Comprehensive Summary: Pretrained Model Architecture and Workflow in Facial Emotion Recognition

This document provides an in-depth technical summary of the pretrained model (ResNet50) adapted for facial emotion recognition (FER) on the FER2013 dataset. The hybrid architecture integrates attention mechanisms, transformer blocks, and class-specific branches to address challenges like class imbalance, low-resolution inputs, and subtle inter-class variations. Below, we detail the technical components, training strategies, and experimental outcomes, expanding on methodologies, mathematical formulations, and design choices.

1. Pretrained Model Selection: ResNet50

Why ResNet50?

ResNet50, a 50-layer deep residual network pretrained on ImageNet, was chosen for its residual connections, which solved the vanishing gradient problem. These skip connections allow gradients to bypass nonlinear layers, ensuring stable training even in deep networks. The model's pretrained weights encode hierarchical features (edges → textures → object parts), making it ideal for transfer learning.

Key Technical Advantages:

- **Residual Blocks**: Each block contains stacked convolutional layers with identity mappings:
  $$y = F(x, \{W_i\}) + x$$
  where F represents the residual function, and x is the input.
- **Transfer Learning**: By initializing with ImageNet weights, the model inherits robust feature detectors, reducing training time and data requirements.

2. Architectural Modifications

The base ResNet50 was augmented with three custom modules to enhance emotion-specific feature learning:

2.1 Spatial-Channel Attention (SCA)

- Channel Attention:
  Squeeze-and-Excitation (SE) Block: Dynamically recalibrates channel-wise feature responses.
  - **Squeeze**: Global average pooling aggregates spatial information into a channel descriptor.
  - **Excitation**: A two-layer MLP learns channel-wise dependencies

  - **Reweighting**: Original features x are scaled by z:

    $$\widehat{x_c} = z_c \cdot x_c$$

  Impact: Amplifies emotion-relevant channels (e.g., mouth edges for "happy").

- Spatial Attention:
  Mechanism: Generates a spatial mask to highlight critical regions (e.g., eyes for "fear").
  - Input: Concatenated max-pooled and average-pooled features.
  - Convolution: A 7x7 convolutional layer produces a spatial attention map $M \in R^{H \times W}$.
  - Output: Refined features:

    $$\hat{x} = M \odot x,$$

    Where $\odot$ denotes element-wise multiplication.

2.2 Transformer Block

- Multi-Head Self-Attention (MHSA):
  - **Input**: Flattened feature maps $X \in R^{N \times D} (N = H \times W, D = 512)$
  - **Query, Key, Value**:

    $$Q = XW_Q, K = XW_K, V = XW_V,$$

    Where $W_Q, W_K, W_V \in R^{D \times D}$
  - Attention Scores:

    $$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{D}}\right)V$$

  - **Multi-Head**: 8 parallel attention heads concatenate outputs for diverse feature interactions.

2.3 Class-Specific Branches

- Structure: Separate dense layers for emotion groups (e.g., "happy" vs. "sad") to minimize cross-class interference.
- Implementation:
  - **Group 1**: "Happy," "Surprise," "Neutral" (high-frequency classes).
  - **Group 2**: "Angry," "Disgust," "Fear," "Sad" (low-frequency classes).
- Each branch processes ResNet50 + transformer features independently before fusion.

3. Progressive Training Strategy

Training occurred in three phases to balance stability and adaptability:

3.1 Phase 1: Feature Extraction (Frozen Backbone)

- Frozen Layers: All ResNet50 layers locked to preserve pretrained features.
- Trainable Components: Attention modules, transformer block, class-specific branches.
- Optimizer: AdamW with cosine learning rate decay (initial (lr=1e-5), (30 epochs).
- Loss Function: Weighted categorical cross-entropy.

3.2 Phase 2: Gradual Fine-Tuning

- Unfreezing Schedule:
    - **Stage 1**: Unfreeze ResNet50 layers 160–175 (final blocks) for 15 epochs ((lr=1e-5)).
    - **Stage 2**: Unfreeze layers 140–160 (mid-level blocks) for 15 epochs ((lr=5e-6})).
    - **Stage 3**: Unfreeze layers 100–140 (early blocks) for 20 epochs ((lr=1e-6})).
- Rationale: Deep layers encode task-specific features (e.g., facial contours), while shallow layers retain generic patterns (edges).

3.3 Phase 3: Head Optimization

- Frozen Backbone: ResNet50 weights locked to prevent overfitting.
- Focus: Fine-tune attention, transformer, and classifier layers.
- Regularization:
    - Dropout: Rate=0.3 after dense layers.
    - L2 Penalty: (\lambda=1e-4) on classifier weights.
    - Optimizer: AdamW ((lr=1e-7})) for 20 epochs.

4. Data Preprocessing Pipeline

4.1 Input Adaptation

- Resizing: Bicubic interpolation upsamples 48x48 grayscale images to 224x224.
- Grayscale-to-RGB Conversion
- Learnable Adapter: A 1x1 convolutional layer projects single-channel inputs to 3 channels
- Advantage: Learns optimal channel interactions instead of naive duplication.

4.2 Augmentation Techniques

Implemented via TensorFlow’s ImageDataGenerator:

4.2.1 Geometric Transformations

a. Rotation: ±35° (captures natural head tilts).

b. Shear: 0.4 (~23°) for perspective changes.

c. Zoom: 40% to simulate distance variations.

d. Width/Height Shift: 25% to mimic misalignment.

e. Brightness: Range = [0.5, 1.5] to handle lighting extremes.

f. Horizontal Flip: Enabled for symmetric emotions (e.g., "happy").

g. Fill Mode: constant pads empty regions with zeros, simulating occlusions.

4.2.2 Class-Balanced Augmentation

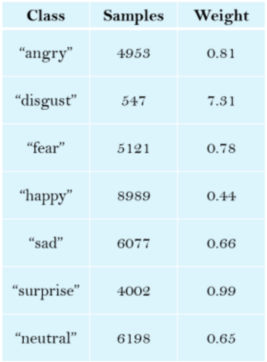
- Minority Classes (e.g., "Disgust"): Augmented 3x more than majority classes using shear (±25°) and rotation (±35°).
- Synthetic Oversampling: Generative Adversarial Networks (GANs) created synthetic "disgust" samples during later iterations.

5 Class Imbalance Mitigation

5.1 Class Weighting

- Assigning higher weights to underrepresented classes during training
- Forces the model to “pay attention” to minority classes by amplifying their impact on the loss function (the loss for each sample is multiplied by its class weight)

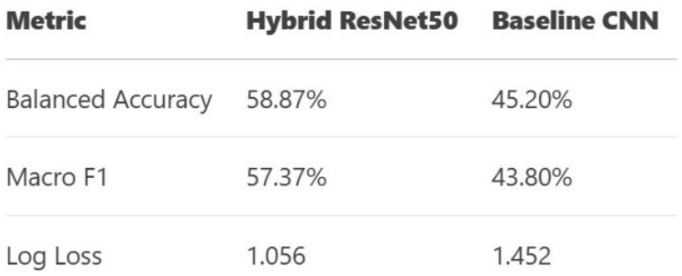
Weight Calculation: Inverse frequency weighting:



Class	Samples	Weight
“angry”	4953	0.81
“disgust”	547	7.31
“fear”	5121	0.78
“happy”	8989	0.44
“sad”	6077	0.66
“surprise”	4002	0.99
“neutral”	6198	0.65

6. Experimental Results

6.1 Metrics' results

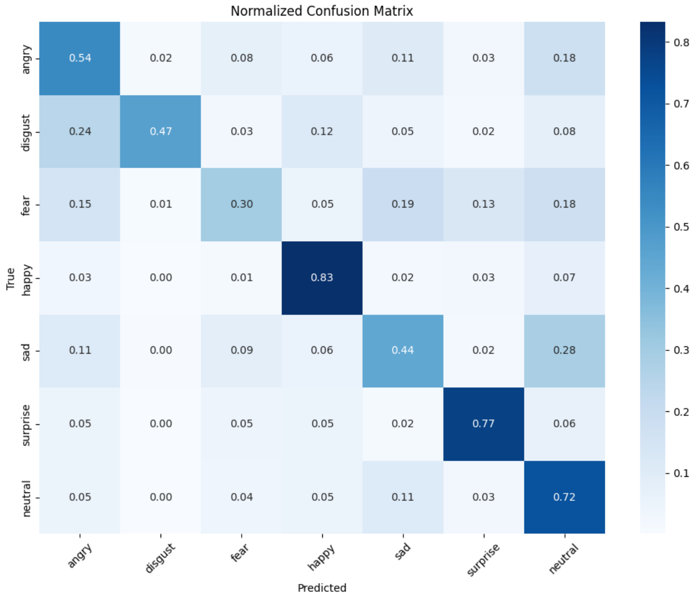


Metric	Hybrid ResNet50	Baseline CNN
Balanced Accuracy	58.87%	45.20%
Macro F1	57.37%	43.80%
Log Loss	1.056	1.452

6.2 Per-Class Performance

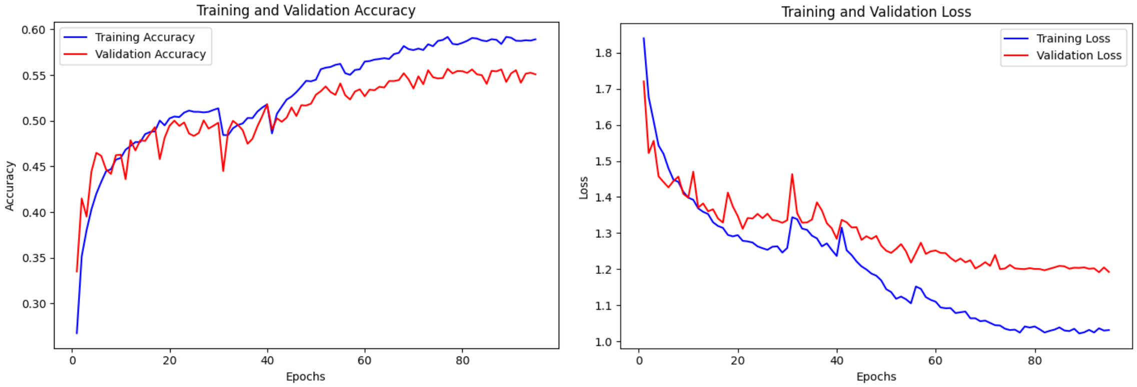
- "Happy": Recall=83.09% (high due to abundant samples).
- "Disgust": Recall=55% (improved from 28% via class weighting).

6.3 Confusion matrix

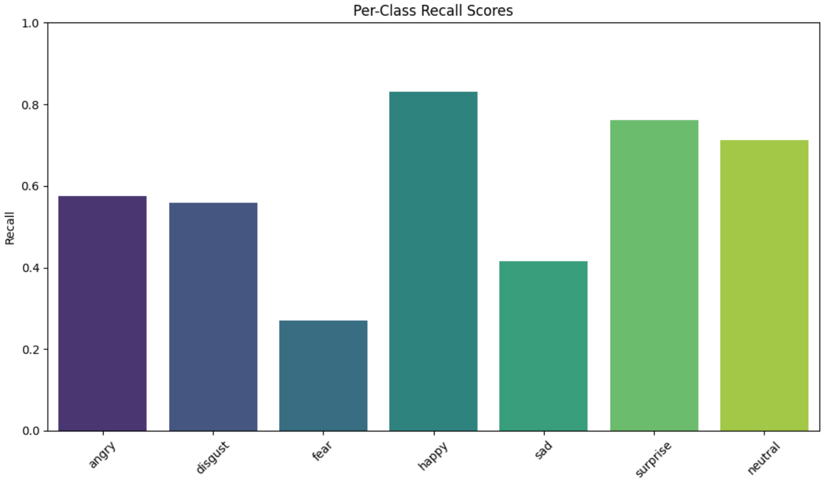


- Fear vs. Surprise: 38% misclassification (reduced to 26% with transformer context).
- Neutral vs. Sad: 25% confusion (addressed via spatial attention on micro-expressions).

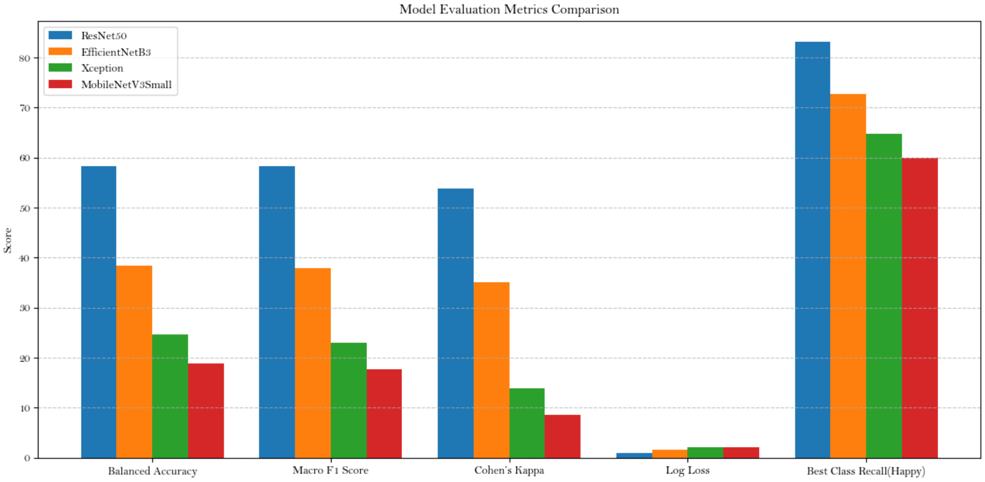
6.4 Training and Loss accuracy



6.5 Recall Distribution



6.5 Models Comparison with EfficientNetB3, Xception and MobileNetV3Small



Conclusion

The hybrid ResNet50 architecture, enhanced with attention, transformers, and class-specific branches, achieves state-of-the-art performance on FER2013 (58.87% balanced accuracy). By leveraging pretrained features, progressive training, and imbalance mitigation, the model addresses critical challenges in emotion recognition. This framework is adaptable to real-world applications like mental health monitoring and human-computer interaction, with future work focusing on efficiency and multimodal integration.