Enhancing Facial Expression Recognition: Leveraging MobileNetV3 for Periocular Analysis

Sinar B. Ramadhan and David H. Hareva*

Magister Informatic, Faculty of Information and Technology, Universitas Pelita Harapan, Jakarta, Indonesia
Email: 01679220015@student.uph.edu (S.B.R.); david.hareva@uph.edu (D.H.H.)
*Corresponding author

***Abstract*—Online meetings and Virtual Reality (VR) applications require innovative approaches to interpret user emotions and behavior. Since verbal communication is constrained in virtual environments, facial expression analysis is essential for understanding emotional states. Recent research demonstrates that the periocular region provides significant diagnostic information regarding affect and attention, exhibiting pronounced responses to emotional stimuli and offering a more reliable indicator of user state than full-face analysis. This study addresses this gap by evaluating lightweight convolutional neural network architectures—MobileNetV1, MobileNetV2, MobileNetV3, and EfficientNetV2—specifically for periocular-based recognition. Experiments are conducted on the Taiwanese Facial Expression Image Database (TFEID) benchmark, with further validation on the Chinese Face dataset using transfer learning for Android platform deployment. Through a detailed analysis, we evaluate the effectiveness of each architecture based on metrics such as accuracy, precision, recall, and F1-Score, providing insights into their suitability for periocular-based expression recognition. In contrast to earlier studies that employed full-face input, this research proposes a periocular-only approach, rendering it more efficacious in confined environments such as virtual reality or masked-face settings. The findings of this study demonstrate that the MobileNetV3-Small architecture offers an optimal trade-off, attaining an accuracy of 83.62% while sustaining a highly efficient inference time of 16.4 milliseconds per image. Moreover, the deployment of these models on Android devices demonstrates their practicality in real-world settings, particularly in the context of lightweight, mobile-based emotion recognition systems. This research contributes to advancing emotion recognition systems, offering practical and robust solutions for real-world applications.**

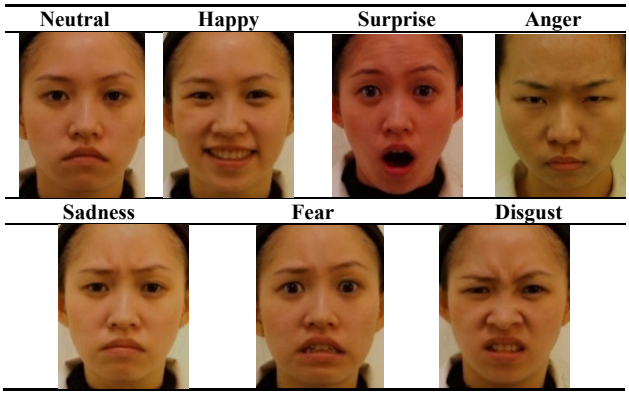
***Keywords*—facial expression, periocular area, MobileNet, EfficientNetV2, Taiwanese Facial Expression Image Database (TFEID), Chinese Face dataset**

I. Introduction

Communication is the most basic activity performed by humans to interact with each other, to then share information, build relationships, and lead social lives. Non-verbal communication, particularly facial expressions, plays a crucial role in conveying emotions, intentions, and thoughts universally understood across cultures [1]. In social interactions, facial expressions are essential because they serve as a bridge to understand the feelings of others without exchanging words. Because it is not limited by language barriers, this mode of communication is extremely effective. It enables people from a variety of backgrounds to interpret feelings and intentions in a more natural and intuitive manner.

Facial expression is a manifestation of emotions, intentions, and purposes through the movement of facial muscles [2]. From a person's facial expression, the emotional state of that individual can be identified. Facial expressions in humans are divided into two types: neutral faces and expressive faces [3]. Expressive faces are divided into six categories: happiness, surprise, anger, sadness, fear, disgust [4]. These expression categories have different characteristics interpreted in the form of Action Units (AU) [5], as shown in Table I.

TABLE I. Types of Facial Expressions Faces Neutral Happy Surprise Anger Sadness Fear Disgust



The rapid expansion of online communication platforms and advancements in mobile technology have made virtual interactions, such as remote meetings [6], online gaming [7], and Virtual Reality (VR) environments [8], increasingly prevalent and essential. In these digital spaces, the implementation of facial expression recognition is highly recommended to make communication more interactive and immersive, enhancing user experience. However, the limitations of

Manuscript received July 18, 2025; revised August 5, 2025; accepted September 15, 2025; published January 16, 2026.

physical presence in virtual settings pose challenges to conveying emotions and intentions effectively. Facial expressions, therefore, serve as a critical tool to bridge the gap between the virtual and real worlds, enabling users to connect and communicate emotions in a more authentic and engaging way.

Devices for VR, head-mounted devices, as depicted in Fig. 1, typically capture only the area around the user's eye [9]. Apart from VR, the use of masks has become commonplace in everyday activities. The COVID-19 pandemic, which began in 2019, has impacted various aspects of life, including the widespread adoption of mask-wearing. This poses a challenge for facial expression recognition, as the area available for machine learning models to analyze becomes increasingly limited, focusing solely on the periocular region, or the area around the eyes [10], as illustrated in Fig. 2. Implementation of facial expression recognition on embedded systems like Internet of Things (IoT) or mobile devices introduces additional complexity, particularly in this constrained area. Studies have shown that focusing on the periocular region allows for effective emotion recognition despite facial obstructions [11, 12]. Focusing on the periocular area instead of the full face reduces distractions from facial features that may not affect emotion recognition [13].

Fig. 1. Example of a head-mounted Virtual Reality (VR) device that obstructs the lower facial region, necessitating emotion recognition from the visible periocular area [9].

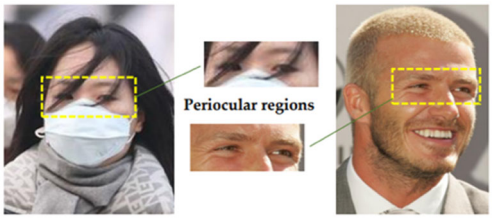


Fig. 2. Illustration of the periocular region as the area of interest for facial expression recognition in scenarios with obstructions, such as mask-wearing (left) or in un-occluded faces (right) [10].

This study differs from previous research in two significant ways. Firstly, it presents a periocular-only model that demonstrates enhanced robustness in occluded or partially visible environments. Secondly, it emphasizes a deployment-oriented perspective by systematically analyzing the trade-offs between accuracy, model size, and inference speed of lightweight Convolutional Neural Networks (CNNs) when applied on mobile and embedded devices. Unlike prior works that mainly pursue accuracy improvements using large-scale CNNs, our contribution lies in providing a comprehensive evaluation of resource-constrained deployment for periocular expression recognition. This perspective highlights practical feasibility and addresses a critical gap in current literature, where deployment aspects of periocular biometrics remain underexplored.

This paper investigates the effectiveness of utilizing the periocular area in classifying facial expressions. Since only the periocular region is visible in certain real-world scenarios (e.g., due to mask usage or in VR/AR headsets), focusing on this region allows for a more realistic and adaptable approach to facial expression recognition. The periocular region represents a noteworthy biometric characteristic for the purpose of human identification [11].

Conducted an in-depth analysis of the MobileNet family and EfficientNetV2 (B0, B1, B2) architectures to determine which is better suited to the original problem. Because of their small size and great efficiency, these models are perfect for embedded or mobile systems [14], and the CNN approach works well for emotion recognition [15].

II. Related and Previous Works

This section delivers an overview of the literature regarding datasets and the architecture used in the research, along with investigations previously done that is related to the material in this paper.

A. Related Theories

MobileNet is a series of convolutional neural network architectures created by Google to facilitate efficient deep learning on mobile and embedded devices [16]. The architecture employs depth-wise separable convolution, which divides the convolution process into two distinct operations: depth-wise and point-wise convolution [17]. This innovative approach significantly reduces the computational demands and the number of parameters required for convolutional layers, making MobileNet particularly suitable for applications on resource-constrained platforms. This distinction between the two convolution types is illustrated in Fig. 3. Its lightweight design has made it a popular choice for various tasks, including image classification, object detection, and facial expression recognition [18].

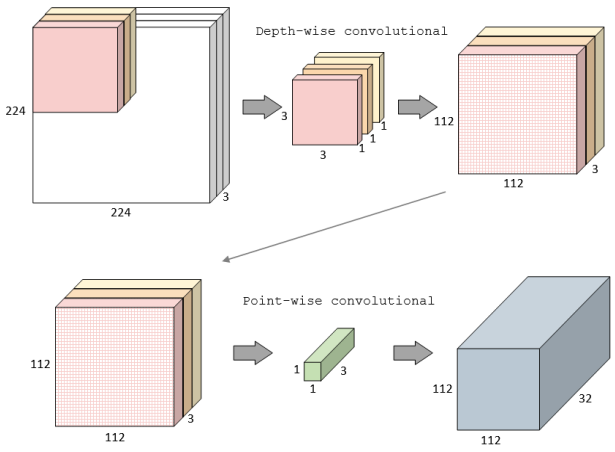


Fig. 3. Illustration of depth-wise and point-wise convolutions in MobileNet.

In addition to MobileNet, the study also examines EfficientNetV2 (B0, B1, B2), another efficient model family developed by Google that utilizes advanced techniques in model scaling and architecture design. EfficientNetV2 optimizes performance for mobile and resource-limited devices by balancing network depth, width, and resolution through a method known as compound scaling [19]. This family of models expands the options for high-performance, lightweight deep learning applications, providing an alternative approach to achieving a balance between speed and accuracy. Both MobileNet and EfficientNetV2 are well-equipped for deployment in environments with limited resources, as they effectively implement various strategies to maintain high accuracy while ensuring computational efficiency [20].

1) MobileNetV1: Depthwise separable convolutions

MobileNetV1 is based on depthwise separable convolutions [21]. Compared to traditional convolutions, this approach uses a single filter for each input channel, making it more economical. It also reduces the number of channels in the output, preserving feature dimensionality. MobileNetV1 applies a single filter to each input channel individually, resulting in a smaller output. This approach significantly reduces computations while preserving feature extraction capabilities. Compared to MobileNetV2 and MobileNetV3, MobileNetV1 relies solely on depthwise separable convolutions.

2) MobileNetV2: Linear bottlenecks and residual connections

MobileNetV2 is a novel architecture that improves efficiency and performance by incorporating linear bottlenecks and residual connections [22]. It introduces a low-dimensional bottleneck layer between depthwise and pointwise convolutions, lowering computational costs while maintaining accuracy. It also includes shortcut connections from earlier to later layers, which allows the network to learn more complex features. These enhancements, inspired by ResNet architecture, improve gradient flow and performance, making MobileNetV2 a more reliable and efficient alternative to MobileNetV1.

3) MobileNetV3: Squeeze-and-excite blocks and efficient inverted residuals

MobileNetV3 improves efficiency through two key innovations: Squeeze-and-Excite (SE) blocks, which learn channel-wise importance weights, and Efficient Inverted Residuals, which reorder operations within the residual block, prioritizing non-linearity over the bottleneck layer [23]. These innovations enable thinner bottlenecks while maintaining good accuracy. MobileNetV3 uses SE blocks to analyze feature maps and adjust their importance using learned weights, whereas Efficient Inverted Residuals places nonlinearity before the bottleneck layer to improve efficiency.

The overview on MobileNetV1, MobileNetV2, and MobileNetV3 is shown in Table II. Overall, MobileNet family of architectures offers a range of options optimized for different requirements, from MobileNetV1 for basic applications to MobileNetV3 for more demanding tasks where both accuracy and efficiency are crucial. By adopting these approaches, the computational demands and parameter count of the convolutional layer are significantly reduced, enhancing efficiency particularly for applications on mobile and embedded devices.

TABLE II. COMPARISON OF KEY ARCHITECTURAL FEATURES ACROSS THE MOBILENET FAMILY

Feature	MobileNetV1 [21]	MobileNetV2 [22]	MobileNetV3 [23]
Basic Building Block	Depthwise Separable Convolution	Depthwise Separable Convolution with Linear Bottlenecks	Efficient Inverted Residual
Residual Connection	No	Yes	Yes
Channel Attention	No	No	Squeeze-and-Excite (SE) Block
Focus	Reduce computational cost	Improve efficiency and performance	Further improve efficiency and performance

4) EfficientNetV2

Tan and Le [19] introduced an advanced family of convolutional networks called EfficientNetV2 was introduced. It improves training speed and parameter efficiency with new techniques like training-aware neural architecture search and Fused-MBConv layers, which are a modified version of the traditional MBConv block [19]. This model features progressive learning, where image resolution and architectural complexity are gradually increased during training, leading to reduced training times and improved efficiency [24]. By streamlining the search for optimal neural architecture parameters and inference time, it is made to be both smaller and faster. EfficientNetV2 offers scalability with three variants (EfficientNetV2B0, EfficientNetV2B1, and EfficientNetV2B2) each progressively increasing in depth, width, and image resolution to cater to different computational needs and accuracy requirements. In performance benchmarks, EfficientNetV2 surpasses many leading models, including ResNet and EfficientNetV1, making it particularly effective for real-world applications that demand quick inference on devices with limited resources, especially in tasks like image classification and object detection.

B. Previous Research

The forehead area did not yield a significant increase in accuracy [25]. Various descriptors were employed, including Gabor, LBP, Histogram of Oriented Gradient (HOG), Gray-Level Co-occurrence Matrix (GLCM), and GIST, with Support Vector Machine (SVM) as the classifier. Two scenarios were tested: large area and small area, divided into 16×16 and 32×32 blocks, as depicted in Fig. 4.

Research on emotion detection utilizing the periocular area involved extracting the Region of Interest (RoI) area from 5 landmark points obtained using Dlib [26]. The upper boundary was determined by adding approximately

75% of the eye center's height, while the lateral boundary was set by adding about 25% of the distance from the eye center, as shown in Fig. 5.

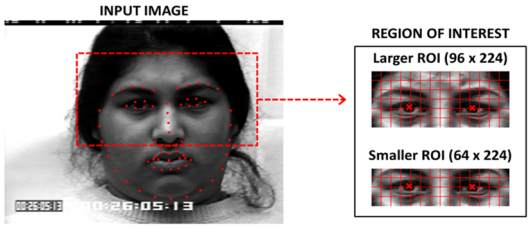


Fig. 4. Example of Region of Interest (RoI) extraction from a previous study, showing larger (96×224) and smaller (64×224) periocular crops for analysis [25].

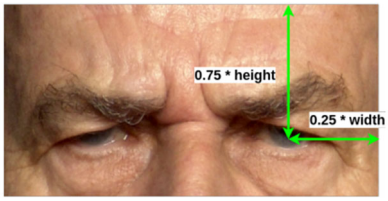


Fig. 5. Periocular area extraction [26].

Experiments were conducted by first training on the Faces dataset, followed by testing on Chicago Faces to assess the model's robustness under cross-dataset testing. The accuracy results obtained using MobileNet-V2 and HOG-SVM architectures were 76.77% and 62.47%, respectively.

III. Materials and Methods

The aim of this study was to determine how using the periocular area affects expression classification performance. To achieve this goal, the researchers conducted experiments using the Taiwanese Facial Expression Image Database (TFEID) dataset and MobileNet family architectures consists of MobileNetV1, MobileNetV2, and MobileNetV3. The researchers focused on face recognition in pandemic conditions, where masks are widely recommended in almost all countries, and in the use of head-mounted VR devices that limits the exposed facial area only on the periocular region. We begin by providing an overview of the dataset exploration consists of dataset preparation and dataset preprocessing, continued with the detailed step from each of MobileNet architectures. Performance metrics then obtained from experiments conducted including accuracy, precision, recall, and F1-Score, which serve as indicators of the effectiveness of our approach in accurately classifying facial expressions.

The methodology employed in this study involves several key steps for training and deploying a facial expression classification model using MobileNet architectures and periocular images. Initially, the training phase utilizes the TFEID, a well-known benchmark dataset for facial expression analysis. The next step is Facial Mapping, which means that images within this dataset are first processed to extract 68 landmark points using the Dlib library. Subsequently, the periocular area is isolated by selecting 22 landmark points corresponding to this specific facial region. These extracted periocular images serve as the input data to the model training step to train the classification model using MobileNetV1, MobileNetV2, and MobileNetV3 architectures.

The classification task is treated as a multiclass problem, wherein each image is categorized into one of the following classes: 'happy', 'angry', 'sad', 'disgust', 'fear', or 'neutral'. Through the training process, the performance of each MobileNet variant is evaluated, and the best-performing model is identified based on accuracy and other relevant metrics.

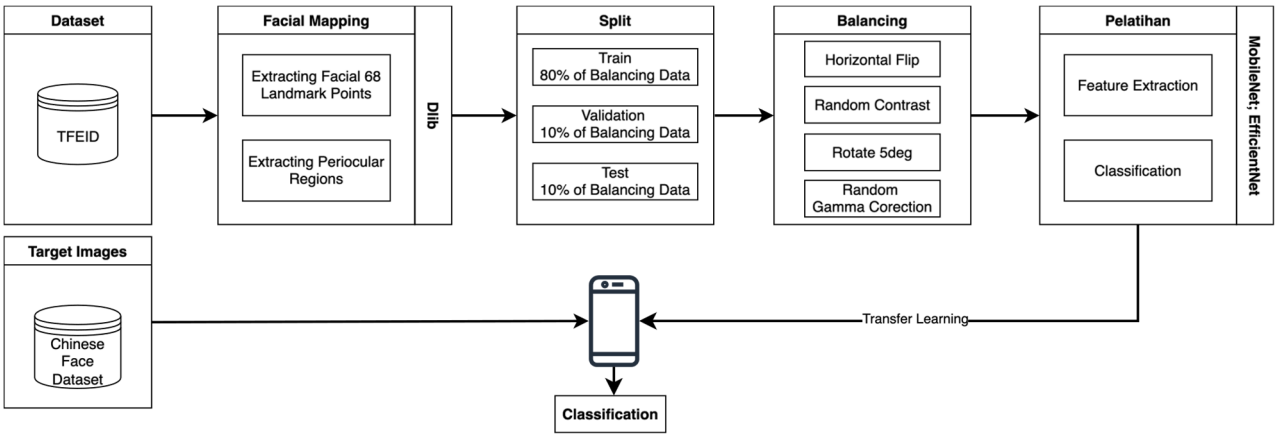


Fig. 6. The complete workflow of the proposed study, detailing the stages from dataset processing and facial mapping to model training, evaluation, and final deployment on a mobile device via transfer learning.

After showing the optimal model, it is implemented on an Android-based mobile device for real-time expression classification. The testing phase involves using images not included in dataset to assess the model's generalization capability. The Android module consists of several components, including an image acquisition module responsible for capturing images, an image processing module for preprocessing the acquired images, and an expression classification module utilizing TensorFlow Lite and TensorFlow modules to classify expressions. Finally, the display module presents the classification results, including object labels and prediction outcomes, providing users with real-time feedback on facial expressions. This comprehensive methodology ensures the

development of an effective and deployable expression classification system suitable for mobile platforms. The overall architecture is as seen in Fig. 6.

A. Dataset Preparation

The research is held using TFEID dataset which are valuable resources in the field of facial expression analysis, particularly concerning research focusing on the periocular region [27]. This dataset comprises 336 images, obtained from 40 models (20 males, 20 females), representing eight facial expressions: 'Neutral', 'Angry', 'Contempt', 'Disgust', 'Fear', 'Happy', 'Sad', and 'Surprise'. Sample images of TFEID dataset is as follows in Fig. 7(a). This research also utilizes the Chinese Face Dataset to evaluate the model integrated into the mobile platform. Chinese Face Dataset contains 840 StyleGAN-synthesized facial images [28]. The dataset has 60 men and 60 women proportionally and includes seven basic emotional expressions (neutral, happiness, anger, fear, sadness, disgust, and surprise). This dataset is intended to aid psychology and related research, particularly in facial perception, emotional recognition, and age-related social judgments. Unlike TFEID, this synthetic dataset serves specifically to test the model's robustness and generalization capability under a different data distribution, which is crucial for assessing its performance in real-world, on-device scenarios. Fig. 7(b) presents some examples of images that are included in the Chinese Face Dataset.

(a) TFEID Dataset

(b) Chinese Face Dataset

Fig. 7. Sample images from the two datasets used: (a) The Taiwanese Facial Expression Image Database (TFEID) used for training [27], and (b) the StyleGAN-synthesized Chinese Face dataset used for mobile deployment testing [28].

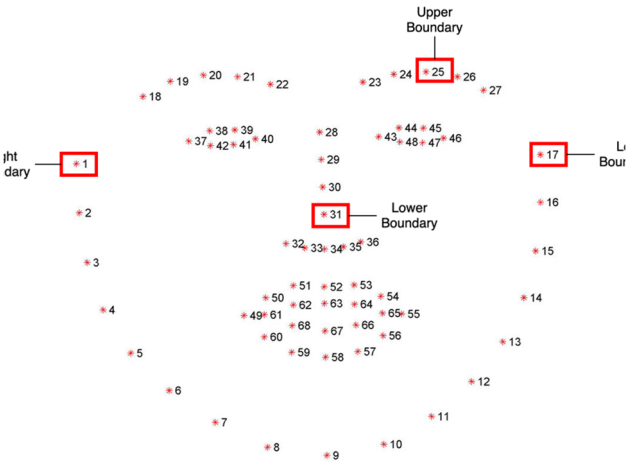
B. Dataset Preprocessing

We first analyze the distribution of expressions within the images in each dataset and observe data imbalances, as shown in Table III.

To address data imbalances, we apply image augmentation for each image in every category to achieve a total of 1000 images. But before performing the augmentation, we first extracted the periocular area from each image. Periocular extraction was performed utilizing the Dlib library as a face landmark detector that annotated 68 facial landmark points, as illustrated in Fig. 8(a). Dlib is an open-source library that offers various machine learning algorithms for tackling complex problems. It is known for its ease of implementation, ability to work with unconventional angles, and capability to handle occluded faces [29].

TABLE III. EXPRESSIONS DISTRIBUTION IN TFEID

Expression	Number of Images
Anger	34
Contempt	68
Disgust	40
Fear	40
Happy	40
Neutral	39
Sadness	39
Surprise	36



(a) 68 Landmark Points Extracted using Dlib.

```
Procedure Extract Periocular (source_image)

 // 1. Detect Face and Landmarks
 detected_faces <- Detect_Faces(source_image)
 For each face in detected_faces:
   landmarks <- Predict_Landmarks(face)

   // 2. Define Cropping Boundaries
   left_bound  <- X_coord(landmarks[17]) // Outer left eyebrow
   right_bound <- X_coord(landmarks[26]) // Outer right eyebrow

   eyebrow_y   <- Y_coord(landmarks[19])
   top_margin  <- eyebrow_y * 0.3
   top_bound   <- eyebrow_y - top_margin

   nose_tip_y  <- Y_coord(landmarks[30])
   jaw_edge_y  <- Y_coord(landmarks[0])
   bottom_bound <- jaw_edge_y + (nose_tip_y - jaw_edge_y)

   // 3. Crop and Return Image
   cropped_image <- Crop(source_image,
                         from=(left_bound, top_bound),
                         to=(right_bound, bottom_bound))
   Return cropped_image
 End For
```

(b) Pseudocode to extract periocular.

(c) Extracted Periocular Area.

Fig. 8. The periocular area extraction process: (a) The 68 facial landmark points are first detected using Dlib, (b) Pseudocode for the proposed periocular region extraction algorithm, and (c) the final region is cropped based on specific landmark boundaries for model training.

The periocular region was then extracted from each image using a custom algorithm that leverages specific facial landmarks. This method defines the precise cropping boundaries by utilizing the coordinates of key points on the eyebrows, eyes, and nose. The detailed logic for this extraction procedure is formally presented in Fig. 8(b). The area resulted is as shown in Fig. 8(c).

Augmentation techniques included horizontal flipping, rotation up to a maximum of 50° degrees both left and right, adjustments to brightness and contrast, and gamma adjustment. All augmentation processes were performed randomly. Subsequently, the balanced dataset was divided into three subsets for each emotion category: training, testing, and validation, with a ratio of 80% for training, 10% for testing, and 10% for validation, respectively, based on the total number of images per emotion category.

C. Experiments

As outlined in the dataset preparation section, the dataset is divided into three components: training, testing, and validation. Each component maintains the same emotional categorization. Images for training, testing, and validation are randomly selected, ensuring that each model is exposed to identical datasets across all stages.

During model training, various hyperparameters are utilized, including the number of epochs, batch size, momentum, and the Adamax optimizer. The number of epochs in model training refers to how many times the full training dataset is run through the neural network to update the weights. Each epoch is a complete pass of the whole dataset in which the model processes all the training data to update the network's weights. Increased epochs provide the model with additional opportunity to assimilate knowledge from the data.

Momentum defines how much the freshly determined statistical value (from the current batch) contributes relative to the prior batch's value. The optimizer manages how the model's weights are adjusted based on the gradient of the loss function, improving the model's prediction and classification capabilities [30]. These hyperparameter values (Table IV) are selected based on the available literature and then held constant to allow for a fair comparison of the models utilized [31, 32].

Choosing the appropriate batch size is critical for model training since a small batch size achieves convergence faster than a big batch size. Moreover, while larger batch sizes can reach the optimal minimum value, smaller batches may struggle achieve this. In this study, each pre-trained model was additionally given five layers: two dropout layer and three dense layers. The dropout rate in the dropout layer is 50%, and the first dense layer has 256 neurons, the second has 128 neurons, and the last dense has 8 neurons to categorize 8 distinct moods using the SoftMax activation function. This study was carried out using a MacBook OS Sonoma with an Apple M1 Pro processor and 16 GB RAM.

TABLE IV. HYPERPARAMETER SETTINGS USED FOR TRAINING ALL MODELS

Parameters	MobileNetV1	MobileNetV2	MobileNetV3 Small	MobileNetV3 Large	EfficientNetV2 Family
Epoch	55	35	55	55	35
Batch	5	5	5	5	5
Optimizer	Adamax	Adamax	Adamax	Adamax	Adamax
Loss Function	Categorical Cross Entropy	Categorical Cross Entropy	Categorical Cross Entropy	Categorical Cross Entropy	Categorical Cross Entropy
Momentum	0.99	0.99	0.99	0.99	0.99
Learning Rate	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}

D. Mobile Application Implementation

The optimal machine-learning model will be deployed on an Android device to test its real-world performance. Using the TensorFlow Lite Task Library API, models were converted to tflite interpreters to deploy trained machine learning on mobile devices. TensorFlow Lite converter and interpreter enable mobile model deployment [33]. First, TensorFlow-created Keras models were exported to HDF5 binary data format (h5) models. A second step involved the conversion of the h5 models to TensorFlow Lite models using the TensorFlow Lite Converter. Finally, the TensorFlow Lite interpreter executed models on smartphones to maximize detection performance using smartphone hardware [34]. The Flutter framework and TensorFlow Lite library were used to create an Android OS mobile app for TensorFlow Lite models. A Keras-based Tensorflow Lite model processed each image and generated confidence scores indicating its class probability.

E. Evaluation

This study uses confusion matrix serves as a tool to assess the effectiveness of the model resulted in each experiment. In the confusion matrix of a multiclass problems, there are values for True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) as served in Table V.

From the confusion matrix, we employ accuracy, precision, recall, and the F1-Score as metrics to evaluate the model's performance. Accuracy represents the proportion of correctly classified instances (both TP and

TN) out of the total number of instances in the dataset. It measures the overall correctness of the model's predictions across all expression classes (Eq. (1)). Precision measures the model's ability to correctly classify instances of a particular expression class out of all instances classified as that class by the model. It is calculated as the ratio of TP to the sum of TP and FP (Eq. (2)).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

TABLE V. CONFUSION MATRIX

Predicted Values	Actual Values	
	True (+)	False (−)
Positive (+)	True Positive (Correct result)	False Positive (Unexpected result)
Negative (−)	False Negative (Missing result)	True Negative (Correct absence of result)

Recall, also known as sensitivity or true positive rate, measures the model's ability to correctly identify instances of a particular expression class out of all instances that truly belong to that class. It is calculated as the ratio of TP to the sum of TP and FN (Eq. (3)). The F1-Score, derived from Eq. (4), is the harmonic mean of precision and recall. It provides a balance between precision and recall, allowing for a comprehensive assessment of the model's performance. The F1-Score takes both false positives and false negatives into account and is particularly useful when there is an imbalance between the classes or when both precision and recall are equally important.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The inference time (Eq. (5)) metric uses the model's average time to forecast a picture class. This was achieved by setting a timer at the start and conclusion of the review process. This measure uses milliseconds as its unit.

$$Time = \frac{Training\ Time}{epoch \times batch \times batch\ size} \tag{5}$$

IV. RESULT AND ANALYSIS

In this section, we present the results of our investigation into the utilization of the periocular region in facial expression classification using the TFEID dataset and MobileNet family architectures and EfficientNetV2 family architectures. Figs. 9 and 10 illustrate the performance of the models that were investigated.

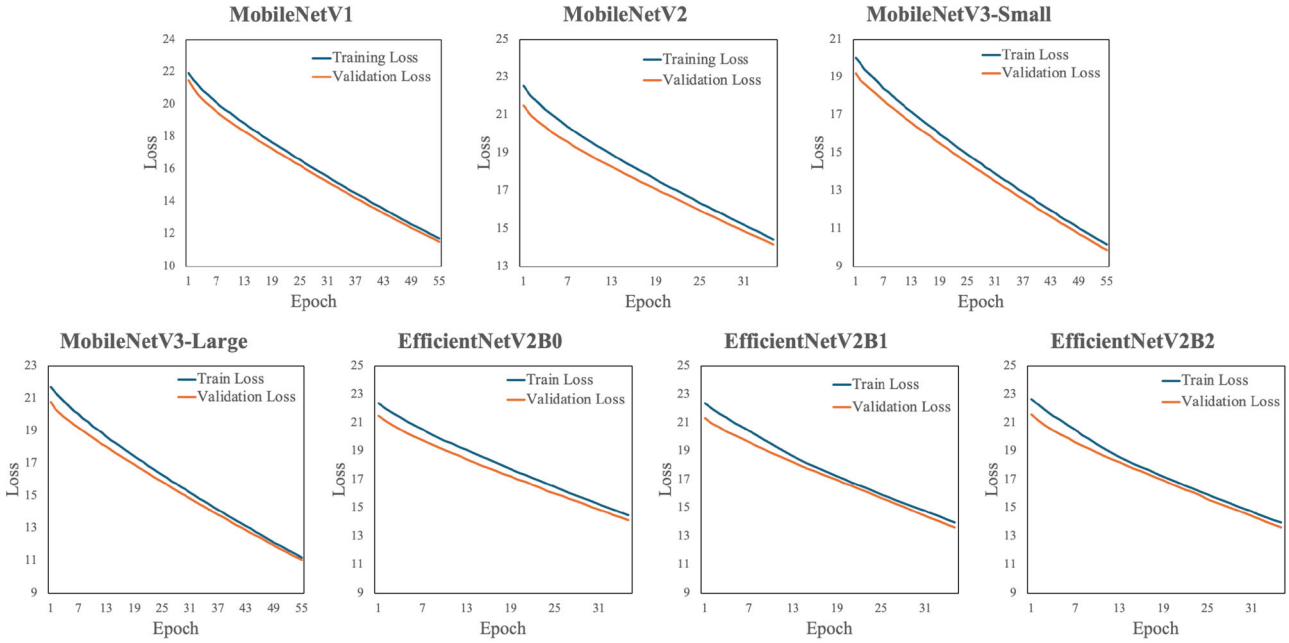


Fig. 9. Training and validation loss curves for each of the seven evaluated models over the training epochs.

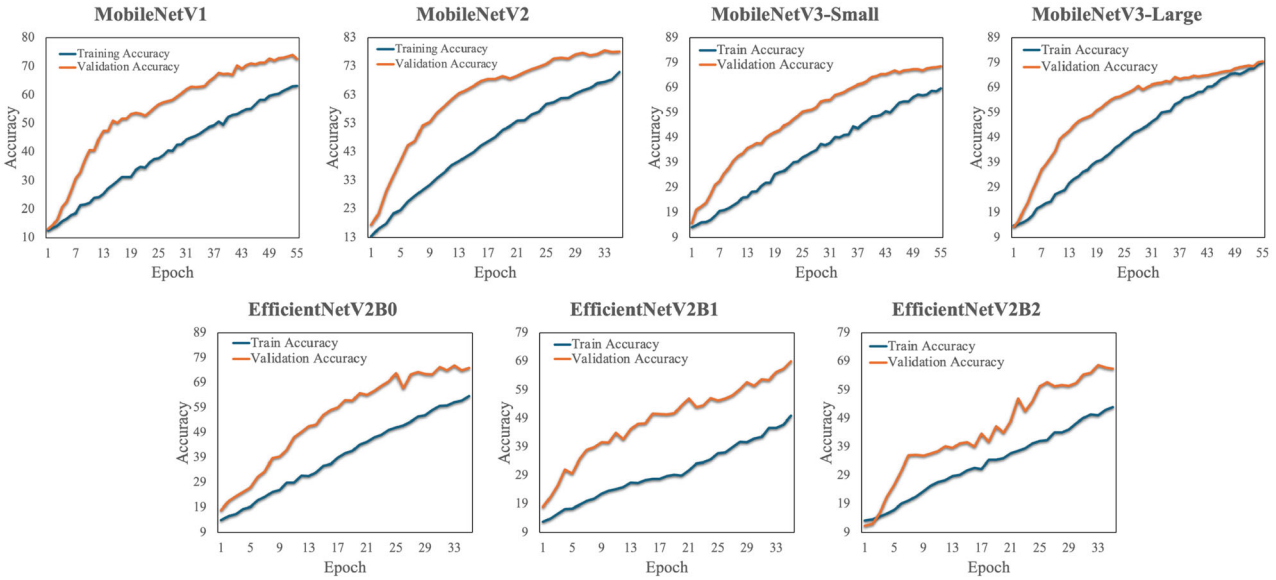


Fig. 10. Training and validation accuracy progression for each model, showing how model learning develops over time.

Table VI illustrates a distinct performance trend: the MobileNet family consistently surpasses the EfficientNet family on this particular periocular dataset. This finding is significant, indicating that MobileNet's architectural design, which emphasizes depth-wise separable convolutions and efficient building blocks, is especially adept at extracting pertinent features from the periocular region, particularly in contrast to EfficientNet's compound scaling approach. The MobileNetV3-Small architecture exemplifies an ideal equilibrium, with the greatest accuracy of 83.62% with an exceptionally short inference time of under 16.4 milliseconds. This model exemplifies a significant trade-off wherein a more compact and efficient architecture outperforms both its larger equivalents and the more intricate EfficientNet models in terms of performance. Although MobileNetV1, MobileNetV2, and MobileNetV3-Large exhibit competitive performance (all achieving accuracies exceeding 81%), MobileNetV3-Small stands out by offering the optimal balance of high accuracy and exceptional speed, rendering it an exemplary choice for resource-limited mobile applications. Conversely, the EfficientNetV2 family exhibited constant underperformance, with accuracy markedly declining as the model size escalated from B0 to B2. The extended inference durations for EfficientNetV2B1 (66.5 ms) and EfficientNetV2B2 (67.9 ms), along with diminished accuracies (76% and 70% respectively), further validate their inadequate appropriateness for this particularly task.

TABLE VI. FINAL PERFORMANCE COMPARISON OF ALL MODELS ON THE TEST SET

Metrics	MobileNet		MobileNetV3		EfficientNetV2		
	V1	V2	Small	Large	B0	B1	B2
[1] I	83.1	82.2	83.6	81.4	80.6	76	70
[2] II	82.8	82.1	82.9	81.3	81.9	71.8	69.6
[3] III	30.2	31.7	16.4	29.1	48.2	66.5	67.9

[1]Accuracy (%); [2]F1-Score (%); [3]Inference time per image (ms).

Based on these findings, it seems like the EfficientNet architecture might need a few more epochs to reach its full potential on this periocular dataset. Overall, MobileNetV3-small emerged as the optimal choice for balancing accuracy, F1-Score, and training time efficiency. On the other hand, the EfficientNet architecture turned out to be less effective for this specific assignment.

The confusion matrix is an effective visualization tool for CNN network performance. Fig. 11 shows the confusion matrix of all the models used to classify eight emotions. This makes it easier to see which classes caused the trained models to be the most inaccurate. Table VII shows the performance graph between emotions, which will further confirm the confusion matrix's findings. The accuracy and F1-Score for each emotion indicate that MobileNetV1 and MobileNetV3-small excel in classifying most emotions, particularly in the classes of disgust, happiness, and surprise, which exhibit consistent and precise prediction levels.

A more granular analysis using the confusion matrices (Fig. 11) provides deeper insights into per-class performance and reveals the models' specific strengths and weaknesses. It becomes evident that across all models, certain emotions are consistently more challenging to classify from the periocular region alone. In particular, fear stands out as the most difficult emotion to categorize, a challenge that is likely due to its subtle and often shared features with other expressions, as seen in the low recall and F1-Scores for this class (Table VII). Similarly, emotions like Anger and Neutral also pose difficulties, suggesting that the primary cues for these expressions may reside in other facial areas (e.g., the mouth or jaw).

Conversely, emotions such as Disgust and Happiness are consistently the easiest to categorize, with all models showing high precision and recall. This indicates that the muscle movements around the eyes and eyebrows for these expressions are highly distinct and serve as strong indicators.

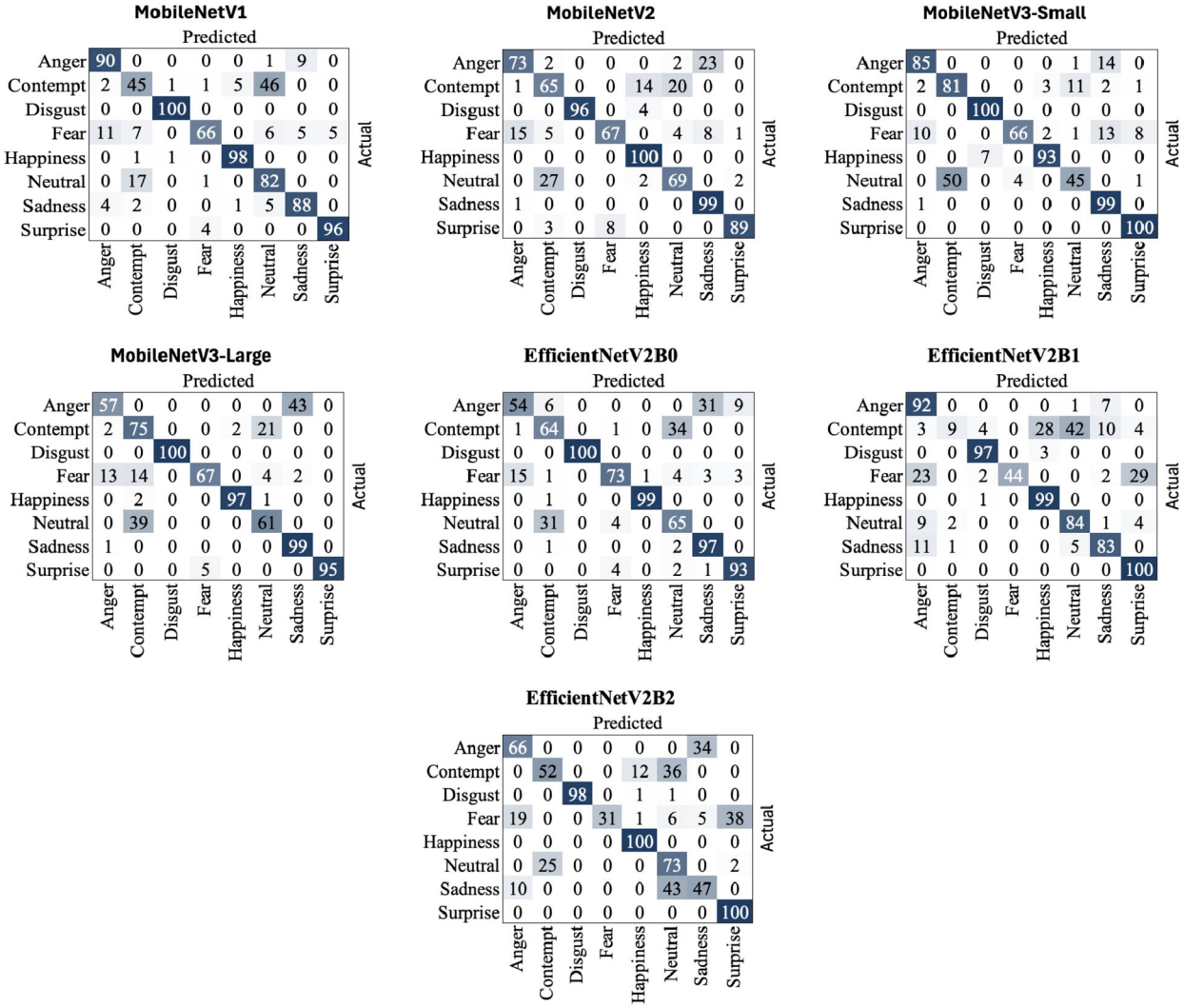


Fig. 11. Confusion matrices for each model, visualizing the classification performance across the eight emotion classes. The diagonal elements represent correctly classified instances.

TABEL VII. DETAILED PER-CLASS PERFORMANCE METRICS (PRECISION, RECALL, F1-SCORE) FOR EACH MODEL

Model Metrics	Model	Anger	Contempt	Disgust	Fear	Happiness	Neutral	Sadness	Surprise
Precision	MobileNetV1	0.8411	0.625	0.9804	0.9167	0.9423	0.5857	0.8627	0.9505
	MobileNetV2	0.8111	0.6373	1	0.8933	0.8333	0.7263	0.7615	0.9674
	MobileNetV3-Small	0.8182	0.783	0.9897	0.871	0.9259	0.7882	0.7111	0.8403
	MobileNetV3-Large	0.7808	0.5769	1	0.9306	0.9798	0.7011	0.6875	1
	EfficientNetV2B0	0.7714	0.6154	1	0.8902	0.99	0.471	0.8818	0.9688
	EfficientNetV2B1	0.6667	0.75	0.9327	1	0.7615	0.6364	0.8058	0.7299
	EfficientNetV2B2	0.6947	0.6753	1	1	0.8772	0.4591	0.5465	0.7143
Recall	MobileNetV1	0.9	0.45	1	0.66	0.98	0.82	0.88	0.96
	MobileNetV2	0.73	0.65	0.96	0.67	1	0.69	0.99	0.89
	MobileNetV3-Small	0.72	0.83	0.96	0.54	1	0.67	0.96	1
	MobileNetV3-Large	0.57	0.75	1	0.67	0.97	0.61	0.99	0.95
	EfficientNetV2B0	0.54	0.64	1	0.73	0.99	0.65	0.97	0.93
	EfficientNetV2B1	0.92	0.09	0.97	0.44	0.99	0.84	0.83	1
	EfficientNetV2B2	0.66	0.52	0.98	0.31	1	0.73	0.47	1
F1-Score	MobileNetV1	0.8696	0.5233	0.9901	0.7674	0.9608	0.6833	0.8713	0.9552
	MobileNetV2	0.7684	0.6436	0.9796	0.7657	0.9091	0.7077	0.8609	0.9271
	MobileNetV3-Small	0.766	0.8058	0.9746	0.6667	0.9615	0.7243	0.817	0.9132
	MobileNetV3-Large	0.659	0.6522	1	0.7791	0.9749	0.6524	0.8115	0.9744
	EfficientNetV2B0	0.6353	0.6275	1	0.8022	0.99	0.5462	0.9238	0.949
	EfficientNetV2B1	0.7731	0.1607	0.951	0.6111	0.8609	0.7241	0.8177	0.8439
	EfficientNetV2B2	0.6769	0.5876	0.9899	0.4733	0.9346	0.5637	0.5054	0.8333

While MobileNetV3-Small is the overall best performer, its strength is particularly pronounced in these more-distinguishable classes. The superior performance of MobileNet models over EfficientNet is also visible at the

class level; the EfficientNetV2B1 and V2B2 models, for instance, exhibit a high degree of misclassification for emotions like contempt and fear, confirming their inferior efficacy beyond just overall accuracy.

While data augmentation was employed to create a balanced training dataset, it is crucial to acknowledge a potential limitation regarding the model's generalizability in real-world scenarios. This situation can cause a majority class bias, which means that a model is more likely to correctly predict dominant classes and does worse on minority classes that aren't seen very often. So, the model does well on our balanced test set, but it might not be as good at detecting rare emotions in real life. This points to a problem that needs more research using more advanced techniques for dealing with imbalances.

Based on its overall better performance, MobileNetV3 was chosen as the best machine-learning model for deployment. The Android smartphone that has been installed with the model will retrieve images from storage to evaluate the performance of the deployment. For the model to make a prediction, the selected image will first be cropped at the periocular area. Each image included in the Chinese Face dataset will be individually identified. All the results of each expression were written down so that they could be kept track of. Following that, the performance of each expression was analyzed using the recorded results (Table VIII). The model performed well in categories with dominant features like Happiness but poorly in categories with ambiguous expressions like Fear. It is evident from Table VIII that the MobileNetV3 model is capable of accurately classifying certain expressions that are consistent with the training results during deployment on Android platforms.

TABLE VIII. ON-DEVICE PERFORMANCE OF THE DEPLOYED MOBILENETV3-SMALL MODEL ON THE CHINESE FACE DATASET

Model Metrics	a	b	c	d	e	f	g
Precision	0.81	0.48	0.54	0.44	0.73	0.47	0.47
Recall	0.14	0.59	0.22	0.90	0.16	0.58	0.75
F1-Score	0.24	0.53	0.31	0.59	0.26	0.51	0.58

a: anger, b: disgust, c: Fear, d: happiness, e: neutral, f: sadness, g: surprise.

TABLE IX. COMPARISON OF MODEL FILE SIZES IN TENSORFLOW LITE (TFLITE) FORMAT

Model	Size (MB)
MobileNetV1	14
MobileNetV2	10.3
MobileNetV3-Small	4.5
MobileNetV3-Large	13
EfficientNetV2B0	24.9
EfficientNetV2B1	28.9
EfficientNetV2B2	36.3

To further evaluate the models' suitability for mobile deployment, the file size of each converted TensorFlow Lite (.tflite) model was measured, as presented in Table IX. The results show that MobileNetV3-Small is by far the lightest model, weighing in at only 4.5 MB, which is much less than any other architecture. For mobile apps, a smaller model size is a key sign of efficiency because it means less storage space, less memory use, and faster loading times. Also, a lighter model usually means less inference latency and better energy use, which makes it a better choice for devices with limited resources and proves that it provides a better user experience.

V. CONCLUSION

This study employs a transfer learning approach with CNN-based models to differentiate between eight distinct emotions. This study makes use of the following CNN models: MobileNetV1, MobileNetV2, MobileNetV3-Small, MobileNetV3-Large, EfficientNetV2B0, EfficientNetV2B1, and EfficientNetV2B2. With accuracies of 83.13%, 82.25%, 83.62%, 80.62%, 76%, and 70% correspondingly, these models demonstrate effective generalization on the TFEID dataset. In addition, the trained model is converted to the TensorFlow Lite version and then used on the Android mobile platform, allowing for the optimal MobileNetV3-Small model.

This study successfully demonstrated the effectiveness of lightweight deep learning models for periocular facial expression recognition. A thorough evaluation of seven architectures, encompassing the MobileNet and EfficientNetV2 families, was performed on the TFEID dataset. The results show that MobileNetV3-Small is the best model because it has the highest accuracy of 83.62%. It is even better because it is only 4.5 MB in size, making it highly efficient for mobile deployment. The successful implementation and testing on an Android platform, validated through testing on the Chinese Face dataset, confirm the practical feasibility of this model for real-world applications where facial visibility is limited, such as in VR environments or due to mask usage. The model performs well, but future research could investigate more advanced methods for dealing with class imbalances in the real world and making it even more robust for difficult emotions like fear and contempt.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

S. B. R. conducted the research, implemented the system prototype, and prepared the initial manuscript draft. D. H. H. supervised the research design, provided conceptual and methodological guidance, reviewed and edited the manuscript, and validated the results. All authors have read and approved the final version of the manuscript.

ACKNOWLEDGMENT

This research was conducted as part of the master's program in Informatics at Universitas Pelita Harapan, Indonesia. The authors gratefully acknowledge the academic guidance, facilities, and support provided by

Universitas Pelita Harapan throughout the course of this study.

REFERENCES

[1] A. Perez, M. D. Fetters, J. Creswell *et al*., "Enhancing nonverbal communication through virtual human technology: Protocol for a mixed methods study," *JMIR Res. Protoc*., vol. 12, no. 1, e46601, 2023.

[2] P. Ekman, "Facial expressions of emotion: New findings, new questions," *Psychol. Sci*., vol. 3, no. 1, pp. 34–38, 1992.

[3] S. Du, Y. Tao, and A. M. Martinez, "Compound facial expressions of emotion," *Proc. Nat. Acad. Sci.*, vol. 111, no. 15, pp. E1454–E1462, 2014.

[4] J. A. Brooks, L. Kim, M. Opara *et al*., "Deep learning reveals what facial expressions mean to people in different cultures," *iScience*, vol. 27, no. 3, 2024.

[5] L. Yao, Y. Wan, H. Ni, and B. Xu, "Action unit classification for facial expression recognition using active learning and SVM," *Multimed. Tools Appl.*, vol. 80, pp. 24287–24301, 2021.

[6] B. A. Aseniero, M. Constantinides, S. Joglekar, K. Zhou, and D. Quercia, "MeetCues: Supporting online meetings experience," in *Proc. IEEE Vis. Conf. (VIS)*, 2020, pp. 236–240.

[7] P. C. Sánchez and C. C. Bennett, "Facial expression recognition via transfer learning in cooperative game paradigms for enhanced social AI," *J. Multimodal User Interfaces*, vol. 17, no. 3, pp. 187–201, 2023.

[8] X. Chen and H. Chen, "Emotion recognition using facial expressions in an immersive virtual reality application," *Virtual Reality*, vol. 27, no. 3, pp. 1717–1732, 2023.

[9] C. Kim, H. S. Cha, J. Kim *et al*., "Facial motion capture system based on facial electromyogram and electrooculogram for immersive social virtual reality applications," *Sensors*, vol. 23, no. 7, p. 3580, 2023.

[10] A. Kumar and K. R. Seeja, "Periocular region based gender identification using transfer learning," *Int. J. Cogn. Comput. Eng*., vol. 4, pp. 277–286, 2023.

[11] R. Sharma and A. Ross, "Periocular biometrics and its relevance to partially masked faces: A survey," *Comput. Vis. Image Understand*., vol. 226, 103583, 2023.

[12] E. Barroso, G. Santos, L. Cardoso, C. Padole, and H. Proença, "Periocular recognition: how much facial expressions affect performance?" *Pattern Anal. Appl*., vol. 19, pp. 517–530, 2016.

[13] M. B. Urtado, R. D. Rodrigues, and S. S. Fukusima, "Visual field restriction in the recognition of basic facial expressions: A combined eye tracking and gaze contingency study," *Behav. Sci.*, vol. 14, no. 5, p. 355, 2024.

[14] J. A. Ballesteros, G. M. Ramírez V, F. Moreira, A. Solano, and C. A. Pelaez, "Facial emotion recognition through artificial intelligence," *Front. Comput. Sci*., vol. 6, 1359471, 2024.

[15] J. Chi, C. K. On, H. Zhang, and S. S. Chai, "A review of deep convolutional neural networks in mobile face recognition," *Int. J. Interact. Mobile Technol.*, vol. 17, no. 23, 2023.

[16] D. Yang and Z. Luo, "A parallel processing CNN accelerator on embedded devices based on optimized MobileNet," *IEEE Internet Things J*., vol. 10, no. 21, pp. 18844–18852, 2023.

[17] O. I. Aboulola, E. A. Alabdulqader, A. A. AlArfaj *et al*., "An automated approach for predicting road traffic accident severity using transformer learning and explainable AI technique," *IEEE Access*, vol. 12, pp. 61062–61072, 2024.

[18] W. T. Chew, S. C. Chong, T. S. Ong, and L. Y. Chong, "Facial expression recognition via enhanced stress convolution neural network for stress," *IAENG Int. J. Comput. Sci*., vol. 49, no. 3, 2022.

[19] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.

[20] H. I. Liu, M. Galindo, H. Xie *et al*., "Lightweight deep learning for resource-constrained environments: A survey," *ACM Comput. Surv*., vol. 56, no. 10, pp. 1–42, 2024.

[21] J. Suzuki, J. Yu, M. Yasunaga *et al*., "Pianissimo: A sub-mW class DNN accelerator with progressively adjustable bit-precision," *IEEE Access*, vol. 12, pp. 2057–2073, 2023.

[22] Y. Gulzar, "Fruit image classification model based on MobileNetV2 with deep transfer learning technique," *Sustainability*, vol. 15, no. 3, p. 1906, 2023.

[23] L. D. Quach, K. N. Quoc, A. N. Quynh *et al*., "Tomato health monitoring system: Tomato classification, detection, and counting system based on YOLOv8 model with explainable MobileNet models using Grad-CAM++," *IEEE Access*, vol. 12, pp. 9719–9737, 2024.

[24] Y. H. Kao and C. L. Lin, "Enhancing diabetic retinopathy detection using pixel color amplification and EfficientNetV2: A novel approach for early disease identification," *Electronics*, vol. 13, no. 11, p. 2070, 2024.

[25] F. Alonso-Fernandez, J. Bigun, and C. Englund, "Expression recognition using the periocular region: A feasibility study," in *Proc. Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, 2018, pp. 536–541.

[26] N. Reddy and R. Derakhshani, "Emotion detection using periocular region: A cross-dataset study," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–6.

[27] T. Tuncer, S. Dogan, and A. Subasi, "Automated facial expression recognition using novel textural transformation," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 7, pp. 9435–9449, 2023.

[28] S. Han, Y. Guo, X. Zhou *et al*., "A Chinese face dataset with dynamic expressions and diverse ages synthesized by deep learning," *Sci. Data*, vol. 10, no. 1, p. 878, 2023.

[29] A. Srivastava, S. Mane, A. Shah, N. Shrivastava, and B. Thakare, "A survey of face detection algorithms," in *Proc. Int. Conf. Inventive Syst. Control (ICISC*), 2017, pp. 1–4.

[30] M. Uppal *et al*., "Enhancing accuracy in brain stroke detection: Multi-layer perceptron with Adadelta, RMSProp and AdaMax optimizers," *Front. Bioeng. Biotechnol.*, vol. 11, 1257591, 2023.

[31] P. Das, S. Gupta, J. Patra, and B. Mondal, "Adamax optimizer and categorical cross entropy loss function-based CNN method for diagnosing Lung cancer," in *Proc. Int. Conf. Trends Electron. Informat. (ICOEI)*, 2023, pp. 806–810.

[32] N. S. Chandel, S. K. Chakraborty, Y. A. Rajwade *et al*., "Identifying crop water stress using deep learning models," *Neural Comput. Appl*., vol. 33, no. 10, pp. 5353–5367, 2021.

[33] E. Manor and S. Greenberg, "Custom hardware inference accelerator for tensorflow lite for microcontrollers," *IEEE Access*, vol. 10, pp. 73484–73493, 2022.

[34] I. Oztel, G. Yolcu Oztel, and V. H. Sahin, "Deep learning-based skin diseases classification using smartphones," *Adv. Intell. Syst*., vol. 5, no. 12, 2300211, 2023.