Engineering Applications of Artificial Intelligence 136 (2024) 108983

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Benchmarking deep Facial Expression Recognition: An extensive protocol with balanced dataset in the wild

Gianmarco Ipinze Tutuianu [a,1], Yang Liu [b,*,1], Ari Alamäki [a], Janne Kauttonen [a]

[a] Haaga-Helia University of Applied Sciences, Ratapihantie 13, 00520, Helsinki, Finland
[b] University of Oulu, Pentti Kaiteran Katu 1, 90570, Oulu, Finland

ARTICLE INFO

Keywords:
Facial expression recognition
TensorFlow implementation
Deep neural network
Pre-trained model
Practical deployment

ABSTRACT

Facial expression recognition (FER) is crucial in enhancing human-computer interaction. While current FER methods, leveraging various open-source deep learning models and training techniques, have shown promising accuracy and generalizability, their efficacy often diminishes in real-world scenarios that are not extensively studied. Addressing this gap, we introduce a novel in-the-wild balanced testing facial expression dataset designed for cross-domain validation, called BTFER. We rigorously evaluated widely utilized networks and self-designed architectures, adhering to a standardized protocol. Additionally, we explored different configurations, including input resolutions, class balance management, and pre-trained strategies, to ascertain their impact on performance. Through comprehensive testing across three major FER datasets and our in-depth cross-validation, we have ranked these network architectures and formulated a series of practical guidelines for implementing deep learning-based FER solutions in real-life applications. This paper also delves into the ethical considerations, privacy concerns, and regulatory aspects relevant to the deployment of FER technologies in sectors such as marketing, education, entertainment, and healthcare, aiming to foster responsible and effective use. The BTFER dataset and the implementation code are available in Kaggle and Github, respectively.

1. Introduction

Facial expression recognition (FER) allows machines to interpret human emotions using computer vision techniques through images and videos and has diverse applications, including intelligent marketing (Khan et al., 2024), AR/VR devices (Somarathna et al., 2023), mental state measurement (Li et al., 2022), and social interactions (Raamkumar and Yang, 2022). Driven by deep learning, current FER techniques have notably enhanced recognition accuracy and generalization capabilities, leveraging large-scale annotated datasets and advanced computational resources (Ullah et al., 2022)- (Thanathamathee et al., 2023).

Recent research in Facial Expression Recognition (FER) has progressively focused on optimizing model structures and training strategies to enhance accuracy and efficiency. For example, studies have introduced architectures that integrate convolutional neural networks (CNNs) with attention mechanisms to better capture subtle emotional cues from facial expressions (Liu et al., 2020, 2021, 2023). Pan et al. (Liu et al., 2021) advances this further by integrating attention mechanism blocks within a CNN framework, optimizing performance across multiple datasets. MTAC (Liu et al., 2024) enhances model robustness by addressing uncertain expressions through confidence estimation, weighted regularization, auxiliary tasks, and a re-labeling strategy, demonstrating significant improvements over existing methods on various benchmarks. These models underscore a trend towards hybrid architectures that leverage both traditional convolutional layers and transformer mechanisms, aimed at improving the granularity and accuracy of emotion detection across variable real-world conditions.

However, the practical deployment of FER systems still faces significant challenges. Facial expressions in daily life are diverse and mostly unseen for FER models trained on specific datasets due to environmental variations, i.e., illuminations and obstructions, and domain differences, i.e., unseen expressions and demographic factors, often leading to classification errors (Xue et al., 2022; Singh and Kapoor, 2023). Additionally, variations in data quality, attributable to differences in camera specifications or data compression, can severely impair system performance (Lo et al., 2024). Furthermore, imbalances in public FER datasets can skew the representation of certain emotions, thereby exacerbating model overfitting (Liu et al., 2024; Zeng et al.). To address

* Corresponding author.
E-mail address: yang.liu@oulu.fi (Y. Liu).
[1] Equal contribution.

https://doi.org/10.1016/j.engappai.2024.108983
Received 8 November 2023; Received in revised form 15 June 2024; Accepted 12 July 2024
Available online 18 July 2024

these issues, existing literature suggests adopting strategies such as enlarging data collections, incorporating multi-modal inputs, employing data augmentation, and implementing techniques to balance datasets, either individually or in combination (Greco et al., 2023a; Xue et al., 2023).

Despite varying methodologies involving different backbone networks, datasets, and training strategies, identifying the true drivers of performance enhancement remains challenging. While some studies have explored various network architectures, they conclude that certain model families excel on specific datasets, which offers limited practical utility (Greco et al., 2023b; Yang et al., 2021). In response, a detailed protocol needs to be proposed for evaluating practical FER deployment effectively and addressing the following research questions (RQs).

- RQ1: Whether different resolutions exhibit FER performance differences in practical settings?
- RQ2: Whether pre-trained weights increase FER performance in practical settings?
- RQ3: Whether built-in balancing strategies improve FER performance in practical settings?

To address the above RQs, we conduct a rigorous study by establishing a cross-domain benchmark with a uniform experimental protocol and thus provide practical insights for FER applications and guide the development of new models. The contribution of this paper is summarized as follows.

- A new class-balanced in-the-wild facial expression dataset called BTFER, including 2100 images with annotations of seven basic emotions, is collected for cross-domain validation. It significantly enhances the ability to accurately recognize and interpret emotional expressions across diverse demographic groups and varying environments.
- A unified evaluation protocol is designed to systematically evaluate the performance of different FER models, including both publicly available and self-designed network architectures and training strategies, by employing a standardized BTFER dataset which allows for consistent and comparable results.
- Extensive experiments are conducted to investigate three critical research questions using the BTFER dataset and the evaluation protocol, exploring the impact of image resolution, model size, and the efficacy of different model architectures under constrained computational resources, respectively.
- Thorough analysis is undertaken based on robust validation of the models discussed and comprehensive evaluation of their performance across diverse scenarios. The insights derived from it significantly enhance the understanding of FER technologies, offering promising recommendations in terms of future research directions, practical deployments in various domains, and ethical considerations.

2. Balanced Testing of Facial Expression Recognition dataset

To enhance the training of advanced deep models for FER, several datasets have been developed under in-the-wild conditions, including RAF-DB (Li et al., 2017), FER 2013 (Goodfellow et al., 2013), and AffectNet (Mollahosseini et al., 2019), with examples illustrated in Fig. 1.

The RAF-DB dataset comprises facial images captured in real-world scenarios, depicting a broad spectrum of emotions exhibited by individuals in natural settings. It includes a variety of emotions such as anger, disgust, fear, happiness, sadness, surprise, and neutral, with each image measuring 100 $\times$ 100 pixels.

The FER2013 dataset is extensively utilized and contains over 35,000 grayscale images of facial expressions across seven categories: anger, disgust, fear, happiness, sadness, surprise, and neutral. Each image is 48 $\times$ 48 pixels.

The AffectNet, another significant dataset, offers a vast collection of facial expression images from uncontrolled environments, spanning a diverse range of emotions and intensities. It provides detailed annotations, including eight emotional labels, as well as valence, arousal, and dominance ratings. However, the image sizes in this dataset are not standardized.

Despite the availability of these large-scale annotated training datasets, challenges persist due to variations in image size and imbalanced categories. For instance, the RAF-DB dataset contains 887 disgust samples compared to 5957 happiness samples, as detailed in Table 1. Such inherent biases may lead to suboptimal performance in certain classes and pose risks of overfitting. Moreover, it is common for existing studies to employ pre-trained backbones that necessitate resizing images to a uniform resolution, yet few studies have explored the effects of this resizing process on FER accuracy.

In pursuit of a fair comparison of the performance of existing models using in-the-wild facial expression images under practical conditions, we have compiled a novel dataset, termed Balanced Testing of Facial Expression Recognition (BTFER), to enhance our understanding of potential issues such as overfitting, biases, robustness, and generalizability. Specifically, we detail the methodology employed in the development of the BTFER dataset as follows.

- Stage 1: Initially, we utilized the Google API of Images Download Library to assemble a control dataset from a broad collection of internet images (see: https://github.com/hardikvasa/google-images-download). We employed specific search queries like '*white man*

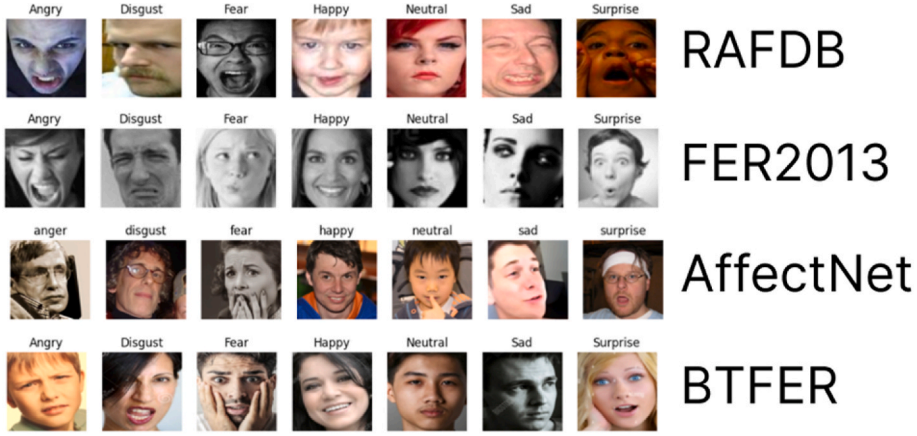

Fig. 1. Sample images for four datasets.

Table 1
Datasets used in this work with descriptions.

Dataset Name	Expressions	No Images per class	No. Total of Images (Dataset)	Resolution
Real-world affective Database (RAF-DB)	Angry	867	15,339	100 × 100
	Disgust	877		
	Fear	355		
	Happy	5957		
	Neutral	3204		
	Sad	2460		
	Surprise	1619		
FER 2013 (Facial Expression Recognition, 2013 Dataset)	Angry	4953	35,887	48 × 48
	Disgust	547		
	Fear	5121		
	Happy	8989		
	Neutral	6198		
	Sad	6077		
	Surprise	4002		
AffectNet	Angry	3353	25,938	512 × 512
	Disgust	2480		
	Fear	3289		
	Happy	4791		
	Neutral	4878		
	Sad	3164		
	Surprise	3983		
BTFER (Our Dataset)	Angry	300	2100	256 × 256
	Disgust	300		
	Fear	300		
	Happy	300		
	Neutral	300		
	Sad	300		
	Surprise	300		

angry', '*Asian men happy*', '*black women sad*', '*Latino women surprised*', and '*Indian kid smile*' to ensure a diverse representation of ethnicities, ages, and genders. The diversity of the dataset is derived from various ethnic groups including White, Black, Asian, Latino, and Indian, as well as gender (i.e., female, male), and age categories (i.e., middle-aged, adult, toddler, child, and elderly). It also encompasses a broad range of facial expressions such as happy, smiling, laughing, sad, crying, sorrowful, fearful, scared, screaming, neutral, serious, angry, surprised, amazed, and disgusted, among others. Additionally, these expressions are listed in Spanish to enhance the search tool's capability to yield a more extensive array of results.

- Stage 2: The images thus gathered underwent a rigorous manual cleaning process to maintain dataset integrity. During this phase, we eliminated non-human faces, irrelevant images, duplicates, and other problematic elements (incl. no faces, emojis, words, etc.) that might introduce bias or noise. In addition, personal identifiers were removed from the dataset to ensure anonymity. Data was anonymized to prevent the identification of individuals from the facial expressions captured.
- Stage 3: Subsequently, the MTCNN (Zhang et al., 2016) and OpenCV are employed to automatically identify and isolate facial regions within the images. This step involved extracting and cropping the detected faces to focus solely on the relevant region of interest. Each cropped facial image was resized to a uniform dimension of 256 × 256 pixels, incorporating an additional 20 pixels around the region of interest to preserve context and maintain facial integrity, thus minimizing alignment discrepancies. A second-round manual clean is applied to remove those images that OpenCV wrongly detects as faces.
- Stage 4: A representative sample consisting of 300 images from each category is randomly selected to construct a testing dataset with seven facial expression classes for cross-validation experiments.

As shown in Table 1, the constructed BTFER dataset consists of 2100 images, with 300 images for each of the seven basic emotions. This balanced size allows for more uniform training and testing, reducing the bias often seen in datasets with uneven class distributions. In addition, the dataset involves samples across various demographics by using specific search queries, in terms of diverse ethnicities, ages, and genders, specifically curated to represent seven basic emotions. Each image in the dataset has been resized to a consistent dimension close to the common input resolution of existing networks to maintain standardization, which helps in reducing preprocessing discrepancies that might arise due to varying image sizes. The balanced nature and the controlled collection method make the BTFER dataset suitable for cross-domain validation. This means the dataset can be more effectively used to test the generalization capabilities of FER models across different backgrounds and environments that reflect real-world conditions. The BTFER dataset will be integral in our practical protocol to evaluate all selected models in subsequent experiments.

3. Proposed FER evaluation protocol and implementation

3.1. Overview

In addition to the BTFER dataset, we propose a detailed protocol for the practical evaluation of various deep FER models. This protocol, which guides model development and testing, is illustrated in Fig. 2. Initially, the protocol involves augmenting a given dataset to facilitate the tuning of model parameters in subsequent steps. Following this, the augmented dataset is input into various network architectures with adjustments made to hyperparameters. Subsequently, specific training settings are determined to conduct advanced evaluations during the model training phase. Ultimately, the efficacy of the trained models is assessed using the BTFER dataset to gauge their practical FER performance. This protocol has been consistently applied across all model architectures and datasets to ensure standardized testing conditions.

3.2. Public network architectures

Following the above protocol, we evaluate a total of twenty-three CNN architectures both on the three public datasets and the collected BTFER dataset, as detailed in Tables 2 and 3. According to recent reviews in the field of FER (Li et al., 2022) and computer vision (Amiri et al., 2024), we select seven widely used CNNs and their variants (i.e., VGG (Simonyan and Zisserman, 2015), MobileNet (Howard et al., 2017), EfficientNet (Tan, 2013), ResNet (He et al., 2016), DenseNet (Huang et al., 2017), and Inception (Szegedy et al., 2015)) and three classical CNNs (i.e., LeNet (LeCun et al., 1998), AlexNet (Krizhevsky et al., 2017), and ZFNet (Zeiler and Fergus, 2014)). Most of these architectures are accessible through the TensorFlow Keras library, while others are documented in related research papers (i.e., RepVGG (Ding et al., 2021), LeNet5 (LeCun et al., 1998), GoogLeNet (Szegedy et al., 2015)). Next, we provide concise descriptions of each CNN family. For further details and Python implementations, please refer to our repository at https://github.com/gipinze/ResearchModels_GIT.

VGG (Simonyan and Zisserman, 2015) is a common CNN architecture used as a backbone of FER models. In this work, **VGG16**, including thirteen convolutional layers interspersed with multiple max-pooling layers and culminated fully connected layers, is applied. In addition, **VGG19**, an expansion of VGG16, is used, which maintains a similar architecture but incorporates additional convolutional layers, thus enhancing representational capacity albeit at a greater computational expense. Moreover, **VGGNet** serves as an evaluated model due to its uniform architecture and deep structure. As a novel variant, **RepVGG** (Ding et al., 2021) is also introduced since its re-parameterized design simplifies the model while maintaining competitive accuracy and enhancing interpretability.

MobileNet (Howard et al., 2017) is a lightweight CNN architecture designed for efficient inference on mobile and edge devices, which is

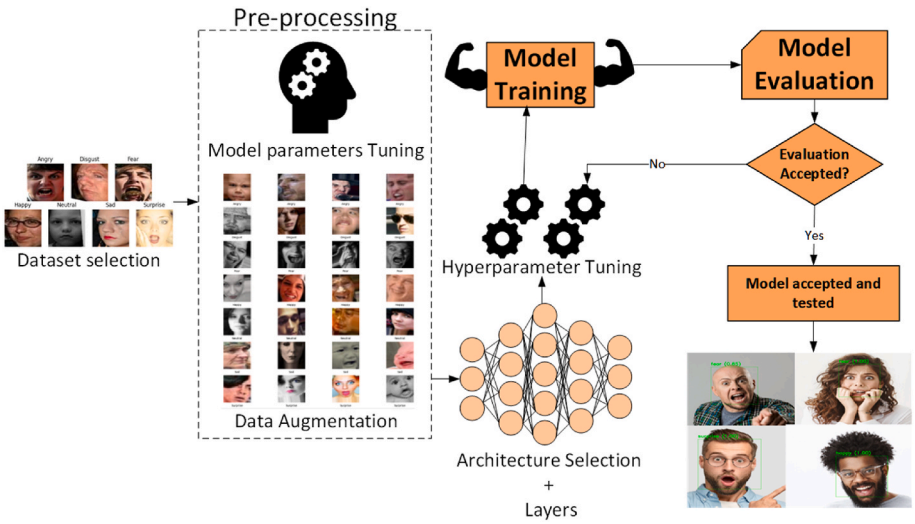

Fig. 2. Pipeline of the evaluation protocol.

involved in our evaluation. It uses depth-wise separable convolutions to minimize computational complexity while preserving accuracy. Besides, we also involve the **MobileNetV2**, which enhances the original version by incorporating inverted residual blocks and linear bottlenecks, boosting both efficiency and performance.

EfficientNet (Tan, 2013) is another CNN model developed for an optimal balance between accuracy and computational efficiency through compound scaling of depth, width, and resolution. Here, we choose the **EfficientNetV2_B0**, which is the base architecture in this family.

ResNet (He et al., 2016) is a commonly used backbone CNN with effective residual connections addressing the vanishing gradient problem. In this work, we use **ResNet50** and **ResNet18** that has 50 layers and 18 layers, respectively. The **ResNet18** provides a lighter architecture for limited computational resources. Notably, we add extra layers to improve the initial performance of the **ResNet18** in our experiments, which are *Dropout* (0.2) + *Denselayer* (1024) with *ReLU* activation and one last *Dropout* layer before the *SoftMax*, to prevent overfitting and achieve higher accuracy. Additionally, a neural architecture search-based CNN optimized for mobile devices, termed **NasNetMobile** (Zoph et al., 2018) is also considered. It utilizes a cell-based structure with skip connections, like the residual blocks, to deliver competitive accuracy with reduced computational complexity.

DenseNet (Huang et al., 2017) is a densely connected CNN architecture that ensures each layer feeds forward to every other layer, enhancing feature reuse, reducing parameter count, and facilitating efficient and accurate model training. Here, we apply the standard **DenseNet121** as one of our evaluation models.

Inception (Szegedy et al., 2015) is a milestone CNN architecture with repeating blocks where the output of a block acts as an input to the next block. We apply the **GoogLeNet** in our study because it features *Inception* modules that allow the network to select from multiple convolutional filter sizes in each block. Furthermore, **InceptionV3** is exploited since it introduces several enhancements including Label Smoothing, factorized 7 × 7 convolutions, and an auxiliary classifier to enhance label propagation throughout the network. Alternatively, **Xception** (Chollet, 2017), named for *Extreme Inception*, adapts the *Inception* architecture by using depth-wise separable convolutions to better capture local and global dependencies.

SeNet (Hu et al., 2020) is introduced because it incorporates squeeze-and-excitation blocks that enable dynamic channel-wise feature recalibration, optimizing network performance.

In addition, a few classical networks are also evaluated as follows. **LeNet5** (LeCun et al., 1998) is an early CNN design primarily developed for handwritten digit recognition and is instrumental in popularizing CNNs. **AlexNet** (Krizhevsky et al., 2017) consists of five convolutional layers, max-pooling layers, and fully connected layers, showcasing the capabilities of deep learning for large-scale image classification tasks. **ZFNet** (Zeiler and Fergus, 2014) comprises convolutional, pooling, and fully connected layers, demonstrating the effectiveness of deep networks for image classification tasks.

3.3. Self-designed network architectures

Apart from the above publicly available networks, we additionally propose several self-designed architectures considering the requirement of practical FER application, i.e., **Sequential 3 Conv**, **Sequential 4 Conv**, **Sequential Simple**, **Ensemble**, **Modular**, and **Single-model-multi-branch (SMMB)**, to enrich the benchmark protocol. The architecture details of each self-designed network are elaborated as follows.

Sequential 3 Conv consists of three main convolutional blocks. Specifically, 1) Block-1 has one *Conv2D* layer (32 filters, 3 × 3 kernel size, *ReLU* activation, and '*same*' padding), another *Conv2D* layer (64 filters, 3 × 3 kernel size, *ReLU* activation), and one *MaxPooling2D* layer (2 × 2 *Pooling*, 0.25 *Dropout*); 2) Block-2 has two more *Conv2D* layers (128 filters, 3 × 3 kernels, *ReLU* activation), and one *MaxPooling2D* layer; 3) Block-3 has one *Conv2D* layer (512 filters, 3 × 3 kernel, *ReLU* activation, and *Batch Normalization*), and one *MaxPooling2D* layer; 4) One *GlobalAveragePooling2D* layer, one *Dense* layer (1024 units, and *ReLU* activation, 0.5 *Dropout*), and one final *Dense* layer (*C* units for class classification, and *SoftMax* activation). Fig. 3 illustrates the layers of the **Sequential 3 Conv** architecture.

Sequential 4 Conv consists of four main convolutional blocks. Specifically, 1) Block-1 has one *Conv2D* layer (32 filters, 3 × 3 kernel size, *ReLU* activation, and '*same*' padding), another *Conv2D* layer (64 filters, 3 × 3 kernel size, *ReLU* activation), and one *MaxPooling2D* layer (2 × 2 *Pooling*, 0.25 *Dropout*); 2) Block-2 has two more *Conv2D* layers (128 filters, 3 × 3 kernels, *ReLU* activation), and one *MaxPooling2D* layer; 3) Block-3 has two additional *Conv2D* layers (256 filters, 3 × 3 kernels, *ReLU* activation), and one *MaxPooling2D* layer; 4) Block-4 has another one *Conv2D* layer (512 filters, 3 × 3 kernels, *ReLU* activation, *Batch Normalization*, 0.2 *Dropout*); 5) One *GlobalAveragePooling2D* layer, one *Dense* layer (1024 units, and *ReLU* activation, 0.5 *Dropout*), and one final *Dense* layer (*C* units for class classification, and *SoftMax* activation). Fig. 4 illustrates the layers of the **Sequential 4 Conv** architecture. The **Sequential 3 Conv** and the **Sequential 4 Conv** consist of three and four convolutional blocks, respectively, each progressively increasing the number of filters and incorporating dropout layers to prevent

Table 2
Performance comparison in models with 48 x 48 resolution images. Bolded values indicate the top-3 values. * = Flattening pooling. IN: pre-trained on ImageNet, VF: pre-trained on VGGFace, SC: training from scratch.

48 × 48			Simple Balancing								Conservative Balancing							
			Unseen Validation Set				Cross-domain BTFER				Unseen Validation Set				Cross-domain BTFER			
Model	PARAM	Dataset	P	R	F1	Acc	P	R	F1	Acc	P	R	sF1	Acc	P	R	F1	Acc
VGG16 - IN	~15M	RAF-DB	0.78	0.77	0.77	0.77	0.51	0.52	0.51	0.52	*0.79*	*0.79*	*0.79*	0.79	*0.52*	*0.52*	*0.50*	0.52
		FER2013	0.68	0.68	0.68	0.68	0.60	0.54	0.51	0.54	**0.68**	**0.68**	**0.68**	**0.68**	**0.59**	**0.54**	**0.51**	**0.54**
		AffectNet	**0.65**	**0.62**	**0.62**	**0.62**	**0.46**	**0.44**	**0.40**	**0.44**	0.65	0.63	0.64	0.63	0.46	0.46	0.44	0.44
VGG16 - VF	~15M	RAF-DB	0.69	0.64	0.66	0.72	0.39	0.37	0.35	0.45	0.75	0.73	0.74	0.75	0.39	0.38	0.37	0.46
		FER2013	0.66	0.66	0.66	0.66	0.55	0.48	0.45	0.48	0.66	0.65	0.65	0.65	0.54	0.50	0.46	0.50
		AffectNet	0.60	0.60	0.60	0.60	0.46	0.46	0.43	0.46	0.61	0.60	0.60	0.60	0.48	0.45	0.43	0.45
VGG16 - SC	~15M	RAF-DB	0.76	0.73	0.74	0.73	0.38	0.38	0.36	0.47	0.77	0.76	0.76	0.76	0.52	0.51	0.49	0.51
		FER2013	0.67	0.67	0.66	0.67	0.61	0.54	0.50	0.54	0.68	0.68	0.68	0.68	0.59	0.52	0.48	0.52
		AffectNet	0.63	0.56	0.57	0.56	0.41	0.41	0.34	0.41	0.61	0.57	0.57	0.57	0.40	0.37	0.33	0.37
VGG19	~20M	RAF-DB	***0.79***	***0.78***	***0.78***	**0.78**	**0.54**	**0.54**	**0.52**	**0.54**	0.80	0.78	0.79	0.78	0.54	0.54	0.52	0.54
		FER2013	**0.68**	**0.68**	**0.68**	**0.68**	**0.61**	**0.55**	**0.53**	**0.55**	0.67	0.67	0.66	0.67	0.55	0.52	0.48	0.52
		AffectNet	0.66	0.63	0.64	0.63	0.45	0.44	0.39	0.44	0.66	0.64	0.64	**0.64**	0.45	0.44	0.40	**0.44**
MobileNet	~6M	RAF-DB	0.72	0.70	0.71	0.72	0.43	0.43	0.41	0.42	0.74	0.72	0.72	0.72	0.43	0.42	0.41	0.42
		FER2013	0.64	0.62	0.63	0.67	0.46	0.41	0.36	0.41	0.64	0.63	0.63	0.63	0.42	0.41	0.38	0.41
		AffectNet	0.60	0.57	0.58	0.60	0.45	0.44	0.44	0.36	0.63	0.62	0.62	0.62	0.35	0.36	0.30	0.36
MobileNetV2	~5M	RAF-DB	0.72	0.71	0.71	0.71	0.45	0.41	0.38	0.41	0.69	0.65	0.66	0.65	0.37	0.37	0.35	0.37
		FER2013	0.61	0.61	0.61	0.61	0.38	0.35	0.32	0.35	0.63	0.62	0.62	0.62	0.47	0.41	0.38	0.41
		AffectNet	0.59	0.57	0.58	0.57	0.47	0.46	0.46	0.46	0.61	0.60	0.60	0.60	0.34	0.34	0.29	0.34
EfficientNetV2_B0	~6M	RAF-DB	0.69	0.61	0.63	0.61	0.40	0.35	0.33	0.35	0.71	0.68	0.68	0.68	0.48	0.43	0.41	0.43
		FER2013	0.61	0.59	0.59	0.59	0.52	0.47	0.42	0.47	0.62	0.61	0.61	0.61	0.48	0.43	0.37	0.43
		AffectNet	0.57	0.53	0.54	0.53	0.39	0.38	0.33	0.38	0.57	0.50	0.51	0.50	0.39	0.38	0.31	0.38
NasNetMobile	~4,5M	RAF-DB	0.70	0.68	0.68	0.68	0.38	0.39	0.34	0.39	0.69	0.67	0.67	0.67	0.38	0.39	0.35	0.39
		FER2013	0.61	0.60	0.60	0.60	0.34	0.30	0.25	0.30	0.62	0.62	0.62	0.63	0.34	0.29	0.25	0.29
		AffectNet	0.51	0.43	0.43	0.43	0.36	0.33	0.29	0.33	0.55	0.50	0.51	0.50	0.34	0.30	0.26	0.38
ResNet50	~24M	RAF-DB	0.72	0.68	0.69	0.68	0.47	0.43	0.42	0.43	0.73	0.72	0.71	0.72	0.46	0.42	0.39	0.42
		FER2013	0.65	0.64	0.64	0.64	0.49	0.39	0.34	0.39	0.65	0.64	0.65	0.64	0.50	0.45	0.41	0.45
		AffectNet	0.60	0.58	0.58	0.58	0.41	0.38	0.33	0.38	0.59	0.55	0.56	0.55	0.35	0.38	0.31	0.38
DenseNet121	~7M	RAF-DB	0.74	0.69	0.70	0.69	0.49	0.42	0.40	0.42	0.75	0.74	0.75	0.74	0.47	0.45	0.43	0.45
		FER2013	0.65	0.64	0.64	0.64	0.57	0.47	0.42	0.47	0.66	0.66	0.66	0.66	0.51	0.45	0.41	0.45
		AffectNet	0.62	0.57	0.58	0.57	0.50	0.48	0.47	0.48	0.63	0.59	0.60	0.36	0.53	0.52	0.52	0.29
AlexNet*	~21,5M	RAF-DB	0.61	0.53	0.56	0.53	0.36	0.32	0.30	0.32	0.65	0.63	0.63	0.63	0.39	0.38	0.36	0.38
		FER2013	0.49	0.48	0.47	0.47	0.42	0.43	0.39	0.43	0.60	0.59	0.58	0.59	0.61	0.50	0.44	0.50
		AffectNet	0.51	0.50	0.49	0.50	0.33	0.33	0.29	0.43	0.57	0.50	0.50	0.50	0.38	0.36	0.29	0.38
Ensemble	~2M	RAF-DB	0.70	0.64	0.66	0.64	0.44	0.40	0.39	0.40	0.70	0.66	0.66	0.66	0.44	0.40	0.36	0.40
		FER2013	0.62	0.61	0.59	0.61	0.45	0.35	0.29	0.35	0.64	0.63	0.62	0.63	0.62	0.48	0.45	0.41
		AffectNet	0.66	0.63	0.63	0.63	0.45	0.41	0.36	0.41	0.66	0.63	0.62	0.63	0.43	0.41	0.41	0.41
Sequential 3Conv	~1,3M	RAF-DB	0.67	0.59	0.61	0.59	0.43	0.37	0.36	0.37	0.75	0.72	0.72	0.72	0.50	0.46	0.44	0.46
		FER2013	0.63	0.60	0.60	0.60	0.58	0.49	0.46	0.49	0.66	0.66	0.66	0.66	0.61	0.53	0.51	0.53
		AffectNet	0.65	0.60	0.61	0.60	0.41	0.39	0.32	0.39	0.65	0.59	0.60	0.59	0.44	0.41	0.34	0.41
Sequential 4Conv*	~3M	RAF-DB	0.72	0.65	0.66	0.65	0.50	0.47	0.47	0.47	0.78	0.76	0.77	0.76	0.57	0.54	0.51	0.54
		FER2013	0.62	0.61	0.60	0.61	0.54	0.52	0.50	0.52	0.67	0.66	0.66	0.66	0.62	0.48	0.45	0.48
		AffectNet	0.64	0.59	0.60	0.59	0.37	0.38	0.31	0.38	0.65	0.61	0.61	0.61	0.41	0.40	0.34	0.40
Modular	~5M	RAF-DB	0.71	0.63	0.65	0.63	0.43	0.41	0.40	0.41	0.74	0.72	0.72	0.72	0.47	0.45	0.43	0.45
		FER2013	0.63	0.60	0.60	0.60	0.58	0.49	0.46	0.49	0.65	0.65	0.65	0.65	0.60	0.53	0.49	0.53
		AffectNet	0.62	0.56	0.58	0.56	0.37	0.39	0.32	0.39	0.60	0.56	0.57	0.56	0.40	0.38	0.31	0.38
RepVGG	~26,5M	RAF-DB	0.68	0.60	0.62	0.60	0.38	0.32	0.30	0.32	0.74	0.72	0.72	0.72	0.47	0.46	0.44	0.46
		FER2013	0.65	0.65	0.64	0.65	0.56	0.38	0.34	0.38	0.64	0.64	0.64	0.65	0.40	0.30	0.27	0.20
		AffectNet	0.62	0.58	0.59	0.58	0.39	0.39	0.33	0.39	0.62	0.58	0.59	0.58	0.41	0.42	0.36	0.42
Sequential Simple*	~2.5	RAF-DB	0.75	0.67	0.70	0.67	0.50	0.48	0.48	0.48	0.76	0.75	0.75	0.75	0.51	0.50	0.48	0.50
		FER2013	0.62	0.61	0.60	0.61	0.59	0.51	0.47	0.51	0.66	0.66	0.66	0.66	0.64	0.55	0.51	0.55
		AffectNet	0.62	0.58	0.59	0.53	0.39	0.39	0.33	0.40	0.57	0.46	0.46	0.46	0.36	0.37	0.30	0.37

(*continued on next page*)

Table 2 (*continued*)

48 × 48			Simple Balancing							Conservative Balancing						
			Unseen Validation Set				Cross-domain BTFER			Unseen Validation Set				Cross-domain BTFER		
Model	PARAM	Dataset	P	R	F1	Acc	P	R	F1	Acc	P	R	sF1	Acc	P	R	F1	Acc
SMMB	~5,5M	RAF-DB	0.73	0.66	0.68	0.66	0.39	0.37	0.36	0.37	0.73	0.69	0.70	0.69	0.50	0.46	0.44	0.46
		FER2013	0.61	0.52	0.53	0.52	0.58	0.49	0.45	0.49	0.64	0.63	0.63	0.63	0.57	0.51	0.47	0.51
		AffectNet	0.58	0.52	0.53	0.52	0.39	0.39	0.33	0.39	0.52	0.46	0.47	0.46	0.37	0.34	0.30	0.34
VGGNet	~15M	RAF-DB	0.67	0.61	0.63	0.61	0.41	0.38	0.37	0.38	*0.78*	*0.77*	*0.77*	0.77	*0.53*	*0.51*	*0.49*	0.51
		FER2013	0.58	0.58	0.58	0.58	0.39	0.31	0.28	0.31	0.67	0.66	0.66	0.66	0.59	0.54	0.52	0.54
		AffectNet	0.53	0.47	0.47	0.47	0.32	0.32	0.25	0.37	0.55	0.52	0.52	0.52	0.32	0.33	0.26	0.33
LeNet5*	~0,2M	RAF-DB	0.62	0.53	0.56	0.53	0.40	0.35	0.35	0.35	0.66	0.65	0.65	0.65	0.42	0.37	0.36	0.37
		FER2013	0.50	0.48	0.48	0.48	0.45	0.45	0.43	0.45	0.53	0.53	0.52	0.53	0.42	0.40	0.37	0.40
		AffectNet	0.55	0.53	0.53	0.53	0.32	0.33	0.28	0.33	0.55	0.49	0.50	0.49	0.33	0.32	0.28	0.32
ResNet18 + Dropout + Dense	~12M	RAF-DB	0.77	0.69	0.71	0.74	0.54	0.52	0.52	0.54	***0.82***	***0.81***	***0.81***	***0.81***	***0.61***	***0.60***	***0.58***	**0.56**
		FER2013	0.68	0.67	0.68	0.67	0.55	0.49	0.45	0.49	0.68	0.68	0.68	0.68	0.60	0.53	0.49	0.53
		AffectNet	0.63	0.57	0.56	0.57	0.40	0.40	0.32	0.41	**0.71**	**0.71**	**0.71**	**0.57**	**0.62**	**0.61**	**0.60**	**0.44**
ZFNET	~21,5M	RAF-DB	0.70	0.67	0.68	0.61	0.42	0.39	0.37	0.40	0.70	0.67	0.68	0.67	0.42	0.39	0.37	0.39
		FER2013	0.57	0.57	0.56	0.56	0.53	0.49	0.45	0.51	0.57	0.56	0.56	0.61	0.58	0.51	0.48	0.52
		AffectNet	0.53	0.56	0.55	0.55	0.35	0.37	0.36	0.36	0.70	0.67	0.68	0.55	0.42	0.39	0.37	0.37
GoogLeNet	~6M	RAF-DB	0.73	0.69	0.70	0.71	0.50	0.47	0.47	0.50	0.74	0.72	0.72	0.74	0.54	0.52	0.50	0.55
		FER2013	0.65	0.64	0.65	0.65	0.54	0.43	0.41	0.54	0.66	0.66	0.66	0.66	0.62	0.55	0.52	0.55
		AffectNet	0.59	0.55	0.51	0.53	0.43	0.40	0.32	0.42	0.63	0.58	0.59	0.56	0.48	0.43	0.37	0.44

overfitting. This contrasts with traditional architectures like VGG, which use uniform blocks without such progressive complexity or dropout regularization.

Sequential Simple consists of two basic convolutional blocks. Specifically, 1) Block-1 has one *Conv2D* layer (32 filters, 3 × 3 kernel size, *ReLU* activation, and input shape of (Li et al., 2024; Li et al., 2024; Li et al., 2022)), another *Conv2D* layer (64 filters, 3 × 3 kernel size, *ReLU* activation), and one *MaxPooling2D* layer (2 × 2 *Pooling*, 0.1 *Dropout*); 2) Block-2 has one *Conv2D* layer (128 filters, 3 × 3 kernel size, and *ReLU* activation), one *MaxPooling2D* layer, another *Conv2D* layer (64 filters, 3 × 3 kernel size, and *ReLU* activation), and one *MaxPooling2D* layer; 3) One *Flatten* layer, one *Dense* layer (512 units, and *ReLU* activation, 0.2 *Dropout*), and one final *Dense* layer (*C* units for class classification, and *SoftMax* activation). Fig. 5 illustrates the layers of the **Sequential Simple** architecture. Designed for computational efficiency, this architecture uses fewer convolutional layers with lower filter counts, making it suitable for deployment in resource-constrained environments, unlike heavier models like ResNet or Inception.

Ensemble comprises three CNNs, each following a similar stack of convolutional layers. Specifically, 1) CNN-1 has three *Conv2D* layers (32/64/128 filters, 3 × 3 kernel size, *Batch Normalization*, and *ReLU* activation), double *MaxPooling2D* layer (2 × 2 *Pooling*, 0.25 *Dropout*), one *GlobalAveragePooling2D* layer, and *Dense* layer (512 units, and 0.25 *Dropout*); 2) The output of CNN-1 is passed through a *Dense* layer with *SoftMax* activation for classification, while CNN-2 and CNN-3 follow similar patterns with variations in convolutional layers; 3) The outputs of all three models are averaged to produce the ensemble prediction. An ensemble model combines the predictions of multiple individual networks with diverse architectures or parameters, which can learn features from different perspectives to improve the overall performance. Fig. 6 illustrates the layers of the **Ensemble** architecture. This model combines predictions from multiple individual networks with diverse architectures, enhancing performance by leveraging different feature extraction capabilities, which contrasts with single-model architectures, offering improved robustness and accuracy.

Modular consists of a local feature sub-network and a global feature sub-network. Specifically, 1) The local feature sub-network includes three *Conv2D* layers, three *MaxPooling2D* layers, two *Flatten* layers, and two *Dropout* layers. These layers extract local features from the input images; 2) The global feature sub-network is similar but uses larger kernel sizes in the convolutional layers; 3) The outputs of the two sub-networks are then concatenated. The model has a final *Dense* layer with *SoftMax* activation for multi-class classification. Overall, the model is composed of modular building blocks with a total of 24 layers, allowing flexibility in designing and incorporating different components or subnetworks for specific tasks or domains. It offers a modular and customizable approach to constructing new networks. Fig. 7 illustrates the layers of the **Modular** architecture. It incorporates both local and global feature extraction sub-networks, allowing for a more detailed and comprehensive analysis of facial expressions. This modular approach offers flexibility and customization for specific tasks, differing from monolithic architectures like AlexNet or VGG.

SMMB consists of a single network with multiple branches, where each branch specializes in extracting features or making predictions for specific subtasks or classes. It enables the network to learn diverse representations and perform multiple tasks simultaneously. Our Single model multi-branch has two branches: face detection and emotion recognition. Specifically, 1) The shared convolutional block includes three *Conv2D* layers (32/64/64 filters, 3 × 3 kernel size, and *ReLU* activation); 2) The face detection branch has one *MaxPooling2D* layer, one *Conv2D* layer (128 filters), one *Flatten* layer, and *Dense* layer (64 units); 3) The emotion recognition branch has one *Conv2D* (128 filters), one *MaxPooling2D* layer, another *Conv2D* layer (256 filters), one *Multi-head Attention* layer, one *Layer Normalization*, one *Dense* layer (128 units, 0.5 *Dropout*), and one *Flatten* layer; 4) Both branches are merged using the concatenate layer followed by one final *Dense* layer (*C* units for class

Table 3
Performance comparison in models with 224 x 224 resolution images. Bolded values indicate the top-3 models. * = Flattening pooling. IN: pre-trained on ImageNet, VF: pre-trained on VGGFace, SC: training from scratch.

224 × 224			Simple Balancing								Conservative Balancing							
			Unseen Validation Set				Cross-domain BTFER				Unseen Validation Set				Cross-domain BTFER			
Model	PARAM	Dataset	P	R	F1	Acc	P	R	F1	A	P	R	F1	Acc	P	R	F1	Acc
VGG-16 - IN	~15M	RAF-DB	0.83	0.82	0.82	0.82	0.62	0.59	0.59	0.59	0.83	0.83	0.83	0.83	0.59	0.56	0.54	0.56
		FER2013	0.68	0.67	0.67	0.67	0.61	0.54	0.49	0.54	0.69	0.69	0.69	0.69	0.62	0.57	0.54	0.57
		AffectNet	0.71	0.71	0.71	0.71	0.64	0.60	0.58	0.60	0.40	0.69	0.69	0.69	0.64	0.62	0.62	0.62
VGG-16 - VF	~15M	RAF-DB	0.84	0.83	0.84	0.83	0.61	0.60	0.58	0.60	0.85	0.85	0.85	0.85	0.62	0.60	0.59	0.60
		FER2013	**0.77**	**0.72**	**0.73**	**0.71**	**0.54**	**0.54**	**0.52**	**0.58**	0.68	0.69	0.69	0.72	0.59	0.57	0.56	0.56
		AffectNet	0.70	0.71	0.70	0.71	0.68	0.63	0.62	0.63	0.70	0.70	0.70	0.70	0.69	0.66	0.66	0.66
VGG-16 - SC	~15M	RAF-DB	0.82	0.81	0.81	0.82	0.59	0.57	0.55	0.56	0.82	0.82	0.82	0.82	0.58	0.56	0.54	0.56
		FER2013	0.70	0.67	0.69	0.69	0.60	0.57	0.56	0.56	0.70	0.69	0.69	0.69	0.64	0.56	0.52	0.56
		AffectNet	0.71	0.71	0.71	0.71	0.62	0.60	0.59	0.60	0.71	0.71	0.71	0.71	0.60	0.58	0.58	0.58
VGG-19	~20M	RAF-DB	0.80	0.80	0.80	0.81	0.56	0.54	0.52	0.59	0.82	0.82	0.82	0.82	0.59	0.57	0.55	0.57
		FER2013	0.69	0.68	0.68	0.68	0.60	0.55	0.53	0.55	0.70	0.69	0.69	0.69	0.65	0.57	0.54	0.57
		AffectNet	0.72	0.72	0.72	0.72	0.65	0.63	0.62	0.63	**0.72**	**0.72**	0.72	0.72	**0.60**	**0.56**	0.54	0.56
ResNet50 - IN	~24M	RAF-DB	0.83	0.82	0.82	0.82	0.61	0.56	0.54	0.56	0.82	0.82	0.82	0.82	0.59	0.57	0.59	0.59
		FER2013	0.69	0.72	0.68	0.68	0.57	0.55	0.50	0.50	0.69	0.69	0.69	0.69	0.62	0.52	0.48	0.52
		AffectNet	0.72	0.71	0.71	0.71	0.60	0.56	0.55	0.56	0.70	0.70	0.70	0.70	0.57	0.55	0.55	0.55
ResNet50 - VF	~24M	RAF-DB	**0.86**	**0.86**	**0.86**	**0.86**	**0.64**	**0.62**	**0.61**	**0.62**	**0.87**	**0.87**	**0.87**	**0.87**	**0.63**	**0.60**	**0.58**	**0.60**
		FER2013	0.71	0.70	0.70	0.70	0.61	0.56	0.54	0.56	0.71	0.71	0.71	0.71	0.66	0.61	0.58	0.61
		AffectNet	0.66	0.65	0.64	0.71	0.67	0.48	0.41	0.56	0.71	0.72	0.71	0.72	0.67	0.61	0.59	0.61
MobileNet	~6M	RAF-DB	0.80	0.78	0.79	0.78	0.56	0.52	0.48	0.52	0.81	0.81	0.81	0.81	0.56	0.54	0.51	0.54
		FER2013	0.67	0.67	0.67	0.67	0.52	0.48	0.44	0.48	0.70	0.70	0.70	0.70	0.58	0.52	0.49	0.52
		AffectNet	0.73	0.71	0.71	0.71	0.61	0.56	0.55	0.56	0.72	0.71	0.71	0.71	0.60	0.54	0.54	0.54
MobileNetV2	~5M	RAF-DB	0.79	0.78	0.78	0.78	0.56	0.53	0.51	0.53	0.78	0.77	0.77	0.77	0.58	0.51	0.48	0.51
		FER2013	0.66	0.66	0.66	0.66	0.54	0.42	0.38	0.42	0.68	0.68	0.68	0.68	0.58	0.53	0.50	0.53
		AffectNet	0.70	0.70	0.70	0.70	0.60	0.56	0.56	0.56	0.69	0.69	0.68	0.69	0.60	0.55	0.55	0.55
Xception	~21.5M	RAF-DB	0.81	0.80	0.81	0.80	0.58	0.54	0.52	0.54	0.81	0.81	0.81	0.81	0.60	0.52	0.48	0.52
		FER2013	0.70	0.70	0.70	0.70	0.56	0.49	0.45	0.49	0.71	0.71	0.70	0.71	0.59	0.53	0.49	0.53
		AffectNet	0.72	0.71	0.71	0.71	0.62	0.61	0.6	0.61	0.72	0.72	0.72	0.72	0.62	0.61	0.61	0.61
SeNet50	~27M	RAF-DB	0.85	0.85	0.84	0.85	0.62	0.59	0.57	0.59	0.86	0.86	0.86	0.86	0.62	0.59	0.57	0.59
		FER2013	0.70	0.69	0.69	0.69	0.61	0.54	0.51	0.54	**0.71**	**0.71**	**0.71**	**0.71**	**0.66**	**0.62**	**0.60**	**0.62**
		AffectNet	**0.74**	**0.74**	**0.74**	**0.74**	**0.65**	**0.60**	**0.58**	**0.60**	0.71	0.69	0.69	0.69	0.63	0.61	0.61	0.61
EfficientNetV2B0	~6M	RAF-DB	0.74	0.71	0.72	0.71	0.57	0.57	0.55	0.57	0.72	0.71	0.71	0.71	0.60	0.58	0.55	0.58
		FER2013	0.56	0.54	0.53	0.54	0.42	0.41	0.33	0.41	0.68	0.66	0.66	0.66	0.51	0.42	0.36	0.42
		AffectNet	0.72	0.72	0.72	0.72	0.60	0.58	0.57	0.58	0.70	0.69	0.69	0.69	0.58	0.56	0.55	0.56
NasNetMobile	~5M	RAF-DB	0.75	0.75	0.74	0.75	0.53	0.48	0.45	0.48	0.77	0.76	0.76	0.76	0.55	0.49	0.45	0.49
		FER2013	0.66	0.70	0.67	0.67	0.49	0.48	0.49	0.49	0.68	0.67	0.67	0.67	0.54	0.49	0.45	0.49
		AffectNet	0.70	0.69	0.69	0.69	0.59	0.57	0.57	0.57	0.69	0.68	0.67	0.68	0.59	0.55	0.53	0.55
InceptionV3	~26M	RAF-DB	0.80	0.80	0.80	0.80	0.57	0.52	0.50	0.52	0.81	0.82	0.81	0.82	0.57	0.54	0.52	0.54
		FER2013	0.73	0.69	0.69	0.69	0.51	0.40	0.35	0.40	0.70	0.70	0.70	0.70	0.52	0.49	0.43	0.49
		AffectNet	0.71	0.71	0.71	0.71	0.61	0.59	0.59	0.59	0.72	0.72	0.72	0.72	0.61	0.60	0.60	0.60
DenseNet121	~7M	RAF-DB	0.82	0.82	0.82	0.82	0.60	0.55	0.54	0.55	0.83	0.82	0.82	0.82	0.62	0.60	0.58	0.60
		FER2013	0.69	0.69	0.69	0.69	0.59	0.50	0.45	0.50	0.40	0.70	0.70	0.70	0.44	0.52	0.46	0.52
		AffectNet	0.70	0.72	0.71	0.71	0.64	0.61	0.61	0.61	**0.72**	**0.72**	**0.72**	**0.72**	**0.60**	**0.55**	**0.61**	**0.61**
RepVGG	~34M	RAF-DB	0.79	0.78	0.79	0.78	0.56	0.56	0.54	0.56	0.81	0.81	0.81	0.81	0.56	0.55	0.54	0.55
		FER2013	0.70	0.67	0.69	0.69	0.60	0.44	0.43	0.44	0.70	0.69	0.69	0.69	0.59	0.47	0.43	0.47
		AffectNet	0.71	0.70	0.71	0.70	0.62	0.61	0.61	0.61	0.72	0.71	0.71	0.71	0.64	0.62	0.61	0.62
Sequential Simple	~22M	RAF-DB	0.66	0.61	0.63	0.61	0.45	0.43	0.43	0.43	0.71	0.69	0.69	0.69	0.50	0.47	0.46	0.47
		FER2013	0.60	0.60	0.60	0.60	0.57	0.48	0.43	0.48	0.63	0.62	0.62	0.62	0.45	0.36	0.31	0.36
		AffectNet	0.66	0.66	0.66	0.66	0.54	0.52	0.52	0.52	0.66	0.66	0.66	0.66	0.54	0.52	0.52	0.53
VGGNet	~12M	RAF-DB	0.81	0.77	0.78	0.77	0.56	0.55	0.54	0.55	0.83	0.82	0.82	0.82	0.59	0.57	0.55	0.57
		FER2013	0.67	0.67	0.67	0.67	0.66	0.58	0.58	0.58	0.70	0.70	0.70	0.70	0.65	0.60	0.58	0.60
		AffectNet	0.70	0.69	0.69	0.69	0.58	0.57	0.56	0.57	0.70	0.70	0.70	0.70	0.58	0.56	0.55	0.56
ResNet18	~6M	RAF-DB	0.78	0.75	0.76	0.77	0.55	0.53	0.52	0.53	0.80	0.81	0.82	0.82	0.56	0.56	0.56	0.56
		FER2013	0.67	0.67	0.67	0.67	0.61	0.54	0.51	0.54	0.70	0.70	0.70	0.70	0.68	0.63	0.60	0.63
		AffectNet	0.70	0.68	0.68	0.68	0.59	0.56	0.57	0.56	0.69	0.69	0.68	0.69	0.62	0.56	0.53	0.56
GoogLeNet	~89M	RAF-DB	0.76	0.75	0.76	0.75	0.60	0.60	0.59	0.60	0.79	0.79	0.79	0.79	0.58	0.58	0.55	0.58
		FER2013	0.68	0.68	0.68	0.68	0.68	0.63	0.60	0.63	0.69	0.68	0.68	0.68	0.68	0.62	0.59	0.62
		AffectNet	0.7	0.69	0.69	0.69	0.59	0.58	0.58	0.58	0.70	0.69	0.7	0.69	0.61	0.61	0.61	0.61
Modular	~103M	RAF-DB	0.75	0.73	0.74	0.66	0.53	0.50	0.48	0.48	0.75	0.73	0.74	0.73	0.53	0.50	0.48	0.50
		FER2013	0.59	0.58	0.58	0.58	0.52	0.49	0.42	0.49	0.64	0.64	0.64	0.64	0.58	0.53	0.48	0.53
		AffectNet	0.68	0.67	0.67	0.67	0.56	0.56	0.56	0.56	0.67	0.66	0.67	0.66	0.55	0.54	0.54	0.54

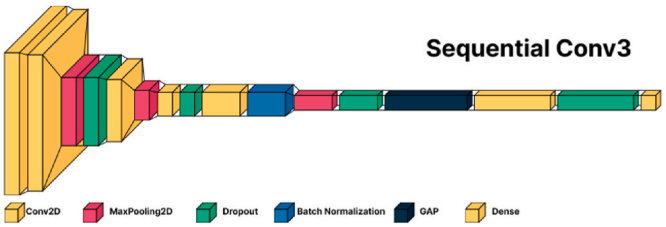

Fig. 3. Architecture of sequential 3 Conv.

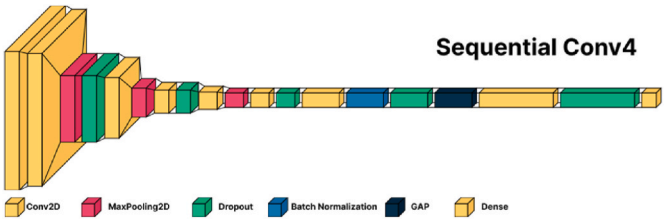

Fig. 4. Architecture of sequential 4 Conv.

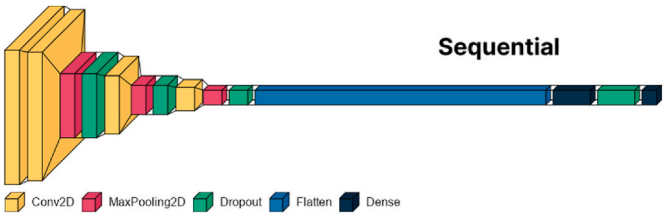

Fig. 5. Architecture of sequential simple.

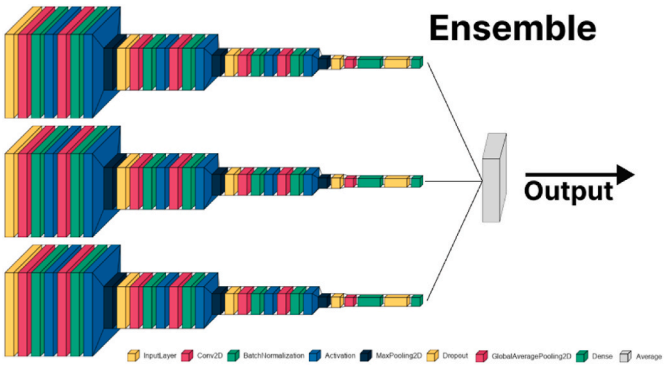

Fig. 6. Architecture of ensemble.

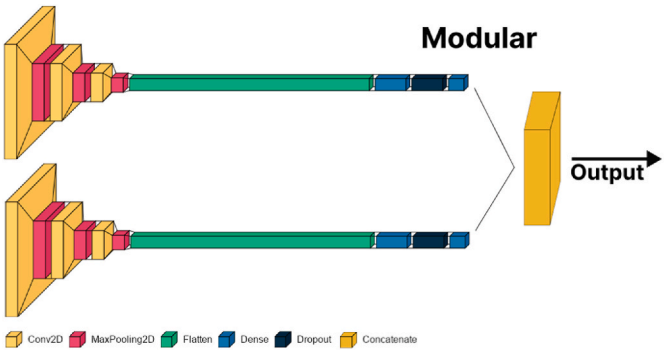

Fig. 7. Architecture of modular.

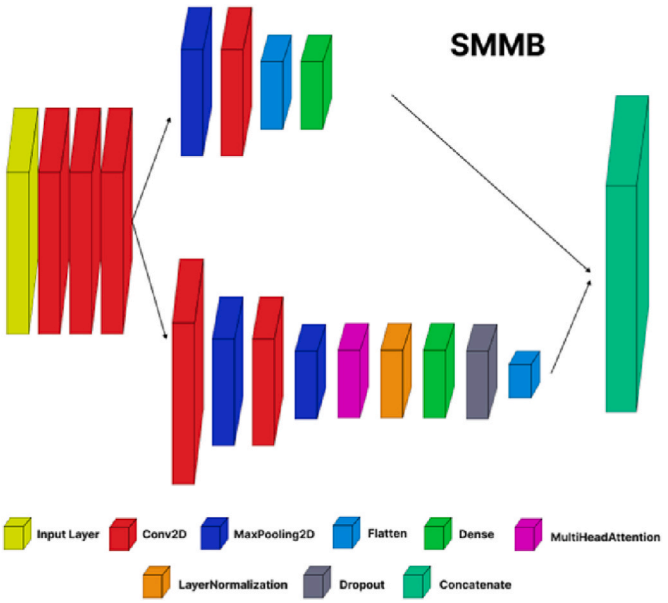

Fig. 8. Architecture of SMMB

classification, and *SoftMax* activation). Fig. 8 illustrates the layers of the **SMMB** architecture. This design enables the network to learn diverse representations, improving performance across multiple FER tasks simultaneously, a capability not typically found in traditional single-branch networks.

3.4. Class balancing managements

As previously noted, public datasets utilized in FER typically exhibit significant class imbalances. Fig. 9 shows the class distribution in the training set and test set of the RAF-DB dataset. Such disparities can lead to overfitting and impede the model's generalization capabilities when deployed. To address this issue and RQ3, several strategies are available, including oversampling, under-sampling, and class weighting. Class weighting is a prevalent technique in machine learning where each class within a classification problem is assigned a specific weight or importance factor. During the training process, these weights are employed to modulate the influence of each class on the model's loss function or optimization algorithm. Weights are generally assigned based on the distribution of classes within the training dataset; minority classes receive higher weights while majority classes are assigned lower weights. The primary aim of class weighting is to counteract the adverse effects of class imbalance, thereby enhancing the model's capacity to deliver accurate predictions across all classes, even with imbalanced data. In this work, we implemented two class weighting approaches: simple class weighting and conservative class weighting.

Simple Class Weighting: This method addresses class imbalances by enhancing the model's sensitivity to less represented classes. In

Fig. 9. RAF-DB class distribution.

datasets with significant imbalances, this approach might lead to overfitting. Consequently, it proves more effective in relatively balanced datasets, such as AffectNet. Specifically, we perform simple class weighting as the following steps: 1) Initially, we established a dictionary '*class_to_idx*' that maps class labels to their respective integer indices. This mapping facilitates the numerical understanding of class labels, useful for several purposes including the inverse mapping with '*idx_to_class*'; 2) Subsequently, we assessed the number of images per class within the training dataset. This data is vital as it reveals the class distribution, crucial for addressing class imbalances; 3) Lastly, we assigned a weight of 1 to the most represented class (the class with the highest number of samples) and computed the class weights for the others. These weights are calculated based on the relative imbalances among the classes, where underrepresented classes receive higher weights by dividing the count of the most common class by the counts of each class, thus ensuring that classes with fewer samples are prioritized. Figs. 10 and 11 present the class distribution before and after simple weighting balance and the weight dictionary, respectively.

Conservative Class Weighting: This technique addresses class imbalance by directing the model's focus towards minority classes during training, while preventing the weights from becoming excessively large, thereby avoiding overfitting. It carefully balances the need to mitigate class imbalance against the risks of potential overfitting. The code computes class weights based on the distribution of classes in the training dataset. Initially, it counts the number of samples in each class using the Counter class. Subsequently, it calculates the class weights by comparing the sample count of each class to that of the most common class. The weights are capped at a maximum value of 3 to avoid extreme values, making them particularly useful for training machine learning models in the presence of class imbalance. This method proved to be highly effective for significantly imbalanced datasets, such as RAF-DB, and less impactful for datasets like AffectNet. Figs. 12 and 13 present the class distribution before and after conservative weighting balance and the weight dictionary, respectively.

3.5. Pretrained weights

Models pre-trained on large-scale datasets can significantly enhance performance in downstream tasks through effective finetuning. To explore the RQ2, we utilize three different pertaining strategies in this study. ImageNet and VGGFace are commonly used pre-trained datasets in FER studies. In contrast to finetuning, training a model from scratch involves learning from the beginning, without the advantage of any pre-trained weights.

ImageNet is an extensive image database designed to train various convolutional neural networks. Specifically, our models utilize ImageNet-1k as pre-trained weights, a subset of ImageNet that includes 1000 diverse classes featuring animals, objects, and means of transport, among others.

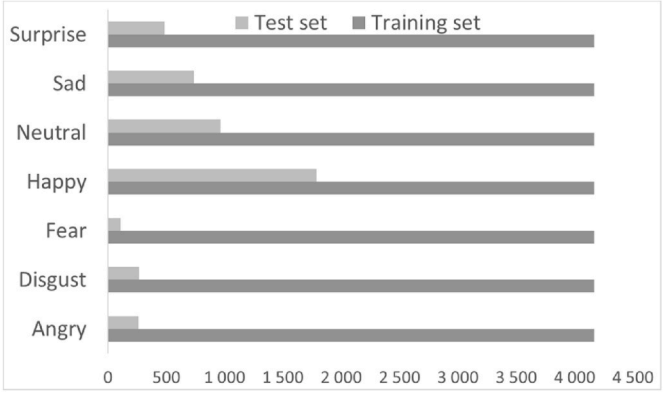

Fig. 10. RAF-DB Simple class weighting balance.

Class name	Class index	weight
Angry	0	6.88
Disgust	1	6.80
Fear	2	16.81
Happy	3	1.00
Neutral	4	1.86
Sad	5	2.42
Surprise	6	3.68

Fig. 11. RAF-DB simple class weighting dictionary.

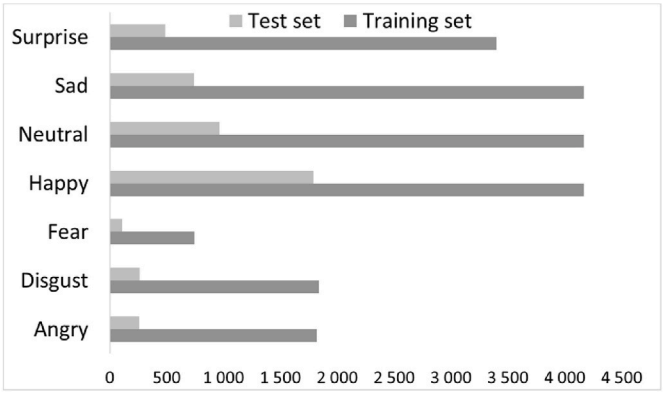

Fig. 12. RAF-DB Conservative Simple class weighting balance.

Class name	Class index	weight
Angry	0	3.00
Disgust	1	3.00
Fear	2	3.00
Happy	3	1.00
Neutral	4	1.86
Sad	5	2.42
Surprise	6	3.00

Fig. 13. RAF-DB conservative class weighting dictionary.

VGGFace is another dataset, tailored for face identification tasks. It is possible to integrate VGGFace pre-trained weights into our models by incorporating the Keras-VGGFace library and employing the VGG16 architecture for training. This setup allows us to leverage VGGFace pre-trained weights effectively.

For all the pre-trained models, we froze the fully connected layers and added new layers with the same hyperparameters. Two pooling techniques are applied. The *GlobalAveragePooling2D* computes the average of the existing values across the spatial dimensions not increasing the number of parameters, while the *Flatten* converts multi-dimensional tensors into a single one-dimensional vector by stacking (compressing or flattening) its contents. Then, one *Dense* layer (256 units, *ReLU* activation, and 0.2 *Dropout*), another *Dense* layer (256 units, and *ReLU* activation), and the final *Dense* layer ('*num_classes*' units, and *SoftMax* activation) are added for classification.

4. Experiments

In this section, we employ the established BTFER dataset and the proposed FER evaluation protocol to conduct experiments for investigating various factors that affect the performance of open-source and self-designed network architectures for FER and answer RQs, in terms of image resolution, class balancing, and pre-trained weight.

4.1. Metrics

We utilize several key metrics to evaluate the performance of the FER models, including **Acc**uracy (the ratio between correctly classified samples against the total number of samples), **P**recision (the ratio of true positives to the sum of true positives and false positives), **R**ecall (the ratio of true positives to the sum of true positives and false negatives), and **F1** Score (the harmonic mean of precision and recall). Notably, we additionally report the **Cross-domain Accuracy** based on our evenly distributed BTFER dataset to further reveal the performance of both public and self-designed models encountering unseen data.

4.2. Implementation

To simulate the scenario of practical application, all experiments are executed on a local workstation running Windows 11 with 64 GB RAM at 3200 MHz and two NVIDIA GeForce RTX GPUs with 12 GB memory. All networks are implemented with Python 3.9 on TensorFlow 2.10. We perform the same data augmentation during all model training. The batch size is 32 per GPU. Unlike current FER methods that employ data augmentation during the training phase, this study adheres to Keras guidelines for pre-processing images. These guidelines recommend specific transformations such as re-scaling (1/255), a rotation range of 20°, width and height shifts of 0.2, a zoom range of 0.2, and enabling horizontal flips (Abbas and Chalup, 2019). To ensure fair comparisons, the same hyperparameters are applied across all models: a learning rate of 1e-4, a reduction in learning rate based on validation loss with patience of 10 epochs, a reduction factor of 0.50, a minimum learning rate of 1e-10, and early stopping after 20 epochs. The training process also follows TensorFlow Keras's recommendations concerning pre-processing, model freezing, and fine-tuning. This involves using pre-trained weights, freezing the last interconnected layers, adding an equivalent number of new layers, and training for 30 epochs or until further learning ceases (an early stopping strategy). After the initial training phase, the best model (as determined by our checkpoint) is selected, and all layers, including those recently added, are unfrozen. Training then continues until it is halted by the early stopping mechanism.

4.3. Performance evaluation related to resolutions (RQ1)

As mentioned in Sec. 1, the impact of different input resolutions is few discussed in existing FER studies. Although the camera sensor of current devices can achieve extremely high resolutions, the region size of human faces in images/videos still could be very limited. Intuitively, higher image resolutions contain more information and should enhance FER performance. In this subsection, we evaluate deep FER models regarding two resolution sizes, i.e., 48 × 48 and 224 × 224, keeping the same model and the dataset. For each network architecture, we use RAF-DB, AffectNet, and FER2013 as training datasets, respectively, and test the model on both the original validation sets of the same dataset and our collected BTFER dataset, respectively. Tables 2 and 3 present a broad spectrum of outcomes. In general, models trained with the RAF-DB dataset achieve the highest accuracy.

Higher resolutions equal more accuracy? To further explore the RQ1, we perform the accuracy comparison between the two resolution settings. When comparing models trained with 48 × 48 resolution images to those trained with 224 × 224 resolution images, we observe differences in accuracy of approximately 0.06 and 0.09 for RAF-DB validation and cross-domain BTFER, respectively, as illustrated in Fig. 14. Consequently, increasing the resolution results in accuracy improvements of approximately 9% and 19% for validation and testing, respectively. This outcome is anticipated due to the availability of more facial details at higher resolutions, which significantly enhanced FER performance. Similar observations have been reported in reference (Abbas and Chalup, 2019), where the accuracy is 60.45% with 48 × 48 resolution, compared to 74.57% with 400 × 400 resolution. For real-life applications, this has significant implications for developing models that generalize better across different environments and conditions. For instance, security cameras might capture lower-resolution images, while smartphones used for social media might provide high-resolution images. Knowing the minimum effective resolution helps in designing FER systems that are adaptable to varied real-world scenarios and can aid in optimizing computational resources.

4.4. Performance evaluation related to pre-trained weights (RQ2)

In this subsection, we aim to compare the performance of pre-trained weights with those trained from scratch. To achieve this, we evaluate the performance of various models by employing all datasets available in our study. The experiments are conducted using three distinct initial states while training the **VGG16** model. Pre-trained weights from VGGFace in Keras, sourced from the Keras-VGGFace library, are utilized. Similarly, two distinct input resolutions, i.e., 48 × 48 and 224 × 224, are used for better analysis, respectively.

Different pre-trained weights, different performance? This experiment examines the impact of utilizing pre-trained weights on the FER performance. Specifically, we train the **VGG16** model using three distinct sets of initial weights: ImageNet, VGGFace, and a random initialization from scratch. For models trained with low-resolution images (48 × 48 pixels), the most effective approach is to employ ImageNet pre-trained weights. This is followed by training the model from scratch with randomly initialized weights, and then using VGGFace pre-trained weights, as referenced in source (Lo et al., 2024) and illustrated in Figs. 15 and 16. The superiority of ImageNet weights can be attributed to their broader generalization capabilities across diverse image types. In contrast, VGGFace weights, which are specifically optimized for high-resolution facial data, exhibit limited generalization when applied to lower-resolution images. The influence of different sampling strategies in training is minimal, with conservative sampling providing a slight improvement (up to 0.03) over simple class weighting. Increasing the image resolution to 224 × 224 pixels, we observe a more pronounced effect of pre-trained weights, especially those from VGGFace,

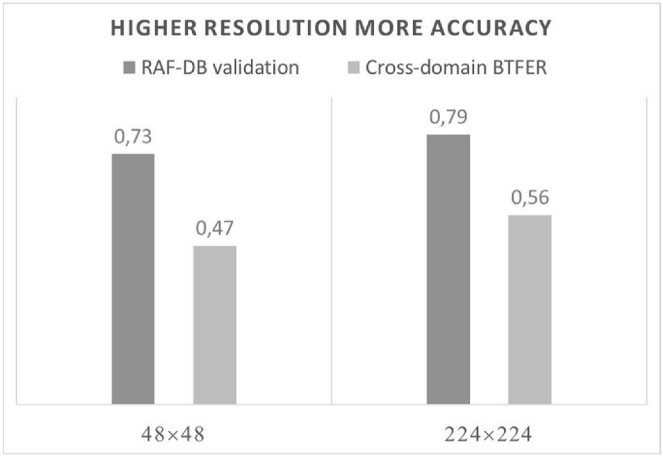

Fig. 14. Performance comparison between models trained on RAF-DB with 48 × 48 and 224 × 224 inputs.

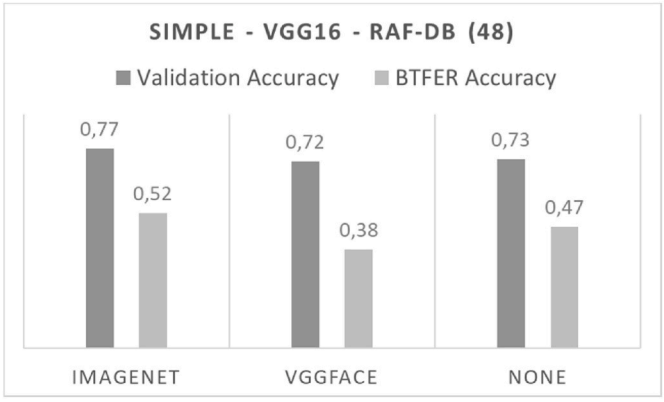

Fig. 15. Pre-trained weights accuracy–Simple 48 × 48.

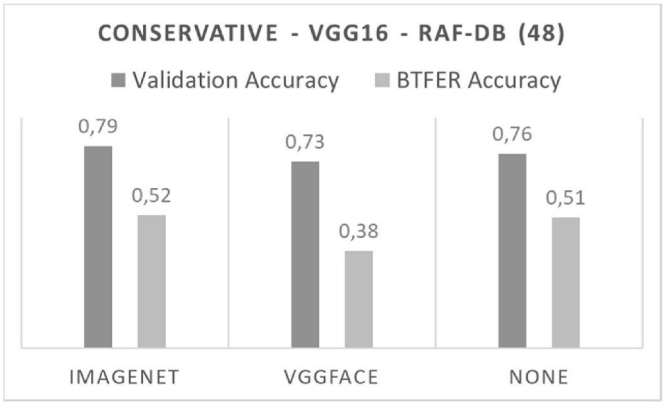

Fig. 16. Pre-trained weights accuracy–Conservative 48 × 48.

on both validation precision and performance against our BTFER dataset. These outcomes are detailed in Figs. 17 and 18. The enhancement in model performance with VGGFace pre-trained weights at this higher resolution can be attributed to the closer alignment of these weights with the characteristics of our dataset, allowing for more effective feature extraction from the images.

Less parameters, good performance? In practical settings, computational resources are often limited. It is essential, therefore, to investigate methods for achieving high accuracy with more compact models. For a resolution of 48 × 48, our experiments demonstrate that it is feasible to achieve high accuracy (over 70%) using network architectures with fewer parameters, as illustrated in Fig. 19. The optimal results are

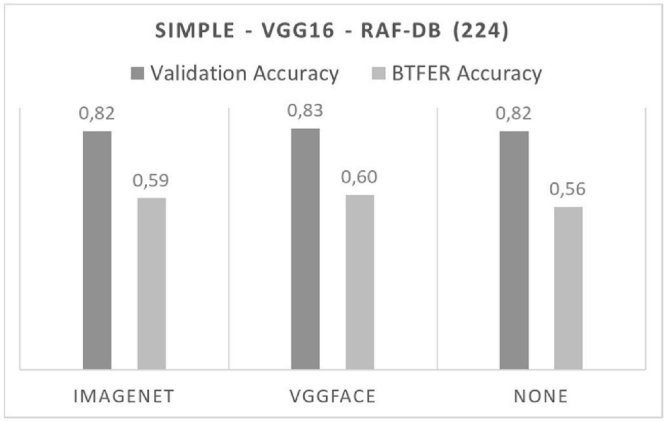

Fig. 17. Pre-trained weights accuracy –Simple 224 × 224.

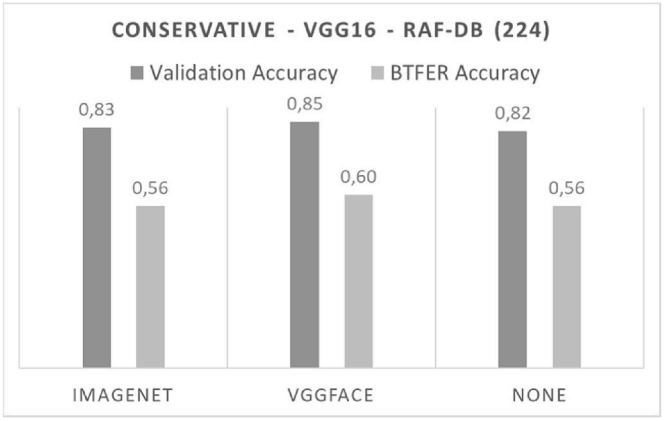

Fig. 18. Pre-trained weights accuracy–Conservative 224 × 224.

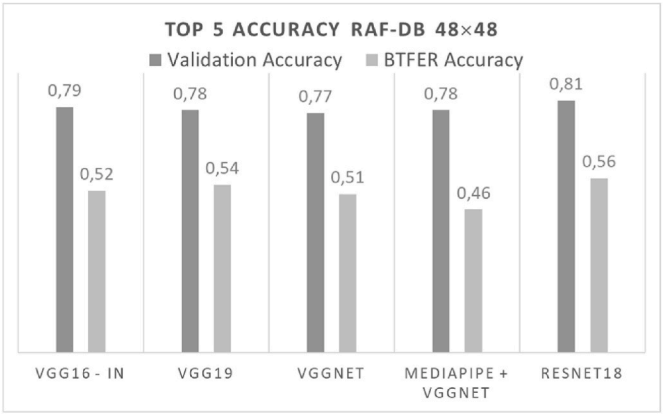

Fig. 19. Top-5 models with less parameter on RAF-DB – 48 × 48.

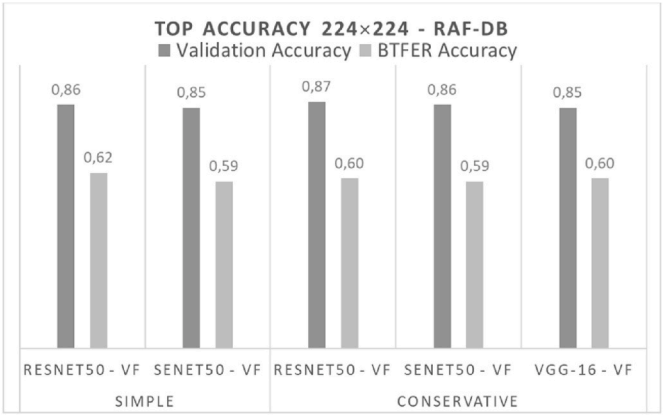

Fig. 20. Top-5 models with less parameter on RAF-DB – 224 × 224.

Table 4
Accuracy comparison of top 5 best models RAF-DB – 48 × 48.

Model	Parameter Scale	Validation Accuracy	BTFER Accuracy
VGG16 - IN	~15M	0,79	0,52
VGG19	~20M	0,78	0,54
VGGNet	~15M	0,77	0,51
MediaPipe + VGGNet	~5,5M	0,78	0,46
ResNet18	~12M	0,81	0,56

achieved with the RAF-DB dataset at both tested resolutions. Table 4 shows that the model achieving the highest performance is **ResNet18**. Models using the **VGG** backbone also exhibit superior performance. A plausible explanation for this is that the **VGG** architecture is capable of extracting features from images, even those of poor or compromised quality, as referenced in (Greco et al., 2023a). If there is a need to prioritize processing speed over accuracy, smaller models such as **Mediapipe + VGGNet** or **Sequential 4 Conv** (ranked sixth) can be considered as they still deliver satisfactory performance. At a resolution of 224 × 224, testing revealed that models that perform exceptionally well are those trained on face datasets (i.e., VGGFace). As depicted in Fig. 20 and outlined in Table 5, the five most effective models are all pre-trained using VGGFace.

4.5. Performance evaluation related to sample annotation (RQ3)

Well-labeled datasets, higher performance? After training various models using three public datasets, it is observed that RAF-DB, despite its smaller size (approximately 15,000 images), consistently achieves higher accuracy compared to FER 2013 (approximately 35,000 images) and AffectNet (approximately 27,000 images), as depicted in Figs. 21–24. This pattern hosts true across different class balance strategies, with RAF-DB outperforming the others in each scenario. These findings underscore the significance of precise labeling, as referenced in (Liu et al., 2024). When the models are trained using images of 224 × 224 pixels, RAF-DB continues to demonstrate superior accuracy. In these tests, AffectNet ranks second, surpassing FER 2013, which can be attributed to differences in image resolution. FER2013 typically uses a resolution of 48 × 48 pixels, while AffectNet uses 512 × 512 pixels, possibly mitigating some of its labeling inaccuracies. In conclusion, the categorization of facial expressions that do not accurately represent certain emotions can detrimentally impact the model's ability to generalize. For example, a face incorrectly labeled as 'happy' might more accurately convey a neutral expression.

Class balance strategy affects more on models trained from scratch? In 48 × 48 Resolution, training models from scratch highlights the significant impact of our class balance strategy, as depicted in Fig. 25. This effect is more pronounced compared to models that utilize pre-trained weights, shown in Fig. 26. The pronounced difference can be attributed to the absence of prior learning. A model trained from scratch must develop all features and representations from the beginning; thus, incorporating a class balance strategy may lead to overfitting. Essentially, if a model is being trained from scratch, any strategy that mitigates overfitting is likely to yield superior results, regardless of the type of rebalancing strategy employed, whether it be *Conservative* or *Simple* weighting balance. In the 224 × 224 Resolution, models are trained at a higher resolution in similar configurations, as illustrated in Figs. 27 and 28. The disparity in performance between models employing simple weighting class balance strategies has diminished. Comparing this with the 48 × 48 resolution, it is evident that in higher resolution training, the impact of the class balance strategy is less crucial than having a well-labeled and balanced dataset.

4.6. Practical implications for model selection

To offer a deeper analysis of why certain network architectures ranked higher in the study of FER, it is essential to consider both the architectural strengths and weaknesses in practical scenarios. As detailed in Table 6, for low-resolution images (i.e., 48 × 48), the optimal strategy for achieving high accuracy involves utilizing **ResNet18**, supplemented with two dropout layers and a robust dense layer. One possible explanation is that **ResNet** models employ residual connections to alleviate the vanishing gradient problem, allowing for deeper networks without a loss in performance. This is particularly beneficial in FER, where deeper feature hierarchies often capture more nuanced emotional information across a diverse range of facial expressions, especially in in-the-wild datasets where expression variance is significant. It is also demonstrated when using a higher resolution (i.e., 224 ×

Table 5
Accuracy comparison of top 5 best models RAF-DB – 224 × 224.

Balance	Model – Pretraining	Validation Accuracy	BTFER Accuracy
Simple	ResNet50 - VF	0,86	0,62
	SeNet50 - VF	0,85	0,59
Conservative	ResNet50 - VF	0,87	0,60
	SeNet50 - VF	0,86	0,59
	VGG-16 - VF	0,85	0,60

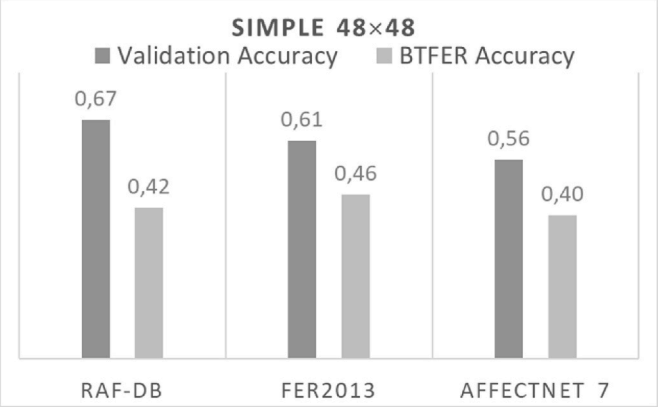

Fig. 21. Average accuracy with simple class weighting 48 × 48.

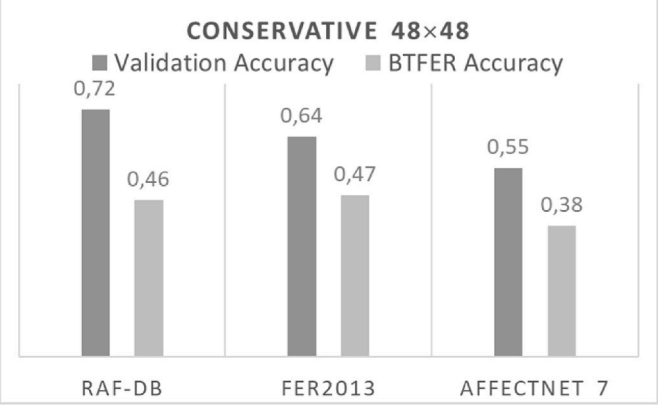

Fig. 22. Average accuracy with conservative class balance 48 × 48.

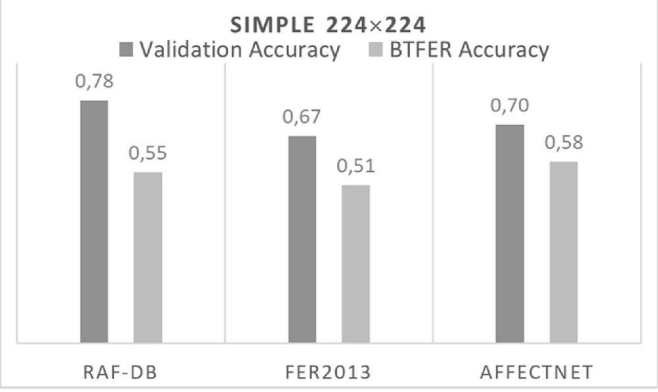

Fig. 23. Average accuracy with simple class weighting 224 × 224.

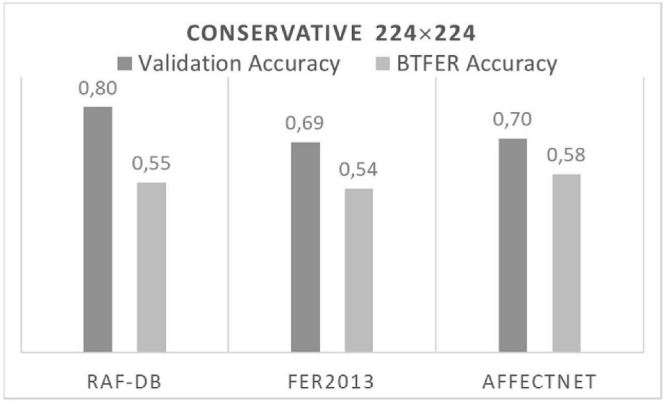

Fig. 24. Average accuracy with conservative class balance 224 × 224.

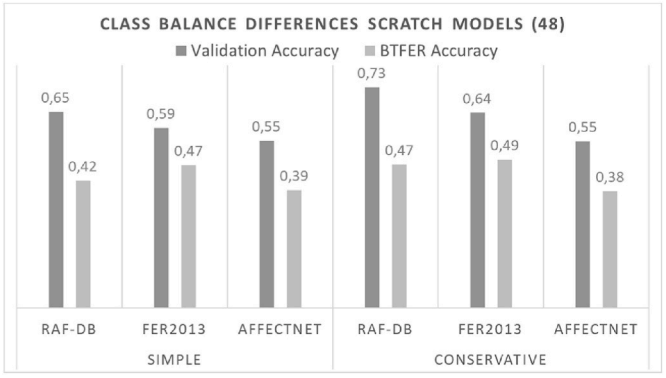

Fig. 25. Average accuracy without pre-trained weights 48 × 48.

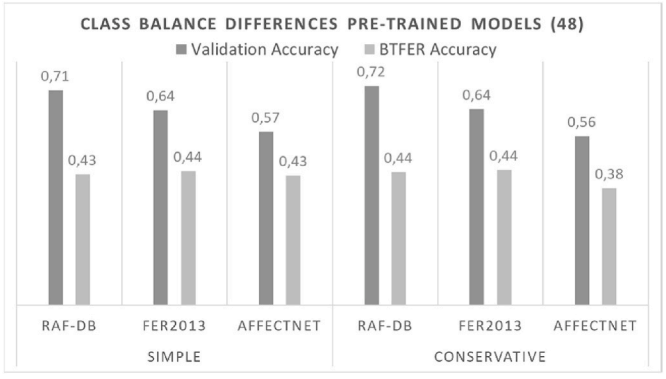

Fig. 26. Average accuracy with pre-trained weights 48 × 48.

224), the best accuracy is achieved using **ResNet50** with VGGFace pre-trained weights. An alternative and more user-friendly option is to employ **VGG16** with ImageNet pre-trained weights, as depicted in Table 2, without the necessity of constructing a **ResNet18** model from scratch. The **VGG** architecture is straightforward yet effective in capturing essential features, making them robust against overfitting with sufficient training data and regularization, a common challenge in practical FER applications. Our self-designed models demonstrate competitive performance across various metrics (e.g., accuracy, F1 score) when evaluated on the BTFER dataset, showing strengths in scenarios with imbalanced data and real-world diversity. The architectures are designed to offer a range of computational complexities, from lightweight models suitable for mobile devices (e.g., **Sequential Simple**) to more complex models for high-accuracy applications (e.g.,

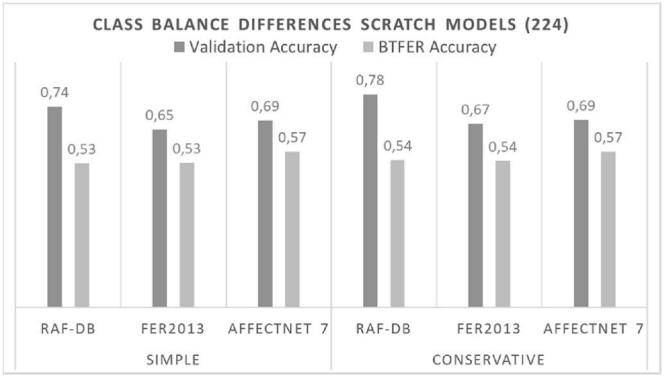

Fig. 27. Average accuracy without pre-trained weights 224 × 224.

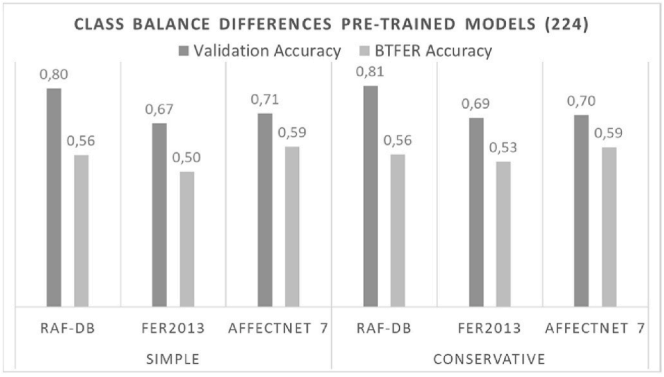

Fig. 28. Average accuracy with pre-trained weights 224 × 224.

Table 6
Best performance models with 48 × 48 and 224 × 224.

	Model	Num Parameter	Validation Accuracy	BTFER Accuracy
48 × 48	ResNet18	~12M	0,81	0,55
	Sequential 4 Conv	~3 M	0,76	0,54
224 × 224	ResNet50	~24M	0,85	0,62
	DenseNet121	~7M	0,82	0,60

Ensemble). This versatility ensures that our designs can be effectively deployed across different hardware environments without compromising performance. For instance, our self-designed **Sequential 4 Conv** and the public **DenseNet121** achieve competitive performance with small-scale parameters, which point out their potential in practical applications, as shown in Table 6.

In general, when selecting a model for practical FER applications, it is crucial to balance the model architectural strengths with the specific requirements and constraints of the deployment environment. For instance, if deployment is intended for mobile devices, architectures like **MobileNet** or **EfficientNet** are preferable due to their efficiency, even though it may compromise the model's ability to learn more complex and subtle features necessary for accurate FER across diverse expressions and conditions. However, for applications requiring high accuracy and complexity, such as psychological analysis or security systems, more robust architectures like **ResNet** or **VGGNet** might be more suitable. The choice of architecture should also consider the dataset's characteristics and the need for generalization across different demographic groups and environmental conditions. Models that provide extensive pre-training options, such as those compatible with ImageNet or VGGFace weights, offer valuable transfer learning opportunities to

enhance performance on specific FER tasks. Architectures like **Inception** have complex and computationally intensive modules that can lead to high memory consumption, limiting their deployments in constrained environments where simplicity and speed are prioritized.

4.7. Practical implications for model deployment on edge devices

Besides general devices (e.g., laptops, servers), deploying FER models on edge devices broadens the adoption of FER technologies across various industries, such as healthcare, retail, and education (Aljaafreh et al., 2023). Insights from our experiments also suggest a few issues in deploying optimal FER models, including model conversion, resource constraints, and real-time processing. Intuitively, model compression techniques, like TensorFlow Lite and PyTorch Mobile, can be applied to convert standard models into a format that is more efficient for execution on mobile and edge devices, like ResNet50 with VGGFace pre-trained weights, on resource-constrained platforms, such as Raspberry Pi (Sajjad et al., 2019). The process of converting a full FER model to a lite format involves quantization, which can reduce the model size and improve inference speed but may also impact the model's accuracy (Zhen et al., 2021). On the other hand, edge devices typically have limited computational resources, including lower processing power, memory, and storage. Efficiently optimizing models to fit within these constraints without significantly sacrificing performance is crucial. Techniques like model pruning, architecture search, and lightweight architecture design are often employed (Lin et al., 2022). Finally, ensuring that the model can process data in real-time with minimal latency is essential for many applications, especially those involving user interaction. Optimizations need to focus on reducing inference time, which can be challenging given the hardware limitations of edge devices.

However, the conversion and optimization process can impact the overall accuracy and performance of FER models in several ways. Quantization and other optimization techniques can introduce small losses in accuracy due to reduced precision in computations. It is important to evaluate these trade-offs and determine acceptable accuracy thresholds for specific applications. For instance, depending on the specific use case and hardware constraints, choosing the right model architecture is crucial. Lightweight architectures like **MobileNetV2** or **EfficientNet** can provide a good balance between accuracy and efficiency. Our study also highlights the critical role of class balance strategies in training models from scratch for edge environments, where data imbalance is common. Using pre-trained weights such as VGGFace can help in reducing sensitivity to class imbalances and achieving high accuracy with less need for extensive training data. Developers can prioritize advanced class balancing methods to improve model performance in constrained settings.

5. Challenges of practical FER applications

5.1. Open directions of future FER research

Novel deep architectures: The future of the FER field is poised for rapid advancements that build upon the foundations established by CNNs. While CNNs have demonstrated significant success in FER, the trajectory of the field is likely to encompass more sophisticated deep learning architectures. Approaches like Multi-Task CNNs, Vision Transformers, Multimodal implementations, and attention mechanisms are anticipated to gain prominence (Liu et al., 2024; Singh and Kapoor, 2023; Xie et al., 2023), this is possible to visualize when looking at the most recent papers such as (Xue et al., 2022; Chen et al., 2022). This includes exploring hybrid models that combine the strengths of CNNs for feature extraction with sequence models like RNNs to better capture the temporal dynamics of expressions in videos from diverse sources. These advanced architectures can capture temporal dependencies and contextual information, enabling better feature extraction and understanding of facial expressions. However, the challenge lies in the increased computational demands of these complex models compared to traditional CNNs.

Advanced training techniques: In our empirical evaluations, we observed that models pre-trained on large, diverse datasets like ImageNet showed better generalization capabilities when fine-tuned on targeted FER datasets. However, when these models are deployed on datasets with fundamentally different characteristics (e.g., from posed to spontaneous expressions), performance can degrade significantly unless specific adaptation strategies, such as domain adaptation or transfer learning techniques, are employed. To overcome these challenges, future research should focus on developing more adaptable training schemes that can learn domain-invariant features. As our research centered on fixed resolutions, the development of models that can adapt to varying resolutions could enhance accuracy by ensuring robust data utilization and generalization across different environments. Similar studies in other domains have sought to identify the ideal resolution for specific problems to optimize FER accuracy while managing computational resources efficiently (Zhao et al., 2021). Additionally, the pursuit of finding the optimal balance between computational power and accuracy in different image resolutions holds promise.

Diverse datasets: Additionally, there is a pressing need for creating and utilizing balanced datasets that reflect a broad spectrum of demographic and emotional diversity. This approach will help in training models that are not only high-performing but also fair and unbiased across various user groups. The future FER landscape should also prioritize the development of larger, balanced, and unbiased datasets to ensure models' robustness and generalization across diverse scenarios. Exploring other architectural paradigms like transformers and multi-task learning for FER could further enhance the field's capabilities (Liu et al., 2024). Besides, the notion of capturing micro-expressions to create a more nuanced representation of facial expressions is also gaining traction (Pan et al., 2023). Developing datasets that encompass a wide range of micro-expressions presents a challenge, yet addressing this could significantly boost the accuracy and applicability of FER models (Li et al., 2022).

5.2. Deployment of FER models in different domains

Marketing: In the marketing sector, FER technology is employed to analyze consumer reactions to products and advertisements on both retail and online platforms (Khan et al., 2024). The insights derived from this analysis are utilized to refine marketing strategies and tailor content, enhancing resonance with target demographics. FER models in this domain may process data either offline (post-exposure) or in real-time (during exposure). The primary objective here is not the precise detection of emotional states but rather the identification of general sentiment trends, such as through A/B testing of advertisements.

Education: FER technology proves advantageous in the educational field for monitoring student engagement and emotional responses, especially in remote or hybrid learning settings (Lawpanom et al., 2024; Li et al., 2024). These models must process data in real time to provide instant feedback to educators on student engagement levels. Challenges include customizing FER technology for diverse educational contexts and varied student demographics, ensuring robust performance across different backgrounds and lighting conditions typical of home learning environments. Moreover, it is crucial to integrate FER seamlessly into educational platforms, avoiding disruptions in the learning process and concerns about surveillance.

Entertainment: In the entertainment industry, FER assists creators in assessing audience reactions to films, games, shows, and performances, which aids in content modification to maximize viewer satisfaction (Witherow et al., 2023). FER is also incorporated into consumer hardware, such as gaming consoles and VR/AR devices, to evaluate user emotions (Somarathna et al., 2023). This application particularly requires models that perform fast inference on limited hardware, favoring

simpler models where high accuracy is less critical.

Healthcare: The application of FER in healthcare is vital for discerning patient emotions, which supports diagnostics and monitoring, particularly in mental health and pain assessment (Werner et al., 2022; Rong et al., 2022). Here, high accuracy and reliability are imperative to ensure that emotional evaluations positively impact patient care and avoid misinterpretations. These models generally necessitate local processing to adhere to privacy and regulatory standards concerning health data (Heidari et al., 2022). The challenge lies in integrating FER into clinical settings in a manner that respects patient privacy and improves communication between patients and healthcare providers without supplanting traditional observational techniques. Monitoring patients with contactless multimodal systems is an application area where FER technologies could be adopted.

Law Enforcement and Security: In law enforcement and security, FER is utilized to identify individuals who may pose a threat based on their emotional states or to pinpoint persons of interest in crowded settings (Sajjad et al., 2019). This application requires the utmost accuracy to prevent wrongful identification and ensure public safety. The models must perform robustly under various environmental conditions and be capable of real-time processing for immediate threat assessment and response. A significant challenge is ensuring that FER technology deployment is aligned with ethical standards and civil liberties, emphasizing transparency and accountability.

5.3. Ethical considerations in adopting FER models

Deploying FER models across various practical applications presents a complex landscape of ethical considerations. Central to these concerns are issues related to privacy, data security, and compliance with diverse regulatory frameworks (Nair et al., 2022). To this end, specific privacy and concerns and legal frameworks need to be addressed, as well as how these considerations might influence the design and deployment of FER systems.

Deep FER systems often require extensive data collection, including video footage and images that capture individuals' facial expressions. This raises significant privacy concerns as individuals may not be aware of or have consented to the recording and analysis of their facial data. The storage of such data, especially when it includes personally identifiable information (PII), presents risks of data breaches and unauthorized access. Such data collection raises concerns about consent, data minimization, and the potential for misuse of personal information. To mitigate these risks, developers should implement stringent data protection measures, such as de-identification (Cheng et al., 2022) and anonymization (Heidari et al., 2024), securing data storage and transmission, and ensuring that data collection processes are transparent. Furthermore, potential misuse, including surveillance, profiling, and discrimination, can also have severe implications for individual freedoms and rights.

In addition, the deployment of FER technologies must adhere to a variety of legal standards across different regions. For instance, in the European Union, the General Data Protection Regulation (GDPR) imposes strict guidelines on the processing of personal data, including biometric information. In 2024, the European Commission has taken facial recognition technology under special regulation by the AI Act. It mandates that individuals have the right to know how their data is being used, and organizations must ensure that data is collected and processed lawfully, transparently, and for a specific purpose. This regulation significantly influences the design of FER systems to ensure compliance with data protection principles. Like the GDPR, the California Consumer Privacy Act (CCPA) in the United States provides consumers with rights regarding the collection and use of their personal data. FER systems deployed in regions covered by the CCPA must ensure that they comply with these regulations, including providing transparency about data collection and giving consumers the right to opt-out. Developers should conduct comprehensive legal reviews to ensure their FER applications comply with these regulations and include mechanisms for users to provide informed consent, access the data held about them, and request data deletion. Ethical considerations are especially required in dealing with critical sectors, such as healthcare and education which are dealing with sensitive user data. These guidelines should focus on ensuring fairness, accountability, and transparency in the deployment of FER systems.

CRediT authorship contribution statement

Gianmarco Ipinze Tutuianu: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation. **Yang Liu:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Ari Alamäki:** Writing – original draft, Supervision, Investigation, Formal analysis. **Janne Kauttonen:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This research was supported by the AI Forum project funded by the Finnish Ministry of Education and Culture (OKM/116/523/2020). Dr. Yang Liu appreciated the support of the Finnish Cultural Foundation for North Ostrobothnia Regional Fund under Grant 60231712, and in part by the Instrumentarium Science Foundation under Grant 240016.

References

Abbas, A., Chalup, S., 2019. The impact of image resolution on facial expression analysis with CNNs. In: Proceedings of the International Joint Conference on Neural Networks. https://doi.org/10.1109/IJCNN.2019.8852264.

Aljaafreh, A., Abadleh, A., Alja'afreh, S.S., Alawasa, K., Almajali, E., Faris, H., 2023. Edge deep learning and computer vision-based physical distance and face mask detection system using jetson xavior NX. Emerging Science Journal 7, 70–80. https://doi.org/10.28991/ESJ-2023-SPER-05.

Amiri, Z., Heidari, A., Navimipour, N.J., Unal, M., Mousavi, A., 2024. Adventures in data analysis: a systematic review of Deep Learning techniques for pattern recognition in cyber-physical-social systems. Multimed. Tool. Appl. 83, 22909–22973. https://doi.org/10.1007/s11042-023-16382-x.

Chen, R., Zhou, W., Li, Y., Zhou, H., 2022. Video-based cross-modal auxiliary network for multimodal sentiment analysis. IEEE Trans. Circ. Syst. Video Technol. 32, 8703–8716. https://doi.org/10.1109/TCSVT.2022.3197420.

Cheng, K.H.M., Yu, Z., Chen, H., Zhao, G., 2022. Benchmarking 3D face de-identification with preserving facial attributes. In: Proceedings - International Conference on Image Processing, ICIP. IEEE Computer Society, pp. 656–660. https://doi.org/10.1109/ICIP46576.2022.9897232.

Chollet, F., 2017. Xception: deep learning with depthwise separable convolutions. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition. CVPR. https://doi.org/10.1109/CVPR.2017.195.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J., 2021. RepVgg: making VGG-style ConvNets great again. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 13728–13737. https://doi.org/10.1109/CVPR46437.2021.01352.

Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., Bengio, Y., 2013. Challenges in representation learning: a report on three machine learning contests. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642-42051-1_16.

Greco, A., Strisciuglio, N., Vento, M., Vigilante, V., 2023a. Benchmarking deep networks for facial emotion recognition in the wild. Multimed. Tool. Appl. 82 https://doi.org/10.1007/s11042-022-12790-7.

Greco, A., Strisciuglio, N., Vento, M., Vigilante, V., 2023b. Benchmarking deep networks for facial emotion recognition in the wild. Multimed. Tool. Appl. 82, 11189–11220. https://doi.org/10.1007/s11042-022-12790-7.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2016.90.

Heidari, A., Toumaj, S., Navimipour, N.J., Unal, M., 2022. A privacy-aware method for COVID-19 detection in chest CT images using lightweight deep conventional neural network and blockchain. Comput. Biol. Med. 145 https://doi.org/10.1016/j.compbiomed.2022.105461.

Heidari, A., Jafari Navimipour, N., Dag, H., Unal, M., 2024. Deepfake Detection Using Deep Learning Methods: A Systematic and Comprehensive Review, vol. 14. Wiley Interdiscip Rev Data Min Knowl Discov. https://doi.org/10.1002/widm.1520.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E., 2020. Squeeze-and-Excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. 42 https://doi.org/10.1109/TPAMI.2019.2913372.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

Khan, U.A., Xu, Q., Liu, Y., Lagstedt, A., Alamäki, A., Kauttonen, J., 2024. Exploring contactless techniques in multimodal emotion recognition: insights into diverse applications, challenges, solutions, and prospects. Multimed. Syst. 30 https://doi.org/10.1007/s00530-024-01302-2.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60. https://doi.org/10.1145/3065386.

Lawpanom, R., Songpan, W., Kaewyotha, J., 2024. Advancing facial expression recognition in online learning education using a homogeneous ensemble convolutional neural network approach. Appl. Sci. 14, 1156. https://doi.org/10.3390/app14031156.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86. https://doi.org/10.1109/5.726791.

Li, S., Deng, W., Du, J.P., 2017. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition. CVPR. https://doi.org/10.1109/CVPR.2017.277.

Li, Y., Wei, J., Liu, Y., Kauttonen, J., Zhao, G., 2022. Deep learning for micro-expression recognition: a survey. IEEE Trans Affect Comput 13, 2028–2046. https://doi.org/10.1109/TAFFC.2022.3205170.

Li, Y., Liu, Y., Nguyen, A., Shi, H., Vuorenmaa, E., Järvelä, S., Zhao, G., 2024. Interactions for socially shared regulation in collaborative learning: an interdisciplinary multimodal dataset. ACM Trans Interact Intell Syst. https://doi.org/10.1145/3658376.

Lin, X., Kim, S., Joo, J., 2022. FairGRAPE: fairness-aware GRAdient pruning mEthod for face attribute classification. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-031-19778-9_24.

Liu, Y., Zhang, X., Lin, Y., Wang, H., 2020. Facial expression recognition via deep action units graph network based on psychological mechanism. IEEE Trans Cogn Dev Syst 12, 311–322. https://doi.org/10.1109/TCDS.2019.2917711.

Liu, Y., Zhang, X., Zhou, J., Fu, L., 2021. SG-DSN: a semantic graph-based dual-stream network for facial expression recognition. Neurocomputing 462, 320–330. https://doi.org/10.1016/j.neucom.2021.07.017.

Liu, Y., Zhang, X., Li, Y., Zhou, J., Li, X., Zhao, G., 2023. Graph-based facial affect analysis: a review. IEEE Trans Affect Comput 14, 2657–2677. https://doi.org/10.1109/TAFFC.2022.3215918.

Liu, Y., Zhang, X., Kauttonen, J., Zhao, G., 2024. Uncertain facial expression recognition via multi-task assisted correction. IEEE Trans. Multimed. 26, 2531–2543. https://doi.org/10.1109/TMM.2023.3301209.

Lo, L., Ruan, B.K., Shuai, H.H., Cheng, W.H., 2024. Modeling uncertainty for low-resolution facial expression recognition. IEEE Trans Affect Comput 15, 198–209. https://doi.org/10.1109/TAFFC.2023.3264719.

Mollahosseini, A., Hasani, B., Mahoor, M.H., 2019. AffectNet: a database for facial expression, valence, and arousal computing in the wild. IEEE Trans Affect Comput 10, 18–31. https://doi.org/10.1109/TAFFC.2017.2740923.

Nair, D.G., Nair, J.J., Jaideep Reddy, K., Aswartha Narayana, C.V., 2022. A privacy preserving diagnostic collaboration framework for facial paralysis using federated learning. Eng. Appl. Artif. Intell. 116 https://doi.org/10.1016/j.engappai.2022.105476.

Pan, H., Xie, L., Wang, Z., 2023. C3DBed: facial micro-expression recognition with three-dimensional convolutional neural network embedding in transformer model. Eng. Appl. Artif. Intell. 123 https://doi.org/10.1016/j.engappai.2023.106258.

Raamkumar, A.S., Yang, Y., 2022. Empathetic conversational systems: a review of current advances, gaps, and opportunities. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2022.3226693.

Rong, Q., Ding, S., Yue, Z., Wang, Y., Wang, L., Zheng, X., Li, Y., 2022. Non-contact negative mood state detection using reliability-focused multi-modal fusion model. IEEE J Biomed Health Inform 26, 4691–4701. https://doi.org/10.1109/JBHI.2022.3182357.

Sajjad, M., Nasir, M., Ullah, F.U.M., Muhammad, K., Sangaiah, A.K., Baik, S.W., 2019. Raspberry Pi assisted facial expression recognition framework for smart security in law-enforcement services. Inf. Sci. 479, 416–431. https://doi.org/10.1016/j.ins.2018.07.027.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Singh, N., Kapoor, R., 2023. Multi-modal Expression Detection (MED): a cutting-edge review of current trends, challenges and solutions. Eng. Appl. Artif. Intell. 125 https://doi.org/10.1016/j.engappai.2023.106661.

Somarathna, R., Bednarz, T., Mohammadi, G., 2023. Virtual reality for emotion elicitation - a review. IEEE Trans Affect Comput 14, 2626–2645. https://doi.org/10.1109/TAFFC.2022.3181053.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2015.7298594.

Tan, Q.V.L.M., 2013. EfficientNet: rethinking model scaling for convolutional neural networks mingxing. Can. J. Emerg. Med. 15.

Thanathamathee, P., Sawangarreerak, S., Kongkla, P., Nizam, D.N.M., 2023. An optimized machine learning and deep learning framework for facial and masked facial recognition. Emerging Science Journal 7, 1173–1187. https://doi.org/10.28991/ESJ-2023-07-04-010.

Ullah, Z., Qi, L., Hasan, A., Asim, M., 2022. Improved deep CNN-based two stream super resolution and hybrid deep model-based facial emotion recognition. Eng. Appl. Artif. Intell. 116 https://doi.org/10.1016/j.engappai.2022.105486.

Werner, P., Lopez-Martinez, D., Walter, S., Al-Hamadi, A., Gruss, S., Picard, R.W., 2022. Automatic recognition methods supporting pain assessment: a survey. IEEE Trans Affect Comput 13, 530–552. https://doi.org/10.1109/TAFFC.2019.2946774.

Witherow, M.A., Samad, M.D., Diawara, N., Bar, H.Y., Iftekharuddin, K.M., 2023. Deep adaptation of adult-child facial expressions by fusing landmark features. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2023.3297075.

Xie, Y., Tian, W., Yu, Z., 2023. Robust facial expression recognition with transformer block enhancement module. Eng. Appl. Artif. Intell. 126 https://doi.org/10.1016/j.engappai.2023.106795.

Xue, F., Wang, Q., Tan, Z., Ma, Z., Guo, G., 2022. Vision transformer with attentive pooling for robust facial expression recognition. IEEE Trans Affect Comput. https://doi.org/10.1109/TAFFC.2022.3226473.

Xue, F., Wang, Q., Tan, Z., Ma, Z., Guo, G., 2023. Vision transformer with attentive pooling for robust facial expression recognition. IEEE Trans Affect Comput 14, 3244–3256. https://doi.org/10.1109/TAFFC.2022.3226473.

Yang, K., Wang, C., Sarsenbayeva, Z., Tag, B., Dingler, T., Wadley, G., Goncalves, J., 2021. Benchmarking commercial emotion detection systems using realistic distortions of facial image datasets. Vis. Comput. 37, 1447–1466. https://doi.org/10.1007/s00371-020-01881-x.

Zeiler, M.D., Fergus, R., 2014. LNCS 8689 - Visualizing and Understanding Convolutional Networks.

D. Zeng, Z. Lin, X. Yan, Y. Liu, F. Wang, B. Tang, Face2Exp: Combating Data Biases for Facial Expression Recognition, n.d. https://github.com/danzeng1990/Face2Exp.

Zhang, K., Zhang, Z., Li, Z., Qiao, Y., 2016. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. 23, 1499–1503. https://doi.org/10.1109/LSP.2016.2603342.

Zhao, Z., Liu, Q., Zhou, F., 2021. Robust lightweight facial expression recognition network with label distribution training. www.aaai.org.

Zhen, P., Chen, H.-B., Cheng, Y., Ji, Z., Liu, B., Yu, H., 2021. Fast video facial expression recognition by a deeply tensor-compressed LSTM neural network for mobile devices. ACM Trans. Internet Technol. 2 https://doi.org/10.1145/3464941.

Zoph, B., Vasudevan, V., Shlens, J., V Le, Q., 2018. Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2018.00907.