Real-time Facial Expression Recognition using Convolutional Neural Network on Mobile Device

Erick[1], Keiko Kimberly Octavina[2], Gede Putra Kusuma[3]

[1,2,3] *Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480*

E-mail address: erick007@binus.ac.id, keiko.octavina@binus.ac.id, inegara@binus.edu

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Implementation of facial expression recognition can help improve human-computer interaction in various aspects, such as education, entertainment, health and more. In this study, convolutional neural networks (CNN) were designed and implemented to recognize facial expression. The FER2013 dataset was used to train the models which have seven different emotion classes: anger, disgust, fear, happiness, sadness, surprise and neutral. The purpose of this study is to compare the computational load of 23 different CNN models for the facial expression recognition task on a mobile device. In this study, we compare ResNet101V2, MobileNet, and EfficientNetV2B3 as the top three candidate models among the other 23 models that we have tried, achieving the highest overall accuracy on the testing set. The highest overall accuracy is achieved by the EfficientNetV2B3 model at 61.9%, while the MobileNet model has the lowest overall accuracy at 58.8%. We then compare computational load based on average inference time, peak CPU usage, and peak memory usage on a mobile device. The results show that MobileNet has the lowest computational load but the lowest overall accuracy. On the other hand, EfficientNetV2B3 has the highest overall accuracy with less computing load than MobileNet. Therefore, we recommend EfficientNetV2B3 for real-time facial expression recognition using CNN on mobile devices.

Keywords: ***Facial Expression Recognition, Convolutional Neural Networks, FER2013 Dataset, Mobile Devices, EfficientNetB2V3 Model***

1. INTRODUCTION

Humans normally communicate verbally, but they may also express their emotions and highlight specific portions of their speech through body language. In human social interactions, Facial expressions, is an essential component of communication, are one of the keyways humans convey their emotions. Natural human-machine interfaces may make use of face detection, which can also be applied in behavioral science and therapeutic conditions [1].

The existence of computer vision is expected to be able to help detect some of those expressions in the most effective way. Applications that can recognize facial expression can help users communicate more effectively and receive more personalized services [2]. Recently, mobile devices with embedded cameras have become an important in people's lives. Numerous applications for both personal and professional use have emerged from the widespread and increased use of mobile cameras built into smartphones. Which aroused the interest in human-computer interaction through user's facial expression.

One example of FER implementation is the interpretation of a player's emotions in a game, which can significantly influence the gaming experience. Facial expression recognition (FER) serves as a method to detect a player's emotions and dynamically adjust the level of difficulty in the game. For example, the game's difficulty variables can be decreased when the player exhibits a relaxed facial expression in response to signs of stress or anxiety. This can be achieved through the integration of FER technology into the game.

Therefore, FER has become essential especially to improve human-computer interaction (HCI) in areas such as autopilot, education, surveillance, and psychological analysis in computer vision [3]. In order to automate human emotion recognition (ER) in smart homes, smart hospitals, and smart cities, the internet of things (IoT) has to get better. In the age of HCI (human-computer interaction), it is critical to have computational tools that can recognize human emotions automatically. As human-

machine interaction grows, computers must be able to accurately interpret human emotion in order for it to react appropriately to a given situation. Consequently, the CNN model was developed and enhanced to perform Facial Expression Recognition tasks.

Nonetheless, these CNN models are limited to certain specific scenarios due to their network's complexity and large number of parameters. This condition makes the models limited due to high hardware requirements. Mobile terminals and embedded devices have challenges in meeting their hardware requirements for implementing the model [4]. Therefore, it motivates us to do experiments and comparation CNN models performance on mobile devices.

In this study we conduct a Facial Expression Recognition, which is the ability to recognize facial expression that transmit fundamental emotions like fear, happiness, disgust, and others. FER2013 dataset is one of the emotion recognition datasets that encompasses the challenging difficulties that will be used in this study. The primary difficulties that FER from other image classification tasks are the similarities and contrasts between human facial expressions within and between classes [5]. There are challenging conditions in FER especially when the subjects are posed at certain angles or when the subject faces are partially covered by other objects.

According to recent studies, rapid progress has been achieved in recent years for the automatic recognition of facial emotions because of current developments in computer vision and machine learning approaches. However, recent studies mainly focusing on improving model for better overall accuracy to achieve state-of-the-art results [6], [7], [8], [9].

Therefore, the focus of this study is to compare and evaluate several model's performance such as evaluation of the overall accuracy on PC computational load on mobile in recognizing facial expressions that are trained on FER2013 Dataset. Some models that we have tried to do Facial Expression Recognition task on FER2013 dataset are DenseNet, EfficientNetV1, EfficientNetV2, Inception, MobileNet, MobileNetV2, ResNet, ResNetV2, and Xception with the same data augmentations and experiments. Finally, we choose the top three models with highest overall accuracy on test set, which are Mobile-Net, Resnet101V2, EfficientNetV2B3 with overall accuracy 58.8%, 60.3%, and 61.9% for further comparison.

One of the models that achieve highest overall accuracy is EfficientNetV2B3. EfficientNetV2 is a new family of CNN with faster training speed and a more efficient number of parameters. EfficientNetV2 has shown great improvement in times, faster in training speed and better parameter efficiency. MobileNet also has achieved the top three highest overall accuracy on our experiments. MobileNet was implemented with attention module and dropout layer to enhance model's ability in feature extraction and prevent overfitting [10]. One of the top three models with the highest overall accuracy is Resnet101V2. Residual Neural Network or we usually call ResNet was created to address the neural network degradation problem. Through cross-layer feature fusion, ResNet further improves its capacity to extract network features, and as the network become complex, network performance progressively gets better[11].

As we mentioned before, we selected the top three models with the highest overall accuracy on the test set for further comparison. Further comparison will measure by calculating the average inference time in milliseconds (ms), the utilization of the Central Processing Unit (CPU), and the Random Access Memory (RAM) consumption in Megabyte (MB) of those top three models on mobile device. The purpose of this research is to evaluate and compare lightweight CNN models that can perform well on mobile devices to counter the problem related to limitation in CNN models due to hardware requirements.

2. RELATED WORKS

A. Previous works related to Image Classification

Deep learning has been implemented widely in the case of computer vision. One of the challenges encountered is Image Classification using deep learning. Many images classification task that has been solved using Convolutional Neural Network (CNN), a class of deep learning models to process grid-like data such as images. As time goes by, with the growth of computing resources and datasets CNN has developed as an applicable tool for feature extraction and image classification. CNN has been successfully and efficiently used in variety of pattern and image recognition, such as, gesture recognition [12], face expression recognition [3], [6], [7], [10], [13], object classification [4], and generating scene description.

CNN commonly consists of convolutional, pooling, and fully connected layer built together to help extract features, reduce dimensionality, while maintaining important features. Varied techniques have been implemented for further performance, for instance Rectified Linear Unit (ReLU) activation has replaced Sigmoid activation function to prevent gradient dispersion and speed up training [14] and has been used broadly for image classification task. Also, many pooling methods implemented to reduce the input sample size and helps generalization [15].

CNN model LeNet-5 was successfully implemented as the first CNN architecture introduced in 1998 to helped handwritten recognition case [16]. As time goes

by CNN have developed a lot with the success of AlexNet in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) to classify high-resolution images in 2012 [17]. Researchers have been creating and improving different algorithms to optimize the training process and achieve better results.

A research conduct by [18], evaluated ResNet (34B, 34C, 50, 101, 152), VGG, PReLu-Net, GoogLeNet, and BN-Inception on ImageNet validation set the results are ResNet models has lower error percentage than another model. The 50/101/152-layer ResNets are more accurate than the 34-layer ones by considerable margins. Though ResNet 101-layer and 152-layer ResNets have three more-layer blocks which make them deeper (11.3 billion FLOPs), ResNet 101-layer and 152-layer still have lower complexity than VGG 16/19 networks (15.3/19.6 billion FLOPs).

Other research from [19], propose a new family of convolutional networks that have faster training speed and better number of parameters efficiently. In this study, EfficientNetV2 has outperformed previous models on ImageNet with 85.7% top-1 overall accuracy while training 3x - 9x faster and being up to 6.8x smaller than previous model. EfficientNetV2 architecture added fused MB-Conv in the early layers, this model also prefers smaller 3x3 kernel size and removes the last stride-1 stage in the original EfficientNet.

Though there is no standard guideline on deciding which optimizer, but the results have shown applicable optimization algorithm that can create better model's results [20]. Other technique implementation that can enhance model's performance are dropout regularization and data augmentation as the technique for manipulating data without losing the essence of the data to help prevent overfitting [21], [22].

B. Previous works related to Facial Expression Recognition

CNN has been developed significantly with the success of AlexNet [23] in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) to classify high-resolution images. The availability of open-source large datasets has helped the improvement of the experiment. Some of the well-known dataset for FER task is JAFFE [24], FER2013 dataset [6], FERPlus dataset [26], AffectNet dataset [27], Extended CohnKanade Dataset (CK+) dataset [28], etc. In this study we conduct an experiment with FER2013. FER2013 was introduced at ICML 2013, it consists of various faces in terms of age, face, pose, and other factors. FER2013 has become one of the benchmarks in comparing model performance in FER.

Many CNN models have developed and achieved results starts from between 65% and 72.7% [29], [30], [31], [32], [33], [34]. This study from [29], have experimented with trained three separate CNNs and ensembled them to enhance the performance and their best single-network overall accuracy is 62.44% on FER2013 dataset.

Another novel CNN model has started to improve especially in FER task with FER2013 dataset, consists of a 34-layer ResNet [33] from [35], but without the initial convolutional and pooling block, combined with dropout after the final pooling layer. Although other models compared in that work have fewer parameters, ResNet shows better overall accuracy in 72.4%. This study also experimented with several architectures on FER2013 reaching 71.6% overall accuracy with Inception and 72.7% with VGG.

Other study by [2], have conducted several models such as MobileNetV1, MobileNetV2, MobileNetV3-small, have achieved overall accuracy with 85.40%, 77.18%, and 73.24% on FERPlus dataset (elicited from FER2013). Those three MobileNet series models also experimented on (RAF-DB) Real-world Affective Faces Database consists of seven emotion categories (except for contempt) with performance 81.62% for MobileNetV1, 67.77% for MobileNetV2, and 68.29% MobileNetV3-small. All these models can be applied to mobile and embedded devices.

A study from [36] have evaluated that MobileNetV1 and MobileNetV2 have fewer parameter than other models such as Xception and VGG models on two datasets (FER2013, AffectNet). This study compared several models on FER2013 dataset, the results showed that Xception achieved overall accuracy of 67.4%, MobileNet and MobileNetV2 have achieved overall accuracy of 61.8% and 62.1%, and DenseNet-40 with overall accuracy 66.6%, while VGG-pretrain achieved highest overall accuracy with 70.1% but VGG-pretrain have quite big number of parameters (9.4 M). All the models tested have implemented the same preprocessing protocol for comparison which is histogram equalization, and normalization using mean and standard deviation on training pixels. CNN model usually has large computational load and memory requirements. Therefore, lightweight deep neural network has developed to encounter this problem called MobileNet series. Which of these models can be run on mobile and embedded devices.

Another study conducted by [37] experimenting FER using DenseNet model that is built with one convolution

layer, three dense blocks, and one FC layer. The research resulted with overall accuracy of 63.5% for the DenseNet trained with FER2013 dataset and 85.4% for the KDEF dataset.

An experiment conducted by [9] using a modified VGG11 with batch normalization on FER2013 dataset and thoroughly tune all hyperparameters to create an optimized model for facial emotion recognition. The best overall accuracy of 73.06% is obtained by experimenting with different optimizers and learning rates, surpassing earlier single-network accuracies. To increase overall accuracy to 73.28%, this study also uses Cosine Annealing to conduct extra tuning model and combine training and validation data.

Previous research has explored various models for the FER task, as summarized in Table 1. In this study, we will utilize a subset of these models that have been shown to be effective in previous work. In addition, we aim to incorporate other models that have proven successful for FER, as shown in Table 2.

TABLE I. SUMMARY OF PREVIOUS OVERALL ACCURACY WORKS

Methods	Dataset	Overall Accuracy
CNN Ensemble Model [29]	FER2013	62.44%
Resnet 34-layer [33]	FER2013	72.40%
Inception [33]	FER2013	71.60%
VGG [33]	FER2013	72.70%
MobileNetV1 [2]	FER2013	85.40%
MobileNetV2[2]	FER2013	77.18%
MobileNetV3-small [2]	FERPlus	73.24%
MobileNetV1 [2]	RAF-DB	81.62%
MobileNetV2 [2]	RAF-DB	67.77%
MobileNetV3-small [2]	RAF-DB	68.29%
Xception [36]	FER2013	67.40%
MobileNetV1 [36]	FER2013	61.80%
MobileNetV2 [36]	FER2013	62.10%
DenseNet-40 [36]	FER2013	66.60%
DenseNet [37]	FER2013	63.50%
DenseNet [37]	KDEF	85.40%
VGG [9]	FER2013	73.28%

3. METHODS

A. Dataset

We used the dataset used in the Challenges in Representation Learning: Facial Expression Recognition Challenge competition held at Kaggle [6]. This dataset consists of about 28,000 training images, and 7,000 images for testing. These images are structured in 48x48 pixel gray-scale images of faces. The images can be classified into seven classes, including Angry, Disgusted, Fearful, Happy, Neutral, Sad, and Surprised. One of the limitations of this dataset is its unbalanced class distribution as we can see in Fig. 1. As a result of this unbalanced distribution, the models sometimes struggle to accurately identify certain facial expressions.

B. Model Development on PC

To conduct the model training for this study, we used the Keras framework in conjunction with Python 3.9.

1) Data Preprocessing

At this stage, the dataset is divided into three parts: training, validation, and testing. As the dataset taken from Kaggle did not include a validation set, 25% of the training set was allocated for validation purposes. This allowed for the assessment of the model's performance and prevented overfitting, where the model becomes too specialized and fails to generalize to new data. Two data augmentation techniques were then employed: random horizontal flip and random translation. These techniques can be used to generate new image variations, improving the model's ability to identify and categorize objects. The final step is to resize the images. Before entering the input layer of the model, the training set images are resized to 224x224.

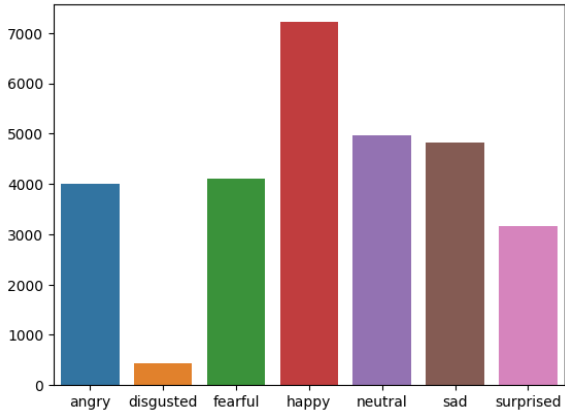

Figure 1. FER2013 class distribution

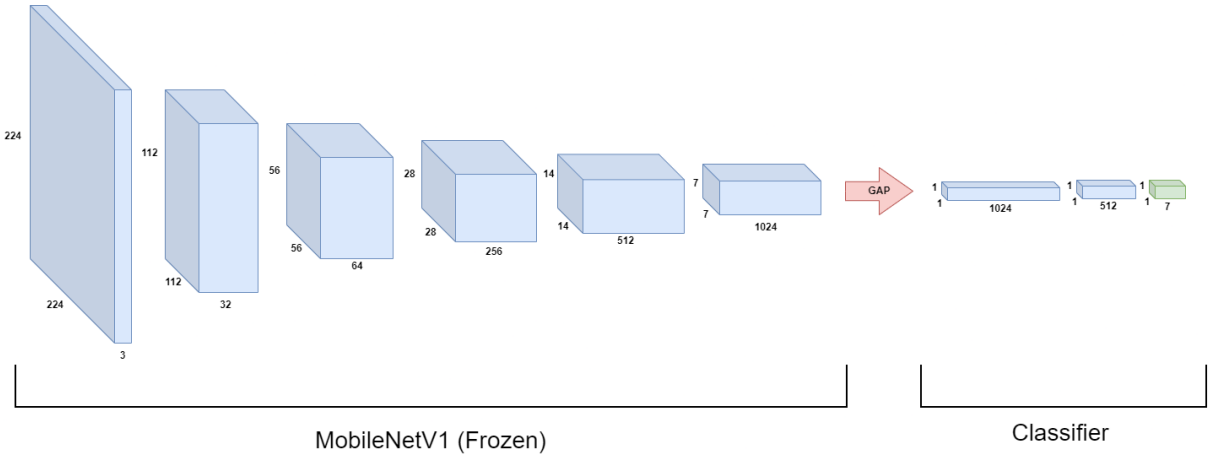

Figure 2. Model development architecture example on MobileNet

2) Training Model

To perform model training, the architecture used in this study consists of a feature extractor and a classifier as depicted in Fig. 2. The feature extractor was a pre-trained model that has been trained using the ImageNet dataset which can be seen in Table 2. In order to use the pre-trained model as a feature extractor, the top part of the pre-trained model that acts as the classifier is not included, then the layers in the pre-trained model that have been cut are frozen, this ensures that the feature extractor cannot learn new information during the training process. The advantage of using a pre-trained model as a feature extractor is that it can learn common features because it is trained using a larger and more general dataset, and it saves training time because we do not have to train the model from scratch.

Furthermore, the classifier architecture consists of two series of dense layers and dropout layers, and then ends with a dense layer which functions as a classifier, as shown in Table 3. Once the model architecture is built, the model is trained using Adam's optimizer, which is initialized with a learning rate of 1e-3. When the learning rate reaches a plateau, the learning rate is reduced, however we ensured that the learning rate never falls below 1e-5. In addition, the model incorporates early stopping to avoid overfitting the training data. This is achieved by monitoring the overall accuracy of the validation data over successive epochs.

TABLE II. THE PRE-TRAINED MODELS WE USED

Model Name	No. of Parameters (Millions)
MobileNetV3Small	2.9
MobileNetV2	3.5
MobileNet	4.3
NasNetMobile	5.3
MobileNetV3Large	5.4
EfficientNetV2B0	7.2
DenseNet121	8.1
EfficientNetV2B1	8.2
EfficientNetV2B2	10.2
DenseNet169	14.3
EfficientNetV2B3	14.5
DenseNet201	20.2
EfficientNetV2S	21.6
Xception	22.9
InceptionV3	23.9
ResNet50V2	25.6
ResNet50	25.6
ResNet101V2	44.7
ResNet101	44.7
EfficientNetV2M	54.4
Resnet152	60.4
ResNet152V2	60.4
EfficientNetV2L	119

TABLE III. LAYERS USED IN CLASSIFIER ARCHITECTURE

Model Name	Output Shape
Dense 1	(None, 1024)
Dropout 1	(None, 1024)
Dense 2	(None, 512)
Dropout 2	(None, 512)
Dense 3	(None, 7)

C. Evaluation Design on PC

After the model is trained on training and test data, it must be evaluated on test data to assess the model's performance. This evaluation is done using an overall accuracy metric calculated using equation (1), where *TCi* is the true classification of instances in dataset, and *n* is the total number of instances in the dataset. This formula is used to determine how effective the model's performance is compared to the training data. Typically, the overall accuracy of a model is determined by comparing the predicted results to the actual results. The closer the predicted results are to the actual results, the better the overall accuracy of the model.

$$Overall\ Accuracy = \frac{\sum_{i=1}^{7} TC_i}{n} \quad (1)$$

D. Evaluation Design on Mobile Device

To conduct an evaluation on mobile, we selected the top three candidates from each model based on their overall accuracy. The models were then converted into TensorFlow Lite, which is optimized for use on mobile and embedded devices. We also applied post-training quantization during conversion to minimize CPU latency, reduce power consumption, and decrease model size while maintaining high overall accuracy. Additionally, the three Tensorflow Lite models are deployed on mobile applications developed using Flutter with the *tflite_flutter* package to evaluate the computational load on mobile devices. The computational load is determined by three main metrics: inference time, peak CPU usage, and peak memory usage.

1) Inference time

Inference Time refers to the duration it takes for the model to process and generate an output for a given input. This metric is significant as it determines the speed at which the model can function and yield outcomes.

TABLE IV. MOBILE DEVICE SPECIFICATIONS USED FOR EVALUATION

Criteria	Configuration
Memory	6 GB
Processor	Octa-core (2x2.0 GHz Kryo 460 Gold & 6x1.7 GHz Kryo 460 Silver)
Graphics processing unit	Adreno 612

2) Peak CPU usage

CPU usage is an important metric for evaluating model performance as it indicates the level of processing power required for the model to operate. Models with high CPU usage may be inefficient in terms of computing resources and may not be suitable for deployment in resource-restricted settings.

3) Peak memory usage

Memory usage serves as a metric that gauges the amount of memory required by the model to store parameters and interim computations. Models with high memory usage may demand substantial resources and may not be well-suited for application in environments with limited memory resources.

To measure these three metrics, the following steps are performed: (1) For each class in the test dataset, take five sample images at random, resulting in a total of 35 images. (2) Make predictions on these images. (3) Calculate the average inference time, peak CPU usage, and peak memory usage during the prediction process.

These steps were repeated 10 times to calculate the average of the three metrics. This evaluation was performed on a mobile device running the Android 11 operating system with the specifications described in Table 4. The Android Studio Profiler was used to measure the memory usage and the CPU Profiler because this tool measures only per selected application or process without being disturbed by other applications or processes running in the background, thus allowing an accurate evaluation of the model performance.

4. RESULTS

A. Evaluation Results on PC

Table 5 shows the overall accuracy results for the testing data, revealing interesting findings. Among the models examined, MobileNet, ResNet101, and EfficientNetV2B3 are the three candidate models. EfficientNetV2B3 achieved the highest overall accuracy at 61.9% with a relatively low parameter count of 14.5 million. Despite having the highest number of parameters at 44.7 million, the ResNet model achieved the second-

best overall accuracy at 60.3%. MobileNet, with the least number of parameters at 4.3 million, exhibited the lowest overall accuracy among the mentioned models, standing at approximately 58.8%.

TABLE V. SUMMARY OF OVERALL ACCURACY OF 23 MODELS

Model Name	Overall Accuracy	No. of Parameters (millions)
DenseNet121	0.531	8.1
DenseNet169	0.557	14.3
DenseNet201	0.548	20.2
EfficientNetV2B0	0.601	7.2
EfficientNetV2B1	0.594	8.2
EfficientNetV2B2	0.598	10.2
EfficientNetV2B3	**0.619**	**14.5**
EfficientNetV2L	0.584	119
EfficientNetV2M	0.577	54.4
EfficientNetV2S	0.612	21.6
InceptionV3	0.511	23.9
MobileNet	**0.588**	**4.3**
MobileNetV2	0.546	3.5
MobileNetV3Large	0.579	5.4
MobileNetV3Small	0.521	2.9
NasNetMobile	0.464	5.3
ResNet101	0.564	44.7
ResNet101V2	**0.603**	**44.7**
Resnet152	0.601	60.4
ResNet152V2	0.559	60.4
ResNet50	0.591	25.6
ResNet50V2	0.596	25.6
Xception	0.538	22.9

TABLE VI. COMPUTATIONAL LOAD OF MOBILENET, EFFICIENTNETV2B3, AND RESNET101

Model Name	Average Inference Time (ms)	Peak CPU Usage	Peak Memory Usage (MB)
MobileNet	323.04	22%	36.09
EfficientNetV2B3	363.04	21%	34.34
ResNet101	532.15	21%	36.86

B. Evaluation Results on Mobile

The mobile evaluation used three candidate models with the highest overall accuracy: MobileNet, EfficientNetV2B3, and ResNet101. Based on the data in Table 6, there was no significant difference in the average CPU utilization among the three models, with results ranging from 21% to 22%. In addition, this study found that the memory utilization of the three models was almost the same, with the average utilization ranging from 34 MB to 37 MB. The models showed comparable efficiency in terms of CPU and memory usage. In contrast to the inference time metric, the MobileNet model has the lowest average inference time of 323 ms, EfficientNetV2B3 has a slightly longer average inference time of 363 ms, and the ResNet101 model has the longest average inference time of 532 ms.

C. Evaluation Results Summary

As can be seen in Table 7, the EfficientNet model has an advantage over MobileNet in terms of overall accuracy. This is due to the fact that the EfficientNet model has a more complex architecture, which allows it to learn more complicated features. On the other hand, the MobileNet model has lower overall accuracy because it has a simpler architecture, which limits its ability to learn complex features. Finally, the ResNet model has the highest parameterization of the three models, although it has a fairly good overall accuracy.

These results show that MobileNet and EfficientNet are suitable for fast, real-time preprocessing applications with sufficient overall accuracy. On the other hand, the ResNet101 model may not be suitable for applications that require real-time processing and fast response. This is because the ResNet101 model has more parameters than the other two models, which requires more time to perform inference.

TABLE VII. OVERALL ACCURACY AND COMPUTATIONAL LOAD OF MOBILENET, EFFICIENTNETV2B3, AND RESNET101

Model Name	Overall Accuracy	No. of Parameters (Millions)	Average Inference Time (ms)	Peak CPU Usage	Peak Memory Usage (MB)
MobileNet	0.588	4.3	323.04	22%	36.09
EfficientNetV2B3	0.619	14.5	363.04	21%	34.34
ResNet101V2	0.603	44.7	532.15	21%	36.86

5. CONCLUSION AND FUTURE WORKS

In this study, we implemented and evaluated several lightweight CNN models to perform facial expression recognition. Then, to evaluate their computational load on mobile devices when performing facial expression recognition, three models with the best overall accuracy were selected, namely MobileNet, EfficientNetV2B3, and ResNet101.

This evaluation shows that EfficientNetV2B3 has the highest overall accuracy of 61.9% with a computational load of 363.04 ms average inference time, 21% peak CPU utilization, and 34.34 MB peak memory utilization. Then, the ResNet101 model has a lower overall accuracy than EfficientNetV2B3 at 60.3% and has the highest computational load compared to other models with an average inference time of 532.15 ms, peak CPU utilization of 21%, and peak memory utilization of 36.86 MB. The last model, MobileNet, has the lowest overall accuracy of 58.8%, but has the lowest computational load, with an average inference time of 323.04 ms, peak CPU utilization of 22%, and peak memory utilization of 36.09 MB. These evaluation results show that although EfficientNetV2B3 and ResNet101 have the best overall accuracy for facial expression recognition, other factors such as speed or low computational load are also important for real-world applications, so models such as MobileNet should be considered.

There are several things that need to be studied to carry out further development of this research. The dataset used to train and evaluate the model is currently only using the FER2013 Dataset, even though the model evaluation results have quite good performance and computational load, the overall accuracy is still below 80%. Therefore, it is hoped that in future research we can try to train and evaluate models using a more diverse dataset regarding facial expression recognition.

ACKNOWLEDGMENT

The authors would like to thank the creators and collectors of the FER2013 dataset for their efforts in collecting this data, which played an important role in supporting our research on real-time facial expression recognition with CNN on mobile devices.

REFERENCES

[1] S. Jia, S. Wang, C. Hu, P. J. Webster, and X. Li, "Detection of Genuine and Posed Facial Expressions of Emotion: Databases and Methods," *Frontiers in Psychology*, vol. 11. Frontiers Media S.A., Jan. 15, 2021. doi: 10.3389/fpsyg.2020.580287.

[2] Y. Nan, J. Ju, Q. Hua, H. Zhang, and B. Wang, "A-MobileNet: An approach of facial expression recognition," *Alexandria Engineering Journal*, vol. 61, no. 6, pp. 4435–4444, Jun. 2022, doi: 10.1016/j.aej.2021.09.066.

[3] Z. Y. Huang *et al.*, "A study on computer vision for facial emotion recognition," *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-35446-4.

[4] C. Szegedy *et al.*, "Going Deeper with Convolutions," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.4842

[5] J. Le Ngwe, K. M. Lim, C. P. Lee, and T. S. Ong, "PAtt-Lite: Lightweight Patch and Attention MobileNet for Challenging Facial Expression Recognition," Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.09626

[6] I. J. Goodfellow *et al.*, "Challenges in Representation Learning: A report on three machine learning contests," Jul. 2013, [Online]. Available: http://arxiv.org/abs/1307.0414

[7] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition," *IEEE Access*, vol. 7, pp. 64827–64836, 2019, doi: 10.1109/ACCESS.2019.2917266.

[8] H. Ding, S. K. Zhou, and R. Chellappa, "FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition,"

Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.06591

[9] Y. Khaireddin and Z. Chen, "Facial Emotion Recognition: State of the Art Performance on FER2013," 2021.

[10] Y. Nan, J. Ju, Q. Hua, H. Zhang, and B. Wang, "A-MobileNet: An approach of facial expression recognition," *Alexandria Engineering Journal*, vol. 61, no. 6, pp. 4435–4444, Jun. 2022, doi: 10.1016/j.aej.2021.09.066.

[11] J. Liang, "Image classification based on RESNET," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Oct. 2020. doi: 10.1088/1742-6596/1634/1/012110.

[12] G. R. S. Murthy and R. S. Jadon, "Hand gesture recognition using neural networks," in *2010 IEEE 2nd International Advance Computing Conference, IACC 2010*, 2010, pp. 134–138. doi: 10.1109/IADCC.2010.5423024.

[13] V. Bettadapura, "Face Expression Recognition and Analysis: The State of the Art," 2012.

[14] IEEE Signal Processing Society, *2013 IEEE International Conference on Acoustics, Speech, and Signal Processing : proceedings : May 26-31, 2013, Vancouver Convention Center, Vancouver, British Columbia, Canada*. 2013.

[15] O. Bayat, S. Aljawarneh, H. F. Carlak, International Association of Researchers, Institute of Electrical and Electronics Engineers, and Akdeniz Üniversitesi, *Proceedings of 2017 International Conference on Engineering & Technology (ICET'2017) : Akdeniz University, Antalya, Turkey, 21-23 August, 2017*. 2017.

[16] Y. Lecun, L. Eon Bottou, Y. Bengio, and P. H. Abstract|, "Gradient-Based Learning Applied to Document Recognition," 1998.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012. [Online]. Available: http://code.google.com/p/cuda-convnet/

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385

[19] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.00298

[20] R. Sun, "Optimization for deep learning: theory and algorithms," Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.08957

[21] B. Han POSTECH and J. Sim, "BranchOut: Regularization for Online Ensemble Tracking with Convolutional Neural Networks."

[22] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015, [Online]. Available: http://arxiv.org/abs/1502.03167

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." [Online]. Available: http://code.google.com/p/cuda-convnet/

[24] F. Y. Shih, C.-F. Chuang, and P. S. P. Wang, "PERFORMANCE COMPARISONS OF FACIAL EXPRESSION RECOGNITION IN JAFFE DATABASE," 2008.

[25] I. J. Goodfellow *et al.*, "Challenges in Representation Learning: A report on three machine learning contests," Jul. 2013, [Online]. Available: http://arxiv.org/abs/1307.0414

[26] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution," Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.01041

[27] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," Aug. 2017, doi: 10.1109/TAFFC.2017.2740923.

[28] 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, CVPRW 2010*, 2010, pp. 94–101. doi: 10.1109/CVPRW.2010.5543262.

[29] K. Liu, M. Zhang, and Z. Pan, "Facial Expression Recognition with CNN Ensemble," in *Proceedings - 2016 International Conference on Cyberworlds, CW 2016*, Institute of Electrical and Electronics Engineers Inc., Nov. 2016, pp. 163–166. doi: 10.1109/CW.2016.34.

[30] S. Minaee, M. Minaei, and A. Abdolrashidi, "Deep-emotion: Facial expression recognition using attentional convolutional network," *Sensors*, vol. 21, no. 9, May 2021, doi: 10.3390/s21093046.

[31] M.-I. Georgescu, R. T. Ionescu, and M. Popescu, "Local Learning with Deep and Handcrafted Features for Facial Expression Recognition," Apr. 2018, doi: 10.1109/ACCESS.2019.2917266.

[32] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going Deeper in Facial Expression Recognition using Deep Neural Networks," Nov. 2015, doi: 10.1109/WACV.2016.7477450.

[33] C. Pramerdorfer and M. Kampel, "Facial Expression Recognition using Convolutional Neural Networks: State of the Art," Dec. 2016, [Online]. Available: http://arxiv.org/abs/1612.02903
[34] R. T. Ionescu and C. Grozea, "Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition," 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:17024133
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.03385
[36] Institute of Electrical and Electronics Engineers, *IEEE 21st International Workshop on Multimedia Signal Processing, MMSP 2019 : Aloft Kuala Lumpur Sentral Hotel, Kuala Lumpur, Malaysia, September 27-29, 2019.*
[37] K. C. O'nell, R. Saxe, and S. Anzellotti, "Recognition of identity and expressions as integrated processes," 2019.

Erick is a graduate student of Master of Computer Science Department in Bina Nusantara University. His research interests include computer vision, machine learning, and image processing.

Keiko Kimberly Octavina is a graduate student of Master of Computer Science Department in Bina Nusantara University. Her research interests include computer vision, machine learning, and image processing.

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Head of Department of Master of Computer Science, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R – A*STAR, Singapore. His research interests include computer vision, deep learning, face recognition, appearance-based object recognition, gamification of learning, and indoor positioning system.