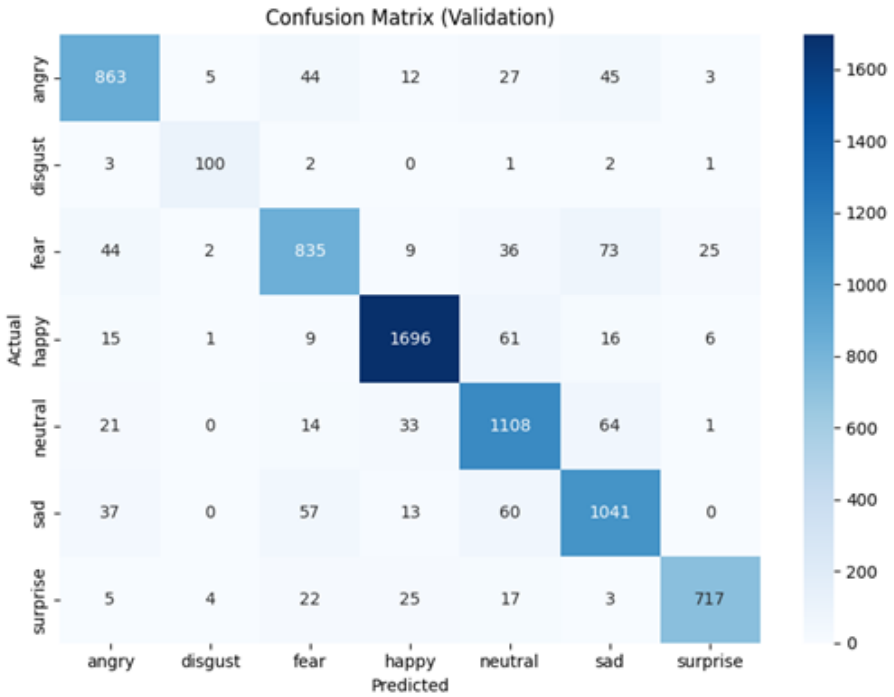

Figure 3: Validation metrics or confusion matrix .

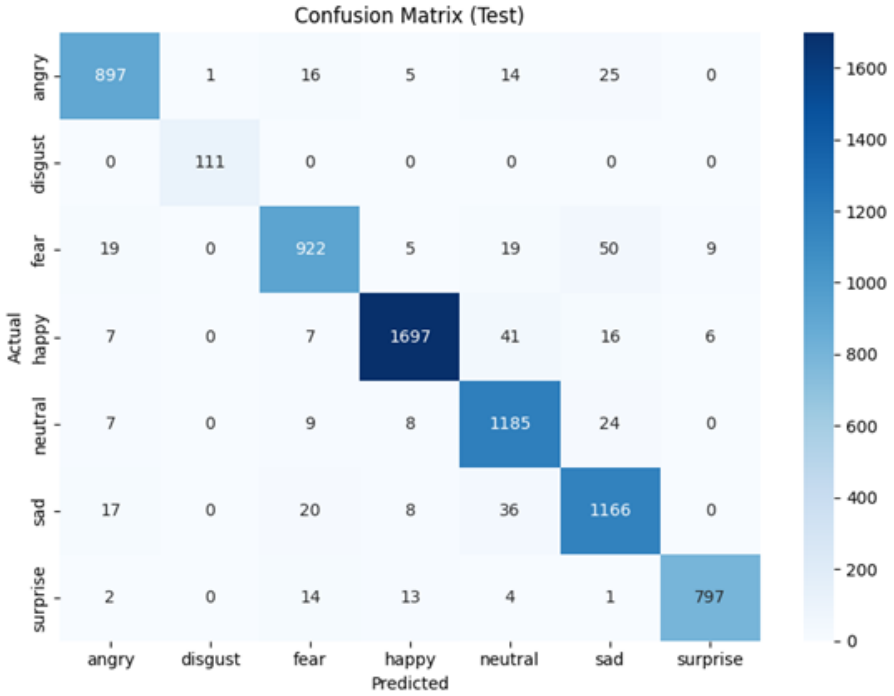

Figure 4: Test metrics .

8 How we progressed from last week

In week 3 we used only FER-2013 and trained for 20 epochs (10 frozen, 10 unfrozen). The best accuracy then was 67.08%. We reduced the model head size and doubled dropout to reduce overfitting, added EDFER, and increased training to 30 epochs (10 frozen + 20 unfrozen), which improved final performance.

Classification report:

	precision	recall	f1-score	support
angry	0.9452	0.9363	0.9407	958
disgust	0.9911	1.0000	0.9955	111
fear	0.9332	0.9004	0.9165	1024
happy	0.9775	0.9566	0.9670	1774
neutral	0.9122	0.9611	0.9360	1233
sad	0.9095	0.9350	0.9221	1247
surprise	0.9815	0.9591	0.9702	831
accuracy			0.9439	7178
macro avg	0.9500	0.9498	0.9497	7178
weighted avg	0.9445	0.9439	0.9440	7178

Figure 5: classification report.

9 Sources

- FER-2013 dataset (source).
- EDFER (extended dataset) — internal / extended dataset notes.