Facial Emotion Recognition via ResNet-50 with Convolutional Block Attention Module

Deep Learning (COMP6826001)

Semester 5

Anggota:

1. Daniel Crispalito - 2702377726
2. Haikal Asfa Audri - 2702342496
3. Rizkyan Alif Malikulsyah - 2702369402

Universitas Bina Nusantara

Abstract

This project implements a deep learning-based Facial Emotion Recognition (FER) system capable of classifying human facial expressions into seven discrete categories. The model is built using a Residual Network - 50 (ResNet-50) with Attention mechanism using architecture with batch normalization and dropout regularization to improve generalization performance. Training was conducted on a dataset of 100x100 RGB facial images. The resulting model achieved a final testing accuracy of 82% and a loss of 0.9862, indicating reasonably good performance for a baseline FER model, though with room for improvement. The project includes dataset preprocessing, model development, experimental evaluation, and analysis of limitations and opportunities for future enhancements.

1. Introduction

Facial emotion recognition has been a significant research domain in Artificial Intelligent especially in Computer Vision because it enables computers to see like human eyes and can classify emotion based on facial expression. The ability to recognize emotions automatically allows intelligent systems to better understand human behavior and emotional context.

This technology has the potential to help solve real-world problems across many fields, including healthcare, education, public safety, customer service, and social research. In healthcare, FER can support early detection of mental health conditions, stress monitoring, and patient emotion assessment during treatment. In education, it can provide insights into student engagement, learning experience, and emotional well-being in classrooms or online learning environments. In public safety, FER can assist surveillance systems by identifying aggressive or distressful emotions in crowds as part of risk mitigation. Meanwhile, in customer service, emotion-aware systems can improve user experience by adapting responses based on customer emotions. Additionally, FER contributes to social and behavioral research by enabling large-scale, non-intrusive analysis of human emotional feedback without relying solely on verbal communication.

This project explores a deep learning approach to emotion classification by combining the strong feature extraction capability of ResNet-50 with attention mechanism of Convolutional Block Attention Mechanism (CBAM). ResNet-50 is used as the backbone network to learn hierarchical facial features, while CBAM enhances the model by focusing on the most informative spatial regions and channel representations, allowing the system to better capture subtle emotional cues.

By implementing this architecture, the project aims to build a reliable FER system that not only classifies emotions effectively but also leverages attention-based learning to optimize performance in human-centered AI applications.

2. Related Work

Facial expression recognition (FER) has progressed from psychology-driven studies to advanced deep learning–based computer vision systems. Early foundational work by Ekman and Friesen demonstrated that a set of basic facial expressions are universal across cultures, forming the theoretical basis for categorical emotion recognition models [1]. These findings motivated the development of early computational FER systems and standardized evaluation protocols such as FERET, which provided benchmark datasets and evaluation methodologies for facial analysis algorithms [2]. However, early FER approaches relied heavily on handcrafted features and shallow classifiers, limiting their robustness in unconstrained environments.

Traditional FER methods commonly employed handcrafted feature descriptors such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gabor filters combined with classifiers including Support Vector Machines (SVMs) and Hidden Markov Models [3], [4]. While these methods achieved moderate success under controlled conditions, they struggled with variations in illumination, head pose, occlusion, and inter-subject facial diversity. The inability of handcrafted features to capture high-level semantic and contextual information restricted their scalability to real-world scenarios.

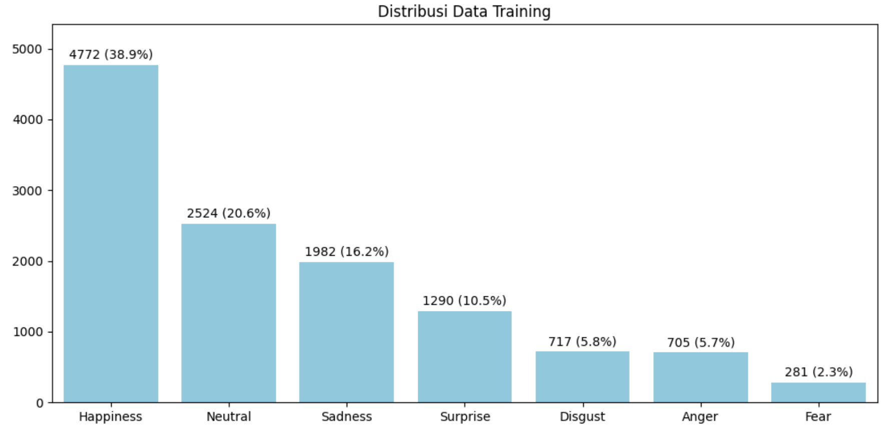
The emergence of deep convolutional neural networks (CNNs) significantly advanced FER performance by enabling end-to-end feature learning directly from raw image data. Architectures such as VGGNet, GoogLeNet, and ResNet have demonstrated superior representation learning capabilities compared to traditional approaches [5], [6]. In particular, ResNet introduced residual connections that alleviate the vanishing gradient problem, allowing deeper networks to be trained effectively. Several studies have shown that ResNet-based backbones outperform shallow CNN architectures in FER tasks, especially when modeling subtle facial expressions and complex emotional patterns [7].

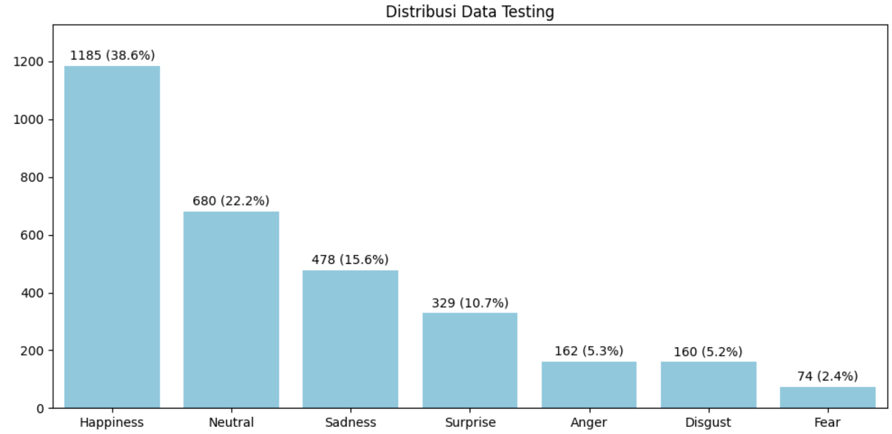
More recent research has incorporated attention mechanisms to further enhance FER performance by guiding networks to focus on emotionally salient facial regions. Attention-based modules enable models to emphasize critical areas such as the eyes, eyebrows, and mouth while suppressing irrelevant background information. The Convolutional Block Attention Module (CBAM), which sequentially applies channel and spatial attention, has been shown to improve feature discrimination across various computer vision tasks [8]. In the context of FER, integrating attention mechanisms with deep CNN backbones has resulted in improved recognition accuracy and robustness, particularly for fine-grained emotion classification [9]. These findings motivate the adoption of a ResNet-50 architecture enhanced with CBAM in this project as a strong yet interpretable baseline toward state-of-the-art FER systems.

3. Methodology

3.1. Dataset

This project uses a public dataset, namely the Real-world Affective Faces Database (RAF-DB), which is sourced from Kaggle. The total number of available data samples is 15,339, consisting of 12,271 training samples and 3,068 testing samples. The dataset is divided into seven classes: Surprise, Fear, Disgust, Happiness, Sadness, Anger, and Neutral. The class distribution is as follows:





All available images have a resolution of 100×100 pixels and are in RGB color format.

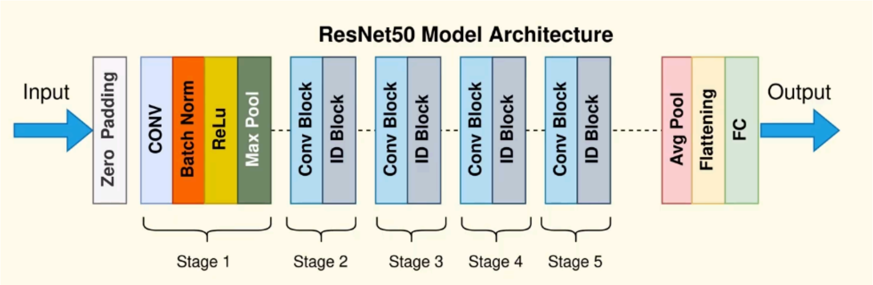
3.2. Preprocessing

The previously analyzed dataset undergoes a comprehensive preprocessing procedure that includes several stages.

- Duplicate images are identified using the MD5 hashing method, in which each image file is converted into a binary representation and its MD5 hash value is computed. Images with identical hash values are considered duplicates, and only one instance is retained.
- Data augmentation is applied, consisting of resizing all images to 224 × 224 pixels, adjusting brightness by 20% and contrast by 20% relative to their original values through color jittering, and performing pixel normalization using a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225] means that each RGB image is normalized on a per-channel basis by subtracting the corresponding mean value and dividing by the corresponding standard deviation.
- The test dataset is further divided to create a validation set, with a split ratio of 50% for validation and 50% for testing.

3.3. Model Architecture

3.3.1 Residual Network - 50 (ResNet-50)



Residual Network-50 (ResNet-50) is a convolutional neural network (CNN) model that consists of 50 layers and incorporates residual learning with skip connections, which effectively addresses the vanishing gradient problem. ResNet-50 employs an approach that allows the network to learn corrections to the input rather than a full mapping, by learning a residual function F(x), which represents the difference between the desired mapping and the input. Therefore, this relationship can be expressed as:

$$F(x) = H(x) - x$$

Thus, the output of the residual block is given by:

$$H(x) = F(x) + x$$

A skip connection is a shortcut path that connects the input of a residual block directly to the output of that block through an element-wise addition operation. Mathematically, the gradient with respect to the input can be derived as:

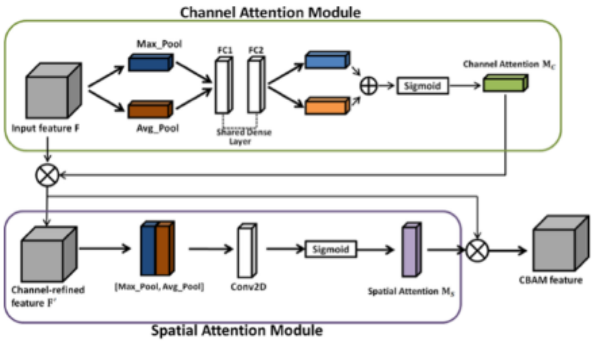
$$\partial H(x)/\partial x = \partial F(x)/\partial x + 1$$

The presence of the +1 constant ensures that the gradient does not become zero.

The data flow in ResNet-50 proceeds linearly from left to right through five main stages:

1. **Initial Stage (Preprocessing & Stage 1):**
   The input data is first processed using zero padding to preserve spatial dimensions, then passed through an initial convolutional (CONV) layer with a 7×7 filter. After that, the data goes through Batch Normalization to stabilize the training process, followed by a ReLU activation function, and Max Pooling to reduce the initial spatial dimensions.
2. **Feature Extraction Stage (Stage 2 – Stage 5):**
   The data passes through a series of residual blocks, which are divided into two types:
   a. **Convolutional Block (Conv Block):** Used when there is a change in dimensions, typically at the beginning of each stage.
   b. **Identity Block (ID Block):** Used when the input and output dimensions are the same, allowing information to “skip” directly across layers through a shortcut connection.
3. **Final Stage (Output):**
   After features are deeply extracted up to Stage 5, the data passes through Average Pooling to aggregate feature values, followed by a flattening process into a one-dimensional vector. Finally, a fully connected (FC) layer or dense layer determines the final classification of the image.

3.3.2 Convolutional Block Attention Module (CBAM)



CBAM is a lightweight attention module that is integrated into a Convolutional Neural Network (CNN) to help the model focus on the most important features by enhancing relevant feature maps and suppressing less important ones. The main idea of CBAM is divided into two components, namely:

1. **Channel Attention Module :**

   The Channel Attention Module (CAM) in CBAM aims to determine which features (*what*) are the most important by exploiting the inter-channel relationships within the feature map. Each channel is considered a feature detector; therefore, not all channels contribute equally to the final representation.
   CBAM combines average pooling and max pooling because:

   - **Average pooling** captures global contextual information.
   - **Max pooling** highlights the most distinctive and salient features.

   The channel attention process begins by aggregating spatial information from the feature map using two approaches, namely global average pooling and global max pooling. Global average pooling captures global contextual information from each channel, while global max pooling emphasizes the most salient feature responses. The combined use of these two pooling methods enables the module to obtain a richer channel representation compared to using a single pooling operation.

   The outputs of both pooling operations are then passed through a multi-layer perceptron (MLP) with shared weights. This MLP is designed to model inter-channel dependencies while maintaining parameter efficiency through a dimensionality reduction mechanism. The outputs from the two pooling paths are subsequently fused and transformed into a channel attention map that reflects the importance of each channel.

   The resulting channel attention map is finally applied to reweight the original feature map by enhancing channels with higher contributions and suppressing those that are less relevant.

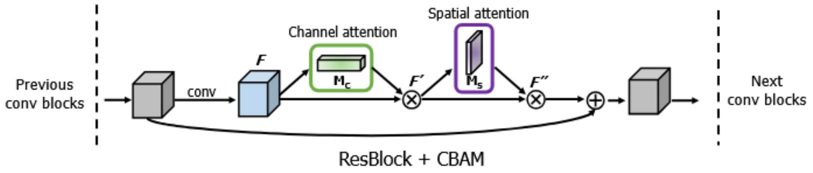
2. **Spatial Attention Module :**

   The Spatial Attention Module in CBAM aims to determine *where* the most important information is located within the feature map. Unlike channel attention, which focuses on selecting the type of features, spatial attention exploits inter-spatial relationships to highlight informative regions and suppress less relevant areas.

This process begins by taking the ***channel-refined feature F′*** as the main input. To capture spatial relationships among pixels, the module applies two pooling operations simultaneously along the channel axis: max pooling to capture the most salient features and average pooling to obtain mean information across all channels. The outputs of these pooling operations are then concatenated into a compact feature descriptor. This combined descriptor is further processed using a convolutional layer (Conv2D), which extracts spatial patterns and produces a two-dimensional feature map. The resulting map is passed through a sigmoid activation function to normalize its values to the range between 0 and 1, forming the Spatial Attention Map (Ms).

In the final stage, the spatial attention map is multiplied with the input feature 'F′. The result of this operation is the refined CBAM feature representation.

3.3.3 ResNet - 50 with Light CBAM



In the ResNet-50 architecture enhanced with the Convolutional Block Attention Module (CBAM), the CBAM module can conceptually be integrated into every residual block to enhance feature representations through channel and spatial attention mechanisms. However, applying CBAM to all residual blocks significantly increases model complexity and computational cost.

Therefore, this study adopts a Light CBAM strategy, in which the CBAM module is applied only to the last residual block of layer4, which corresponds to the stage that extracts high-level semantic features. This approach aims to achieve a balance between improved feature representation and computational efficiency by focusing the attention mechanism on the most discriminative features for classification.

With this strategy, attention is not applied to all layers of the ResNet-50 architecture, but selectively to the final stage of the convolutional network. The remaining components of the network strictly follow the standard ResNet-50 architecture, as described in Section 3.3.1, including the use of residual learning and skip connections to ensure training stability and to mitigate the vanishing gradient problem.

3.4. Training

3.4.1 Training Method

The training strategy employed in this study is **Two-Stage Transfer Learning with Progressive Fine-Tuning.** In the first stage, all parameters of the ResNet-50 backbone with Light CBAM are frozen, except for the fully connected classification layer. This stage aims to adapt the classifier to the target dataset distribution, leverage generic features learned from the ImageNet dataset, and prevent catastrophic forgetting of the pretrained weights.

In the second stage, fine-tuning is performed by unfreezing all network parameters and retraining the model using a lower learning rate. This process allows the model to adjust high-level feature representations, particularly in layer4 and the CBAM module, thereby optimizing the feature extractor for the specific characteristics of the target domain. The objective of this stage is to further enhance the model's discriminative capability and improve final classification accuracy.

To mitigate overfitting and ensure efficient training, an early stopping mechanism is applied with a patience value of 10. The stopping criterion is based on validation accuracy, where training is terminated if no improvement is observed over a predefined number of consecutive epochs.

3.4.2 Training Configuration

The model was compiled and trained using the following configuration :

Component	Description
Optimizer	Adam optimizer was used to update model parameters during training.
Loss Function	Categorical Crossentropy was employed to measure the difference between predicted and true emotion class distributions.
Evaluation Metric	Accuracy was used as the primary metric to evaluate classification performance.

Maximum Epochs	The model is initially set to train for 10 epoch in stage 1 (where all the ResNet layers are frozen), and 70 epoch in stage 2 (where all the layers are unfrozen)
Model Checkpoint	ModelCheckpoint was used to save the model with the highest validation accuracy during training.
Early Stopping	EarlyStopping was applied to terminate training if the validation loss showed no improvement for 10 consecutive epochs.
Learning Rate	A learning rate of **0.001** is used during the first training stage, while a smaller learning rate of **1×10^{-5}** is applied in the second stage.

3.5. Evaluation Metrics

The metrics used to measure the model's performance include :

- **Accuracy** : Measures the total proportion of correct predictions

  Accuracy = TP+TN / (TP+TN+FP+FN)
- **Precision** : Measures the proportion of correct positive results (True Positives) out of all results that the model predicted as positive.

  Precision = TP / (TP + FP)
- **Recall** : Measures the proportion of correct positive results (True Positives) out of all samples that are truly positive.

  Recall = TP / (TP + FN)
- **F1 - Score** : Harmonic mean of Precision and Recall.

  F1 - Score = 2 * (Precision * Recall) / (Precision + Recall)
- **Confusion Matrix** : Table used to describe the performance of a classification model on a set of test data for which the true values are known.

4. Implementation

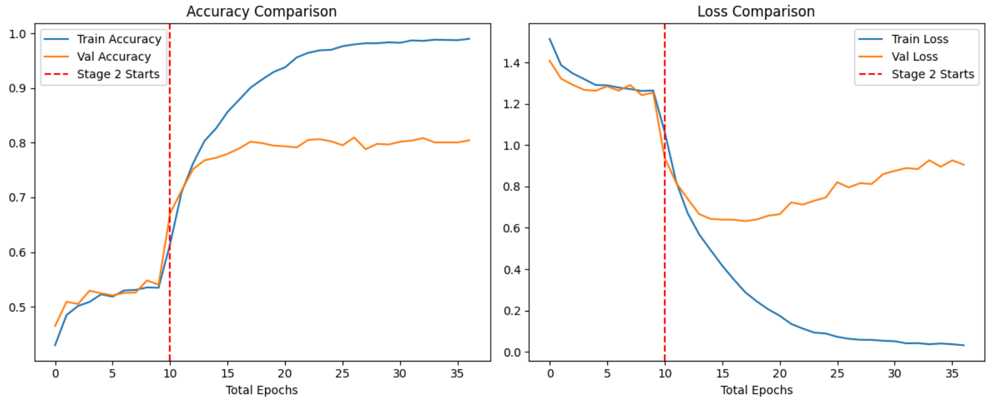
4.1. System details

Framework and Library used to develop our project:

Framework/Library	Description
Python	Programming language that used to develop the model.
PyTorch	Python framework to build and to train the model

Torchvision	Library used for loading the ResNet-50 Model and performing image transformation operation
Pandas	Library used for reading csv file
kagglehub	Library used for downloading dataset from kaggle.
PIL	Library used for opening image
NumPy	Library used for numerical computation.
Matplotlib	Library used for data and model visualization.
Seaborn	Library used for visualization
Scikit-learn	Library used for splitting dataset into train and validation, and model evaluation
Flask	Python framework used for deploying the ResNet-50+CBAM model.

4.2. Results



The model was trained for 37 epochs and divided into 2 stages. In stage 1, the ResNet layers were frozen, it only trained the classifier layer. The model trained for 10 epochs. Both accuracy and loss on the training and validation data showed consistent improvement, although accuracy remained around 50%.

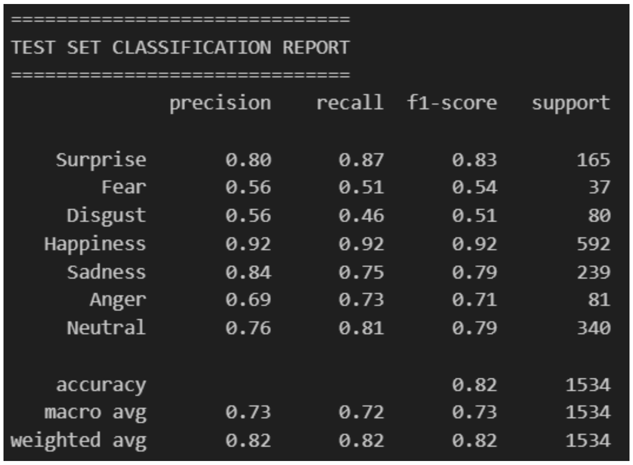
In stage 2, all layers were unfrozen, so it trains all the layers. The model is initially set to train for 70 epochs, but the early stopping mechanisms stopped the training at epoch 27. In total, the model learned for 37 epochs.

After entering stage 2, the model gained a sharp performance increase, both its accuracy and loss. However by around epoch 12-13, the train accuracy continued to increase almost achieving 1 while the validation accuracy stagnated.

Similarly, the train loss is decreased almost to 0, whereas the validation loss is increasing. This indicates that the model might have started to memorize the data instead of learning from it, and might indicate the model is starting to overfit.

The EarlyStopping mechanism monitored the validation loss with a patience of 10 epochs, ensuring the model did not overfit significantly by stopping training when the model stopped improving. It stopped the model training at epoch 37 (stage 2, epoch 27) and the ModelCheckpoint restored the model from epoch 27 (stage 2, epoch 17) which represented the best model to generalize validation data.

The final model evaluated using the test set and achieved an accuracy of 82% and loss around of 0.9862.



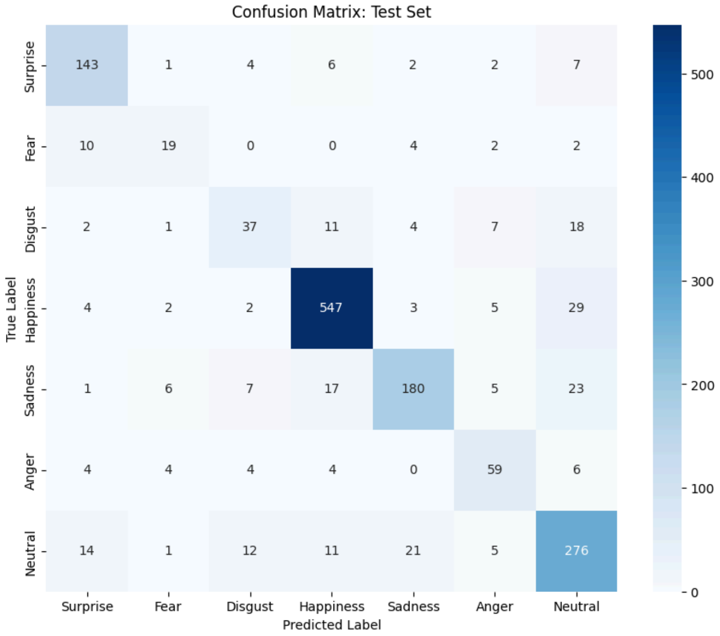
```
==============================
TEST SET CLASSIFICATION REPORT
==============================
              precision    recall  f1-score   support

    Surprise       0.80      0.87      0.83       165
        Fear       0.56      0.51      0.54        37
     Disgust       0.56      0.46      0.51        80
   Happiness       0.92      0.92      0.92       592
     Sadness       0.84      0.75      0.79       239
       Anger       0.69      0.73      0.71        81
     Neutral       0.76      0.81      0.79       340

    accuracy                           0.82      1534
   macro avg       0.73      0.72      0.73      1534
weighted avg       0.82      0.82      0.82      1534
```

The evaluation results indicate that the model performs quite well in several classes, particularly in classes Happy and Surprise, with highest F1 scores. However, there is room for improvement, especially in class Disgust, which has the lowest recall value (0.46). Further adjustments to the model or feature selection could enhance the overall results.

5. Discussion & Limitations



This figure shows the final model's confusion matrix on the test set. From the analysis of the confusion matrix, the model best classification is the "Happiness" class, followed by the "Surprise" class as the second best classification based on the test dataset, likely due to distinct image features, which make the model easier to differentiate them from other images.

On the other hand, the model's worst classification is the "Disgust" class. The model often misclassified "Disgust" class as "Neutral", likely due to the similarity in image features.

The second model's worst classification is the "Fear" class. It is often misclassified as "Surprise", likely because both "Fear" and "Surprise" share similar features such as wide-open eyes and an open mouth.

6. Conclusion & Future Work

6.1. Conclusion

The ResNet-50 with Light CBAM model demonstrates strong performance in facial emotion recognition, achieving a testing accuracy of 82% with a loss of 0.9862. Although the model attains a training accuracy of up to 99%, the gap compared to the best validation accuracy of 0.8096 indicates a tendency toward *overfitting*. Nevertheless, the application of two-stage transfer learning, progressive fine-tuning, and early stopping effectively mitigates overfitting and helps maintain the model's generalization capability.

6.2. Future Work

To further enhance the performance of the FER system, several future improvements can be explored:

1. **Integration of attention mechanisms at multiple stages**

   Future studies may explore applying CBAM to additional residual blocks or comparing Light CBAM with other attention mechanisms, such as SE-Block or Transformer-based attention, to further enhance feature representation.

2. **Cross-dataset and domain adaptation**

   Training and evaluating the model on multiple facial expression datasets or applying domain adaptation techniques may increase robustness across varying data distributions and real-world conditions.

3. **Semi-supervised and self-supervised learning**

   Leveraging unlabeled data through self-supervised or semi-supervised learning approaches could further improve feature extraction capabilities.

7. References

[1] P. Ekman and W. V. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement. Palo Alto, CA, USA: Consulting Psychologists Press, 1978.

[2] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[3] S. Shan, W. Gao, B. Cao, and D. Zhao, “Facial expression recognition based on Local Binary Patterns: A comprehensive study,” Image and Vision Computing, vol. 27, no. 6, pp. 803–816, May 2009.

[4] M. Pantic and L. J. M. Rothkrantz, “Automatic analysis of facial expressions: The state of the art,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1424–1445, Dec. 2000.

[5] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[6] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” in Proc. European Conf. Computer Vision (ECCV), 2018, pp. 3–19.

[9] Y. Li, J. Zeng, S. Shan, and X. Chen, “Occlusion-aware facial expression recognition using CNN with attention mechanism,” IEEE Transactions on Image Processing, vol. 28, no. 5, pp. 2439–2450, May 2019.

8. Appendix

A. Team Contribution

1. Daniel Crispalito :
    - Design and developed model ResNet-50 with LightCBAM
    - Implemented model training and evaluation
2. Rizkyan Alif Malikulsyah :
    - Collected and curated the dataset used in the project
    - Assisted in debugging and fixing implementation issues during model development
3. Haikal Asfa Audri :
    - Performed data preprocessing, including normalization and data preparation
    - Contributed to writing and structuring the project report

Link Drive Demo Video :
https://drive.google.com/file/d/1xLDN1qpFLns710qJEor1LRx9tJU2NvRG/view?usp=sharing

Link Presentasion Slide :
https://www.canva.com/design/DAG7fGJyw8c/4snq4ZmC0n-4u9Xgklvtxg/edit?utm_content=DAG7fGJyw8c&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Link GitHub :
https://github.com/DanielCrispalito/Face_emotion_recog_via_resnet50_withCBAM