EMOTION DETECTION MODEL USING CONVOLUTIONAL NEURAL NETWORK

Jan Allen R. Bernabe
College of Computing and Information Technologies
National University
Manila, Philippines
bernabejr@students.national-u.edu.ph

Christian Bongao
College of Computing and Information Technologies
National University
Manila, Philippines
bongaoci@students.national-u.edu.ph

Justin Neo R. Flores
College of Computing and Information Technologies
National University
Manila, Philippines
floresjr1@students.national-u.edu.ph

Trixia Nicole A. Libunao
College of Computing and Information Technologies
National University
Manila, Philippines
libunaota@national-u.edu.ph

***Abstract*—Facial emotion recognition is a crucial aspect of human-computer interaction, with practical applications in healthcare, security, automotive safety, and psychological assessment. This study presents a deep learning-based approach to facial emotion detection using Convolutional Neural Networks (CNNs) trained on a publicly available dataset from Kaggle, comprising 48x48 grayscale images labeled with seven distinct emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The dataset includes 28,709 training samples and 3,589 testing samples. Our approach involves preprocessing the images, and evaluating its performance based on accuracy and loss metrics across multiple epochs. The trained model is then converted to TensorFlow Lite format and integrated into a mobile application developed in Kotlin using Android Studio, allowing users to upload facial images and receive emotion predictions in real time. The results demonstrate the model's potential for reliable emotion classification, achieving strong predictive performance and usability on mobile platforms. This project underscores the viability of deploying deep learning models for emotion recognition in practical, real-world applications.**

***Index Terms*—Facial Emotion Recognition, Convolutional Neural Networks (CNN), Deep Learning, Human-Computer Interaction (HCI), Mobile Application, TensorFlow Lite, Image Classification**

I. Introduction

Emotions are mental reactions and expressions, typically involving the use of one's facial features. According to Britannica, they are complex experiences of consciousness, bodily sensation, and behavior that reflect the personal significance of a thing, an event, or a state of affairs. Facial and emotion detection and recognition have a plethora of real-life applications, such as determining the medical state and comfort level of a patient, behavior analysis in security systems to prevent potential crimes, monitoring of a driver's emotional state to implement safety features, and more. With these examples, machine learning techniques have emerged to detect and recognize facial emotions through either real-time implementation or analysis of uploaded images.

This study makes use of a dataset from Kaggle, which contains a training set of 28,709 examples and a public test set of 3,589 examples. The pictures are grayscale images of random faces sized at 48×48 pixels, cropped and centered to showcase varied emotions. They are categorized into seven classes: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

One of the primary challenges of facial emotion recognition is the struggles of human-computer interaction in processing images and determining emotion. Constantine et al. (2016) developed a framework for emotion recognition from HCI in natural settings, addressing challenges such as unobtrusive data collection and mapping digital interactions to human activities. Kanna R et al. (2022) proposed a computational model for emotion detection using facial recognition, achieving 88% accuracy with the EMOTIC database. Jain (2019) reviewed automatic emotion identification methods based on facial expressions, discussing their potential applications in enhancing HCI and mental health assessment. Our research focuses on implementing convolutional neural networks to detect facial emotions and expressions and categorize them into one of seven emotion categories. We used plots to showcase model accuracy and loss over training epochs. We also created a mobile Kotlin application in Android Studio that utilizes a TensorFlow Lite model, allowing the user to upload their own images, with the application providing a text prediction. The objective of this study is to develop an accurate facial emotion detection and recognition model that minimizes model loss while maintaining a high confidence rate. By leveraging machine learning-based facial detection methods, we aim to provide a more reliable and accurate facial detection framework for real-life educational and professional purposes.

II. Review of Related Literature

A. Overview of Key Concepts and Background Information

Facial Emotion Recognition (FER) is a critical component of affective computing that aims to enable machines to identify and interpret human emotional states through facial expressions. Emotions are defined as complex experiences involving conscious thought, physiological arousal, and expressive behavior (Britannica, n.d.). In the context of computer vision, these expressions can be captured and analyzed to inform intelligent systems about a user's affective state.

The development of FER systems is particularly valuable in applications such as healthcare monitoring, psychological diagnostics, automotive safety, and security systems. However, the accurate interpretation of emotions remains a challenge due to variations in facial features, lighting conditions, occlusions, and individual differences in expression. Constantine et al. (2016) addressed these challenges by proposing a framework for emotion recognition through human-computer interaction (HCI) in natural environments. Their study highlighted the importance of unobtrusive data collection and the need to contextualize digital interaction data in order to improve the system's emotional inference capabilities.

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have been widely adopted in FER tasks due to their ability to automatically extract hierarchical features from image data. Kanna et. al (2022) demonstrated the effectiveness of CNN-based emotion recognition systems using the EMOTIC dataset, achieving an accuracy rate of 88%. Their approach underscores the strength of deep learning in generalizing across complex emotional expressions and real-world conditions. Similarly, Jain (2019) provided an extensive review of facial expression-based emotion recognition methodologies. The study categorized various machine learning and deep learning techniques employed in FER systems, and discussed their practical applications, including mental health assessment and adaptive user interfaces. Jain emphasized the growing relevance of automated emotion recognition systems in improving HCI and enhancing user-centered technologies. These studies collectively provide a strong conceptual and technical foundation for the present research. The implementation of CNNs, supported by prior empirical successes, informs the methodology for building an accurate FER model. Furthermore, the deployment of such models into mobile applications using lightweight formats like TensorFlow Lite represents a practical advancement in making FER tools more accessible and applicable in real-time environments.

B. Review of Other Relevant Research Papers

Pereira et al. (2024) conducted a comprehensive systematic review of CNN-based facial emotion recognition models, examining 77 research papers involving datasets such as FER2013, CK+, and JAFFE. Their study identified several high-performing CNN architectures like VGGNet, ResNet, and Inception, highlighting key challenges such as emotion ambiguity, variation in lighting conditions, and class imbalance. The authors emphasized the importance of lightweight CNNs and attention mechanisms in improving recognition accuracy and efficiency in real-world applications. Agung et al. (2024) developed a CNN-based facial emotion recognition system using transfer learning with InceptionV3 and MobileNetV2 models. Their study utilized the Emognition dataset, which contains ten emotion categories, expanding beyond the traditional seven. The CNN models were fine-tuned and achieved up to 96Kusno and Chowanda (2023) presented a comparative study involving two CNN architectures applied to the FER2013 and CK+ datasets. The models achieved significantly better accuracy on the CK+ dataset due to its clearer emotional expressions and consistent image quality. In contrast, the FER2013 dataset led to overfitting due to its complexity and noise. The study highlighted the importance of data quality, augmentation, and regularization in training robust CNNs for emotion recognition. Parel Hakim et al. (2024) proposed a real-time facial emotion detection system using CNNs integrated with OpenCV and webcam input. Trained on FER2013, their model achieved 85To address the challenges posed by occlusion, Ghukasyan et al. (2022) introduced a hybrid FER model that combines CNN-based feature extraction with an SVM classifier. Their system was evaluated on a dataset with masked facial expressions and showed superior accuracy compared to traditional SoftMax-based CNNs. This approach is particularly relevant in post-pandemic environments where mask-wearing can obscure key facial features. Several recent studies have explored methods to improve the interpretability of CNN-based facial emotion recognition (FER) systems. One widely adopted approach is Gradient-weighted Class Activation Mapping (Grad-CAM), which provides visual explanations by highlighting regions in facial images that most influence the model's predictions. For instance, Selvaraju et al. (2017) introduced Grad-CAM as a model-agnostic method to generate class-discriminative localization maps, enhancing the transparency of CNN-based decisions. In the context of FER, such visualizations can help identify whether the model is focusing on key facial areas (e.g., eyes, mouth, eyebrows) relevant to emotional expression. While many studies prioritize accuracy and speed, incorporating interpretability techniques is becoming increasingly important for applications in healthcare, education, and safety, where trust and accountability in AI models are critical.

III. Methodology

This study follows a systematic approach to fraud detection using anomaly detection techniques. The methodology consists of several key stages, including data preprocessing, feature engineering, model selection, evaluation, and interpretability analysis.

A. Environment Setup

The development and training of the facial emotion recognition model were conducted using Google Colaboratory (Colab), an online platform that provides a cloud-based Python environment with GPU acceleration. The model was implemented using TensorFlow and Keras, which are widely

used libraries for deep learning tasks. Supporting libraries included NumPy for numerical operations, Matplotlib and Seaborn for visualization, and scikit-learn for evaluation metrics. The dataset was accessed through Google Drive, which was mounted within Colab.

For deployment, the trained model was converted into TensorFlow (TFLite) format and integrated into a mobile application developed using Kotlin in Android Studio, enabling local emotion prediction on user-uploaded facial images.

B. Data Collection and Preprocessing

The dataset utilized in this study was sourced from Kaggle and consisted of grayscale facial images with a resolution of 48x48 pixels. These images were organized into directories based on seven emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The dataset included a total of 28,709 training images and 3,589 test images. To ensure consistent formatting and improve model performance, all pixel values were rescaled to a range between 0 and 1 by dividing each value by 255.

Preprocessing was conducted using the ImageDataGenerator class from the Keras library, which enabled real-time augmentation of the training data. Data augmentation techniques applied to the training images included random rotations, width, and height shifts, zoom operations, and horizontal flips. These augmentations helped prevent overfitting and improved the model's generalization capability. The training set was further divided into training and validation subsets, with 10 percent of the images reserved for validation. Additionally, the emotion labels were transformed into one-hot encoded vectors to facilitate multi-class classification using a softmax activation function in the output layer.

C. Model Implementation

The facial emotion detection model was implemented using a Convolutional Neural Network (CNN) architecture within the Keras framework. The model was designed to process 48x48 grayscale images and classify them into one of the seven defined emotional categories. The CNN architecture included convolutional layers for feature extraction, activation functions to introduce non-linearity, max pooling layers for downsampling, and fully connected layers for classification. The model was compiled using the Adam optimizer and categorical cross-entropy as the loss function, which is appropriate for multi-class classification problems. During training, a batch size of 64 was used, and the model was trained over multiple epochs. To improve the efficiency and performance of the training process, several callback functions were incorporated. EarlyStopping was used to halt training when the validation loss ceased to improve, while ReduceLROnPlateau was applied to decrease the learning rate when performance plateaued. ModelCheckpoint was used to save the version of the model that achieved the best validation accuracy. Throughout the training phase, plots of training and validation accuracy and loss were generated to visualize the model's learning progress. After achieving satisfactory performance, the trained model was converted into TensorFlow Lite format to support efficient deployment in mobile applications, allowing real-time emotion prediction on user-uploaded images through the Android-based application.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 64)	640
batch_normalization (BatchNormalization)	(None, 48, 48, 64)	256
conv2d_1 (Conv2D)	(None, 48, 48, 64)	36,928
batch_normalization_1 (BatchNormalization)	(None, 48, 48, 64)	256
max_pooling2d (MaxPooling2D)	(None, 24, 24, 64)	0
dropout (Dropout)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 24, 24, 128)	512
conv2d_3 (Conv2D)	(None, 24, 24, 128)	147,584
batch_normalization_3 (BatchNormalization)	(None, 24, 24, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 128)	0
dropout_1 (Dropout)	(None, 12, 12, 128)	0
flatten (Flatten)	(None, 18432)	0
dense (Dense)	(None, 256)	4,718,848
batch_normalization_4 (BatchNormalization)	(None, 256)	1,024
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 7)	1,799

Total params: 4,982,215 (19.01 MB)
Trainable params: 4,980,935 (19.00 MB)
Non-trainable params: 1,280 (5.00 KB)

Fig. 1. Layers used in the Model

D. Model Evaluation and Performance Metrics

To assess the model's performance, several evaluation metrics were considered. In this study, we evaluated the model using a test generator to produce accuracy and loss results of how it generated predictions for chosen test images. The notebook produced a classification report on precision, recall, F1-score, and support for each seven classes, their average, macro average, and weighted average values. Additionally, a confusion matrix was generated to show the model's accuracy on making true predictions for each class. Moreover, the method of plotting was utilized to visualize training and validation accuracy and loss per epoch. A bar graph was printed to show validation confidence scores for correct predictions. Lastly, a plot grid was used to showcase sample images, their true answers and the model's predictions, which included the confidence rate of each prediction.

IV. Results and Discussion

A. Key Findings

In this chapter, the study will focus on presenting the findings and their indications about the recognition model

for image classification in determining the face's display of emotion, whether it is angry, disgust, fear, happy, neutral, sad, and surprise. Among the seven various classes, Happy (0.86) demonstrated the highest precision, while detecting an image displaying the Angry emotion was the least. Due to the low amount of images in Disgust, the model struggled to effectively identify faces with such emotion, as shown by a low recall value (0.31).

TABLE I
DATASET OVERVIEW

Metrics	Values
Total Images	28,709
Angry	3,596
Disgust	393
Fear	3,688
Happy	6,494
Neutral	4,469
Sad	4,347
Surprise	2,854

TABLE II
SUMMARY OF MODEL PERFORMANCE METRICS

Features	Precision	Recall	F1-Score	Support
Angry	0.51	0.66	0.58	958
Disgust	0.81	0.31	0.44	111
Fear	0.58	0.31	0.40	1024
Happy	0.86	0.86	0.86	1774
Neutral	0.55	0.70	0.62	1233
Sad	0.52	0.51	0.51	1247
Surprise	0.79	0.75	0.77	831
Accuracy			0.65	7178
Macro Avg	0.66	0.59	0.60	7178
Weighted Avg	0.65	0.65	0.64	7178

B. Figures and Tables

This section lists the key figures and tables that illustrate the results:

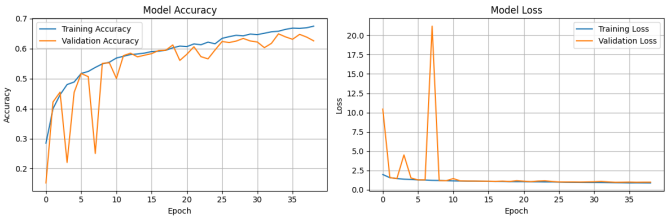

Fig. 2. Model Training and Validation Accuracy and Loss

Fig. 3. Model Validation Confidence Score

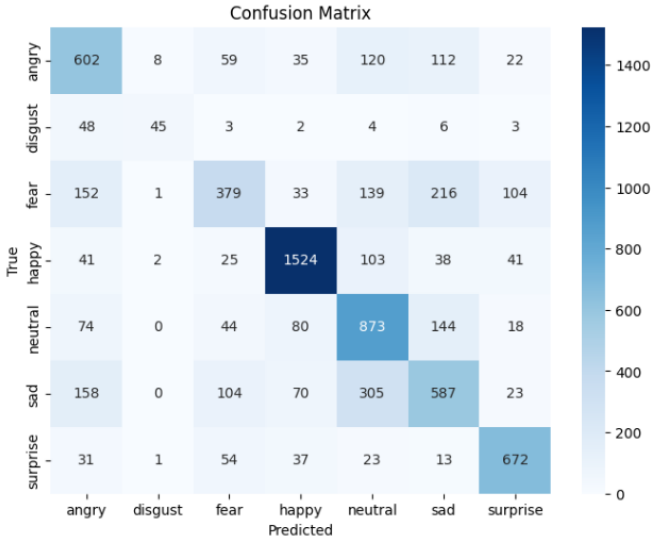

Fig. 4. Confusion Matrix

C. Model Evaluation

In Figure 2, the image shows two plots representing the training progress of the CNN model. In the model accuracy plot, the training accuracy line suggests that the model is learning and improving on the training data, while validation accuracy plateaus around 0.5-0.6 which indicates that the performance is weaker. Likewise, in the model loss plot, the training loss decreases over time, which means that the model is minimizing errors on training data, while the validation loss

rises slightly after epoch 10 which means the model suffers from overfitting.

Figure 3 shows a plotted grid of nine randomly selected images from the dataset, which shows true and predicted labels accompanied by the confidence score of each.

Figure 4 shows the confusion matrix of the model's performance. The model correctly classified 602 instances of Angry and 672 of Surprise. Happy had the highest accuracy with 1,524 correct predictions. In contrast, Disgust showed the lowest accuracy, with only 45 correct classifications, likely due to limited data. Neutral and Sad were often confused, as shown by their misclassifications. Overall, the model performs well on expressive emotions but struggles with subtle or underrepresented ones.

V. Conclusion

This study successfully developed and evaluated a Convolutional Neural Network (CNN) model for facial emotion recognition using a publicly available dataset containing 48x48 grayscale images categorized into seven emotions. Through careful data preprocessing, augmentation, and the application of appropriate evaluation metrics, the model demonstrated promising results, particularly in classifying more expressive emotions like Happy and Surprise with high precision. However, the model struggled with underrepresented or subtle emotions such as Disgust, largely due to data imbalance.

The integration of the trained model into a mobile application using TensorFlow Lite further highlights the real-world applicability of the system, enabling users to receive emotion predictions from facial images in real time. Despite achieving satisfactory performance, the model showed signs of overfitting as indicated by the divergence between training and validation accuracy and loss.

Future work should focus on enhancing model generalization through improved data augmentation, balancing class distributions, and experimenting with advanced architectures such as attention mechanisms or transfer learning. Additionally, incorporating explainability tools like Grad-CAM can improve the transparency of the model's predictions, particularly in critical applications like healthcare and security. Overall, this research demonstrates the potential of CNN-based systems for effective emotion detection and opens pathways for further enhancements in mobile-based emotion recognition technologies.

References

[1] Kaggle, "FER-2013," *Kaggle*, Available: https://www.kaggle.com/datasets/msambare/fer2013/code. Accessed: [May 20, 2025].

[2] Britannica. (n.d.) *Emotion*. https://www.britannica.com/science/emotion

[3] Constantine, W. L., Huynh, D., & DiMicco, J. (2016). *Framework for emotion recognition in human-computer interaction in natural settings.* Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/2858036.2858462

[4] Jain, A. (2019). *Automated emotion identification through facial expression: A review.* International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(1), 143–147.

[5] Kanna, R., Karthikeyan, V., & Suresh, A. (2022). *A computational model for facial emotion detection using the EMOTIC dataset.* International Journal of Advanced Computer Science and Applications, 13(3), 112–118. https://doi.org/10.14569/IJACSA.2022.0130315

[6] R. Pereira, C. Mendes, J. Ribeiro, R. Ribeiro, R. Miragaia, N. Rodrigues, N. Castro, and A. Pereira. (2024). *Emotion Detection Using Computer Vision and Deep Learning: A Systematic Review.* Sensors, vol. 24, no. 11, p. 3484, May 2024. [Online]. Available: https://doi.org/10.3390/s24113484

[7] Agung, E.S., Rifai, A.P. & Wijayanto, T. (2024) *Image-based facial emotion recognition using convolutional neural network on emognition dataset.* Sci Rep 14, 14429. [Online]. Available: https://doi.org/10.1038/s41598-024-65276-x

[8] J. W. Kusno and A. Chowanda. (2024). *Modeling Emotion Recognition System from Facial Images Using Convolutional Neural Networks.* CommIT (Communication and Information Technology) Journal, vol. 18, no. 2, pp. 251–259, Oct. 2024.

[9] G. J. P. Hakim. (2024). *Real-Time Facial Emotion Detection Application with Image Processing Based on Convolutional Neural Network (CNN).* IJEEMCS, vol. 1, no. 4, pp. 27–36, Nov. 2024.

[10] Shahzad, H. M., Bhatti, S. M., Jaffar, A., Akram, S., Alhajlah, M., & Mahmood, A. (2023). *Hybrid Facial Emotion Recognition Using CNN-Based Features.* Applied Sciences, 13(9), 5572. https://doi.org/10.3390/app13095572

[11] R. R. Selvaraju et al. (2020) *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization.* International Journal of Computer Vision, vol. 128, pp. 336–359, 2020. https://doi.org/10.1007/s11263-019-01228-7