



An Efficient Vision Transformer Architecture for Real-Time Facial Expression Recognition

Prepared By

Name: Ammar Mohammad Wassas

ID No.: 253002052

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

College of Informatics

Midocean University

UAE

Supervisor Name: Dr. Mohamed Essa

Collage of Informatics, Midocean University, Comoros

Email: 253002052@st.midocean.edu.km

Chapter 3 Proposed Approach (Methodology)

3.1 Introduction:

This proposed methodology utilizes a hybrid multi-model pipeline aimed at realizing state-of-the-art performance of Facial Expression Recognition from the FER2013 dataset. As depicted in Figure 4.1 below, the overall pipeline consists of face detection by YOLOv8, deep learning feature extraction from a dual-backbone combination of EfficientNet-B7 and GhostNet, efficient feature refinement by guided attention through a Vision Transformer model, and finally classification through a fully connected head trained with a strong loss formulation. The hybrid pipeline is designed based on the complementary characteristics of each component, which include face detection from YOLOv8, high capacity compound scaling through EfficientNet-B7, computational efficiency through redundancy suppression in GhostNet, and modeling of global spatial dependencies through the Vision Transformer. The main motivation for incorporating these four models into a hybrid pipeline is to address the key drawbacks associated with Facial Expression Recognition systems highlighted above, aiming to achieve a target recognition rate of 90% on the FER2013 dataset.

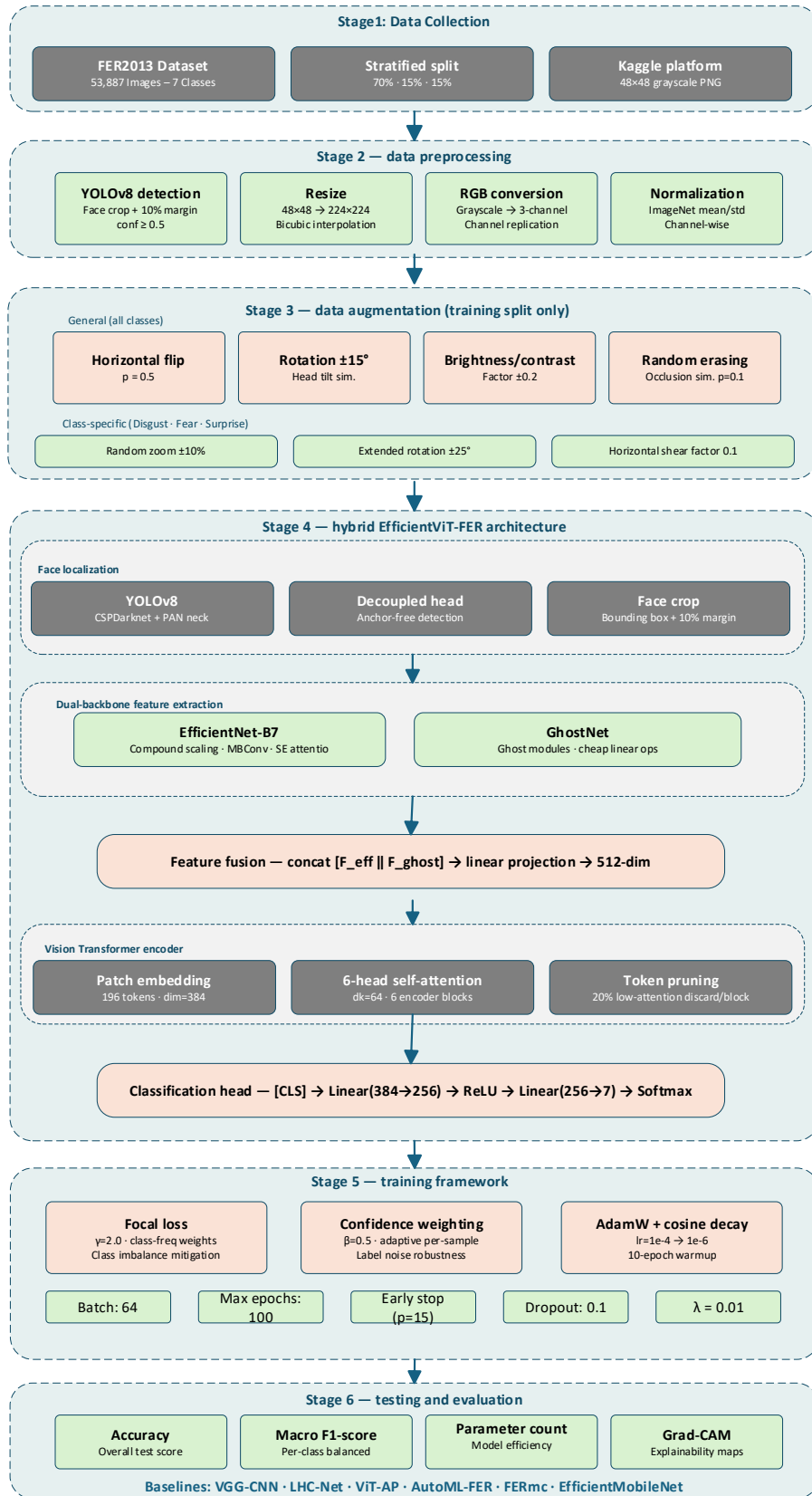


Fig 3.1 — EfficientViT-FER proposed hybrid framework pipeline

3.2 Dataset:

The FER2013 dataset [10] is a widely used in-the-wild FER dataset proposed by Goodfellow et al. at the ICML 2013 Representation Learning Workshop. This dataset consists of 35,887 grayscale facial images with 48×48 pixel resolution labeled into seven emotion classes, namely, Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The FER2013 dataset was created through an automated procedure where queries about emotion were sent to the Google Image Search API, and then facial image detection and alignment were conducted on these returned images.

As compared to the original 70% training and 15%/15% public test/private test train/test split ratio of the FER2013 dataset, in this work, a stratified 70%/15%/15% train/validation/test ratio is used for all seven classes. By stratification, we make sure that the distribution among classes, such as the highly unbalanced 16:1 ratio between the Happy majority class and the Disgust minority class, is preserved proportionally for all three train/test/validation splits, hence ensuring no distribution shift among the two data sets. As a result, there are approximately 25,121 images in the training set, 5,383 images in the validation set, and 5,383 images in the test set. The class distribution is summarized in Table 4.1 below.

TABLE 4.1. FER2013 STRATIFIED DATASET SPLIT (70% – 15% – 15%)

Emotion	Total	Train (70%)	Validation (15%)	Test (15%)	% of Dataset
Angry	4,953	3,467	743	743	13.8%
Disgust	547	383	82	82	1.5%
Fear	5,121	3,585	768	768	14.3%
Happy	8,989	6,292	1,348	1,349	25.1%
Sad	6,077	4,254	912	911	16.9%
Surprise	4,002	2,801	601	600	11.2%
Neutral	6,198	4,339	929	930	17.3%
Total	35,887	25,121	5,383	5,383	100%

3.3 Preprocessing

The preprocessing pipeline is implemented to bring all FER2013 images into a format ready for multi-model hybrid architectures without losing as much expression-discriminative facial information as possible. Preprocessing is performed uniformly across train, validation, and test splits, while data augmentation is only applied to the train split. Preprocessing comprises four sequential steps: detection and cropping of faces using YOLOv8, image standardization, conversion to RGB, and pixel normalization.

3.3.1 Face Detection and Cropping

In particular, raw FER2013 images are initially passed through YOLOv8 [53], which is an advanced framework for real-time object detection, running under the configuration of face detection task, to detect and crop faces of interest from input images. Although FER2013 images have been created with face alignment performed during dataset creation, some residual background areas, inconsistencies in the image framing, and misalignment are present in a subset of images. Using YOLOv8 detector removes these inconsistencies by redetected and recropping of the bounding boxes on faces from the images.

YOLOv8 detector uses CSPDarknet as backbone and PAN as neck for detection heads. In this case, the detection function can be described as:

$$\hat{B} = f_{YOLOv8}(I; \theta_{det})$$

where I represents the original image, θ_{det} represents the pre-trained YOLOv8 face detection weights, and the detected bounding box $\hat{B} = \{x1, y1, x2, y2, conf\}$ corresponds to the predicted face region with its corresponding confidence score. In case of images where no faces are detected confidently with a confidence score of greater than 0.5, such images are kept without any changes in order to not discard them from the training set. The detected bounding boxes are extended by 10% on each side in order to cover all peripheral face regions.

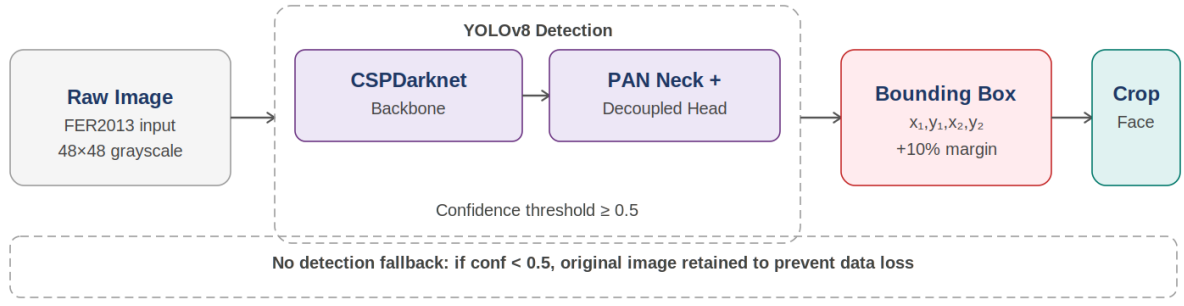


Fig 4.2 — YOLOv8 face detection and cropping pipeline

Fig. 3.2. YOLOv8 face detection and cropping pipeline applied to FER2013 images, showing CSPDarknet backbone, PAN neck, decoupled head, bounding box prediction with 10% margin expansion, and no-detection fallback.

3.3.2 Image Standardization

After detecting faces and crop, all images will be uniformly resized into 224 x 224 dimensions through the method of bicubic interpolation, an image scaling algorithm that retains detailed texture information of the face. In bicubic interpolation, each pixel output value is calculated from the summation of the 16 adjacent input pixels:

$$p_{out}(x, y) = \sum_{i=-1}^2 \sum_{j=-1}^2 p_{in}(x+i, y+j) \cdot w(i) \cdot w(j) \quad (2)$$

where $w(\cdot)$ is the bicubic kernel weight function. Bicubic interpolation is better than bilinear or nearest neighbor interpolation because it provides more smooth upsampling images with better edge and texture representation of the face.

3.3.3 Grayscale-to-RGB Conversion

Since the images for FER2013 are available in the form of grayscale images, and all of the backbones are pre-trained using ImageNet which needs images with three channels, the grayscale images are transformed into a three-channel form:

$$I_RGB = [I_gray, I_gray, I_gray]$$

This conversion maintains complete compatibility with ImageNet pretrained backbone weight parameters without necessitating the need for a separate pretraining phase on single-channel models.

3.3.4 Pixel Normalization

The pixel intensity values are then normalized based on the ImageNet distribution mean and standard deviations in order to ensure that the FER2013 image distribution matches that of the pretrained models:

$$x_norm = (x - \mu) / \sigma$$

With $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$, which denote the channel-wise mean and variance of the ImageNet datasets.

3.4 Data Augmentation

The data augmentation technique is only utilized on the training subset of the FER2013 dataset in order to enhance the ability of our model to generalize, as well as reduce the huge imbalance problem of FER2013. Our data augmentation process consists of two levels: the general augmentation level, where augmentation techniques are applied equally on all training images; and the class augmentation level, where augmentation techniques are applied frequently to minority classes.

3.4.1 General Augmentation Layer

All training images will be subjected to the following operations with the respective probability values at each training epoch:

Random image flip: With probability $P(\text{flip}) = 0.5$, to account for the bilateral symmetry in facial expressions.

Random rotation: Rotation will be performed within the limits defined by the angle $\theta \in [-15^\circ, +15^\circ]$ for a simulated head tilt:

$$x' = x \cos \theta - y \sin \theta, \quad y' = x \sin \theta + y \cos \theta$$

Random brightness and contrast changes: With scaling and offset within the factor α, β :

$$x' = \alpha \cdot x + \beta, \quad \alpha \in [0.8, 1.2], \quad \beta \in [-0.2, 0.2]$$

Random erasing: With probability $P(\text{erase}) = 0.1$, erasing an image portion with zeros.

3.4.2 Class-Specific Augmentation Layer

More augmentations are then performed with increased probability on training examples belonging to the Disgust, Fear, and Surprise classes, thereby decreasing the effective class imbalance ratio:

Random zoom $\pm 10\%$: Random scale changes to simulate variations in face-camera distance.

Extended rotation $\pm 25^\circ$: An extra rotation, offering more geometrical variety than in the general case of $\pm 15^\circ$.

Horizontal shear $s=0.1$: To simulate changes in perspective of the face:

$$x' = x + s \cdot y, \quad y' = y$$

Fig. 3.3. Two-layer data augmentation strategy: general augmentation applied to all training classes (top layer), and class-specific augmentation applied exclusively to minority emotion categories Disgust, Fear, and Surprise (bottom layer).

3.5 Model Architecture

The suggested architecture design comprises four main components arranged sequentially, namely: face detection and localization through the use of YOLOv8, deep feature extraction from dual backbones (EfficientNet-B7 and

GhostNet), refinement of features with global attention by an efficient Vision Transformer encoder, and a classification layer for generating the output emotions. In accordance with Figure 4.4, the architecture design is created to ensure that every part brings a different ability: high-level local feature extraction, efficient ghost feature extraction, attention modeling, and multi-classification.

Fig. 3.4. Detailed EfficientViT-FER hybrid architecture: YOLOv8 face localization → EfficientNet-B7 and GhostNet dual-backbone feature extraction → feature concatenation and projection → Vision Transformer encoder with token pruning → classification head → predicted emotion label.

3.5.1 YOLOv8 Face Detection

The use of YOLOv8 [53] has two purposes within the proposed system: first, it acts as a pre-processing tool for face detection and image cropping during the training process; and second, it acts as an online face localization mechanism when processing images, which allows the entire system to function without the need for any pre-cropping of the input image. The YOLOv8 uses anchor-free detection and is backed by the CSPDarknet model and a path aggregation network.

3.5.2 EfficientNet-B7 Backbone

EfficientNet-B7 [35] acts as the backbone for efficient feature extraction, producing rich facial features via compound scaling of network depth, width, and input image resolution, where:

$$d = \alpha^{\phi}, \quad w = \beta^{\phi}, \quad r = \gamma^{\phi}$$

with the following condition to be satisfied:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \quad \text{with } \alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

where d , w , and r represent the scaling factors of the network depth, width, and resolution respectively, with ϕ being the compound scaling factor having the value 7 for the EfficientNet-B7. The network architecture consists of Mobile Inverted Bottleneck Convolutional Blocks, each having channel-wise squeeze-and-excitation mechanisms:

$$F_{se} = \sigma(W_2 \cdot \delta(W_1 \cdot GAP(F)))$$

where GAP stands for global average pooling, σ represents the sigmoid activation function, δ represents the ReLU activation function, and W_1 and W_2 are the squeeze and excitation weights matrices, respectively. This model is initialized with pre-trained ImageNet weights and fine-tuned using the end-to-end training approach. The original model classification head is modified to a feature projection head outputting a 512D facial feature vector.

Fig. 3.5. EfficientNet-B7 compound scaling of depth, width, and resolution with constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ (left panel), and Mobile Inverted Bottleneck Convolution block with squeeze-and-excitation channel attention (right panel).

3.5.3 GhostNet Backbone

On the other hand, GhostNet [52] acts as our secondary feature extraction backbone, providing a more computationally efficient complementary representation of faces using the architecture of the Ghost modules. Using ghost modules, all of the feature maps can be generated by first generating a limited number of intrinsic feature maps by standard convolution and the rest of the feature maps by cheap linear transformations:

$$Y' = \text{Conv}(X; W_0)$$

$$Y_i = \Phi_i(y'_s), \quad \forall i = 1, \dots, s \cdot n'$$

Where Y' refers to the set of intrinsic feature maps generated by a standard convolution layer with weight W_0 , Φ_i refers to the inexpensive depth-wise linear operation acting on an intrinsic feature map y'_s while s refers to the ratio that determines the number of ghost maps in relation to the intrinsic feature maps.

This design results in a reduction in the computation cost of FLOPs by a factor of s compared to a convolutional layer with equivalent output dimensions.

3.5.4 Dual-Backbone Feature Fusion

The resulting 512 dimensional feature vectors from EfficientNet-B7 and GhostNet backbone architectures are merged using concatenation followed by a linear projection that results in a combined 512 dimensional feature vector:

$$F_{fused} = W_{proj} \cdot [F_{eff} \parallel F_{ghost}] + b_{proj}$$

where $F_{eff} \in \mathbb{R}^{512}$ and $F_{ghost} \in \mathbb{R}^{512}$ represent the feature vectors from EfficientNet-B7 and GhostNet respectively; $\parallel$ represents concatenation operation, while $W_{proj} \in \mathbb{R}^{512 \times 1024}$ is the learned weight matrix for projection. BatchNorm and ReLU activations are performed after projection.

3.5.5 Vision Transformer Encoder

The fused 512-Dimensional feature vector is converted into a series of patch tokens which undergo encoding through a vision transformer encoder to model the dependencies between the different facial patches. In addition, a learnable [CLS] token is attached to the patch token series and the following learnable positional encodings are applied to the tokens:

$$Z_0 = [x_{cls} \parallel X] + E_{pos}$$

The token sequence undergoes encoding via $L=6$ Transformer encoder layers, where each encoder layer involves the operations of multi-headed self-attention, followed by feed-forward neural networks and pre-layernorm:

$$Z'_l = MSA(LN(Z_{l-1})) + Z_{l-1}$$

$$Z_l = FFN(LN(Z'_l)) + Z'_l$$

Multi-headed self-attention with $H=6$ heads is computed as follows:

$$MSA(Z) = \text{Concat}(\text{head}_1, \dots, \text{head}_H) \cdot W_O$$

$$\text{head}_h = \text{Attention}(ZW_h^Q, ZW_h^K, ZW_h^V)$$

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) \cdot V$$

Where $d_k = d_{\text{model}} / H = 64$ is the per-head dimension. Pruning tokens is performed after every encoder layer, in which 20% tokens with the least attention weights relative to the [CLS] token will be pruned. The feed-forward network contains two linear layers with GELU activation:

$$FFN(x) = W_2 \cdot \text{GELU}(W_1 x + b_1) + b_2$$

with $W_1 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ where $d_{\text{ff}} = 768$, and $W_2 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$.

3.5.6 Classification Head

The output vector for the last CLS token, $Z_L^{\{\text{cls}\}}$, goes through a classification head made up of two layers:

$$\begin{aligned} \text{logits} &= W_2 \cdot \text{ReLU}(W_1 \cdot Z_L^{\{\text{cls}\}} + b_1) + b_2 \\ \hat{y} &= \text{softmax}(\text{logits}) \end{aligned}$$

where $W_1 \in \mathbb{R}^{256 \times 384}$ and $W_2 \in \mathbb{R}^{7 \times 256}$. The vector of probabilities $\hat{y} \in \mathbb{R}^7$. The predicted emotion label is: $\hat{y}^* = \text{argmax}(\hat{y})$.

3.6 Training Framework

The proposed training pipeline aims at concurrently addressing all the three challenges involved in the FER2013 dataset that is, the presence of class imbalance, noisy labels, and the need for stable fine-tuning of large pre-trained

backbone models by combining focal loss, class frequency re-weighting, sample-wise re-weighting based on confidence scores, and a multi-step optimization paradigm, as shown in Figure 4.7.

Fig. 4.7. Training framework pipeline: training batch → focal loss with class-frequency weights and confidence-based sample weighting → combined weighted loss → AdamW optimizer → cosine annealing scheduler with linear warmup → model weights update.

3.6.1 Focal Loss with Class-Frequency Weighting

Standard cross-entropy loss is substituted with focal loss [54] to cope with extreme class imbalance in FER2013. The standard cross-entropy loss is modified by adding a modulating factor to reduce the importance of easy majority class samples' contribution to the loss value:

$$FL(p_t) = -\alpha_t \cdot (1 - p_t)^\gamma \cdot \log(p_t)$$

where p_t is the model's prediction probability for the correct label, $\gamma=2.0$ is the focusing parameter, and α_t is a frequency-dependent weight:

$$\alpha_t = N_{total} / (C \cdot N_c)$$

where N_{total} is the number of training samples, $C=7$ is the number of emotions, and N_c is the number of samples in class c .

3.6.2 Confidence-Based Sample Weighting

In order to deal with the 15-20% estimated annotation noise level of FER2013, confidence-driven sample weighting is utilized for training. The loss weight w_i per sample is calculated by:

$$w_i = (p\theta(\hat{y}_i = y_i | x_i))^\beta$$

where $p_{\theta}(\hat{y}_i = y_i | x_i)$ represents the current prediction probability by the model for the annotated class y_i for sample x_i and $\beta = 0.5$ is the temperature parameter. The total loss function per sample is:

$$\mathcal{L}_i = w_i \cdot FL(p_{ti})$$

3.6.3 Optimization Configuration

AdamW [55] is used, which involves both adaptive learning rate and decoupled weight decay regularization:

$$\theta_{\{t+1\}} = \theta_t - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon) - \eta \cdot \lambda \cdot \theta_t$$

Cosine learning rate decay is used:

$$\eta_t = \eta_{min} + (\eta_{max} - \eta_{min}) \cdot (1 + \cos(\pi t / T)) / 2$$

where $\eta_{max} = 1 \times 10^{-4}$, $\eta_{min} = 1 \times 10^{-6}$, t refers to the current epoch and T to the total number of epochs. A linear warm-up phase of 10 epochs is used for the first 10 epochs. The full list of training hyperparameters is provided in Table 4.2.

TABLE 3.2. TRAINING HYPERPARAMETERS

Hyperparameter	Value
Optimizer	AdamW
Initial learning rate (η_{max})	1×10^{-4}
Minimum learning rate (η_{min})	1×10^{-6}
Weight decay (λ)	0.01
LR schedule	Cosine annealing
Warmup epochs	10
Batch size	64
Maximum epochs	100
Early stopping patience	15 epochs (validation accuracy)

Dropout rate	0.1
Focal loss γ	2.0
Confidence weighting β	0.5
ViT attention heads (H)	6
ViT token dimension (d_model)	384
ViT FFN inner dimension (d_ff)	768
ViT encoder blocks (L)	6
Token pruning rate (p)	20% per block
EfficientNet-B7 feature dim	512
GhostNet feature dim	512
Fused feature dimension	512
Pretrained weights	ImageNet (all backbones)

3.7 Explainability Analysis

The Gradient-weighted Class Activation Mapping technique [56] is employed to visualize model decision-making for each of the seven FER2013 emotions in the trained EfficientViT-FER architecture. Grad-CAM obtains the gradients of the predicted class score c with respect to feature maps A^k from a chosen layer and utilizes them as weights to obtain a localization map as follows:

$$\alpha_k^c = (1/Z) \sum_i \sum_j (\partial y^c / \partial A_{ij}^k)$$

$$L_{\{Grad-CAM\}}^c = ReLU(\sum_k \alpha_k^c \cdot A^k)$$

In the case of the vision transformer module, Grad-CAM is computed using the outputs of the attention layers in the last encoder module. In the case of the EfficientNet-B7 backbone, Grad-CAM is obtained using the feature maps after the last convolution layer but before the global average pooling layer. The obtained maps are up-sampled to match the input image resolution.

Visualization results from Grad-CAM are obtained for both correctly classified and misclassified samples from the seven emotion categories. Grad-CAM

heatmaps of the correct classification cases provide qualitative insights into anatomical facial regions attended by the model. Grad-CAM heatmaps of misclassified cases provide insights into facial features and inter-category differences leading to recognition errors.

Fig. 4.8. Grad-CAM explainability analysis pipeline: trained EfficientViT-FER model → gradient computation with respect to final encoder block attention features → importance weight computation → ReLU-gated localization heatmap → upsampled saliency overlay shown for all seven emotion categories.

3.8 Evaluation Framework

EfficientViT-FER is tested using a unified multidimensional testing protocol that considers both recognition performance metrics and computational efficiency. The test is done only on the set of 5,383 test images and not involved in any stages of model learning or hyperparameter tuning.

Accuracy: Fraction of images recognized correctly for each of the seven emotion classes, a key recognition performance metric to be compared with existing results from FER2013 benchmarks.

Per-Class F1-Score: Harmonic mean of precision and recall calculated for each individual emotion class; macro average F1-score is defined as follows:

$$F1_macro = (1/7) \sum_{c=1}^7 (2 \cdot P_c \cdot R_c) / (P_c + R_c)$$

Parameter Count: Number of trainable parameters throughout the whole model pipeline (without fixed YOLOv8 network), measured in millions (M).

Inference Latency: Time required to process a single test image in forward pass through the whole model, measured in milliseconds (ms) and frames per second (FPS).

The proposed solution is compared with six benchmark models as follows:

Baseline Model	Reference	Year	FER2013 Accuracy
VGG-CNN	Khairuddin & Chen [15]	2021	73.28%
LHC-Net	Pecoraro et al. [16]	2022	72.96%
ViT-AP	Xue et al. [20]	2022	~73.5%
AutoML-FER	Tang et al. [22]	2025	76.4%
FERmc	Li et al. [23]	2025	Evaluated on FER2013
EfficientMobileNet	Ismail et al. [24]	2026	77.6%

3.9 Summary

The above section provided the entire proposed methodology for the EfficientViT-FER algorithm in a well-defined set of steps. FER2013 data set will be split into train, validation, and testing data using stratified 70–15–15 split. The entire process of preprocessing involves standardizing all images by face detection and cropping based on YOLOv8, image resizing to 224 x 224 pixels using bicubic interpolation, converting grayscale to RGB images, and finally applying standard image net normalization. An augmentation technique involving two layers of transformation is used in which all images undergo general and class-wise augmentations with increased intensity for Disgust, Fear, and Surprise categories.

The hybrid network architecture proposed herein includes four different modules: YOLOv8 for face localization, EfficientNet-B7 for compound scaling feature extraction, GhostNet for efficient complementary feature extraction, and a vision transformer encoder with token pruning for global spatial attention modeling over faces. The dual backbone features are concatenated and projected linearly before being tokenized and passed through the transformer encoder; the [CLS] token representation will then be forwarded into a classification head of two fully connected layers to generate seven emotion classes predictions. The learning framework will simultaneously tackle the problems of class imbalance via the focal loss function along with inverse-frequency class weighting and label noise via confidence sample weighting optimization using AdamW algorithm with cosine annealing and linear warmup schedule. Network interpretability will be

achieved via Grad-CAM visualization of the attention-weighted regions of the facial images over all seven emotions. The model aims at 90% classification accuracy on the FER2013 test dataset, benchmarked against six other representative models under a unified multi-dimensional evaluation paradigm of accuracy, F1-score, number of parameters, and inference speed.

Chapter 4 :Experimental Results and Discussion

Chapter 5 Conclusion and Future Work

References