Efficient Transformers for FER: LoRA Adaptation and Cross-Dataset Evaluation with POSTER

Calderara Beatrice, Lo Presti Federico, Veronese Alex
University of Modena and Reggio Emilia
{300033, 302624, 302474}@studenti.unimore.it

***Abstract*—Facial Emotion Recognition (FER) remains a challenging computer vision task due to the large variability of facial expressions across subjects and conditions, as well as the strong class imbalance that characterizes many public benchmarks. In this work, we build on the POSTER framework and investigate both preprocessing improvements and architectural adaptations to enhance robustness and generalization. We study the impact of data augmentation and imbalance-mitigation strategies, and we explore parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA) to adapt pretrained models to FER with reduced computational cost. In addition, we introduce a retrieval module that extracts and analyzes frames from movie clips, enabling qualitative evaluation in unconstrained scenarios. Finally, we incorporate an explainability component based on Class Activation Maps (CAM) to highlight the image regions that most influence the model's predictions and improve interpretability. Overall, our experiments show that imbalance-mitigation strategies are strongly dataset-dependent, while LoRA consistently provides a favorable trade-off between recognition performance and fine-tuning efficiency.**

Code is available at: https://github.com/alexveronese/POSTER-cv

***Index Terms*—Facial Expression Recognition, Transformers, Low-Rank Adaptation (LoRA).**

I. Introduction

FACIAL expression is one of the most powerful, natural and universal signals for human beings to convey their emotional states and intentions. Numerous studies have been conducted on automatic facial expression analysis because of its practical importance in sociable robots, medical treatment, driver fatigue surveillance, and many other human-computer interaction systems [1]. In the field of computer vision and machine learning, facial expression recognition (FER) is a critical task and has various practical applications in fields such as human-computer interaction, education, healthcare, and online monitoring [2]. Despite the remarkable progress achieved through deep learning techniques, FER remains challenging due to high intra-class variability, subject-dependent facial expressions, and the inherent class imbalance of commonly used datasets.

In this work, we addressed the FER task by taking the POSTER [2] architecture as our main reference. Our goal was to analyze, reproduce, and extend some of the methodological choices proposed in the original paper, while exploring possible improvements across the entire processing pipeline.
Our pipeline consists of a preprocessing stage with image resizing and normalization, followed by data augmentation to improve robustness. We extended the baseline setup by exploring additional data augmentations, strategies to mitigate class imbalance, and facial alignment based on landmark keypoints. On the modeling side, we investigated parameter-efficient fine-tuning via Low-Rank Adaptation (LoRA) [3] to adapt pretrained networks to FER with reduced computational cost. We included an inference module for extracting representative video frames and an explainability component based on Class Activation Maps (CAM) to provide qualitative insight into the regions driving the model's predictions.We also implemented a retrieval module which returned nearest-neighbor examples for face crops extracted from movie frames, enabling qualitative assessment in unconstrained scenarios.

II. Methodology

A. Baseline

The baseline model in this work is **POSTER** (Pyramid crOss-fuSion TransformER) [2], a two-stream facial expression recognition (FER) architecture designed to jointly address three common challenges in FER: inter-class similarity, intra-class discrepancy, and scale sensitivity. The model combines an image stream (capturing global facial appearance) with a landmark stream (capturing sparse expression-relevant keypoints) and enables information exchange between the two via transformer-based cross-fusion attention.

Given an aligned face image $X \in R^{H\times W\times 3}$, POSTER [2] extracts image features $X_{\text{img}} \in R^{P\times D}$ using a CNN backbone (e.g., IR50), which is initialized from face recognition pretraining on MS-Celeb-1M and fine-tuned for FER. In parallel, an off-the-shelf facial landmark detector (e.g., MobileFaceNet) produces landmark features $X_{\text{lm}} \in R^{P\times D}$; during training, the landmark detector is kept frozen to preserve stable landmark outputs.

Rather than simply concatenating modalities, POSTER [2] introduces a cross-fusion multi-head self-attention mechanism in which the two streams exchange query information to guide each other, as shown in Figure 1. Specifically, in the image stream, attention uses a landmark-derived query together with image-derived key/value, while in the landmark stream, attention uses an image-derived query together with landmark-derived key/value. This design encourages the image stream to focus on salient landmark-indicated regions and provides the landmark stream with additional global facial context.

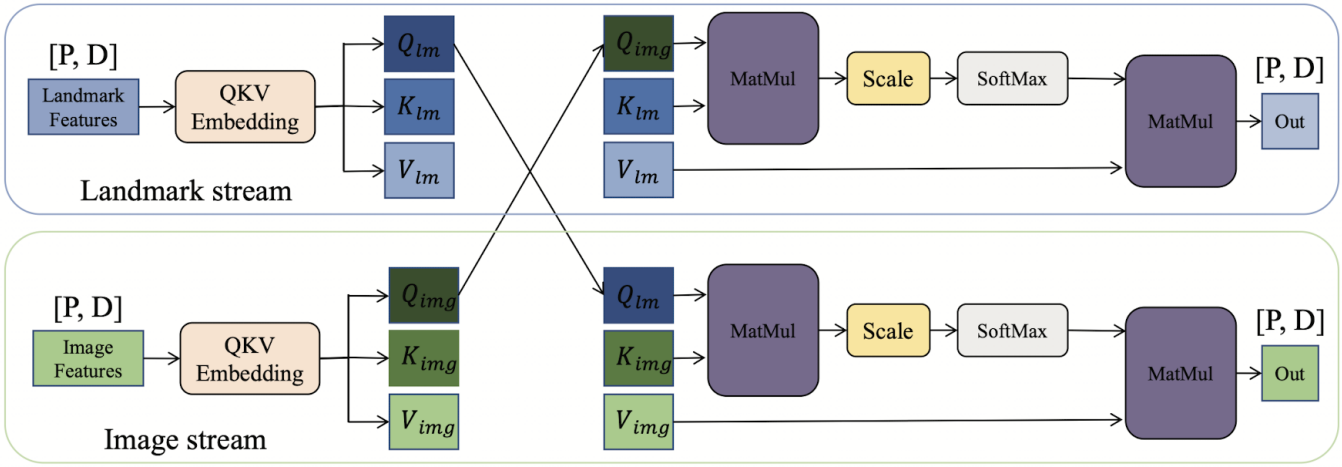


Fig. 1: Cross-fusion multi-head self-attention (MSA) used in POSTER. The landmark stream attends using the image-stream query (and vice versa), enabling mutual guidance between landmark and image features. Adapted from Zheng et al. [2]

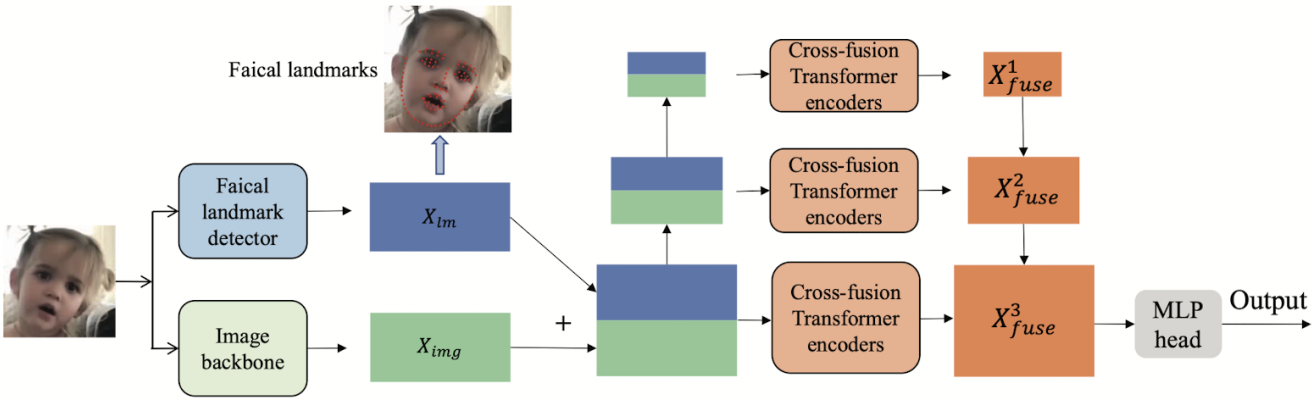


Fig. 2: Overview of the POSTER architecture for facial expression recognition: an image backbone extracts image features X_{img} and a facial landmark detector extracts landmark features X_{lm}; multi-scale (pyramid) features are processed by cross-fusion transformer encoders and the resulting fused representations $\{X^1_{\text{fuse}}, X^2_{\text{fuse}}, X^3_{\text{fuse}}\}$ are aggregated and classified by an MLP head. Adapted from Zheng et al. [2].

To reduce scale sensitivity, POSTER [2] incorporates a feature pyramid strategy by constructing multiple feature levels (large/medium/small) and processing them with separate cross-fusion transformer encoders. In the reference implementation, the pyramid levels use embedding dimensions $D_H = 512$, $D_M = 256$, and $D_L = 128$, and each level applies a stack of transformer encoders before the multi-scale outputs are aggregated for final classification. The complete architecture is shown in Figure 2.

POSTER [2] is trained end-to-end: the image backbone is updated during training, while the landmark detector remains frozen. The reference setup uses label-smoothing cross-entropy loss, batch size 100, learning rate 4×10^{-5}, MLP ratio 2, and drop-path rate 0.01 in the transformer encoders.

B. LoRA

To address the computational overhead and memory constraints associated with the full fine-tuning of the Pyramid Vision Transformer (approximately 71M parameters), we employed **Low-Rank Adaptation (LoRA)** [3]. This technique assumes that, during fine-tuning, the task-specific weight update has a low intrinsic rank, i.e., the adaptation mainly lies in a low-dimensional subspace rather than requiring an unrestricted update of all parameters. For a pre-trained weight matrix $W_0 \in R^{d \times k}$, the update is constrained by representing the side-tuning as a low-rank decomposition:

$$W = W_0 + \Delta W = W_0 + BA \tag{1}$$

where $B \in R^{d \times r}$ and $A \in R^{r \times k}$, and the rank $r \ll \min(d, k)$. During the training phase, W_0 is kept frozen, and only the A and B matrices are updated. In our implementation, LoRA was applied to the query, key, and value (q, k, v) projections, the output projections, and the position-wise feed-forward networks, as shown in Figure 3.

III. Experiments

A. Datasets

We evaluated our method on four widely used facial expression recognition (FER) benchmarks, spanning both controlled

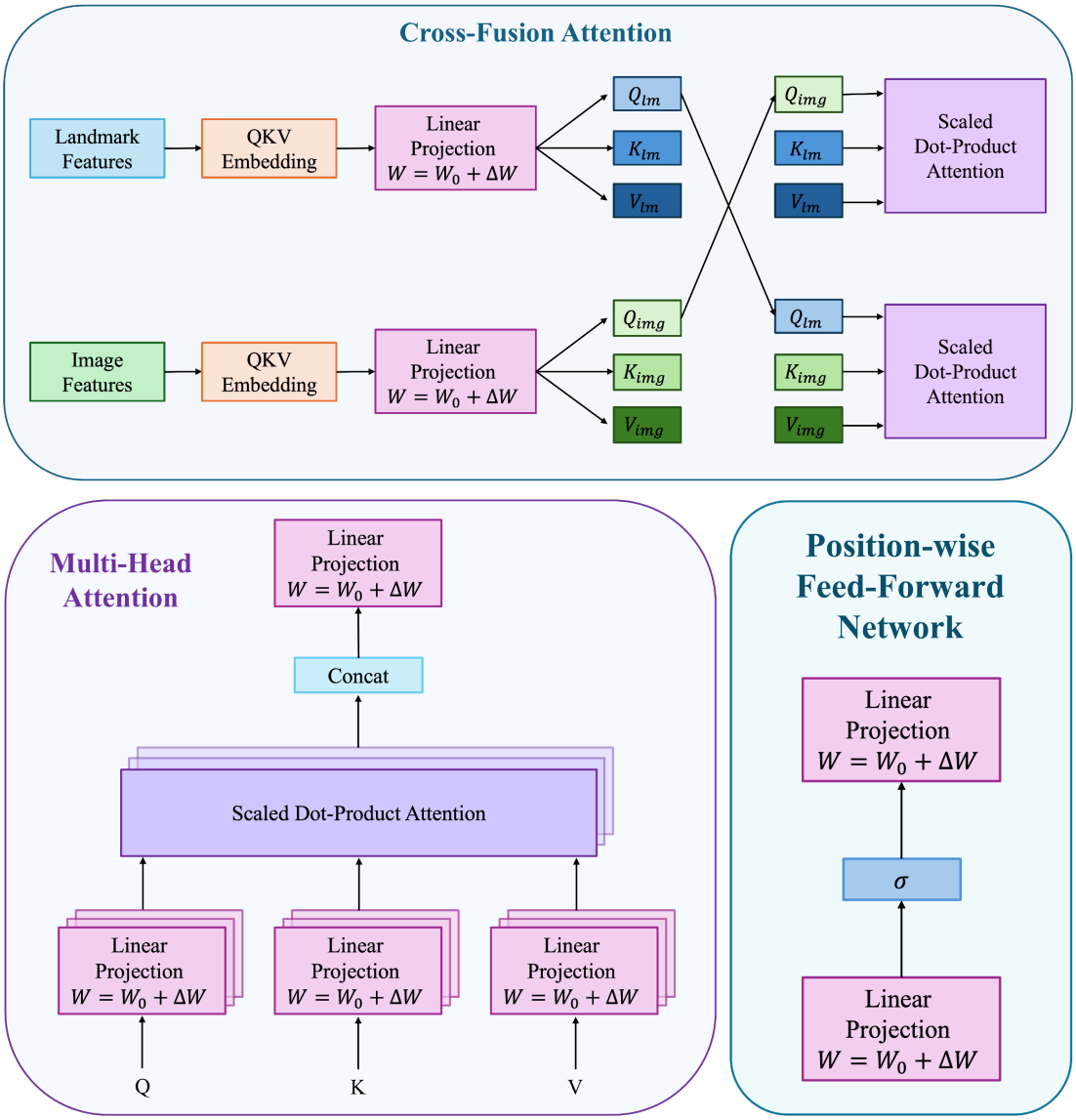


Fig. 3: In this figure, we highlight the specific modules where LoRA [3] is integrated into the architecture, showing that the low-rank adaptation is applied to the linear projection layers of the attention mechanism (Q, K, V and the output projection) and to the position-wise feed-forward layers.

laboratory settings and unconstrained in-the-wild imagery, in order to assess robustness across different data distributions and acquisition conditions. All benchmarks were sourced from publicly accessible online releases, and download links are provided in the appendix.

RAF-DB (Real-world Affective Faces Database) is a large-scale FER benchmark comprising 29,672 real-world facial images collected from the Internet, with substantial variability in subject identity and demographics, illumination, background clutter, occlusions, and head pose. Under the standard basic-expression protocol, 12,271 images are used for training and 3,068 for testing, and all samples are annotated into 7 basic emotion categories: *happiness*, *surprise*, *sadness*, *anger*, *disgust*, *fear*, and *neutral* [4].

CK+ (Extended Cohn–Kanade) is a laboratory-controlled dataset composed of posed facial expression sequences that evolve from a neutral onset to a peak expression, and it is widely used as a clean benchmark to evaluate FER methods under constrained capture conditions with limited background and illumination variation. In this work, we used the CK+ version distributed on Kaggle, which provides a CSV-based representation of face-cropped grayscale images at 48×48 resolution. Since CK+ samples are grayscale, we converted each 48×48 single-channel image into a $48 \times 48 \times 3$ tensor by duplicating the channel three times, ensuring compatibility with RGB-pretrained backbones without altering the underlying intensity information. We casted the task as an 8-category classification problem, using the following emotion labels: *anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise*, *neutral*, and *contempt* [5].

AffectNet is a large-scale in-the-wild dataset collected by querying search engines with emotion-related keywords, con-

taining substantial real-world variability (e.g., pose, illumination, and occlusion) and covering the basic emotion categories; following standard practice, we adopted the 8-class setting and used an *aligned* version of the dataset, meaning that faces were first normalized by a dedicated face aligner to obtain consistent geometry and cropping across samples [6].

Finally, we included **FER2013** as a widely used large-scale FER benchmark featuring unconstrained, in-the-wild facial images with face crops that span broad variations in facial viewpoint, lighting patterns, image quality, and background context. The dataset contains 48×48 grayscale images and follows the canonical split of $28{,}709$ training images and $3{,}589$ public-test images. Since FER2013 provides face crops without an explicit geometric normalization step, we additionally applied our face-alignment preprocessing to obtain an *aligned* version of the dataset, improving consistency in face geometry and cropping across samples. Moreover, because FER2013 is grayscale while our backbone expects 3-channel inputs, we converted each single-channel image to RGB by replicating the intensity values across the three channels as in CK+. The dataset is annotated into 7 basic expression categories: *anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise*, and *neutral* [7].

B. Implementation Details

1) Alignment: To reduce the geometric variability in FER2013 and AffectNet, we inserted an explicit face-alignment step in the input transform pipeline. We adopted a landmark-based strategy using MTCNN [8] (via `facenet-pytorch`), which detects faces and returns five stable keypoints: left/right eye, nose tip, and left/right mouth corner, that remain reliable under moderate pose and illumination changes.

We deliberately restricted alignment to a 2D similarity-like affine model, since the dominant nuisance factors across samples are largely global and approximately planar (translation, in-plane rotation, and isotropic scaling).

Given a set of detected facial landmarks $\{p_i\}_{i=1}^{N}$ in the input image, with $p_i \in R^2$, we defined a corresponding canonical template $\{q_i\}_{i=1}^{N}$ in a fixed output coordinate system of size 224×224, where $q_i \in R^2$ denotes the desired location of the i-th landmark after alignment. We estimated the similarity transform parameters by fitting

$$q_i \approx Ap_i + t, \tag{2}$$

where $A \in R^{2\times 2}$ is the linear component encoding in-plane rotation and uniform scaling, and $t \in R^2$ is the translation vector. Equivalently, OpenCV returns the corresponding 2×3 matrix

$$T = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \end{bmatrix}, \tag{3}$$

which is fully determined by four degrees of freedom: rotation angle θ, scale s, and translations (t_x, t_y). In practice, we computed A and t using `cv2.estimateAffinePartial2D` [9], which fits the optimal transformation in a least-squares sense over the landmark correspondences $\{(p_i, q_i)\}_{i=1}^{N}$. This procedure constrains the mapping to translation, rotation, and isotropic scaling, providing robust normalization of in-plane variations while avoiding excessive distortions. The aligned face is then warped to the canonical frame, enforcing a consistent spatial layout of discriminative regions (especially eyes and mouth) across both datasets.

2) Data Augmentation: We trained all datasets using the same baseline configuration as POSTER [2]. Specifically, each input image is converted to a PIL image, resized to 224×224, converted to a tensor, and normalized with mean 0.5 and standard deviation 0.5 to ensure a consistent preprocessing protocol across benchmarks. During training, we additionally applied **random erasing** with an erased-area scale range of $(0.02, 0.1)$ to improve robustness to partial occlusions and to reduce overfitting to spurious texture cues. Finally, we introduced optional photometric augmentation by applying a **lighting** perturbation module (enabled when `use_lighting` is set) with strength 0.1, which induces random illumination changes to better simulate real-world lighting variability, while keeping the same augmentation pipeline for all datasets to preserve experimental consistency. All ablations reported in the following sections apply these augmentations (and the class-balancing component) uniformly across RAF-DB, CK+, AffectNet, and FER2013 [4]–[7].

3) Sampling Strategy: Since some of the employed datasets exhibit noticeable class imbalance, with some expressions (e.g., *happy*) appearing much more frequently than others, we incorporated a **class-balanced sampling strategy** during training to mitigate bias toward majority classes and to study its impact under a unified protocol. Concretely, we used an `ImbalancedDatasetSampler` that assigns each training sample a weight inversely proportional to the frequency of its class label

$$w_i \propto 1/n_{y_i} \tag{4}$$

where n_{y_i} denotes the number of samples belonging to class y_i. At each epoch, mini-batches are then formed by drawing indices via multinomial sampling with replacement according to these weights, which effectively oversamples minority classes and increases their contribution to the gradient updates without altering the underlying datasets.

4) Loss functions: To train the network, we followed the original POSTER [2] training recipe and optimized a classification objective based on cross-entropy, using both the **standard cross-entropy loss** and its **label-smoothed variant** (Label Smoothing Cross Entropy) with smoothing factor $\epsilon = 0.2$. Label smoothing regularizes the classifier by replacing hard one-hot targets with a convex combination of the ground-truth label and a uniform distribution, which reduces over-confidence and can improve generalization. Concretely, the loss used in this setting is:

$$\mathcal{L}_{CE\_train} \;=\; \mathcal{L}_{CE} \;+\; 2\,\mathcal{L}_{LSCE} \tag{5}$$

To further counter the long-tailed label distribution that commonly arises in in-the-wild FER benchmarks, we replaced the standard cross-entropy objective with a **Class-Balanced (CB) loss** that explicitly amplifies the contribution of under-represented expressions. We computed for each class c an *effective number of samples*

$$E_c \;=\; \frac{1-\beta^{n_c}}{1-\beta}, \tag{6}$$

where n_c is the number of training samples in class c and $\beta \in [0, 1)$ controls how quickly additional samples saturate in value. The per-class reweighting factor is then obtained as the inverse of E_c,

$$w_c \;=\; \frac{1-\beta}{1-\beta^{n_c}}, \tag{7}$$

and, in our implementation, these weights are normalized to keep the average weight close to one across the C classes. Given a minibatch, we formed one-hot targets and replicated the corresponding w_c across the logit dimension so that each training example is scaled according to its ground-truth class before computing the loss. We employed the focal-loss instantiation (CB-Focal), which down-weights easy samples while emphasizing hard and misclassified ones through the focusing term controlled by γ [10]. In all CB-Focal experiments, we set $\beta = 0.9999$ and $\gamma = 2.0$, and optimized the resulting objective

$$\mathcal{L}_{\text{train}} \;=\; \mathcal{L}_{\text{CB-Focal}}. \tag{8}$$

5) Optimization and training schedule: We optimized the network using a two-step Sharpness-Aware Minimization (SAM) procedure, where each iteration consists of two forward–backward passes: after the first backward pass we perform a perturbation step, and after recomputing the loss at the perturbed weights we apply the actual parameter update [11]. We kept the optimizer configuration consistent with the original POSTER [2] training setup, and we reported all experiments using the same learning rate and scheduling strategy to ensure fair comparisons across datasets.

To reduce unnecessary computation and prevent overtraining, we additionally employed early stopping: training is terminated when the monitored validation metric does not improve for a fixed number of consecutive epochs (patience), and the checkpoint with the best validation performance is retained. In our implementation, we set the early-stopping patience to 20 epochs, which allows the model to converge while avoiding wasted training time once performance saturates.

C. Ablation Studies

Table I reports our ablation results on the validation splits for all datasets, using Accuracy (Acc.) and F1 score (F1) as evaluation metrics. Accuracy measures the fraction of correctly classified samples over the entire set, while the F1 score summarizes the trade-off between precision and recall via their harmonic mean, and is therefore more informative when performance differs across classes.

We evaluated each dataset under five configurations: (i) *Standard*, meaning the baseline training pipeline as reported in the original paper [2]; (ii) *Lighting + Random Erasing*; (iii) *Lighting + Random Erasing + Class balancing*; (iv) *Lighting + Random Erasing + CB loss*; and (v) *All*, which combines lighting augmentation, random erasing, class balancing, and CB loss within a single setup. These configurations are designed to disentangle the impact of data augmentation and imbalance-mitigation strategies. In the following, we discussed how these components affect Accuracy and F1 across datasets, and we highlighted the most effective configuration for each benchmark.

The CK+ dataset [5] was not included in the original POSTER [2] benchmark; nevertheless, we incorporated it in our evaluation to broaden the comparison and to assess the robustness of the training pipeline on a widely used laboratory-controlled FER dataset. In Table I, all configurations achieve the same Accuracy (0.9837), while small differences emerge in terms of F1, with the best result obtained when combining class balancing and CB loss (F1=0.9527). This behaviour suggests that, on CK+, the proposed modifications have a limited effect on overall correctness (likely due to the dataset being comparatively easier and closer to saturation), but they can still slightly improve the balance between precision and recall across classes. Class balancing and CB loss are complementary mechanisms to mitigate skewed class distributions: the former increases the contribution of minority classes through sampling, while the latter explicitly re-weights the loss according to class frequency, potentially improving per-class performance. However, using both simultaneously may also lead to *over-compensation* of minority classes, by amplifying noisy or scarce samples, which can reduce stability and may not translate into higher Accuracy even when F1 improves.

This effect appears to be more pronounced on RAF-DB [4], where imbalance-mitigation strategies do not consistently translate into improved validation performance. As reported in Table I, the best configuration on RAF-DB is *Lighting + Random Erasing*, achieving 0.9179 Accuracy and 0.8708 F1, slightly improving over the standard setting (0.9169/0.8668). Adding class balancing or CB loss on top of the same augmentations leads to lower scores, suggesting that the additional reweighting may not be well-aligned with the effective difficulty of the dataset and may amplify hard or noisy samples typical of in-the-wild annotations. Overall, these results indicate that, for RAF-DB, stronger regularization through data augmentation is beneficial, whereas explicitly enforcing balance through class balancing and/or CB loss requires careful tuning to avoid degrading both Accuracy and F1.

On AffectNet [6], lower absolute scores are expected compared to more constrained benchmarks, since AffectNet is a large-scale in-the-wild dataset collected from the web and is known to exhibit substantial appearance variability as well as noisy and imbalanced annotations. In Table I, we observed modest gains when introducing some sort of balance within classes: the configurations that include class balancing and/or CB loss achieve higher Accuracy/F1 (0.7502/0.7215 only using *Class balancing*, 0.7577/0.7303 only using *CB loss* and 0.7529/0.7222 using both) compared to the corresponding settings without imbalance handling (0.7494/0.7188 for *Lighting + Random Erasing*) as well as to the baseline *Standard* configuration (0.7479/0.7160). This trend suggests that AffectNet benefits from explicitly compensating for its skewed class distribution, even though the improvements are small in magnitude. Finally, CB loss yields the strongest result on this dataset (0.7577/0.7303), which further supports the relevance of imbalance-aware objectives when training on AffectNet.

Finally, FER2013 [7] shows the opposite trend compared to AffectNet, suggesting that explicit class-imbalance mitigation is not essential in our setting. FER2013 is composed

TABLE I: Ablation study across datasets (validation).

	CK+		RAF-DB		AffectNet		FER2013	
Setting	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Standard[a]	0.9837	0.9499	0.9169	0.8668	0.7479	0.7160	**0.7470**	**0.7409**
Lighting + Random Erasing	0.9837	0.9499	**0.9179**	**0.8708**	0.7494	0.7188	0.7470	0.7357
Lighting + Random Erasing + Class balancing	0.9837	0.9499	0.9094	0.8625	0.7502	0.7215	0.7409	0.7302
Lighting + Random Erasing + CB loss	0.9837	0.9499	0.9065	0.8526	**0.7577**	**0.7303**	0.7414	0.7361
Lighting + Random Erasing + Class balancing + CB loss	**0.9837**	**0.9527**	0.9117	0.8634	0.7529	0.7222	0.7425	0.7323

[a] We used the original POSTER architecture [2] as our baseline and evaluated it on the above datasets without introducing any additional modifications.

of low-resolution (48×48) grayscale face crops annotated into seven basic emotions, and its limited visual detail can make additional re-weighting strategies more sensitive to noise. As shown in Table I, the best validation performance is obtained with the *Standard* configuration (Acc.=0.7470, F1=0.7409), while adding class balancing reduces both metrics (0.7409/0.7302) and the full configuration further degrades performance (0.7425/0.7323). These results indicate that, on FER2013, the baseline training recipe already provides a strong trade-off between overall correctness and class-wise performance, whereas more aggressive imbalance-aware strategies may over-emphasize scarce or noisy samples and thus hurt generalization on the validation split.

D. Results with LoRA

We applied LoRA [3] to all four evaluated datasets (CK+, RAF-DB, AffectNet, and FER2013 [4]–[7]) in order to study the trade-off between accuracy and parameter efficiency under different data distributions.

1) RAF-DB: We primarily conducted experiments on RAF-DB with two different rank configurations, $r = 32$ and $r = 64$, using a scaling factor $\alpha = 2r$ to ensure numerical stability. This approach reduced the number of trainable parameters from 70.843M to 5.57M ($r = 32$) and 10.62M ($r = 64$), representing only 7.8% and 14.9% of the total model capacity, respectively. The transition from $r = 32$ to $r = 64$ yielded a marginal improvement of +0.75%, suggesting that the emotional feature space in the Pyramid Transformer [2] can be effectively captured within a low-dimensional subspace. Furthermore, we investigated a Hybrid Strategy, where LoRA [3] was combined with the unfreezing of critical components, specifically the normalization layers and the cross-attention fusion modules, as well as the final transformer block. Surprisingly, increasing the trainable parameters to 30.06M did not lead to superior performance (88.04%) compared to the standard $r = 64$ LoRA [3] (88.69%). This outcome indicates that for multi-branch architectures like the one used here, integrating both landmarks and image features, maintaining the integrity of the pre-trained weights in the fusion layers is more beneficial than providing additional degrees of freedom, which may lead to catastrophic forgetting or sub-optimal alignment between the two input streams.

A key observation from our comparative analysis is the divergent behavior of the training loss and accuracy between full fine-tuning and LoRA [3]. While the full fine-tuning approach converges more rapidly, it exhibits a significant gap between training and validation performance, a hallmark of potential overfitting. In contrast, LoRA [3] ($r = 32$ and $r = 64$) acts as an implicit regularizer. By constraining the weight updates to a low-rank subspace, LoRA [3] prevents the model from capturing high-frequency noise in the training data, forcing it to focus on the most salient features for facial expression recognition. As shown in Figure 4, the training accuracy for LoRA [3] models remains closer to the validation accuracy throughout the process. This behavior suggests that the low-rank constraint effectively reduces the Hypothesis Space, making the optimization landscape smoother and the resulting model more robust. For RAF-DB dataset, which contains diverse real-world facial orientations, this regularization is crucial for ensuring that the model generalizes well to unseen subjects rather than memorizing specific training samples.

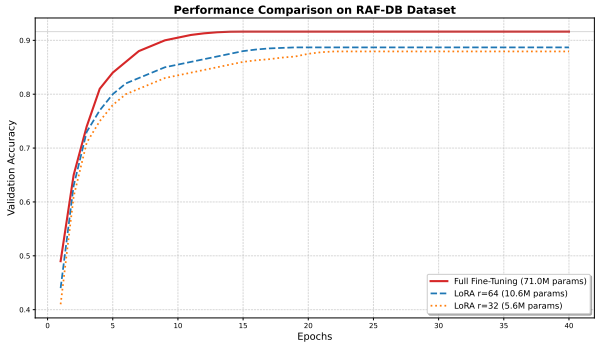


Fig. 4: Validation accuracy on RAF-DB [4] over training epochs for full fine-tuning (71.0M trainable parameters) and LoRA [3] with ranks $r = 64$ (10.6M) and $r = 32$ (5.6M), highlighting the trade-off between accuracy and parameter efficiency.

2) Other datasets: Based on the RAF-DB ablation study, increasing the LoRA [3] rank from $r = 32$ to $r = 64$ consistently improved performance, although with a relatively small gain, indicating that a moderately higher rank provides additional expressive capacity without sacrificing the advantages of LoRA [3] parameter-efficiency. For this reason, and to keep the experimental protocol consistent across benchmarks, we fixed $r = 64$ for all subsequent evaluations on the remaining datasets (CK+, AffectNet, and FER2013). The experimental results across all datasets highlight a compelling trade-off between absolute performance and parameter efficiency. As reported in Table I, full fine-tuning achieves the best overall

TABLE II: Ablation study with LoRA across datasets (validation).

	CK+		RAF-DB		AffectNet		FER2013	
Setting	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Standard	0.9565	0.7760	0.8869	0.8301	0.7056	0.6738	**0.6930**	**0.6758**
Lighting + Random Erasing	0.9565	0.7760	**0.8872**	**0.8318**	**0.7078**	**0.6762**	0.6860	0.6661
Lighting + Random Erasing + Class balancing	**0.9837**	**0.9527**	0.8862	0.8283	0.7063	0.6764	0.6849	0.6662
Lighting + Random Erasing + CB loss	0.9783	0.9322	0.8804	0.8232	0.7060	0.6775	0.6863	0.6659
Lighting + Random Erasing + Class balancing + CB loss	0.9674	0.9132	0.8729	0.8154	0.7062	0.6761	0.6826	0.6653

accuracy; nevertheless, the LoRA-based approach [3] remains highly competitive, retaining on average 95% of the fully fine-tuned performance (Table II). The LoRA [3] variants exhibit only minor fluctuations on AffectNet and FER2013 under the different training configurations, suggesting that the low-rank constraint provides an implicit regularization effect and supports robust generalization across heterogeneous data distributions. On CK+, where the task is closer to saturation, LoRA [3] still attains very high accuracy, but the outcome is more influenced by the imbalance-mitigation strategy, with class balancing yielding the most consistent improvements.

Although full fine-tuning achieves the best absolute performance, LoRA [3] retains a large fraction of the accuracy while consistently reducing the number of trainable parameters, making it a compelling alternative when efficiency and scalability are key requirements.

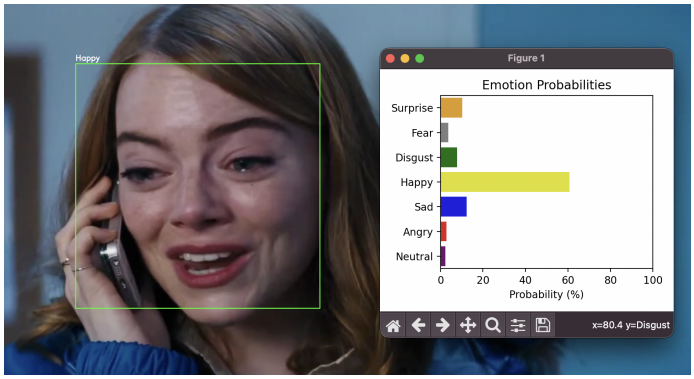
IV. Inference

To qualitatively assess the behavior of our FER models in unconstrained scenarios, we designed a real-time evaluation module on short video clips extracted from movies. Given an input video, the clip is decoded into a sequence of RGB frames which are processed continuously without temporal subsampling. This approach allows for a granular analysis of the model's consistency across consecutive frames and its sensitivity to transient facial expressions.

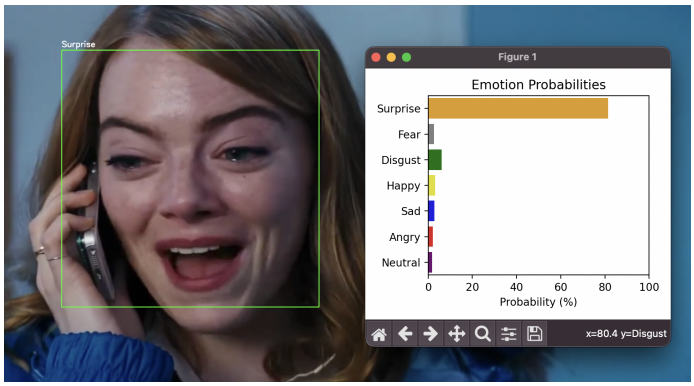
For each frame, we employed a Haar Cascade Classifier based on the Viola-Jones framework [12] to perform subject-specific face detection. This choice ensures a high-throughput, low-latency localization of the facial region, which is essential for maintaining real-time performance during inference. Frames where a face is not successfully detected are discarded. The identified face region is subsequently cropped to ensure the pipeline focuses exclusively on the subject's facial features rather than the background context.

Each cropped face underwent the same test-time preprocessing adopted in our image-based experiments, including landmark-based alignment to a 224×224 canonical frame, resizing, and normalization. This step enforced a consistent geometry across frames and made the video-based evaluation comparable with the dataset benchmarks.

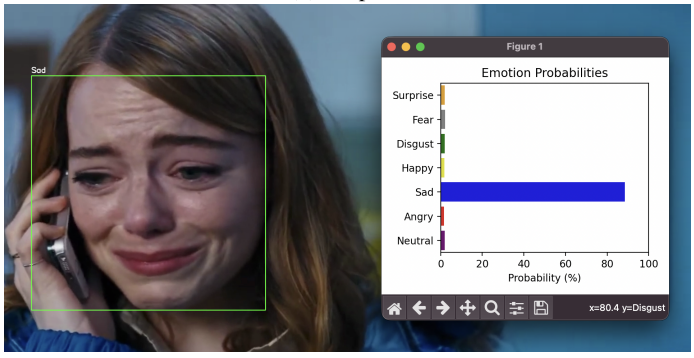
We then performed inference using the trained POSTER-based architecture [2] and loaded the optimized weights from the RAF-DB [4] checkpoints (*Lighting + Random Erasing* configuration). For each processed frame, the network output the logits over the 7 basic expression classes. To provide a more nuanced interpretation of the results, we applied a Softmax activation to convert the model's logits into a normalized



(a) Happy



(b) Surprise



(c) Sad

Fig. 5: Qualitative evaluation using a sequence from *La La Land* (2016). The images show the real-time detection and cropping, while the charts display the Softmax confidence scores for the 7 emotion classes.

(a) Happy (b) Surprise (c) Sad

(d) Happy (e) Surprise (f) Sad

Fig. 6: Qualitative visualization of model attention. The first row shows the original input face images, while the second row reports the corresponding class activation maps.

probability distribution over the classes. The final predicted label is obtained via an $\arg\max$ operation and rendered directly onto the video stream. Simultaneously, a dynamic horizontal bar-plot is generated using the `Seaborn` library to visualize the confidence scores for each emotion class in real-time. This dual visualization allows for a qualitative assessment of the model's certainty and its ability to capture complex or blended emotional states during natural facial movements. The results are shown in Figure 5.

A. Class Activation Maps

To provide an interpretable explanation of our model decisions, we computed class activation maps on test images using a Grad-CAM-style approach. The goal is to localize, in the input face image, the regions that contribute the most to the score of a given expression class, thereby offering a qualitative sanity check that the network focuses on semantically meaningful cues (e.g., mouth, eyebrows, eyes) rather than spurious background patterns.

Operationally, after training we run an additional inference pass in which we selected an internal convolutional representation close to the end of the backbone as *localization layer*. For a chosen class (by default, the predicted class), we computed the gradient of the corresponding logit with respect to the feature maps of this layer. These gradients were spatially aggregated to estimate the importance of each channel, and the final saliency map was obtained by combining the feature maps through these importance weights and retaining only positive evidence (via a ReLU).

The resulting heatmap was resized to the original input resolution and normalized to a fixed range to enable comparisons across images. Finally, we overlayed the heatmap onto the original face image using a color map and alpha blending, obtaining an intuitive visualization of where the model "looks" when producing its prediction.

Across the shown samples, the highlighted regions are predominantly concentrated on semantically meaningful facial areas rather than the background. For *happy* (Figure 6d), the saliency is mainly concentrated at the mouth corners and along the peri-oral region, while also highlighting the widely opened eyes; for *surprise* (Figure 6e) the model focused on the expression wrinkles and surrounding structures in the eye area, which are indicative of widened eyes and eyebrow-related changes. For *sad* (Figure 6f), the saliency shifted toward peri-ocular regions and the mouth corners, matching typical sadness-related deformations. Overall, these visualizations suggested that the network learned to base its decisions on plausible expression-relevant facial cues, supporting the interpretability of the proposed approach.

V. Retrieval

In the retrieval stage, we transformed each input sample into a compact feature representation and then performed a nearest-neighbor search against a reference set. For our setting, the query face crop was preprocessed identically to inference-time inputs, and it was then passed directly to the network. This preprocessing reduced background bias and made the downstream feature space more consistent across frames and subjects.

For each preprocessed face crop x, we forwarded it through our trained model and extracted two outputs: (i) class logits for expression prediction and (ii) an intermediate embedding vector that acted as the retrieval descriptor. Concretely, the model returned

$$(\mathbf{s}, \mathbf{f}) = F(x) \tag{9}$$

Fig. 7: Qualitative retrieval results: given a query face frame (left) and its predicted expression, we retrieved the top-5 nearest neighbors from the memory bank using cosine-similarity search on L_2-normalized embeddings; each neighbor is shown with its ground-truth label and similarity score.

where $\mathbf{s} \in R^C$ denoted the logits over C expression classes and $\mathbf{f} \in R^D$ denoted the learned feature representation used for retrieval.

To enable efficient retrieval, we first constructed a memory bank (gallery) of embeddings by running the model on the full training split with a non-shuffled data loader and saving, for each sample, its feature vector, its label, and its file path. In practice, we extracted the embedding vectors in batches (batch size 32), concatenated them into a single matrix $\mathbf{F} \in R^{N \times D}$, and stored the associated metadata (paths and labels) alongside the index for later lookup.

After computing the gallery embeddings, we applied L_2 normalization and indexed them with FAISS [13] using `IndexFlatIP`, which performed nearest-neighbor search by maximizing the inner product between normalized feature vectors. Because all feature vectors were normalized to unit norm, inner-product search was equivalent to cosine-similarity search; therefore, ranking neighbors by maximum inner product corresponded to retrieving the most similar samples under cosine similarity.

At inference time, given a query embedding $\mathbf{f}_q$, we applied the same L_2 normalization and queried the FAISS index [13] to retrieve the top-k nearest neighbors together with their similarity scores. The retrieved neighbors were used to surface the most similar faces/frames in the dataset and, when needed, to provide supporting evidence for the final prediction by checking whether the neighborhood was consistent with the predicted class.

For each query frame (or image), the retrieval module output a ranked list of the most similar gallery samples together with their similarity scores as shown in Figure 7.

VI. Conclusion

Facial Expression Recognition remains challenging due to large appearance variability and the class imbalance typical of common benchmarks. In this work, we built on the POSTER [2] framework and systematically extended the overall pipeline, analyzing how preprocessing, augmentation, imbalance-mitigation, and parameter-efficient fine-tuning affect robustness and generalization across multiple datasets.

Our results indicate that the impact of imbalance-mitigation is highly dataset-dependent, whereas regularization through data augmentation is generally less disruptive and often improves robustness across settings. We also show that LoRA is a useful parameter-efficient fine-tuning strategy when a lighter model and lower computation are required: although it can yield slightly lower performance than full fine-tuning, it still achieves competitive results. Finally, the interpretability-based evaluation is essential to assess model behavior in unconstrained scenarios, and our qualitative analyses suggest that the model behaves reliably while focusing on meaningful facial regions rather than spurious background cues.

Appendix

To ensure reproducibility, we report below the public URLs used to download the four FER benchmarks employed in our evaluation. All benchmarks were sourced from publicly accessible online releases, and download links are provided here.

- RAF-DB: https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset
- FER2013: https://www.kaggle.com/datasets/msambare/fer2013
- AffectNet: https://www.kaggle.com/datasets/mstjebashazida/affectnet
- CK+: https://www.kaggle.com/datasets/davilsena/ckdataset

References

[1] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE transactions on affective computing*, vol. 13, no. 3, pp. 1195–1215, 2020.

[2] C. Zheng, M. Mendieta, and C. Chen, "Poster: A pyramid cross-fusion transformer network for facial expression recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2023, pp. 3146–3155.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[4] L. Shan and W. Deng, "Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 356–370, 2018.

[5] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *2010 ieee computer society conference on computer vision and pattern recognition-workshops*. IEEE, 2010, pp. 94–101.

[6] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2017.

[7] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *International conference on neural information processing*. Springer, 2013, pp. 117–124.

[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[9] OpenCV, *OpenCV: estimateAffinePartial2D (Function Documentation)*, 2026. [Online]. Available: https://docs.opencv.org/

[10] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

[11] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," 2020.

[12] P. Taunk, V. J. Geddada, P. Priya J, and N. S. Kumar, "Face detection using viola jones with haar cascade," *Test Engineering and Management*, vol. 83, 06 2020.

[13] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.