National College of Ireland

Emotion Recognition with Deep Learning-Based Facial Expressions: Comparative Analysis of Algorithms

MSc Research Project
MSCAI1

Caner OZHAN
Student ID: x23199253

School of Computing
National College of Ireland

Supervisor: Devanshu Anand

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Caner OZHAN
Student ID:	x23199253
Programme:	MSCAI1 **Year:** 2024
Module:	Practicum
Supervisor:	Devanshu Anand
Submission Due Date:	15/11/2024
Project Title:	Emotion Recognition with Deep Learning-Based Facial Expressions: Comparative Analysis of Algorithms
Word Count:	3538
Page Count	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Caner OZHAN
Date:	15/11/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	✓

Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Emotion Recognition with Deep Learning-Based Facial Expressions: Comparative Analysis of Algorithms

Caner OZHAN
x23199253
MSCAI
School of Computing
National College of Ireland

***Abstract* — This study investigates facial expression recognition (FER) as a foundational element of emotion recognition systems, which enhance human-machine interactions by enabling devices to interpret and respond to human emotions. Using the FER2013 dataset and Google's MediaPipe Face Mesh, various machine learning models, including Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), k-Nearest Neighbors (KNN), Random Forests and Logistic Regression, were applied to assess their effectiveness in classifying facial expressions. Results indicate that CNN and Random Forest models achieved the highest accuracy, around 54% and 53%, respectively, with each model demonstrating unique strengths in emotion recognition tasks. This research highlights the role of preprocessing and class balance adjustments in improving model performance and offers insights into enhancing FER systems for real-world applications.**

Keywords — FER(Facial Expression Recognition), Face Mesh, Convolutional Neural Networks (CNNs), Mediapipe, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Logistic Regression

1. Introduction

Emotion recognition systems, especially those focusing on facial expression recognition (FER), play an increasingly vital role in enhancing human-machine interactions by allowing technology to interpret human emotions. Through emotion recognition, machines gain insights into users' psychological and emotional states, fostering empathetic responses that extend beyond functional interactions. This capability is particularly valuable in healthcare, where understanding patient distress can improve monitoring, and in education, where adaptive technologies can respond to student engagement levels. Traditional FER methods, such as the Facial Action Coding System (FACS) and machine learning algorithms like SVM and KNN, often face challenges in real-world scenarios, especially regarding accuracy across diverse populations. However, advancements in deep learning, particularly CNNs, have significantly improved FER performance by automatically extracting features and handling variability in lighting, facial angles, and expressions. This study investigates various machine learning techniques for FER, utilizing the FER2013 dataset and MediaPipe Face Mesh, aiming to identify the most effective model configurations for robust emotion recognition.

2. Literature Review

2.1. An Overview of Emotion Recognition and Facial Expressions

Facial expression recognition (FER) is pivotal in emotion recognition systems, significantly enhancing human-machine interaction by enabling devices to interpret and respond to human emotions effectively. This capability fosters more intuitive and personalized user experiences across various domains, including healthcare, education, and customer service.

Understanding Human Behaviors

Emotion recognition plays an essential role in interpreting human behaviors and enhancing social interactions, especially within the context of human-machine interaction. By recognizing emotions, machines gain the ability to interpret the psychological and emotional states of users, allowing them to engage in more meaningful interactions that go beyond traditional, purely functional responses. In environments such as healthcare, emotional

intelligence in machines can aid in patient monitoring, where understanding patient distress or comfort can be critical. Similarly, in educational settings, adaptive learning technologies using emotion recognition can tailor responses based on student engagement or frustration levels, creating a more supportive learning environment [1].

Emotion recognition contributes to a deeper understanding of human behaviors by analyzing non-verbal cues, such as facial expressions, body language, and vocal tones, which are all integral components of social interactions. These cues provide machines with insights into the subtleties of human communication, enabling them to better comprehend complex social signals that go beyond verbal communication. For instance, recognizing when a person feels confused or disengaged allows the system to adjust its response dynamically, ensuring a more empathetic interaction. This is particularly beneficial in customer service, where emotional understanding can lead to more personalized and satisfactory responses, fostering positive experiences and potentially increasing user retention [2].

In addition to enhancing individual interactions, emotion recognition systems can support group dynamics by tracking and analyzing emotions in collaborative or social settings. For example, emotion-aware systems can monitor group sentiments in collaborative workspaces, identifying moments of tension or cohesion, which helps facilitate smoother teamwork. This capability is valuable in remote work scenarios, where physical cues are limited, yet understanding team morale and engagement levels is crucial for effective management [3].

Moreover, emotion recognition has implications for the development of socially intelligent systems, which can aid individuals with social or communication difficulties, such as those on the autism spectrum. By accurately identifying and interpreting emotions in real-time, such systems can guide users in recognizing others' emotional states, thus enhancing their social engagement skills. Emotion recognition systems also offer significant potential for mental health applications, where recognizing signs of stress, anxiety, or depression can help flag users who may benefit from mental health resources or intervention [4].

Overall, emotion recognition technologies enhance not only human-machine interaction but also contribute to societal well-being by improving communication, empathy, and emotional awareness in various real-world applications. These systems make it possible for machines to participate meaningfully in social contexts, ultimately fostering better understanding and collaboration between humans and technology[1]

Traditional Approaches in Facial Expression Recognition

Traditional methods in facial expression recognition have focused on feature extraction and classification. Techniques such as the Facial Action Coding System (FACS) have been employed to decode individual muscle movements to categorize expressions. Machine learning algorithms like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) have also been utilized for classification tasks based on extracted features. However, these approaches often faced challenges in achieving high accuracy across diverse individuals and real-world scenarios.[5]

Advancements in deep learning have significantly improved the efficacy of facial expression recognition systems. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have enhanced the ability to process and classify facial expressions accurately across diverse demographic groups and in dynamic, real-time environments. These models automatically learn features from vast datasets, allowing them to generalize across different lighting conditions, angles, and facial structures, thereby reducing the limitations of traditional approaches.[6]

2.2. Applications of FER2013

One of the most widely used datasets in FER is FER2013. Introduced during the ICML 2013 Challenges in Representation Learning, FER2013 comprises approximately 30,000 grayscale images of faces, each sized at 48×48 pixels, annotated with seven emotion labels: anger, disgust, fear, happiness, sadness, surprise, and neutrality. This dataset has been instrumental in training and evaluating deep learning models for emotion classification tasks. FER2013 serves as a benchmark for developing and testing various FER models. Researchers have utilized it to train convolutional neural networks (CNNs) and other deep learning architectures, achieving significant advancements in real-time emotion detection systems.[7] Its extensive use has facilitated improvements in human-computer interaction, sentiment analysis, and behavioral studies.

2.3. Face Mesh with Mediapipe

Face mesh technology involves the computational modeling of human facial structures in three dimensions, enabling applications across computer vision, augmented reality (AR), and human-computer interaction. A prominent example is MediaPipe Face Mesh, developed by Google, which estimates 468 3D facial landmarks in real-time using a single camera input without the need for depth sensors. This system employs machine learning to infer 3D facial surfaces, facilitating real-time performance crucial for live applications.

The underlying architecture of MediaPipe Face Mesh comprises two deep neural network models: a face detector that identifies facial regions within an image and a face landmark model that predicts 3D facial landmarks. This dual-model approach enhances accuracy and efficiency in facial feature detection.[8]

Strengths of MediaPipe Face Mesh

- **Real-Time Performance**: MediaPipe Face Mesh delivers real-time facial landmark detection, making it suitable for applications requiring immediate feedback, such as virtual reality and augmented reality environments.
- **Resource Efficiency:** The solution employs lightweight model architectures and GPU acceleration, enabling efficient operation on devices with limited computational resources. **Comprehensive Landmark Detection:** By providing a dense set of 3D landmarks, it enhances the accuracy of facial feature analysis, which is crucial for applications like emotion recognition and facial expression analysis.

Limitations of MediaPipe Face Mesh

- **Dependence on Image Quality:** While effective with low-resolution images, the accuracy of landmark detection can be compromised under poor lighting conditions or significant occlusions.
- **Limited Expression Recognition:** Although it captures detailed facial landmarks, MediaPipe Face Mesh does not inherently classify facial expressions or emotions. Integrating additional models is necessary for comprehensive emotion recognition.

3. Research Methodology

3.1. Dataset

FER2013 (Facial Expression Recognition 2013) dataset is a dataset used for recognizing facial expressions and was created as part of a competition held on Kaggle in 2013. This dataset is widely used in applications such as facial expression classification and emotion analysis.[9]

- **Images:** Consists of grayscale facial images with a resolution of 48x48 pixels.
- **Labels:** Includes seven different emotional expression classes: *happy, sad, surprise, angry, fear, disgust,* and *neutral*.
- **Dataset Splits:** Divided into three subsets: training, validation, and test sets. These subsets are used for training models, evaluating performance, and measuring the overall effectiveness of the model.
- **Total Number of Images:** Contains a total of 35,887 facial images. [Figure 1.]
  - Training Set: Approximately 28,709 images
  - Validation Set: 3,589 images
  - Test Set: 3,589 images

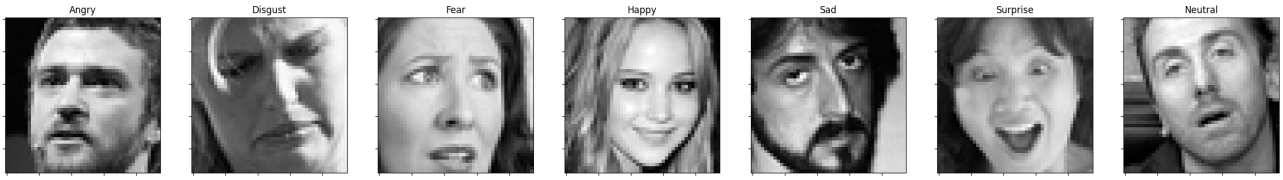


Figure: 1

3.2. Creating a Face Mesh with MediaPipe

Mediapipe, developed by Google, works by detecting faces in images and identifying specific landmarks on the face, such as eyes, nose, and mouth. Using deep learning models and efficient algorithms, it constructs a 3D facial mesh based on 468 key landmarks, allowing for detailed facial analysis in real time.[10]

Advantages Over Pixel-Based Methods

- Real-Time Efficiency: Mediapipe is optimized for fast processing, ideal for applications that need quick responses like augmented reality.
- 3D Landmark Precision: By capturing 3D coordinates, Mediapipe provides more accurate facial movement and expression tracking than pixel-based methods.
- Lower Computation Needs: Compared to pixel-based approaches, Mediapipe is highly efficient, achieving accurate results with fewer resources.

3.3. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of deep learning model specifically designed to process and interpret visual data, such as images. Originally inspired by biological processes in the human visual cortex, CNNs have become fundamental in tasks that involve recognizing spatial hierarchies in data, making them highly effective for image classification, object detection, and semantic segmentation.[11]

A CNN primarily comprises several layers: convolutional layers, pooling layers, and fully connected (dense) layers. Together, these components allow CNNs to capture and learn features across various levels of complexity.

- **Convolutional Layers:** These layers apply a set of learnable filters (kernels) to the input image, performing convolution operations that extract various features such as edges, textures, and patterns.
- **Activation Functions:** Following each convolutional layer, activation functions like ReLU (Rectified Linear Unit) introduce non-linearity into the model, enabling it to learn complex patterns.

- **Pooling Layers:** Pooling operations, such as max pooling, reduce the spatial dimensions of the feature maps, thereby decreasing computational load and controlling overfitting.
- **Fully Connected Layers:** These layers integrate the features learned by the convolutional and pooling layers to perform the final classification or regression tasks.

3.4. k-Nearest Neighbors (KNN)

KNN is a simple, instance-based algorithm for classification and regression. It classifies data points based on the labels of the *k*-closest neighbors in the feature space. The algorithm measures distance (usually Euclidean) to find the closest points.[12]

- **Euclidean Distance:** The Euclidean distance between two points x and y is:

$$f(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

- **Classification Rule:** A data point is assigned to the most common class among its *k*-nearest neighbors.

3.5. Support Vector Machine (SVM)

SVM is a supervised algorithm for classification. It works by finding a hyperplane that best separates data points into different classes. SVM chooses this hyperplane to maximize the margin, or distance, between classes. Mathematically, this separating hyperplane can be represented as:

$$f(x) = wx + b = 0$$

where w is the weight vector, x is the input vector, and b is the bias term.

3.6. Logistic Regression

Logistic Regression (LR) is a widely used statistical method for binary classification problems, though it can be extended to multiclass classification using techniques like one-vs-rest. Unlike linear regression, which is suited for continuous target variables, LR predicts the probability that a given input belongs to a particular class. It achieves this by modeling the log-odds of the outcome as a linear combination of the input features.

At the core of logistic regression lies the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

3.7. Random Forest

Random Forest is a powerful machine learning algorithm widely used for classification and regression tasks. It works by creating multiple decision trees from random subsets of the training data and combining their predictions to improve accuracy and robustness. Each tree independently makes a prediction, and the final output is based on the majority vote in classification or the average prediction in regression.

4. Implementation

4.1. Report on Face Mesh Extraction with MediaPipe

This report evaluates various preprocessing techniques applied to a dataset of 48x48 images to improve MediaPipe's face mesh extraction performance. The preprocessing techniques involve different combinations of resizing, adding borders, or keeping the original image size. The effectiveness of each approach was assessed by the failure rate across training, validation, and testing datasets, with a breakdown by emotion labels.

MediaPipe FaceMesh Configuration

The following parameters were configured for the mp_face_mesh.FaceMesh model in MediaPipe to optimize face mesh detection:

- **static_image_mode**: True - This mode processes each frame as a static image, making it suitable for individual image processing rather than video.
- **max_num_faces**: 1 - The model is set to detect a maximum of one face per image.
- **refine_landmarks**: True - Enables refined landmarks, which improve the detail and accuracy of the detected face mesh.
- **min_detection_confidence**: 0.8 - The model will only process detections with a confidence level of 0.8 or higher.
- **min_tracking_confidence**: 0.8 - A high threshold for tracking confidence ensures that tracked faces meet a high reliability criterion.

Overview of Techniques

1. Adding a 48-pixel border, then resizing to 128x128
2. Adding a 48-pixel border, then resizing to 250x250
3. Only resizing to 128x128
4. Only adding a 48-pixel border
5. Original size (48x48)
6. Resizing to 192x192, then adding a 100-pixel border

Table 1: MediaPipe Face Mesh Extraction Failure Rates

Technique	Train Failures (%)	Val Failures (%)	Test Failures (%)
48 Border + Resize to 128x128	8141 (28.4%)	1047 (29.2%)	987 (27.5%)
48 Border + Resize to 250x250	7953 (27.7%)	1019 (28.4%)	959 (26.7%)
Resize to 128x128 Only	19667 (68.5%)	2451 (68.3%)	2398 (66.8%)
Add 48 Border Only	7894 (27.5%)	1014 (28.3%)	950 (26.5%)
Original Size (48x48)	19782 (68.9%)	2465 (68.7%)	2439 (67.9%)
Resize to 192x192 + Add 100 Border	5553 (19.3%)	700 (19.5%)	668 (18.6%)

Observations

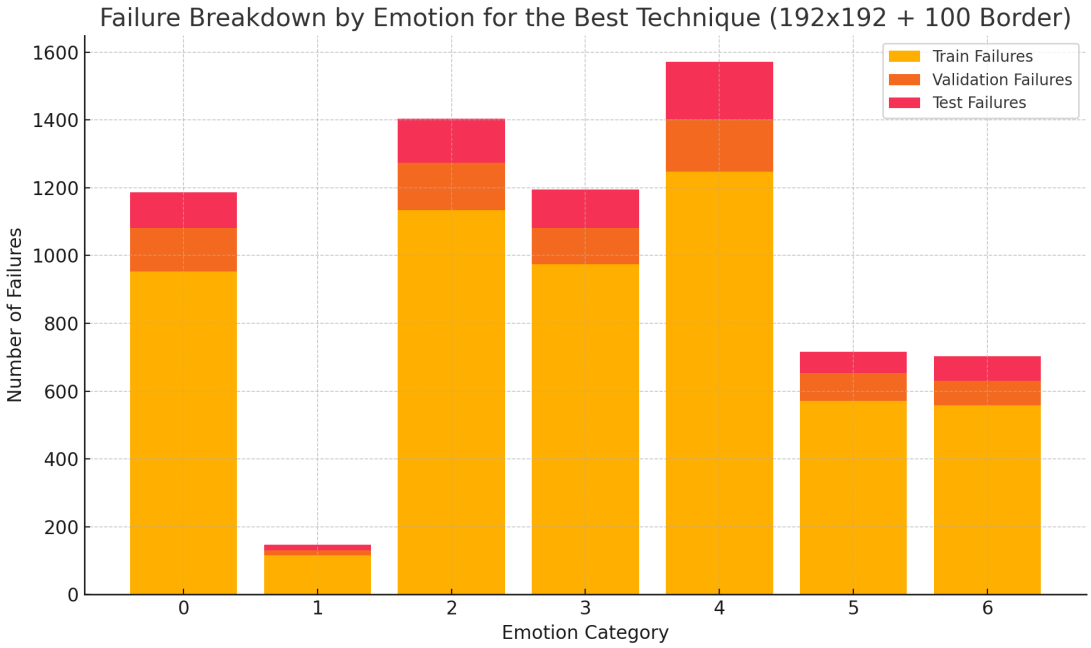
- High Failure Rate for Small or Resized Only Images: Techniques that used the original 48x48 size or simple resizing to 128x128 had the highest failure rates, consistently above 66%.
- Moderate Success with Added Borders and Moderate Resizing: Adding a 48-pixel border (with or without resizing to 128x128 or 250x250) yielded a moderate improvement, lowering the failure rate to around 27%.
- Best Performance with 192x192 Resizing and 100-pixel Border: The most effective technique was resizing to 192x192 and adding a 100-pixel border, with failure rates around 19%, indicating a substantial improvement in mesh extraction.

Emotional Breakdown of Failures

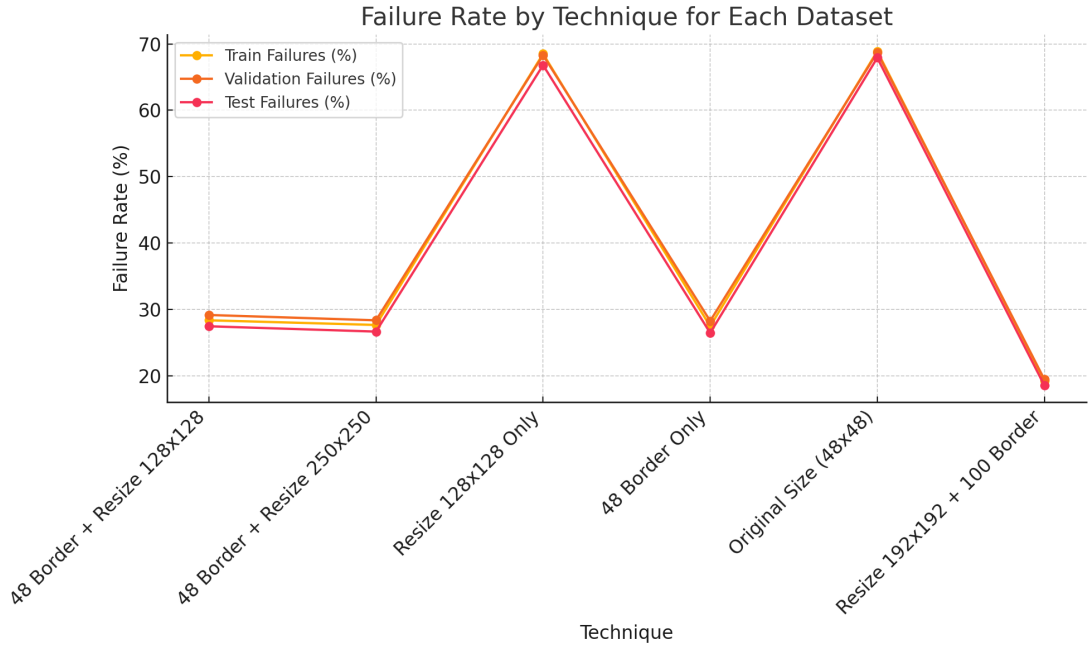
Failure rates vary across different emotion categories (0-6). Techniques using only resizing or the original size had consistently higher failures across all emotions, while the best technique

(192x192 resizing with a 100-pixel border) showed improvement across each emotion category.

I'll now generate graphs to visualize these findings, including:
1. Failure Rate by Technique for Each Dataset (Train, Validation, Test)
2. Failure Breakdown by Emotion for the Best Technique (192x192 with 100 Border)



Graph 1. Failure Rate by Technique for Each Dataset



Graph 2. Failure Breakdown by Emotion for the Best Technique
(0: Angry, 1: Disgust, 2: Fear, 3: Happy, 4: Sad, 5: Surprise, 6: Neutral.)

The graphs provide a clear visual representation of the findings:

1. Failure Rate by Technique for Each Dataset: The first plot shows that the 192x192 resizing with a 100-pixel border technique achieves the lowest failure rates across all datasets, with other techniques showing higher and less consistent performance.
2. Failure Breakdown by Emotion for the Best Technique: The second plot indicates how failure rates for the 192x192 with 100 border technique are distributed across different emotion categories. The majority of failures remain relatively balanced, though certain categories like "4" and "2" see slightly higher failures.

5. Evaluation

5.1. CNN Model Results Report

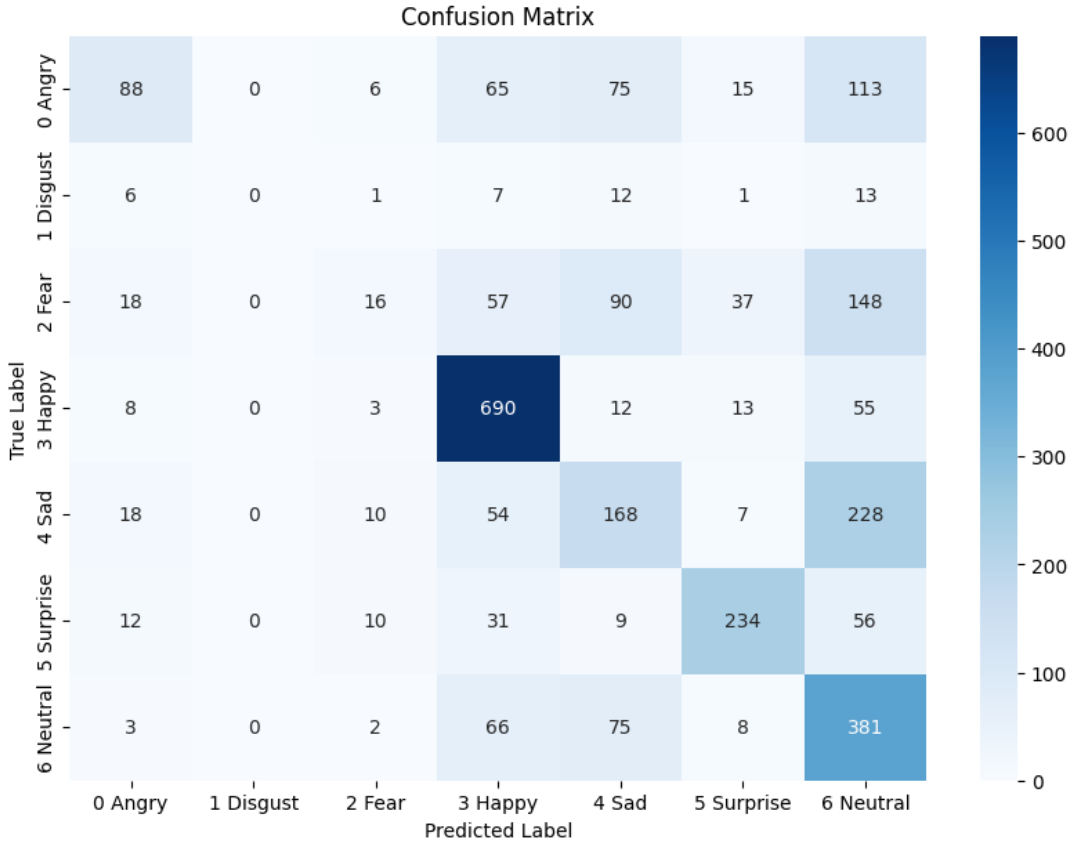
Table 2: CNN Performance Summary

Class	Precision	Recall	F1-Score	Support
Angry 0	0.58	0.24	0.34	362
Disgust 1	0.00	0.00	0.00	40
Fear 2	0.33	0.04	0.08	366
Happy 3	0.71	0.88	0.79	781
Sad 4	0.38	0.35	0.36	485
Surprise 5	0.74	0.66	0.70	352
Neutral 6	0.38	0.71	0.50	535

- **Overall Accuracy:** 0.54
- **Macro Average:**
  - Precision: 0.45
  - Recall: 0.41
  - F1-Score: 0.40
- **Weighted Average:**
  - Precision: 0.53
  - Recall: 0.54
  - F1-Score: 0.50

Based on these results, the model's accuracy is 54%. There is a noticeable disparity in performance between classes. For example, while Class 3 has a high success rate (Precision 0.71, Recall 0.88), Classes 1 and 2 perform significantly lower.

Table 3: CNN Confusion Matrix



- There is significant misclassification among classes. For example, 0 is frequently misclassified as 4 and 6.
- 1 shows poor classification performance, with most samples misclassified. This indicates the model struggles to distinguish this class.
- 3 demonstrates the highest classification success; however, confusion remains high among other classes.

5.2. Logistic Regression Model Results Report

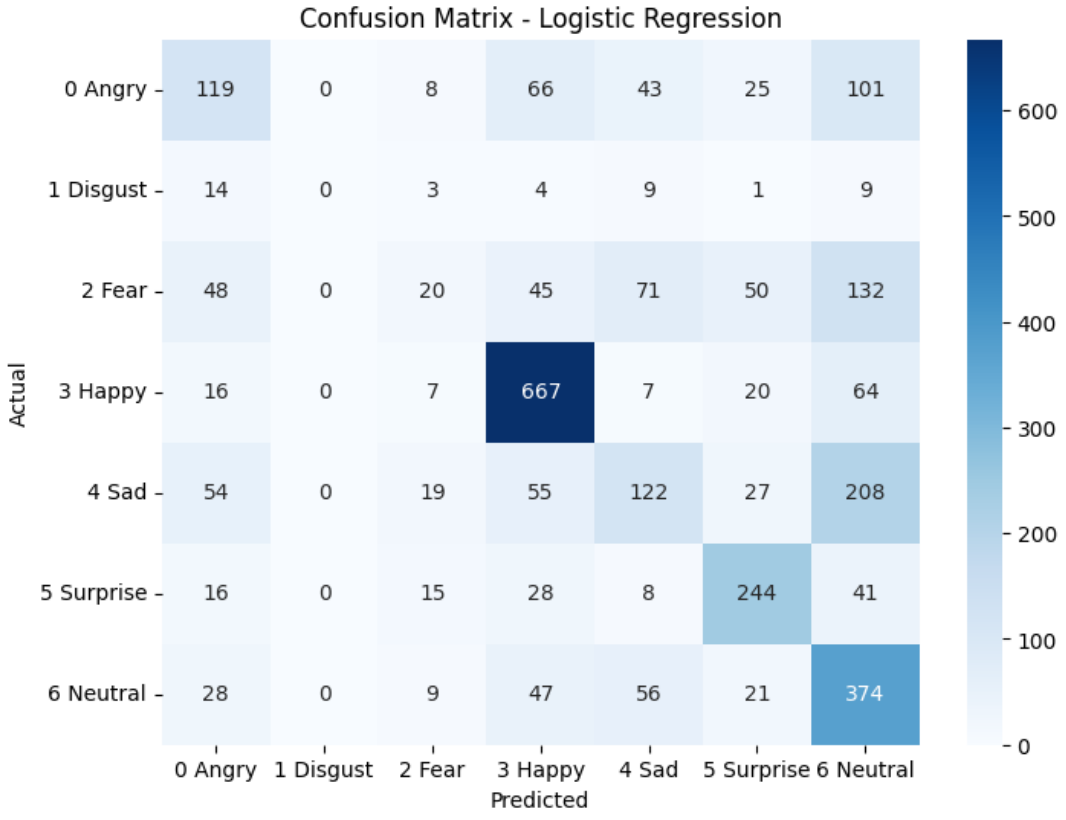
Table 4: Logistic Regression Performance Summary

Class	Precision	Recall	F1-Score	Support
Angry 0	0.40	0.33	0.36	362
Disgust 1	0.00	0.00	0.00	40
Fear 2	0.25	0.05	0.09	366
Happy 3	0.73	0.85	0.79	781
Sad 4	0.39	0.25	0.30	485
Surprise 5	0.63	0.69	0.66	352
Neutral 6	0.40	0.70	0.51	535

• **Overall Accuracy:** 0.53
• **Macro Average:**
  • Precision: 0.40
  • Recall: 0.41
  • F1-Score: 0.39
• **Weighted Average:**
  • Precision: 0.49
  • Recall: 0.53
  • F1-Score: 0.49

The logistic regression model achieved an accuracy of 53%, with varied performance across different classes. 3 stands out with the highest success rate, while other classes, particularly 1, show low performance due to poor precision and recall.

Table 5: Logistic Regression Confusion Matrix



The
confusion matrix indicates high misclassification rates among certain classes:
• Class 0 is frequently confused with Class 6.
• Class 1 is heavily misclassified, indicating poor performance in identifying samples from this class.
• Class 3 shows the best performance, with a high number of correctly classified instances.

5.3. K-Nearest Neighbors (KNN) Model Results Report

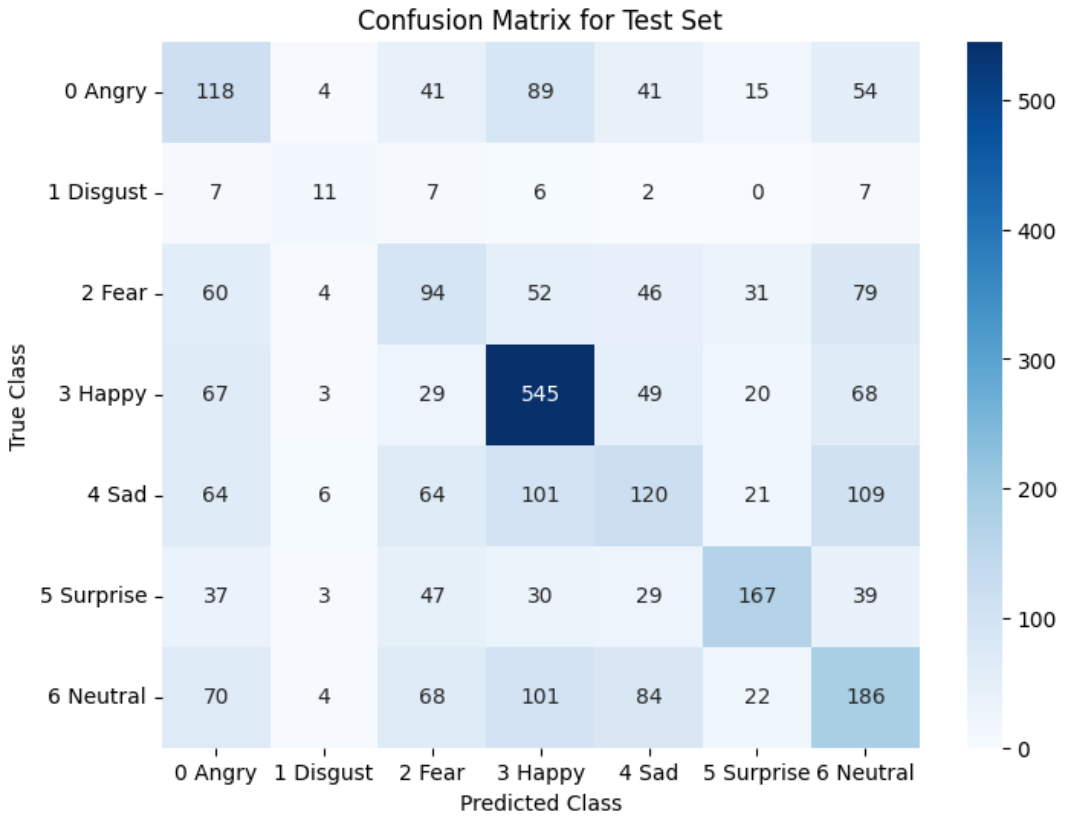
Table 6: K-Nearest Neighbors Performance Summary

Class	Precision	Recall	F1-Score	Support
Angry 0	0.28	0.33	0.30	362
Disgust 1	0.31	0.28	0.29	40
Fear 2	0.27	0.26	0.26	366
Happy 3	0.59	0.70	0.64	781
Sad 4	0.32	0.25	0.28	485
Surprise 5	0.61	0.47	0.53	352
Neutral 6	0.34	0.35	0.35	535

- **Overall Accuracy:** 0.42
- **Macro Average:**
  - Precision: 0.39
  - Recall: 0.38
  - F1-Score: 0.38
- **Weighted Average:**
  - Precision: 0.42
  - Recall: 0.42
  - F1-Score: 0.42

The KNN model yielded an accuracy of 42%, with the highest performance observed in 3, while other classes, such as 0, 4, and 6, show lower precision and recall values.

Table 7: K-Nearest Neighbors Confusion Matrix



The confusion matrix shows:
• 0 and 4 are frequently misclassified across other classes.
• 3 has the highest number of correctly classified instances, indicating better performance on this class.
• Misclassifications are high in classes with similar features or where the boundaries are less clear.

5.4.Support Vector Machine (SVM) Model Results Report

Table 8: Support Vector Machine Performance Summary

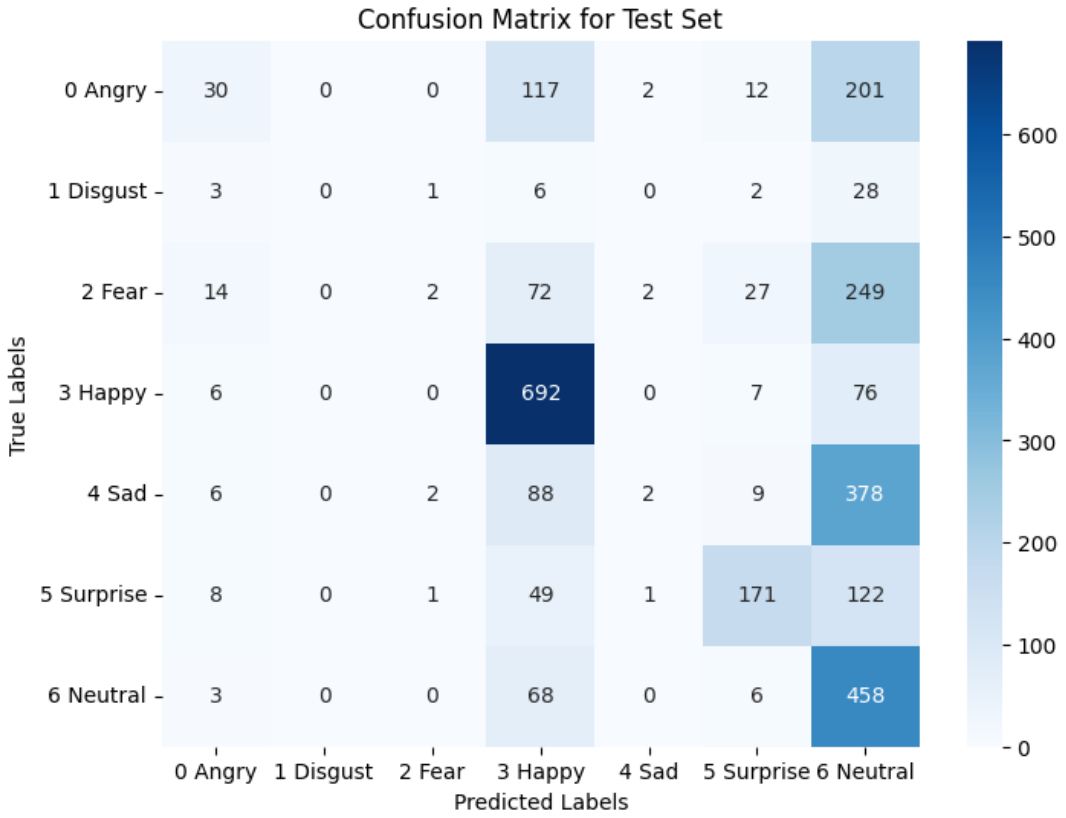
• **Overall Accuracy:** 0.46

Class	Precision	Recall	F1-Score	Support
Angry 0	0.43	0.08	0.14	362
Disgust 1	0.00	0.00	0.00	40
Fear 2	0.33	0.01	0.01	366
Happy 3	0.63	0.89	0.74	781
Sad 4	0.29	0.00	0.01	485
Surprise 5	0.73	0.49	0.58	352
Neutral 6	0.30	0.86	0.45	535

• **Macro Average:**
  • Precision: 0.39
  • Recall: 0.33
  • F1-Score: 0.28
• **Weighted Average:**
  • Precision: 0.46
  • Recall: 0.46
  • F1-Score: 0.37

The SVM model achieved an accuracy of 46%, with 3 demonstrating the highest precision, recall, and F1-score. Conversely, classes such as 0, 2, and 4 show low recall and F1-scores, indicating significant misclassifications.

Table 9: Support Vector Machine Confusion Matrix



The confusion matrix shows:

- 3 and 6 have higher numbers of correctly classified instances, while other classes exhibit significant misclassifications.
- 0, 2, and 4 are frequently misclassified as 6.

5.5.Random Forest Model Results Report

Table 10: Random Forest Performance Summary

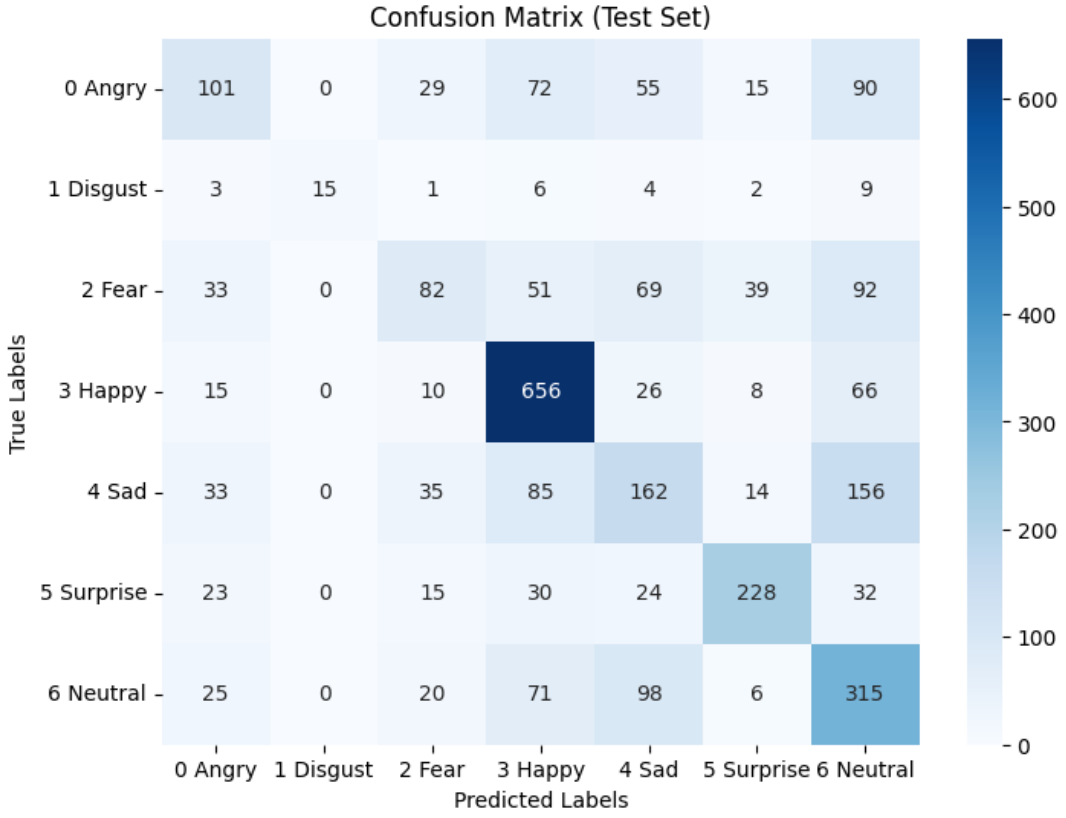
Class	Precision	Recall	F1-Score	Support
Angry 0	0.43	0.28	0.34	362
Disgust 1	1.00	0.38	0.55	40
Fear 2	0.43	0.22	0.29	366
Happy 3	0.68	0.84	0.75	781
Sad 4	0.37	0.33	0.35	485
Surprise 5	0.73	0.65	0.69	352
Neutral 6	0.41	0.59	0.49	535

- **Overall Accuracy:** 0.53

- **Macro Average:**
  - Precision: 0.58
  - Recall: 0.47
  - F1-Score: 0.49
- **Weighted Average:**
  - Precision: 0.53
  - Recall: 0.53
  - F1-Score: 0.52

The Random Forest model achieved an accuracy of 53%, with the best performance observed in Class 3 (precision, recall, and F1-score are relatively high), while Classes 1 and 2 show variable performance due to imbalanced precision and recall values.

Table 11: Random Forest Confusion Matrix



The confusion matrix shows:

- 3 has the highest number of correct classifications, indicating a strong performance in this class.
- Misclassifications are notable for 0, 4, and 6, which are often confused with each other and with neighboring classes.
- 1 demonstrates the highest precision but lower recall, reflecting that while correctly classified when predicted, it is rarely identified.

6. Conclusions

Table 12: A a comparative summary of the different models performance

Model	Accuracy	Key Strengths	Key Weaknesses
CNN	54%	High precision and recall for Class 3, good at capturing complex patterns	Inconsistent across classes, struggles with underrepresented classes (e.g., Class 1)
Logistic Regression	53%	Simple model, decent performance for high-support classes	Low recall for minority classes, limited in capturing complex relationships
KNN	42%	Better performance on Class 3, simple and interpretable	Sensitive to scaling and feature overlap; struggles with multiclass separation
SVM	46%	High recall on Class 3 and Class 6	Poor performance on minority classes; misclassifies frequently with other classes
Random Forest	53%	Good performance for Class 3, captures feature importance, robust to noise	Moderate performance overall; some confusion between similar classes (e.g., Class 0, 4, and 6)

Overall Insights

- Best Overall Performance: CNN and Random Forest achieved the highest accuracy (54% and 53%, respectively). CNN performed well in distinguishing complex patterns, particularly for Class 3, while Random Forest offered a robust and interpretable model with good handling of majority classes.
- Weakest Model: KNN, with an accuracy of 42%, struggled the most with class separation, especially among classes with similar feature distributions. This model may not be suitable for datasets with overlapping class boundaries.
- Class Imbalance Issues: Most models, including Logistic Regression, SVM, and KNN, displayed weaknesses in handling underrepresented classes, often resulting in low recall and precision for these categories.

Recommendations

- Model Selection: For datasets with complex, non-linear relationships and sufficient data, CNN would be recommended. If interpretability and robustness to noisy features are priorities, Random Forest could be preferred.
- Future Improvements: Consider addressing class imbalance through techniques like resampling or adjusting class weights, and apply feature engineering and hyperparameter tuning, particularly for Random Forest and SVM, to further enhance model performance.

This study evaluated the performance of various machine learning models for facial expression recognition (FER), with CNN and Random Forest models achieving the highest accuracy at 54% and 53%, respectively. CNNs demonstrated superior capability in

identifying complex patterns, making them especially effective for distinguishing nuanced expressions within diverse datasets. Random Forest models, on the other hand, offered robustness and interpretability, excelling in managing the majority classes and making them a suitable choice when model transparency and stability are prioritized.

Among the models tested, k-Nearest Neighbors (KNN) performed the weakest, with an accuracy of 42%, struggling particularly with class separation in cases where feature distributions overlapped. This limitation suggests that KNN may not be well-suited for datasets where class boundaries are ambiguous. Additionally, class imbalance posed challenges across most models, including Logistic Regression, SVM, and KNN, leading to reduced recall and precision in underrepresented classes. Addressing these imbalances is critical to enhancing model performance, especially for accurately detecting less common emotional expressions.

Recommendations for future work include selecting CNNs for datasets with complex, non-linear patterns and sufficient data. For applications where interpretability and resilience to noisy data are important, Random Forest may be preferred. Further improvements could involve addressing class imbalance through resampling or class weight adjustments and applying feature engineering and hyperparameter tuning, particularly for Random Forest and SVM models. These enhancements are essential to creating more reliable, accurate, and adaptable FER systems for real-world applications, ultimately advancing the field of emotion recognition and human-machine interaction.

Reference

[1] J. Li, J. Nie, D. Guo, R. Hong, and M. Wang, "Emotion separation and recognition from a facial expression by generating the poker face with vision transformers," IEEE Transactions on Computational Social Systems, 2024.

[2] N. Rawal and R. M. Stock-Homburg, "Facial emotion expressions in human–robot interaction: A survey," International Journal of Social Robotics, vol. 14, no. 7, pp. 1583-1604, 2022.

[3] S. Zhao, G. Jia, J. Yang, G. Ding, and K. Keutzer, "Emotion recognition from multiple modalities: Fundamentals and methodologies," IEEE Signal Processing Magazine, vol. 38, no. 6, pp. 59-73, 2021.

[4] Z. Song, "Facial expression emotion recognition model integrating philosophy and machine learning theory," Frontiers in Psychology, vol. 12, no. 759485, 2021.

[5] A. R. Khan, "Facial emotion recognition using conventional machine learning and deep learning methods: current achievements, analysis, and remaining challenges," Information, vol. 13, no. 6, 2022.

[6] T. Kopalidis, V. Solachidis, N. Vretos, and P. Daras, "Advances in facial expression recognition: A survey of methods, benchmarks, models, and datasets," Information, vol. 15, no. 3, pp. 135, 2024.

[7] Y. Khaireddin and Z. Chen, "Facial emotion recognition: State of the art performance on FER2013," arXiv preprint arXiv:2105.03588, 2021.

[8] MediaPipe Face Mesh, available at: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/face_mesh.md

[9] Kaggle, "Challenges in representation learning facial expression recognition challenge," available at: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/overview

[10] Google AI Edge, "Face Landmarker," available at: https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker

[11] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights into Imaging, vol. 9, pp. 611-629, 2018.

[12] V. S. Prasatha et al., "Effects of distance measure choice on k-NN classifier performance - a review," arXiv preprint arXiv:1708.04321, 2017.