Lightweight Deep Learning Models for Real-Time Facial Emotion Recognition

*Note: Sub-titles are not captured in Xplore and should not be used

1st Haoran Qin
School of Engineering
Vanderbilt University
Nashville, USA
haoran.qin@vanderbilt.edu

***Abstract*—The rise of deep learning and overparameterized non-linear systems has significantly advanced performance in computer vision tasks such as object detection and classification. Vision models like Convolutional Neural Networks (CNNs), Residual Networks (ResNets), and Vision Transformers have made image classification more efficient and accurate, enabling success in complex applications such as facial emotion detection. However, these improvements often come at the cost of increased computational and memory demands, hindering real-time inference. Recently, model compression has emerged as a promising approach to reduce the complexity of large vision models while retaining their performance. In this paper, we present: (1) an ablation study of state-of-the-art vision models on the RAR-DB facial emotion dataset, (2) an evaluation of model compression techniques and their effectiveness in reducing model size and inference time, and (3) the design of a real-time inference pipeline optimized for minimal latency and resource usage.**

***Index Terms*—emotion detection, computer vision, lightweight models, real-time inference, deep learning, embedded AI**

I. Introduction

Facial Expression Recognition (FER) is a key task in computer vision, with applications in areas such as customer service, user experience, and healthcare. It has been extensively studied using a variety of machine learning techniques, from traditional approaches like Support Vector Machines to modern deep learning architectures. In particular, Convolutional Neural Networks (CNNs) have significantly improved FER accuracy by leveraging translation-invariant layers, thereby reducing both the amount of data needed and the complexity required to model visual features. More recently, Vision Transformers (ViTs) have pushed performance further by capturing long-range dependencies using self-attention mechanisms.

State-of-the-art (SOTA) models such as ResEmoteNet incorporate advanced components—like attention mechanisms and squeeze-and-excitation blocks—into highly specialized architectures for emotion recognition. While these models achieve impressive accuracy, they are also highly overparameterized. This overparameterization helps with optimization and generalization during training but imposes heavy burdens during inference, especially in terms of computational load and memory usage. For instance, the ViT-B/16 model has around 86 million parameters, and ResEmoteNet has roughly 80 million parameters, requiring approximately 320MB of storage.

Such overhead introduces two key challenges: (1) inefficiency during real-time inference due to high computational and memory requirements, and (2) inefficiency in information transfer when deploying models across devices or platforms—especially in bandwidth-limited or edge environments. These issues are further exacerbated when training data is scarce. In many FER applications, especially those involving real-world scenarios or specific populations, acquiring large, high-quality labeled datasets is difficult. As a result, models trained on limited data are more susceptible to overfitting, making the use of massive architectures both inefficient and potentially counterproductive.

In this paper, we focus on addressing the first challenge: improving runtime efficiency without significantly sacrificing accuracy. We begin with an ablation study of existing FER models to evaluate whether complex architectures like ResEmoteNet are necessary for achieving strong performance under data-scarce conditions. We then investigate model compression techniques aimed at reducing computational complexity and memory usage, ultimately enabling real-time inference with lightweight FER systems trained on limited data.

II. Related Work

A. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are foundational to modern image classification tasks, significantly improving both accuracy and data efficiency through the use of convolutional and pooling layers. Traditional neural architectures, such as Multi-Layer Perceptrons (MLPs), are not well-suited for image data because they treat each pixel independently, lacking the inductive bias to recognize local patterns or spatial hierarchies. Moreover, MLPs are sensitive to image translations, rotations, and distortions, making them inefficient and prone to overfitting on image tasks.

CNNs address these limitations by introducing convolutional layers that are translation-equivariant, enabling the network to detect spatially local features such as edges, textures, and shapes regardless of their exact position in the image.

Pooling layers then reduce the spatial resolution, providing translation invariance and improving computational efficiency. These extracted features are typically passed through fully connected layers, combining the representational strength of MLPs with the spatial understanding gained through convolution.

CNNs have demonstrated strong generalization performance even with limited data and minimal augmentation, due to their ability to learn hierarchical representations. Popular CNN architectures include VGGNet, AlexNet, DenseNet, and ResNet [1]. In this paper, we focus on ResNet as a robust baseline for our ablation studies.

ResNet encompasses two key designs in modern CNN networks: residual connections and batch normalization. Batch normalization standardizes the inputs of each layer across a mini-batch, reducing internal covariate shift and accelerating convergence during training. Residual connections allow the network to learn identity mappings, mitigating the vanishing gradient problem and enabling the successful training of very deep networks. These skip connections enable gradients to flow more easily through the network, allowing ResNet to maintain stable performance even with many layers. For these reasons, ResNet serves as an effective and efficient baseline in many image recognition tasks, including FER.

B. Vision Transformers

Transformer networks have revolutionized deep learning by introducing the attention mechanism, which enables models to dynamically weigh different parts of the input. In image classification, this allows the model to focus on semantically relevant regions of an image rather than treating all pixels equally. Vision Transformers (ViTs) [2] extend this concept by dividing images into fixed-size patches, linearly embedding them, and feeding the sequence into a transformer encoder—typically adapted from the BERT architecture. The model comprises several components including patch embedding, layer normalization, multi-head self-attention, residual connections, and fully connected layers, all designed to capture both local and global context in images.

However, a major drawback of Vision Transformers is their model size and computational cost. Due to their multiple attention heads and dense final layers, ViTs are often highly overparameterized. Table I lists a few popular ViT architectures and their parameter counts:

Model	Parameters (M)	Input Patch Size
ViT-B/16	86M	16×16
ViT-L/16	307M	16×16
ViT-H/14	632M	14×14

TABLE I
PARAMETER SIZES OF COMMON VISION TRANSFORMER MODELS

To address the inefficiency of large ViT models, Data-efficient Image Transformers (DeiT) were introduced as a lightweight alternative that can be trained with limited data and achieve competitive accuracy. DeiT incorporates a distillation token that allows knowledge transfer from a pre-trained CNN teacher model without requiring massive datasets. This makes it a promising candidate for FER tasks, where labeled data is often scarce. Table II lists a few populat DeiT architectures and their parameter counts:

Model	Parameters (M)	Input Patch Size
DeiT-Ti	5.7M	16×16
DeiT-S	22M	16×16
DeiT-B	86M	16×16

TABLE II
PARAMETER SIZES OF COMMON DATA-EFFICIENT VISION TRANSFORMER (DEIT) MODELS

In this work, we extensively study DeiT as the foundation for building a lightweight, distilled FER model, due to its robustness, competitive performance on small datasets, and the potential redundancy in its parameterization that can be optimized for runtime efficiency.

C. ResEmoteNet

ResEmoteNet [3] adapts the strengths of ResNet and introduces squeeze-and-excitation (SE) blocks to enhance feature modeling for facial emotion recognition. The architecture begins with a shallow convolutional stem incorporating pooling and batch normalization layers, enabling efficient extraction of low-level facial features. Following initial feature extraction, SE blocks are employed to perform channel-wise recalibration, using a max-pooling operation for spatial downsampling and a sigmoid gating mechanism to generate excitation weights. This allows the network to selectively emphasize informative features critical for emotion classification.

Residual connections are integrated throughout the network to preserve feature integrity and ensure smooth convergence during training. By combining residual learning with adaptive attention mechanisms, ResEmoteNet captures both local and global facial patterns essential for robust emotion recognition.

While ResEmoteNet achieves state-of-the-art performance on facial emotion classification benchmarks, it remains a relatively large model, containing over 80 million parameters. Its high capacity also renders it data-hungry and potentially susceptible to overfitting in scenarios involving limited or noisy datasets. In this work, we evaluate the performance and robustness of ResEmoteNet under small-scale, perturbed training conditions, focusing on its practicality for lightweight and real-time inference applications.

D. Model Compression

The success of large transformer networks has highlighted the growing need for model compression techniques to improve runtime efficiency and storage capacity. In this work, we focus on three primary compression strategies: knowledge distillation, pruning, and quantization.

1) Knowledge Distillation: Knowledge distillation [4], often framed as a teacher-student paradigm, is a state-of-the-art technique for transferring information from a large, high-capacity model to a smaller, compressed model. A pretrained

teacher network generates soft target distributions through its softmax outputs, and the student network is trained to match these outputs using a Kullback-Leibler (KL) divergence loss. By simultaneously optimizing for both ground-truth labels and the teacher's soft assignments, the student model learns to replicate the teacher's behavior despite having a reduced number of parameters.

2) Pruning: Pruning techniques aim to remove redundant or less important weights from a trained network, resulting in a sparser architecture. By eliminating unnecessary connections, pruning reduces both the computational cost and the storage footprint of the model. Pruning can be applied globally across all layers or locally within specific layers, depending on the desired sparsity level and performance trade-offs.

3) Quantization: Quantization [5] reduces model storage size by lowering the numerical precision of model parameters, typically converting 32-bit floating point values into lower-bit representations such as 8-bit integers. This process significantly decreases memory usage and can accelerate inference, particularly on hardware optimized for low-precision operations, while aiming to maintain model accuracy.

In this paper, we employ post-hoc model compression strategies to ResEmoteNet and related architectures, targeting both parameter reduction for efficient execution and overall model size reduction for improved storage efficiency. Our goal is to enable lightweight, real-time emotion recognition without substantial loss of classification performance.

III. Method

Our approach is organized into four main stages: data selection and preprocessing, model ablation and selection, model compression, and real-time pipeline development. All of our methods are implemented with Python's PyTorch, OpenCV (cv2), and Torchvision libraries, using Vanderbilt University's ACCRE High Performance Cloud-Computing Platforms.

A. Data Selection and Preprocessing

We utilize the Real-world Affective Faces Database (RAF-DB) [6], which contains approximately 15,339 facial images labeled with seven basic emotion categories. RAF-DB is robust due to its diverse coverage of different facial expressions, demographics, and lighting conditions.

To simulate the noisy and incomplete data conditions often encountered in real-world applications, we augment the test set with two types of perturbations:

- **Geometric Perturbations:** Random horizontal flips and small-angle rotations are applied to simulate variations in camera angles and facial poses.
- **Noise Perturbations:** Gaussian noise and random image distortions are introduced to simulate sensor noise and environmental artifacts.

We define the generated perturbed test image $\tilde{x}$ as:

$$\tilde{x} = \mathcal{T}(x) + \mathcal{N}(0, \sigma^2), \tag{1}$$

where $\mathcal{T}(\cdot)$ denotes random geometric transformations (flip, rotation), and $\mathcal{N}(0, \sigma^2)$ represents additive Gaussian noise with variance σ^2.

This perturbation strategy allows us to better evaluate model robustness and observe potential overfitting behaviors, particularly in overparameterized architectures.

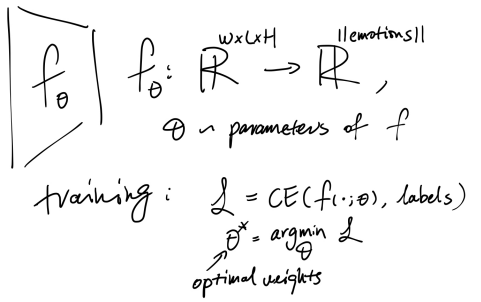

Fig. 1. Pipeline Part A: Model Ablation. For each model f, we train on the RAF-DB dataset using cross-entropy loss between the hard-max of the model's output and the ground-truth labels. We select the best model weights Θ^* for each model and compare their robustness on the perturbed testing set, choosing the best-performing models for the next stage.

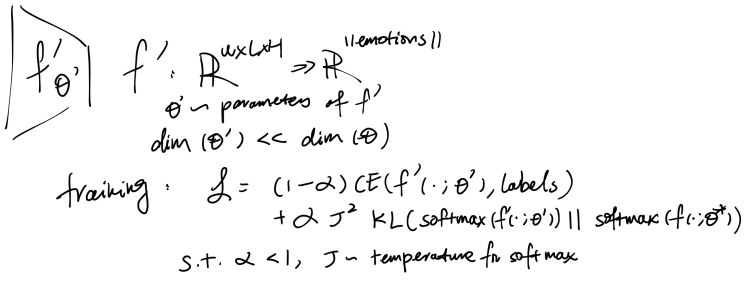

Fig. 2. Pipeline Part B: Model Compression. We define a student model f', a compressed version of the teacher model f, ensuring that the model weights for f' are lower-dimensional than f. During training, we optimize a convex combination of the student's cross-entropy loss for hard assignments and the KL divergence between the soft assignments of teacher and student, weighted by a temperature parameter. We then apply quantization to the student model's weights to further reduce storage complexity. The best-performing compressed model is selected for the final deployment pipeline.

B. Model Ablation and Selection

We first train a baseline Convolutional Neural Network (CNN) consisting of five convolutional layers followed by a fully connected layer, utilizing max-pooling, batch normalization, and dropout for regularization.

We then compare the baseline performance against three deeper architectures:

- ResNet-18 (approximately 11M parameters), trained from scratch and with ImageNet pretraining.
- ResEmoteNet (approximately 80M parameters), the state-of-the-art FER model.
- DeiT-Tiny (approximately 5.7M parameters), a distilled lightweight Vision Transformer.

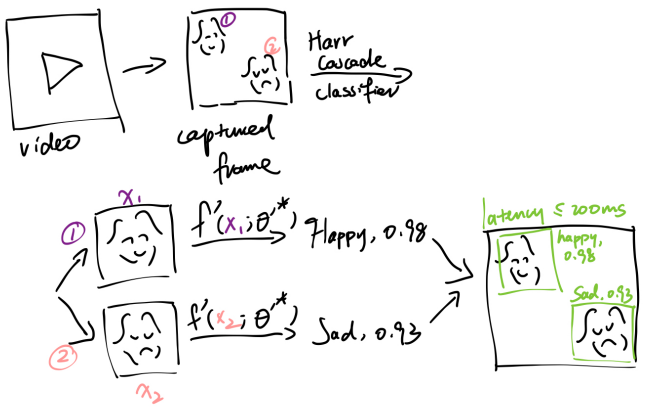

Fig. 3. Pipeline Part C: Real-Time Inference. In the real-time inference pipeline, frames are captured from live video, and faces are detected in each frame using the Haar Cascade Classifier. The best-performing compressed model is then used for real-time emotion inference, with the predicted emotion and confidence displayed back on the frame, achieving an average latency of less than 200 ms.

Training loss, training accuracy, and validation accuracy are recorded to assess performance under limited data conditions, with particular attention to trade-offs between accuracy and computational cost.

C. Model Compression

To further improve runtime and storage efficiency, we explore two compression approaches:

- **Knowledge Distillation:** We train smaller student networks by minimizing both the cross-entropy loss on ground-truth labels and the Kullback-Leibler divergence loss between student and teacher softmax outputs:

$$\mathcal{L}_{\text{total}} = (1-\alpha)\mathcal{L}_{\text{CE}} + \alpha\tau^2\mathcal{L}_{\text{KL}}, \tag{2}$$

  where α controls the balance between objectives and τ denotes the temperature parameter for softening predictions.
- **Pruning and Quantization:** We apply structured pruning to eliminate less important weights, followed by dynamic quantization to reduce precision of model parameters, improving both memory footprint and inference latency.

Post-compression, we evaluate models in terms of parameter count, storage size, and classification performance, selecting the optimal configuration for real-time deployment.

D. Lightweight Model Design

1) Small ResNet: We design a reduced ResNet-20. The architecture includes three groups of residual blocks with progressively increasing channel dimensions ($16 \to 32 \to 64$).

The total number of parameters is computed as follows:

- Initial convolution and batch normalization: 464 parameters.
- Layer1 (three BasicBlocks with 16 channels): 14,016 parameters.
- Layer2 (three BasicBlocks with 32 channels): 55,072 parameters.
- Layer3 (three BasicBlocks with 64 channels): 203,552 parameters.
- Fully connected classification layer: 455 parameters.

Summing across all components yields approximately 273,559, or 0.27M parameters. This compact architecture serves as the student model for knowledge distillation.

2) Small DeiT: We construct a simplified DeiT architecture by removing multi-head attention layers and retaining lightweight feedforward networks (FFNs) for each transformer block. Each FFN consists of a single hidden layer of 192 dimensions followed by a GELU activation.

The approximate parameter count is computed as:

- Patch embedding: $16 \times 16 \times 3 \times 192 = 147,456$ weights (assuming 16×16 patch size).
- Feedforward network: $192 \times 192 = 36,864$ weights.
- Classification head: $192 \times 7 = 1344$ weights.

Including biases and positional encodings, the total parameter count is approximately 0.2M. This model is also trained under the knowledge distillation framework.

These lightweight models are subsequently compressed through pruning and quantization, forming the backbone of our real-time emotion detection system.

E. Real-Time Pipeline Design

Our proposed real-time pipeline consists of three main components: face detection, model inference, and result visualization.

Face Detection: We utilize Haar Cascade classifiers from OpenCV for lightweight and efficient face detection. Haar Cascades are chosen due to their low computational cost, robustness to scale and rotation, and suitability for real-time applications on resource-constrained devices. The detection module continuously processes incoming frames from a camera feed, identifying and cropping regions corresponding to faces.

Model Inference: The cropped face images are forwarded to the distilled and compressed FER model for classification. By leveraging our lightweight ResNet or DeiT models, we ensure fast inference with minimal latency while maintaining high prediction accuracy.

Result Visualization: Predicted emotions and corresponding confidence scores are displayed in real-time. The softmax activation function is applied to model outputs to obtain the probability distribution over emotion classes:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{3}$$

where z_i denotes the output logit for class i and K is the total number of emotion classes. The top predicted label and its associated confidence are overlaid onto the live camera feed.

By integrating compressed FER models with efficient detection and visualization modules, our pipeline achieves high-speed, reliable emotion inference, making it suitable for deployment in real-time embedded AI systems.

IV. Experiments and Results

In this section, we present the experimental design and evaluate the performance of various models. We report both training and validation accuracy, and visualize training dynamics through log-loss and accuracy curves. All models are trained to convergence, typically between 80 and 100 epochs, using the Adam optimizer with a StepLR learning rate scheduler.

A. Data Perturbations

Most state-of-the-art (SOTA) FER methods assume a clean, pre-aligned dataset, such as RAF-DB, for evaluation. However, this assumption rarely holds in real-world applications where datasets are often noisy, distorted, or unrepresentative.

To simulate real-world conditions, we introduce perturbations exclusively to the test set. This contrasts with traditional data augmentation, which enhances the training set. Our perturbation strategy includes:

- **Geometric Perturbations:** Horizontal flips and small rotations to simulate pose variance.
- **Noise Perturbations:** Gaussian noise to simulate poor image quality and sensor artifacts.

For example, Fig. 4 shows a "Disgust" labeled image from RAF-DB, while Fig. 5 displays its perturbed counterpart with horizontal flip and Gaussian noise.

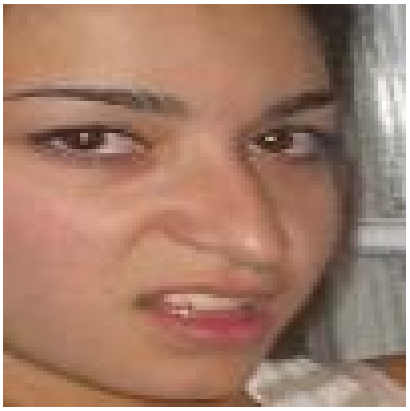

Fig. 4. Original "Disgust" labeled image from RAF-DB

B. Baseline Experiment

The baseline model is a CNN comprising five convolutional layers, batch normalization, dropout regularization, and a final fully connected classification layer. As shown in Figs. 6 and 7, the baseline achieves 46.11% validation accuracy and 44.03% training accuracy, demonstrating limited ability to generalize under noisy test conditions.

Fig. 5. Perturbed image with horizontal flip and Gaussian noise

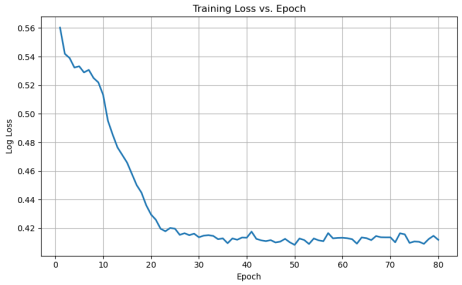

Fig. 6. Log-loss curve (Cross-Entropy) for baseline CNN model

C. Model Benchmarking Experiments

To benchmark performance under perturbed conditions, we evaluate four architectures: ResNet-18 (with and without pretraining), DeiT-Tiny (pretrained), and ResEmoteNet (REN). Training and validation accuracies are summarized in Table III.

TABLE III
Performance comparison across models

Model	# Parameters	Train Acc (%)	Val Acc (%)
ResNet-18	11.71M	84.76	79.79
Pretrained ResNet-18	11.71M	99.09	85.30
Pretrained DeiT-Tiny	5.7M	99.39	84.65
ResEmoteNet	80M	96.59	77.81
REN Baseline (from paper)	80M	–	94.76

Pretraining significantly boosts performance, particularly for deeper networks such as ResNet-18 and DeiT-Tiny. While ResEmoteNet demonstrates strong training accuracy, its validation performance under noisy test conditions, specifically the significant drop of performance from unperturbed dataset, suggests potential overfitting due to its high model capacity. Additionally, all well-performing models exhibit overfitting tendencies, which is expected given the noisy perturbations applied to the test set.

D. Effects of Noise Perturbation on Classification Accuracy

To demonstrate the effect of noise perturbations on final model predictions, we visualize several prediction results with

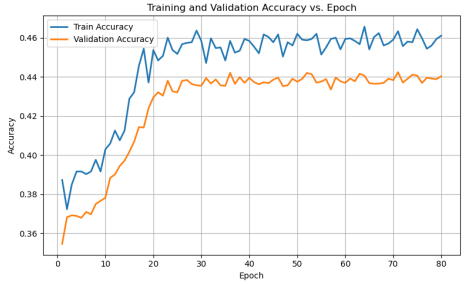

Fig. 7. Training vs. validation accuracy curve for baseline CNN model

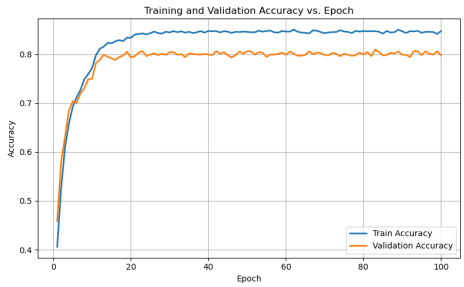

Fig. 8. Training vs. validation accuracy curve for ResNet-18

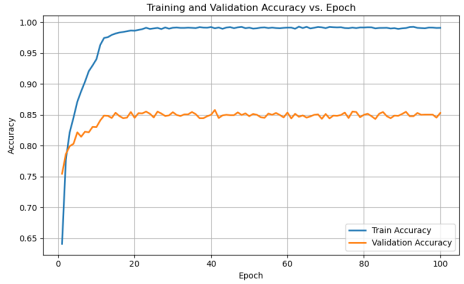

Fig. 9. Training vs. validation accuracy curve for pretrained ResNet-18

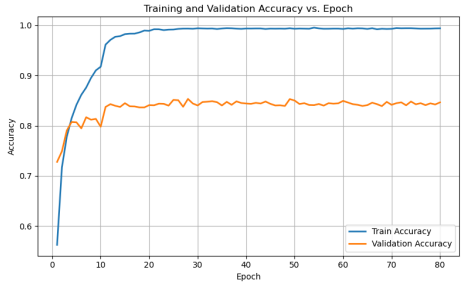

Fig. 10. Training vs. validation accuracy curve for pretrained DeiT-Tiny

and without added noise. Figs. 12 and 13 show the model predictions on a "Happy" image before and after applying small Gaussian noise.

Similarly, Figs. 14 and 15 illustrate the impact of noise, distortion, and rotation on classification outcomes for different images. These examples highlight the vulnerability of overparameterized and overfitted models to even minor perturbations in the input.

E. Model Compression Experiments

The benchmark results indicate that pretrained ResNet-18 and pretrained DeiT-Tiny are the most promising models, achieving high validation accuracy while being substantially overparameterized. This provides a strong motivation for applying knowledge distillation and dynamic quantization to reduce model size and improve efficiency without significant loss of accuracy.

Our compression strategy consists of two stages: (1) knowledge distillation to reduce the number of parameters while retaining predictive performance, and (2) dynamic quantization to further decrease model storage footprint by reducing parameter precision.

1) Knowledge Distillation: We employ the custom-designed Small ResNet, defined in the Methods section, as the distilled student model for ResNet-18. Similarly, we utilize the custom-designed Small DeiT architecture (with attention modules removed) as the distilled student for DeiT-Tiny.

Recall that the total loss $\mathcal{L}_{\text{total}}$ used for distillation is defined as:

$$\mathcal{L}_{\text{total}} = (1-\alpha)\mathcal{L}_{\text{CE}} + \alpha\tau^2\mathcal{L}_{\text{KL}}, \tag{4}$$

where $\mathcal{L}_{\text{CE}}$ denotes the standard cross-entropy loss between predicted and ground-truth labels, and $\mathcal{L}_{\text{KL}}$ denotes the Kullback-Leibler divergence between the student and teacher softmax outputs. The temperature parameter τ softens the probability distributions, while α controls the weighting between the two losses.

Due to limitations in computational resources, extensive hyperparameter tuning of α and τ was not performed; default values were selected to balance soft and hard label learning.

2) Dynamic Quantization: After distillation, we apply dynamic quantization to the student models, converting the majority of parameters from 32-bit floating point (FP32) to 8-bit integer (INT8) precision. This significantly reduces model storage size and inference time, particularly for linear layers, while aiming to preserve predictive accuracy.

3) Compression Results: The results of the compression experiments are summarized in Table IV.

The distilled and quantized models achieve substantial re-

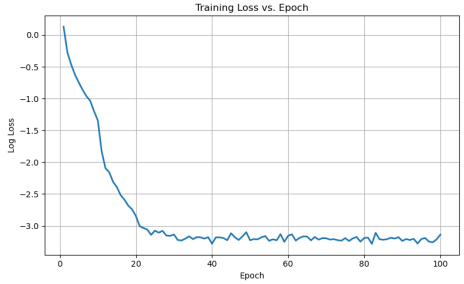

Fig. 11. Log-loss curve (Cross-Entropy) for DeiT

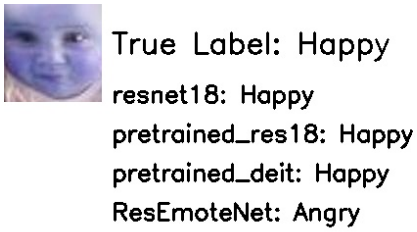

Fig. 12. Model predictions for unperturbed "Happy" image

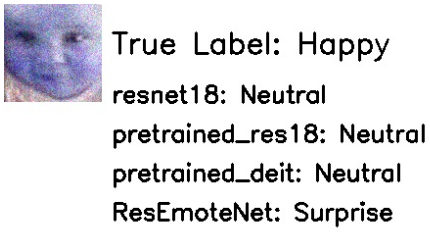

Fig. 13. Model predictions for Gaussian noise-perturbed "Happy" image

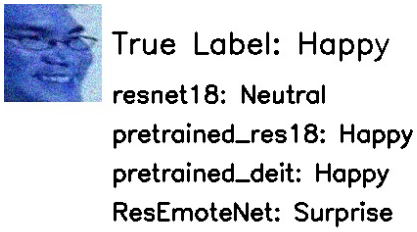

Fig. 14. Effect of noise and distortion on "Happy" prediction labels

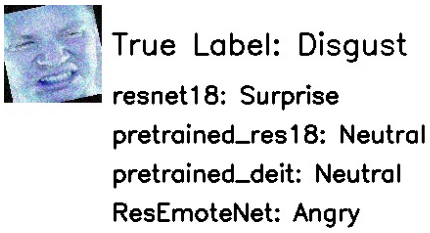

Fig. 15. Effect of distortion and rotation on "Disgust" prediction labels

ductions in both parameter count and storage size. Notably, the distilled DeiT model maintains nearly identical validation accuracy to its full-sized counterpart while achieving over a 20× reduction in model size, demonstrating the effectiveness of combining knowledge distillation and quantization for lightweight real-time FER deployment.

4) Real-Time Pipeline: The real-time pipeline was extensively tested across both human and non-human subjects, including live participants, paintings, and images displayed on walls. Figures 18–21 showcase example outputs of the real-time emotion classification system.

All experiments were conducted under the Vanderbilt VuGuest WiFi network, with the model executed and hosted locally on a MacBook M2 Pro.

Latency measurements reveal an average inference time of approximately 58 milliseconds for single-subject scenarios and around 81 milliseconds when multiple faces were present in the frame. The system demonstrated high classification accuracy in real-world testing, reliably detecting emotional states across varying lighting and environmental conditions. Furthermore, the model successfully identified and classified multiple faces simultaneously with minimal additional latency, highlighting its robustness and suitability for real-time applications.

5) Conclusion: In this work, we demonstrated the feasibility of deploying lightweight, real-time facial emotion recognition systems through a combination of model compression techniques and streamlined model architecture designs. Our results show that compressed models can retain competitive accuracy while significantly improving runtime and storage efficiency, enabling practical deployment in resource-constrained environments.

6) Discussion: **Implications:** The findings suggest that substantial compression of overparameterized models is possible without severely impacting performance, making lightweight FER models viable for real-time applications in edge devices, customer service interfaces, and healthcare monitoring.

Limitations:

- The scope of ablation studies was limited due to computational constraints, preventing exhaustive exploration of hyperparameters and compression strategies.
- Experiments were conducted on a relatively small and synthetic dataset with perturbations, which may not fully capture the diversity and complexity of real-world facial expressions.
- The study only utilized the RAF-DB dataset; generalization to other datasets remains to be evaluated.
- The real-time pipeline has not yet been extensively tested under highly dynamic, crowded, or outdoor conditions, where face detection and classification could be more challenging.

Future Work: Future research could involve expanding the real-time system's testing across larger and more varied datasets, refining the lightweight model architectures through

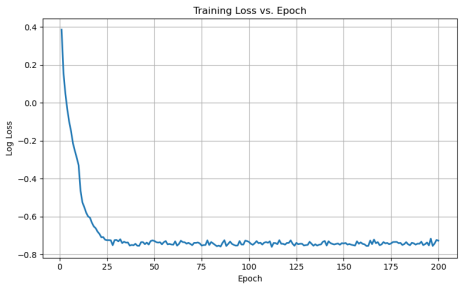

Fig. 16. Small ResNet CE Loss during Distillation

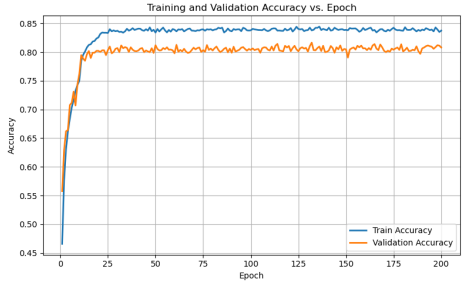

Fig. 17. Train vs. Validation Accuracy during Distillation (Small ResNet)

TABLE IV
MODEL COMPRESSION RESULTS: SIZE AND ACCURACY COMPARISON

Model	# Parameters	Storage Size	Val Acc (%)
Pretrained ResNet-18	11.71M	42.65MB	85.30
Distilled-Quantized ResNet	0.27M	1.03MB	80.08
Pretrained DeiT-Tiny	5.7M	21.08MB	84.65
Distilled-Quantized DeiT	0.2M	0.74MB	84.55

neural architecture search, and optimizing deployment for embedded platforms such as NVIDIA Jetson and Raspberry Pi.

F. Code Availability

The code for model initialization, training procedures, and real-time pipeline implementation is available at https://github.com/niqretnuh/Lightweight_Emotion_Detector.

The dataset has been removed from the repository out of respect for the original authors. Information regarding the RAF-DB dataset can be found at http://www.whdeng.cn/RAF/model1.html#dataset.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[3] A. K. Roy, H. K. Kathania, A. Sharma, A. Dey, and M. S. A. Ansari, "Resemotenet: Bridging accuracy and loss reduction in facial emotion recognition," *IEEE Signal Processing Letters*, vol. 32, pp. 491–495, 2025.

[4] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[5] J. So, J. Lee, D. Ahn, H. Kim, and E. Park, "Temporal dynamic quantization for diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2306.02316

[6] S. Li 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, 2019.

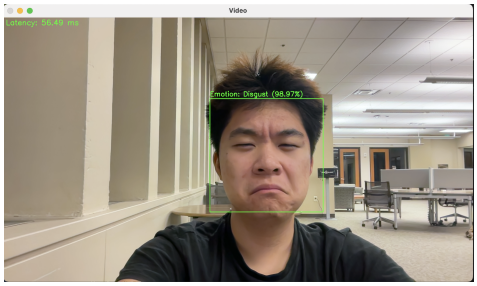

Fig. 18. Human subject emotion classification: "Disgust"

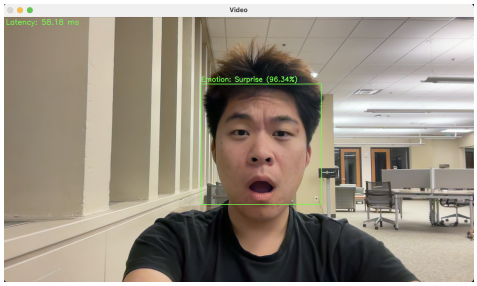

Fig. 19. Human subject emotion classification: "Surprised"

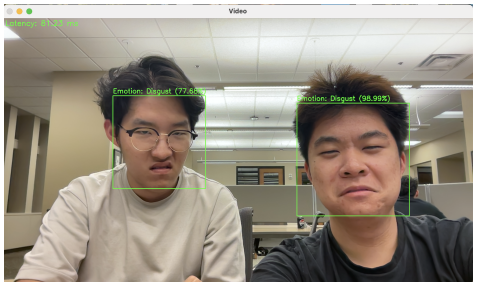

Fig. 20. Multiple human subjects emotion classification

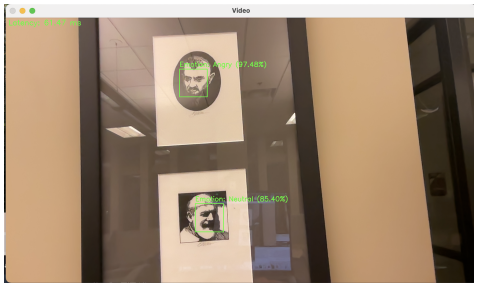

Fig. 21. Emotion classification on non-human subjects (paintings)