2024 18th International Conference on Automatic Face and Gesture Recognition (FG)

LITE-FER: A lightweight facial expression recognition framework for children in resource-limited devices

Erhan Bicer and Hatice Kose
Department of Artificial Intelligence and Data Engineering, Istanbul Technical University, Istanbul, Turkiye

***Abstract*— This study proposes a lightweight facial expression recognition (FER) framework for children that can be used on resource-limited devices such as socially assistive robots interacting with children in real world applications. In this study, knowledge distillation (KD) and unstructured weight pruning (UWP) method are used to achieve lightweight FER models. Effect of joint usage of KD and UWP method on FER is also evaluated by using a pruned teacher model within the teacher-student training paradigm in knowledge distillation method. Experiments are performed utilizing AffectNet, CK+ and CAFE datasets. LITE-FER achieved 89.69% and 77% in CK+ and CAFE respectively in k-fold cross validation strategy. LITE-FER only consists of 113.98K parameters (445.24 KB) which makes the model resource-efficient. LITE-FER can reach up to 173.82 FPS with TensorRT on GPU, and 30.09 FPS with keras on CPU. LITE-FER revealed its maximum inference performance in batch inference with 3213 FPS throughput. Results showed that our proposed model ("LITE-FER") results in comparable accuracy, as well as computational and memory efficiency. The proposed model is planned to be used in assistive robotic systems for children in the future.**

I. INTRODUCTION

Facial expressions of people give hints regarding their emotional status, fatigue, engagement in conversation/interaction and so on. Therefore, developing an automatic FER algorithm has been an active research area. Developed algorithms can be utilized for the aim of detecting aforementioned states of people. In this study, we specifically focus on recognizing emotions through facial expressions. The proposed emotion recognition model will be used to detect children emotions.

Proposed state-of-the-art algorithms in the FER domain are mainly based on convolutional neural networks [4]. Mostly, to achieve state-of-the-art results, proposed deep learning solutions consist of excessive number of parameters which increases the computational load. To this end, lightweight solutions are proposed in the literature by using model compression algorithms. With the aim of resulting in a lightweight neural network that does not require high computational load and memory usage, effects of model pruning [3], [29], [26], [9], [6], [38], knowledge distillation (KD) [5], [15] and neural architecture search (NAS) [33], [8], [17] methods are explored within this research area. Even though there are proposed lightweight solutions for FER, lightweight solutions are not thorougly studied in children FER. In fact, to the best of our knowledge, there is only a single study [3] that proposes lightweight solution by utilizing weight pruning for efficient children FER.

This study presents novel contributions in the literature from several key aspects. Firstly, we introduce the methodology of knowledge distillation (KD) in the child FER domain which has not been tested previously to the best of our knowledge. Secondly, effect of combining unstructured weight pruning (UWP) and knowledge distillation (KD) is explored. Experiments are performed with pruned teacher. Lastly, we provide an open-source lightweight framework for facial expression recognition of children to be used in resource-limited devices. As an open source platform [1] offering ligthweight models for emotion recognition in children, the system can be utilized directly as an assistive service on resource-limited devices such as robots, embedded or mobile devices for children such as child-robot interactions studies, and support the researchers and technology providers in this domain. Proposed approach is published as TensorFlow models and TensorRT models which can be utilized in embedded devices. Performance analysis showed that the proposed lightweight model operates faster compared to complex models on a laptop with i7-12700H and NVidia RTX 3060. Together, these key points highlight the novel and high-impact contributions that our research offers.

II. RELATED WORK

Many solutions have been proposed for the FER problem for a while. In fact, initially, conventional image processing algorithms are utilized to extract features related to human face for emotion and facial landmark recognition [40]. With the development and proven success of deep learning, hand-crafted methods have been discontinued over time [4], even though prior works have also proposed joint approaches that combine hand-crafted and neural networks [27], [7]. To make the neural network focus on significant parts of the face, "attention" concept is utilized in prior works [39], [23]. Minaee et al. [23] integrated attention mechanism within a spatial transformer network. Transfer learning is also widely applied within this area to leverage the information already learned from another domain. Many prior studies benefit from transfer learning by proposing algorithms based on pretrained algorithms (e.g. ResNet-50, VGG-16) [13], [16], [11]. With utilizing the already learned low-level features of pre-trained algorithms, high-level feature extraction of the network can be adapted to the FER task by training parameters of top-most layers [11].

Along with static image solutions for FER, video-based solutions are also explored [24], [20], [12]. When using static

[1] https://github.com/erhanbicerr/LITE-FER

979-8-3503-9494-8/24/$31.00 ©2024 IEEE

images, spatial relation is taken into consideration. Moreover, if a video (consecutive frames of static images) is being used, temporal relation can also be explored. Studies in the video domain mostly utilize both spatial and temporal (spatio-temporal) features regarding input frames. Yet, not all of the frames within a video consist of useful information. Using attention mechanism, proposed algorithms become aware of redundant frames to improve the performance [20], [12].

Even though FER has been an active research area for quite an amount of time, FER studies for children and infants are limited compared to the studies in adult faces. FER should also be addressed in the children domain as the existing solutions for adult FER do not directly apply to children domain due to morphological differences of face muscles [42]. Thus, researchers propose emotion recognition algorithms that are specialized in children faces specifically [30], [37], [32]. In fact, domain adaptation is employed to improve the FER performance in children further than fine-tuning [37]. There are also special cases within child FER domain that needs thoroughly consideration: children with special needs. As their facial expressions differentiate from typically developed children, automatic FER needs to be explored specifically for them [14].

Since state-of-the-art algorithms depend on high number of parameters which require high computational power, lightweight solutions have gained importance to achieve efficient FER frameworks for resource limited devices. Researchers propose lightweight FER models using different approaches of model compression including model pruning [3], [29], [26], [9], [6], [38], knowledge distillation (KD) [5], [15] and neural architecture search (NAS) [33], [8], [17]. Li et al. [17], addressed the inadaptability issue caused by transfer learning from general classification networks into FER and solved by using NAS while providing a lightweight network. In NAS, a search space is defined to find the optimal neural network. Compared to other methods, defining search space requires elaborate consideration whereas other methods are simpler to apply. Cugu et al. employed teacher-student methodology to provide efficient FER models using Inception network as the teacher [5].

III. Methodology

The aim of this study is to propose an efficient FER system to be used within resource-efficient devices such as embedded devices, especially within social robots, and mobile devices. To this end, UWP and KD methods are tested to propose a robust lightweight FER network. Depthwise separable convolutional networks are utilized to be able to benefit from more convolution layers without increasing the number of parameters greatly. Details of aforementioned approaches are given below.

A. Unstructured Weight Pruning (UWP)

UWP approach is adopted to sparsify the weight matrix of the FER model. With this approach, redundant weights are removed by making the value of those weights zero. Throughout the model training, after each step (after processing each image batch), model weights are ordered according to their magnitude, then according to specified sparsity rate for the weight matrices, weights with least magnitudes are replaced with weights with zero magnitude to fulfill the sparsity requirement. This approach falls under the category of unstructured pruning as it makes the weight matrices of the model sparse by zeroing out unnecessary weights instead of removing them [18]. Since neurons are still present after the UWP, hardware and a framework that specialized in accelerating sparse matrix computation is needed. For the sake of achieving such feature, Nvidia's TensorRT framework [2] is used along with Ampere series graphics card (RTX 3060 Laptop GPU). Along with TensorRT, "tflite" module [1] of tensorflow is tested to be able to deploy developed model into mobile devices in future. "tflite" is specifically designated for mobile devices.

B. Knowledge Distillation (KD)

KD is a neural network training technique that aims to transfer knowledge from a complex network into a small network [10]. Complex network in this training strategy is called "teacher network", and small network is called as "student network" in the literature [5], [15]. "Distilling" knowledge from teacher is achieved by incorporating two objective functions in loss function: (1) Kullback-Leibler divergence between the soft (scaled by temperature) softmax outputs of the teacher and student; (2) cross entropy loss between the hard softmax output and the ground truth. We add a focal factor to cross entropy loss for label imbalance by using focal loss. So, overall loss for student is calculated as below:

$$L_{SL} = \alpha * L_F(P^S) + (1-\alpha) * L_D(P^{S'} || P^T),$$

where L_F represents focal loss between outputs of student network and ground truth, L_D represents distillation loss between student and teacher outputs. In loss arguments, $P^{S'}$ and P^T denotes softened probability distribution output of student and teacher network respectively, while P^S represents hard softmax output of student network. L_D is computed using Kullback-Leiber divergence:

$$L_D(P^{S'} || P^T) = \tau^2 * \sum_i^m P_i^T \log \frac{P_i^T}{P_i^{S'}},$$

where τ denotes temperature, m denotes number of classes. Temperature is applied to scale the loss function as suggested in [10], where usage of KD in neural networks is initially introduced. Softened probability distribution is achieved using temperature (τ) hyperparameter. Below, probability distribution computation is given for both teacher and student:

$$P^{S'} = \frac{e^{X_i/\tau}}{\sum_j^m e^{X_j/\tau}}, P^T = \frac{e^{Y_i/\tau}}{\sum_j^m e^{Y_j/\tau}}, P^S = \frac{e^{X_i}}{\sum_j^m e^{X_j}},$$

where X and Y refers to logits of student and teacher networks respectively.

Focal Loss [19] is used to take label imbalance into account by weighting each label inversely as their occurrence. Instead of balancing the data with augmenting and populating, this approach is pursued to make the model comparison with literature fair enough, since previous works using lightweight solutions did not populate the data. Focal loss is calculated as below:

$$L_F(P^S) = \alpha * (1 - P^S)^\gamma * L_{CE}(P^S),$$

where α is weighting factor for each class to deal with class imbalance, γ is utilized to make the model focus more on challenging samples instead of easy ones.

C. Depthwise Separable Convolutions for Efficient Deep Neural Networks

Instead of using standard convolutional layers, using depthwise separable convolutions is more advantageous in terms of number of parameters and the depth of the network. Despite regular convolutional layer, convolution operation is split into two parts as depthwise convolution and pointwise convolution. In depthwise convolution, convolution is applied to each channel separately, and resulting matrices are concatenated. Then, pointwise convolution takes place by applying 1x1 kernels to increase the resulting channel size. This variant of convolutional layer is mainly used in efficient networks. MobileNet also consist of this type of convolutional layers to preserve the computational efficiency while number of layers increases.

IV. Experiments

A. Datasets

AffectNet: AffectNet dataset [25] is used for UWP analysis, since it is widely used in the literature and it is an in-the-wild dataset. AffectNet images are crawled from the Internet. The dataset presents more challenge than posed facial expression dataset due to uncontrolled environment. A portion of the dataset is annotated by human experts which includes 287651 training images and 3500 test images. Emotion annotations are happiness, surprised, sad, fear, anger, neutral and contemptuous. Also, there are arousal and valence annotations, yet the scope of this paper does not include continuous domain, so only emotion labels are used. Contemptuous emotion is removed as many samples are wrongly annotated as stated in a previous study [41].

CK+: In contrast with AffectNet, CK+ [22] dataset contains images captured in a controlled environment. The dataset consists of image sequences captured second by second. First frame represents the neutral state of the subject, while the last frame represents the peak state of the expressed emotion. There are 7+1 (neutral) emotions captured as happiness, surprised, sad, fear, anger, neutral and contemptuous. There are 123 subjects and 592 image sequences in total. Contemptuous emotions are not used as the used child dataset, CAFE [21], do not include contemptuous emotion.

CAFE: CAFE [21] dataset contains posed images of children with age of 2 to 8 years. Each child is expected to express an emotion out of six basic emotions as sadness, happiness, surprise, anger, disgust and fear (and neutral state). Children express emotions both with their mouths open and closed. There are total of 1192 image samples.

B. Weight Pruning Analyses with AffectNet

One of the challenges that AffectNet dataset presents is label imbalance. Emotion frequencies highly differ from one label to another. Happiness and neutral expressions dominate other annotations by far. So, to overcome this distribution imbalance, downsampling is applied on happiness, neutral, sadness and anger, while disgust and fear samples are oversampled (surprise remain same in size). Resulting subset consist of 11409 happiness, 11409 neutral, 12729 sadness, 12441 anger, 14090 surprise, 12756 fear, 11409 disgust samples which results in 86241 images. This approach is adopted by [6] and named as CopyDB as it oversamples the images by "copying" them. Similar oversampling approaches are pursued in several more studies to make the AffectNet dataset balanced [41], [35], [34].

VGG-Face model [28] is used throughout the experiments, since it is pre-trained on a face recognition, which is a similar domain. To fine-tune the VGG-Face model for FER task, the model should adapt its weights to generate relevant high-level features according to the FER task by learning with the AffectNet dataset. Since first convolutional layers extract low level features such as edges and shapes, enabling last layers to be updated is reasonable. To decide whether to update last convolutional block or only last dense layer, experiments are made on CopyDB with using SGD and Adam optimizers. Results showed that when last convolutional layer block is used, validation accuracy can surpass 61% which outperforms only updating last classifier layer. Thus, in following experiments, last convolution block is updated (Fig. 1). Also, since SGD outperforms Adam when convolution layers are enabled to update, SGD is utilized in AffectNet experiments. Comparison of results can be seen in Table I. The best performing model will be used as a baseline model in experiments for CK+ and CAFE, namely Affect-FER. Yet, Affect-FER performed better with Adam in CK+ and CAFE datasets, as explained in upcoming sections. Initially, pruning operation ensure that 50% of the weights are removed, and when the pruning operation has ended, 80% of the weights are removed based on their values. Thus, sparsity rate gradually increases throughout the model training. As it can be seen in Table II, test

TABLE I
EXPERIMENT OF WEIGHT FREEZING ON COPYDB WITH VGG-FACE

Models	Optimizer	Learning Rate	Test Accuracy
Last Layer	SGD	1e-3	53%
Last Layer	Adam	1e-3	56%
Last Conv Block	SGD	1e-3	61%
Last Conv Block	Adam	1e-3	57%
Last Conv Block+1024FC	SGD	1e-3	60%
Last Conv Block+1024FC	Adam	1e-3	54%

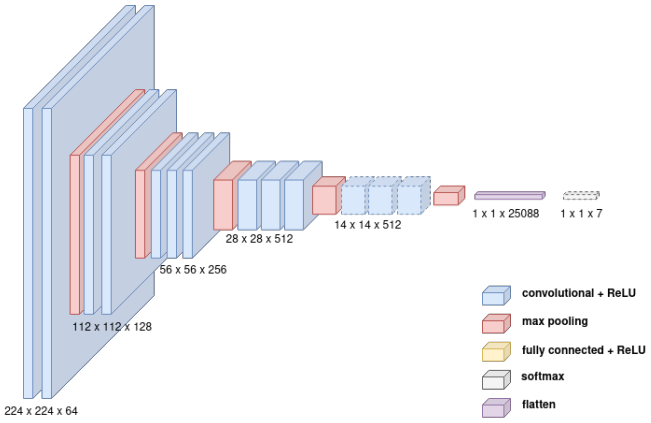

Fig. 1. VGG-Face Model. Layers with dashed lines are enabled for weight updating.

accuracy of the resulting models are comparable to each other, which indicates that UWP method does not affect the model performance negatively. In fact, the highest test accuracy is achieved by pruned model in 10^{-4} learning rate. Most balanced result is achieved by pruned model with 10^{-5} learning rate as train accuracy is the least among others. For experiments with pruned teacher in CK+, teacher will be fine-tuned using the pruned model with 10^{-5} learning rate, namely Affect-FER-P.

TABLE II
EXPERIMENT OF PRUNING ON COPYDB

Prune	Learning Rate	Train Accuracy	Test Accuracy
✓	1e-4	71.28%	**61.06%**
×	1e-4	69.23%	60.91%
✓	1e-5	**65.34%**	59.37%
×	1e-5	71.7%	60%

1) Storage Efficiency Analysis with AffectNet: Since pruning zeroes out redundant weights (setting their magnitude of redundant weights as zero), weights become sparser and the model can be compressed. Thus, models become resource efficient regarding the storage size. After applying the tensorflow strip pruning module and completing the training with pruning, the model is further compressed using gzip. The strip pruning module is required since it removes every variable that pruning uses just during the training phase. Pruning makes the model sparse, hence it is anticipated that it will be compressed more than the model without pruning. Model is changed into a "tflite" model after gzip compression. The model size is then further reduced after being quantized. Pruning reduced the zipped model size from 55 MB into 17 MB. Approximately, 70% of the model size is reduced while keeping the accuracy preserved.

C. Fine-tuning on CK+ Dataset

In KD, teacher network's weights are not updated, only student network's weights are learned throughout the training. Therefore, to have a robust teacher network, VGG-Face is used which is pre-trained on AffectNet dataset to perform FER during our UWP analyses. This choice brings several advantages in the developing of teacher network. Firstly, since the transferred domain is the same, the model would be converging faster than using a model that is pre-trained on another task. Also, datasets that are used in our KD experiments do not include many samples as in AffectNet dataset, thus over-fitting problem may arise if several top-level layers are enabled to update their weights during training. On the other hand, if only the classifier layer is enabled, model capability may not be enough to provide considerable performance if the model is not pre-trained in FER. By using the FER model trained in large dataset such as AffectNet, addressed challenges are overcome. In this regard, Affect-FER and Affect-FER-P are utilized.

Affect-FER is trained using Adam optimizer with learning rate as 10^{-4} is used. The model is set to be trained for 300 epochs. Yet, early stopping technique is used to determinate the training early if the validation loss does not improve for 50 epochs. Similarly, to prevent the model from overfitting, learning rate is diminished by a factor of 0.1 if validation loss does not decrease for 20 epochs. Ultimately, the model with the best validation accuracy is saved on epoch basis. The experiments showed that models have converged before 100 epochs, so for pruning experiments, model is trained for 100 epochs to accelerate training (both validation loss and accuracy is tested for model checkpoint in pruning, and results with minimum validation loss is given as it performed better in knowledge distillation).

Similarly with the previous studies [5], [17], [31] that utilized CK+, 10-fold cross validation is performed to report the model performance. Also, following the same approach as prior studies to preserve the comparability, last 3 frames of a facial expression sequence are utilized as "peak" frames, while the first frame is used to be utilized as "neutral" expressions.

Two different type of approaches are tested in 10-fold cross validation as subject-independent and randomly split folds. When preparing subject-independent folds, it is ensured that images belong to a single subject are not present in both training and validation folds. With such cross-validation strategy, it is aimed to evaluate the accuracy of learned patterns by the model regardless of differences in faces of individuals. In randomly split folds, there is no constraint as the former approach, thus a subject may appear in both training and validation fold. The most apparent downside of this approach is the pleasing results may hinder the model's habit of memorizing individual faces which diminishes the generalization capability of the model (i.e. instead of learning a general pattern regarding emotions, patterns belong to subjects may be memorized and the model may not perform reportedly good in new unseen samples). This approach can be utilized to validate the model's structure by measuring its capability to learn a pattern. For the teacher model in the KD architecture, best performing model out of the subject-independent strategy is used (Affect-FER). For the "pruned teacher" scenario, Affect-FER-P is further fine-tuned in the subject-independent dataset.

TABLE III
10-FOLD CK+ FINE-TUNING RESULTS (S.I: SUBJECT-INDEPENDENT, R.S: RANDOM SPLIT)

Pruned	Strategy	Accuracy	F_1	Precision	Recall
×	S.I	96.89±2.6%	95.75±4.4%	97.18	95.14
✓	S.I	95.29±2.2%	91.53±3.9%	94.46	90.52
×	R.S	99.61±0.4%	99.71±0.4%	99.85	99.59

Table III shows the detailed evaluation results of both subject-independent and random split strategies with accuracy and F_1 metrics. Pruned networks used Affect-FER-P as baseline model, while without pruning models used Affect-FER model as baseline. Results reveal that the Affect-FER model is capable enough to achieve nearly 100% in random split 10-fold cross validation strategy. Also, it can be seen that the degree of accuracy decline in subject-independent is negligible. UWP results in comparable scores compared to networks without pruning. Yet, F_1 score of the pruned networks are slightly less than others in average.

The difference in deviation between two different cross-validation strategies is evident as can be seen from the Table III. Subject-independent cross validation results tend to fluctuate due to varied train-validation splits in terms of label distribution and expressions of subjects. Several subjects express emotions different from the most of the other subjects which causes the model to fail to detect the expressed emotion as the extracted feature vector is distant from the learned pattern for the emotion. An example of this incident can be observed in Fig. 2. Most representative facial action unit constituents of the anger emotion are lowering the brows (Brow lowerer - AU4), tightening eye lid (Lid tightener - AU7) and tightening lips (Lip tightener - AU23). Lid tightener plays significant role in expressing anger emotion [36]. In left upper image within Fig. 2, subject only tightens her lips to express anger emotion, while lid tightening and brow lowering are not fully performed. In the left lower image, the subject lowers his brows in addition to lip tightening which lead the model to increase the probability given to anger emotion. Yet, without the expressive lid tightening, our model is unable to detect the anger emotion. On the right side, three aforementioned action units are performed, and the model can detect the anger emotion. Conclusively, this shows the expression variance between the subjects, which causes relevantly higher standard deviation in subject-independent strategy compared to random split strategy.

To achieve such performance in subject-independent, on-the-fly data augmentation is utilized. Within the data augmentation techniques, rotation, shearing, shifting, flipping are utilized as the pose can differ, but more importantly brightness modifier is utilized since the main visual variation between images in the CK+ is the brightness difference. Whole configuration is listed in Table IV. Data augmentation is only applied to the training batches during training, and the total number of samples is not changed as the augmented samples are replaced with the original ones. This act is done intentionally to preserve the comparability of the model with literature by keeping the total number of samples the same. Trained networks for subject-independent strategy are utilized as teacher networks in the knowledge distillation.

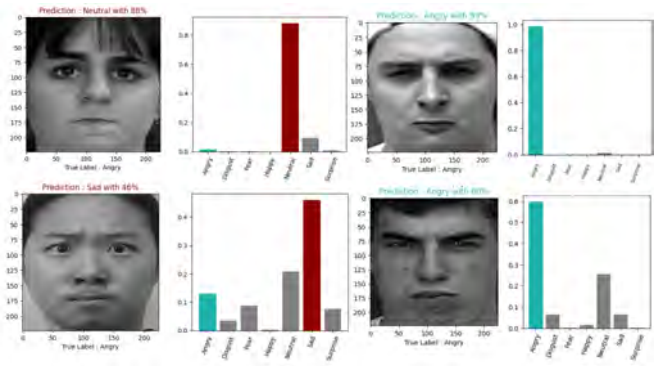

Fig. 2. Visual analysis of falsely predicted images

TABLE IV
DATA AUGMENTATION TECHNIQUES

Technique	Degree/Range
Rotation	-20,+20
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.1
Brightness Range	[0.5,1.5]
Zoom Range	0.1
Horizontal Flip	✓

D. Knowledge-Distillation on CK+ Dataset

A 10 set of teacher models are selected out of best performing Affect-FER cross validation experiments. Average accuracy and standard deviation of those 10 models are reported in Table III. Since subject-independent strategy is followed in all further experiments, models trained with subject-independent strategy is used. For "pruned teacher" scenario, pruned networks are utilized as teacher networks. This model will be utilized to distill knowledge to a more shallow and lightweight neural network, as described in the methodology section.

For the student network, two network architectures are proposed: LITEFER-V1 and LITEFER-V2. LITEFER-V1 is designed to be similar in a sense with MicroExpNet [5], yet it is more lightweight which results in 18.9K (74.09 KB) parameters. Even though the end goal is to propose lightweight models, accuracy of the models are significant as the models should have adequate capability to result in considerable FER performance. Architectural details of the LITEFER-V1 are given in Table V. Each Conv2D block is followed by a batch normalization layer and ReLU activation function. Flatten operation is performed after the maximum pooling operation. ReLU is utilized after the first fully connected layer. Padding is performed to preserve the input shape before the convolutions. Along with additional differences to the architecture compared to MicroExpNet

[5], dropout layer is included as shallow network can easily overfit and become biased towards a class, especially in imbalanced class scenarios like in our case.

In addition to the LITEFER-V1, a more balanced network is proposed regarding considerably low complexity and high capability, named as LITEFER-V2. In this model, instead of standard convolutions, depthwise separable convolutions are utilized. This technique is suited well for a robust lightweight model which has both high capability and considerably small number of parameters. LITEFER-V2 model architecture is visualized in Fig. 3 with layer details as caption. After each depthwise separable convolution block; batch normalization, max pooling with 2x2 shape and 2 stride followed by ReLU activation is performed. Here the depthwise separable convolution block includes a single depth wise convolution and a standard convolution as illustrated in the methodology section. As can be seen from the architecture, with the benefit of using depthwise separable convolutions, the model architecture can contain four convolution blocks without increasing the parameter number dramatically. Different from LITEFER-V1, no paddings are performed to preserve the input shape in convolutions which results in decreased size of convolution output in order to decrease parameters in further layers and compensate the computational load of more deep network. This model contains 113.98K (445.24KB) parameters.

TABLE V
ARCHITECTURE DETAILS OF LITEFER-V1

	LITEFER-V1
Layer	Filter Shape,Stride
Conv2D	4x4x32,4
Conv2D	3x3x16,4
MaxPool2D	2x2,2
Dense	16
Dropout (0.5)	-
Dense	7

Proposed variants of LITEFER models are trained within KD training scheme for 600 epochs with using Adam (early stopping with 100 epochs). Best performing results are achieved with 10^{-4} and 10^{-5} for LITEFER-V1 and LITEFER-V2 respectively. For all LITEFER model variants, hyperparameter optimization is performed with alpha and temperature values as 0.3, 0.4 and 3, 10 respectively. Best performing learning rate for each variant is decided on iterative preliminary experiments to solely focus on hyperparameter search of alpha and temperature. Comparative results are given in Table VI. As can be seen from the Table VI, LITEFER-V2 (L-V2) outperforms the other variant of the model both in accuracy and especially F_1 which is critical metric for the task, since the CK+ is an imbalanced data. There is no meaningful correlation between accuracy and temperature values as previously reported [5]. LITEFER-V2 performed best when alpha value is 0.3 and temperature is 3 while LITEFER-V1 (L-V1) performed its best when alpha is 0.4 and temperature is 3. "Pruned teacher" scenario is also tested using LITEFER-V2 with alpha value as 0.3 and temperature as 3. Resulting student model achieved 87.43% accuracy and 78.24% F_1 score. Since pruned teacher did not improve performance, further analyses were performed by using the teacher network without pruning. Also, pruning is omitted in CAFE experiments. This can be related with high sparsity rate that last convolutional layers have. For convenience, LITEFER-V2 is named as LITEFER.

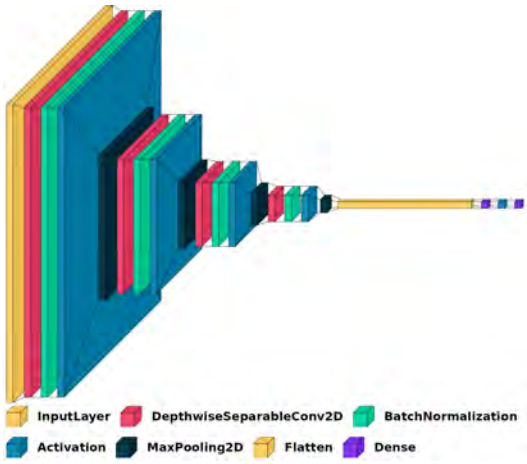

Fig. 3. Visual representation of LITEFER-V2 architecture: 1st Conv: 7x7x32, 2nd Conv: 9x9x64, 3rd Conv: 3x3x32, 4th Conv: 5x5x64, MaxPool2D: (2,2) with 2 stride, 1st Dense: 16 Neurons, 2nd Dense: 7 Neurons

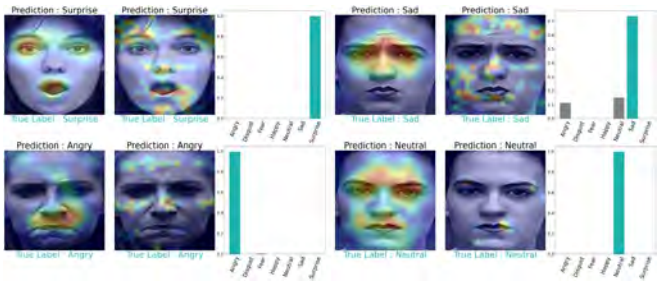

Fig. 4. GradCAM analysis of samples correctly classified by LITEFER (among every pair of face images; left one is the output of the teacher model, and right one is the output of the LITEFER)

GradCAM is utilized to visualize facial regions that LITEFER and teacher model focuses. GradCAM heatmaps and predictions are visualized in Fig. 4 and Fig. 5. As can be seen from Fig. 4, even though predicted labels are same, GradCAM heatmaps are not identical between LITEFER and teacher model. Instead of focusing on a subregion of the face, LITEFER scatters region of interest by focusing on diverse and smaller regions. This outcome can be explained by the number of parameters that LITEFER has. LITEFER aims to classify 7 different emotion with small number of parameters which would lead LITEFER to capture broader patterns to represent the general distribution of the data. On the other hand, this scattered region of interests (ROIs) may reveal interesting patterns that even ground truth does not provide. LITEFER predicts surprise emotion in the upper left image pair in Fig. 5 by focusing on eyebrows which is not focused by teacher model. Although ground truth

is neutral, subject appears as surprised in this image. Yet, due to the same tendency of scattering focus, LITEFER misclassified the happy sample as fear even though the mouth region is focused. ROIs around eyebrows lead the model to portray the subject as she raises her eyebrows. In this sample, teacher model fails also since patterns around mouth regions could not be captured. Overall, this attribute of capturing broad patterns in the lightweight model can lead to false predictions, although model sometimes benefit from this attribute. To improve the robustness of the lightweight models, labels can be categorized as positive negative neutral to match the broad patterns in future studies.

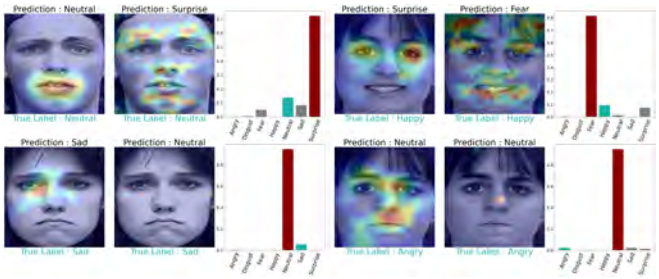

Fig. 5. GradCam analysis of samples misclassified by LITEFER (among every pair of face images; left one is the output of the teacher model, and right one is the output of the LITEFER)

TABLE VI
COMPARATIVE RESULTS OF LITEFER VARIANT MODELS (L-V1: LITEFER-V1, L-V2: LITEFER-V2, TEMP: TEMPERATURE, ACC: ACCURACY, PREC: PRECISION, REC: RECALL)

Model	Alpha	Temp.	Acc.	F_1	Prec.	Rec.
L-V1	0.3	10	81.69%	67.82%	73.39%	68.01%
L-V1	0.3	3	81.66%	68.23%	73.82%	68.22%
L-V1	0.4	3	82.49%	71.91%	78.46%	71.18%
L-V1	0.4	10	81.07%	67.93%	73.90%	68.12%
L-V2	0.3	10	88.88%	81.25%	87.41%	80.94%
L-V2	0.3	3	89.69%	82.56%	87.64%	81.99%
L-V2	0.4	3	88.78%	80.74%	86.31%	80.09%
L-V2	0.4	10	89.34%	81.18%	86.38%	80.59%

Our best performing LITEFER model on CK+ dataset is compared with state-of-the-art studies which propose lightweight solutions for this dataset. As can be seen from the Table VII, our model is the most lightweight model after MicroExpNet [5]. Yet, our LITEFER model compensates this difference with the accuracy near 90%. Auto-FERNET model [17] excels in achieves superior accuracy in CK+ using neural architecture search (NAS). Since NAS can take excessive amount of time to build a robust search space, we do not opt for that approach as our aim is to only validate the student network architecture, on commonly used adult facial expression dataset, that would be developed further for children facial expressions.

E. Fine-tuning on CAFE Dataset

In order to develop a lightweight FER model specialized for children, Affect-FER is further fine-tuned with CAFE [21]. The fine-tuned model is used as the teacher model in the KD training scheme for CAFE dataset. In fine-tuning, model is trained for 100 epochs, as model converges before 100 epochs in experiments with CK+. Training with CAFE also revealed that the model converges before 100 epochs. Early stopping approach is similar, but stopping criteria is reduced to 30 epochs. On-the-fly data augmentation is used as in CK+ dataset showed in Table IV. Resulting teacher model achieved $89.86\% \pm 0.1\%$ accuracy and $88.46\% \pm 0.5\%$ F_1 score. Compared to prior works teacher network achieved comparable score. Simões et al. [32] adapted ConvNeXt model to CAFE dataset, yet number of parameters is not provided in the study [32]. Thus, least number of parameters for that family is used within the table.

TABLE VII
COMPARATIVE RESULTS OF LITEFER MODELS WITH PREVIOUS STUDIES ON CK+ DATASET

Model	Model Size	Compression	Classes	Accuracy
LITEFER	445.24KB	Know. Dist.	7	89.69%
[31]	5.70MB	UWP	8	99.68%
[5]	65K	Know. Dist.	8	81.66%
[17]	2.1MB	NAS	7	98.89%

TABLE VIII
COMPARATIVE RESULTS OF CAFE DATASET

Model	Model Size	Accuracy	F_1
LITEFER	113.98K (445.24KB)	77%	74.2%
Our Teacher	14.89M (56.80MB)	89.86%	88.46%
[3]	1.27M	-	61.34%
[32]	29M	85.92%	-

F. Knowledge-Distillation on CAFE Dataset

Teacher model is not pruned as there is not any strong indication towards accuracy improvement in CK+ experiments. To facilitate the KD process, student model's weights are initialized with using the best performing student model within the experiments with the CK+ dataset. Best performing LITEFER, in CK+ experiments, is achieved with alpha and temperature as 0.3 and 3 respectively. Out of 10 saved models (i.e. since the experiment is 10-fold CV), the one with the most label balanced validation set is selected for weight initialization to improve the generalizability of the model for child facial expression adaptation. Adam is used with 10^{-5} learning rate as 10^{-4} learning rate lead model to overfit the training data. The resulting model achieved 77.1% accuracy and 74.2% F_1 score.

The confusion matrix of LITEFER model is shown in Fig. 6. It can be seen that most misclassified classes are angry, sad and disgust. Angry is mostly misclassified as disgust emotion (32%). Disgust is also mostly classified as angry emotion (25%). This result can be understandable as both class represents negative emotions and expressions can be similar. Sad emotion, on the other hand, is mostly misclassified as neutral emotion. This can be due to the fact that neutral state of several subjects may also be inferred as sad emotional state. Since CAFE dataset providers do not allow for image publishing, samples are not provided.

There is only a single study [3] that proposes lightweight FER model for children by using weight pruning. Details of pruning type is not provided in the study (i.e. whether pruning is performed on filters or individual weights). They achieve 61.34% with the smallest model, also shown in Table VIII. It should be noted that the model they proposed were not trained in CAFE dataset, yet only tested on CAFE.

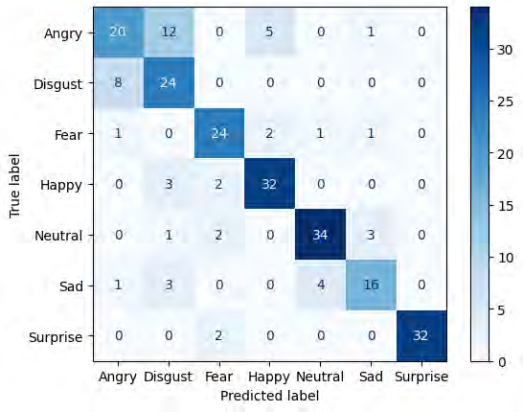

Fig. 6. Confusion matrix of LITEFER on CAFE dataset

G. Inference Speed Analysis

Inference speed of the proposed model LITEFER is measured and compared with complex teacher model in both standard keras (.h5) and TensorRT format. For standard keras format, both CPU and GPU performances are given in Table IX. This latency evaluation is measured on average of 100 iterations of single image prediction. As can be seen from the Table IX, LITEFER outperforms complex teacher network regarding inference latency and throughput. Results also shows that LITEFER operates faster on CPU compared to GPU when used in keras format. This can be explained due to less computations needed to perform inference with LITEFER along with being unable to exploit parellelism provided by GPU in iterative prediction of single image. It can be clearly seen that TensorRT optimizes neural networks such that inference becomes faster. Still, LITEFER results in nearly processes 17 frames more than the teacher network.

TABLE IX
INFERENCE SPEED OF THE MODELS USING SINGLE IMAGE PREDICTION

Model	Device	Format	Latency	Throughput
LITEFER	GPU	Keras	**37 msec**	**26.82 FPS**
Our Teacher	GPU	Keras	47 msec	21.13 FPS
LITEFER	CPU	Keras	**33 msec**	**30.09 FPS**
Our Teacher	CPU	Keras	96.3 msec	10.39 FPS
LITEFER	GPU	TensorRT	**5.8 msec**	**173.82 FPS**
Our Teacher	GPU	TensorRT	6.4 msec	156 FPS

Along with single image prediction, to reveal the maximum speed performance of the LITEFER through TensorRT, inference speed is also measured with batch predictions using data loaders. As can be seen from Table X, LITEFER outperforms teacher also in batch prediction performance. Single batch consisted of 128 images in this evaluation. Average throughput denotes number of images predicted in a second.

TABLE X
INFERENCE SPEED OF THE MODELS USING BATCH PREDICTION

Model	Device	Format	Batch Latency	Throughput
LITEFER	GPU	TensorRT	**10 msec**	**3213 FPS**
Our Teacher	GPU	TensorRT	106.2 msec	301 FPS

V. CONCLUSION

In this study, KD and UWP methods are used to achieve lightweight FER models. It has been shown that with using UWP method, FER models can eliminate redundant individual weights. In both AffectNet and CK+ datasets, the model achieved nearly 80% sparsity rate in topmost convolution and dense layers with preserving the accuracy. Since this approach does not directly remove the parameters from the network, computational load stays the same. To improve the inference efficiency along with storage efficiency (i.e. pruned FER models are compressed better), GPU should have the accelerated sparse-matrix computation feature and pruning should be done in structured manner.

KD approach is utilized to directly acquire a smaller model by transferring knowledge from a complex model. Focal loss is integrated within the general loss to improve robustness towards label imbalance. Two variants of FER model are proposed. LITEFER-V2 outperformed LITEFER-V1 by leveraging the depthwise separable convolution layers. LITE-FER achieved 89.69% and 77% in CK+ and CAFE respectively in k-fold cross validation strategy. LITE-FER only consists of 113.98K parameters (445.24 KB) which makes the model resource-efficient.

The effect of joint usage of KD and UWP method on FER is also evaluated by using a pruned teacher model within the teacher-student training paradigm in KD method. The motivation of this approach is to avoid the distillation of overlearned patterns from the teacher model to the student. However, pruned teacher scenario could not improve the existing LITEFER result, possibly due to the high sparsity rate. Thus, we preferred not to use this approach in children FER experiments.

Chiefly, results showed that among previously proposed lightweight models in the literature, our proposed model ("LITE-FER") results in comparable accuracy, as well as computational and memory efficiency. Proposed LITE-FER model can be implemented in resource-limited devices. In future, proposed model can be used in child-robot interactions or child-computer interactions. With this way, subtle expressions of children that cannot be detected by observers can be detected and behavioral relations can be extracted from the subtle reaction of the children. This model especially would be helpful towards indicating mood of the children with special needs during a therapeutical sessions. To this end, this study can be extended to adapt the model towards children with special needs such as autism spectrum disorder.

REFERENCES

[1] tensorlite, framework for machine learning models in mobile and edge devices. https://www.tensorflow.org/lite. Accessed: 2024-04-22.

[2] Tensorrt, an sdk for high-performance deep learning inference by nvidia. https://developer.nvidia.com/tensorrt. Accessed: 2024-04-22.

[3] A. Banerjee, O. Mutlu, A. Kline, S. Surabhi, P. Washington, and D. Wall. Training and Profiling a Pediatric Facial Expression Classifier for Children on Mobile Devices: Machine Learning Study. *JMIR Formative Research*, 7, 2023.

[4] F. Z. Canal, T. R. Müller, J. C. Matias, G. G. Scotton, A. R. de Sa Junior, E. Pozzebon, and A. C. Sobieranski. A survey on facial emotion recognition techniques: A state-of-the-art literature review. *Information Sciences*, 582:593–617, 2022.

[5] I. Cugu, E. Sener, and E. Akbas. Microexpnet: An extremely small and fast model for expression recognition from face images. In *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2019.

[6] H. Gao, S. An, J. Li, and C. Liu. Deep balanced learning for long-tailed facial expressions recognition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11147–11153, 2021.

[7] M.-I. Georgescu, R. T. Ionescu, and M. C. Popescu. Local learning with deep and handcrafted features for facial expression recognition. *IEEE Access*, 7:64827–64836, 2018.

[8] X. Han. Automatically design lightweight neural architectures for facial expression recognition. In *Proceedings of the 2023 6th International Conference on Machine Vision and Applications*, ICMVA '23, page 101–105, New York, NY, USA, 2023. Association for Computing Machinery.

[9] Z. He and X. Qin. Analysis of facial expressions in class based on lightweight convolutional neural network. In *2022 International Conference on Industrial Automation, Robotics and Control Engineering (IARCE)*, pages 68–74, 2022.

[10] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network, Mar. 2015. arXiv:1503.02531 [cs, stat].

[11] S. Hossain, S. Umer, R. K. Rout, and M. Tanveer. Fine-grained image analysis for facial expression recognition using deep convolutional neural networks with bilinear pooling. *Applied Soft Computing*, 134:109997, 2023.

[12] M. Hu, Q. Chu, X. Wang, L. He, and F. Ren. A two-stage spatiotemporal attention convolution network for continuous dimensional emotion recognition from facial video. *IEEE Signal Processing Letters*, 28:698–702, 2021.

[13] P. Kumar, A. Kishore, and R. Pandey. Emotion recognition of facial expression using convolutional neural network. In J. S. Raj, A. Bashar, and S. R. J. Ramson, editors, *Innovative Data Communication Technologies and Application*, pages 362–369, Cham, 2020. Springer International Publishing.

[14] A. Landowska, A. Karpus, T. Zawadzka, B. Robins, D. Erol Barkana, H. Kose, T. Zorcec, and N. Cummins. Automatic Emotion Recognition in Children with Autism: A Systematic Literature Review. *Sensors*, 22(4), 2022.

[15] K. Lee, S. Kim, and E. Lee. Fast and Accurate Facial Expression Image Classification and Regression Method Based on Knowledge Distillation. *Applied Sciences (Switzerland)*, 13(11), 2023.

[16] B. Li and D. Lima. Facial expression recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering*, 2:57–64, 2021.

[17] S. Li, W. Li, S. Wen, K. Shi, Y. Yang, P. Zhou, and T. Huang. Auto-FERNet: A Facial Expression Recognition Network with Architecture Search. *IEEE Transactions on Network Science and Engineering*, 8(3):2213–2222, 2021.

[18] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, Oct. 2021.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[20] Y. Liu, C. Feng, X. Yuan, L. Zhou, W. Wang, J. Qin, and Z. Luo. Clip-aware expressive feature learning for video-based facial expression recognition. *Information Sciences*, 598:182–195, 2022.

[21] V. LoBue and C. Thrasher. The Child Affective Facial Expression (CAFE) set: validity and reliability from untrained adults. *Front. Psychol.*, 5, Jan. 2015.

[22] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101, 2010.

[23] S. Minaee, M. Minaei, and A. Abdolrashidi. Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, 21(9), 2021.

[24] R. Miyoshi, N. Nagata, and M. Hashimoto. Enhanced convolutional LSTM with spatial and temporal skip connections and temporal gates for facial expression recognition from video. *Neural Computing and Applications*, 33(13):7381–7392, July 2021.

[25] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2019.

[26] F. Najafi, P. Wang, and D. O. Nyabuga. Hybridei: Smartly face detection system in resource constrained edge environment. In *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, volume 1, pages 1–6, 2021.

[27] X. Pan. Fusing hog and convolutional neural network spatial–temporal features for video-based facial expression recognition. *IET Image Processing*, 14(1):176–182, 2020.

[28] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015.

[29] A. Pascual, E. Valverde, J.-I. Kim, J.-W. Jeong, Y. Jung, S.-H. Kim, and W. Lim. Light-FER: A Lightweight Facial Emotion Recognition System on Edge Devices. *Sensors*, 22(23), 2022.

[30] A. Qayyum and I. Razzak. Deep residual neural network for child's spontaneous facial expressions recognition. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, S+SSPR 2020, Padua, Italy, January 21–22, 2021, Proceedings*, page 282–291, Berlin, Heidelberg, 2021. Springer-Verlag.

[31] S. Saurav, A. Saini, R. Saini, and S. Singh. Deep learning inspired intelligent embedded system for haptic rendering of facial emotions to the blind. *Neural Computing and Applications*, 34(6):4595–4623, 2022.

[32] G. Simões, A. Lopes, C. Carona, R. Pereira, and U. J. Nunes. Deep-learning based classification of engagement for child-robot interaction. In *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 112–117, 2023.

[33] M. Verma, M. Mandal, S. Reddy, Y. Meedimale, and S. Vipparthi. Efficient neural architecture search for emotion recognition. *Expert Systems with Applications*, 224, 2023.

[34] K. Wang, X. Peng, J. Yang, S. Lu, and Y. Qiao. Suppressing uncertainties for large-scale facial expression recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6896–6905, 2020.

[35] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *IEEE Transactions on Image Processing*, 29:4057–4069, 2020.

[36] M. Wegrzyn, M. Vogt, B. Kireclioglu, J. Schneider, and J. Kissler. Mapping the emotional face. How individual face parts contribute to successful emotion recognition. *PLOS ONE*, 12(5):e0177239, May 2017. Publisher: Public Library of Science.

[37] M. A. Witherow, M. D. Samad, N. Diawara, H. Y. Bar, and K. M. Iftekharuddin. Deep adaptation of adult-child facial expressions by fusing landmark features. *IEEE Transactions on Affective Computing*, pages 1–12, 2023.

[38] G. Xu, H. Yin, and J. Yang. Facial expression recognition based on convolutional neural networks and edge computing. In *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, pages 226–232, 2020.

[39] Y. Zhang, X. Zou, S. Yu, L. Huang, W. Wang, S. Zhao, and X. Wang. Dnn-cbam: An enhanced dnn model for facial emotion recognition. *Journal of Intelligent and Fuzzy Systems*, 43(5):5673 – 5683, 2022.

[40] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron (vol 13, pg 893, 1999). *International Journal of Pattern Recognition and Artificial Intelligence*, 14:257–257, 03 2000.

[41] Z. Zhao, Q. Liu, and F. Zhou. Robust lightweight facial expression recognition network with label distribution training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3510–3519, May 2021.

[42] Z. Zheng, X. Li, J. Barnes, C.-H. Park, and M. Jeon. Facial expression recognition for children: Can existing methods tuned for adults be adopted for children? In M. Kurosu, editor, *Human-Computer Interaction. Recognition and Interaction Technologies*, pages 201–211, Cham, 2019. Springer International Publishing.