Discrete Expression Recognition with Face Images (CK+ Dataset)

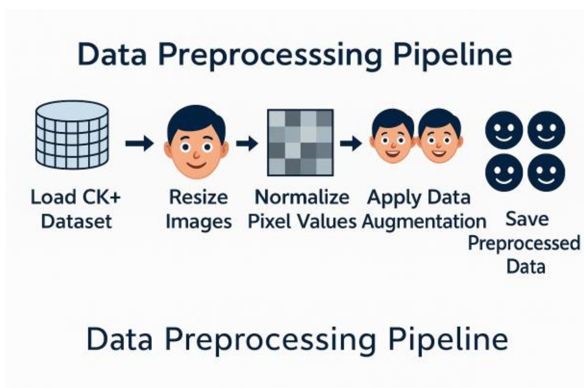
Methodology & Overall Approach:

We implemented a facial expression recognition system using transfer learning with the VGG16 convolutional neural network, pre-trained on ImageNet dataset.

Approach included:

- **Programming Language and Tools Used:**
  - Python 3.11
  - TensorFlow/Keras: Deep learning framework for model development.
  - OpenCV: Image loading, preprocessing, augmentation.
  - scikit-learn: Data splitting and evaluation.
  - Matplotlib and Seaborn: Visualization.
  - Google Colab: Training environment with GPU acceleration.
- **Dataset Overview:**
  - Extended Cohn-Kanade (CK+) Dataset.
  - 981 facial expression images extracted from multiple frames of emotional sequences.
  - Seven emotion categories: Angry, Contempt, Disgust, Fear, Happy, Sadness, Surprise.

- **Data Preprocessing:**



Each image was resized to 224×224 pixels to match the input requirements of the VGG16 architecture. We normalized pixel values by scaling them to a [0, 1] range for faster and more stable model convergence. To enhance generalization and address dataset imbalance, data augmentation techniques such as random rotation, horizontal flipping, and zooming were applied during training. These preprocessing steps helped in creating a robust and diverse training dataset while maintaining the integrity of the emotion labels.

- **Model Architecture:**

In this project, we utilized the VGG16 convolutional neural network, a deep learning architecture pre-trained on the ImageNet dataset. We retained the original convolutional layers of VGG16 to leverage its powerful feature extraction capabilities. We also **fine-tuned the last four convolutional blocks** to allow the model to adjust its deeper features to the facial emotion recognition task. On top of the convolutional base, we added a Global Average Pooling layer, followed by a Dense layer with 512 units and a Dropout layer for regularization. The final classification was performed using a Dense output layer with seven neurons corresponding to the seven emotion classes, activated by a Softmax function. The model was compiled using the **Adam optimizer** and **categorical cross-entropy loss**, which are well-suited for multi-class classification problems. This structure allowed efficient transfer learning while adapting the network specifically for facial emotion recognition.

- **Validation Approach (Training&Testing):**
  To ensure robust model evaluation, we adopted a two-stage validation strategy. First, we split the CK+ dataset using an 80/20 stratified split, allocating 80% for training and 20% for final testing. Within the training set, we applied 5-Fold Stratified Cross-Validation to further assess model generalization during training without touching the final test set. We incorporated EarlyStopping to halt training when validation loss stopped improving, and ReduceLROnPlateau to dynamically adjusted the learning rate for better convergence. This approach minimized overfitting, ensured balanced evaluation across emotion classes, and allowed optimal model fine-tuning.
- **Additional Information:**
  To improve model generalization and combat overfitting, we applied extensive data augmentation techniques such as random rotations, zooming, and horizontal flipping during training. We addressed the class imbalance present in the CK+ dataset by implementing class weighting during model compilation. The final trained model was saved in the mode1.keras format for easier reuse and deployment. For robust evaluation, we analyzed model performance using a confusion matrix, classification reports, and precision, recall, and F1-score metrics, providing a comprehensive view of how well the model performed across all emotion categories.

Initially Proposed Vs Final Approach:

In our initial proposal, we planned to apply basic data augmentation techniques such as random rotations and horizontal flipping to improve model generalization. In the final implementation, we expanded augmentation to include zooming transformations for even better diversity. We had anticipated addressing class imbalance if needed, and during implementation, we actively applied class weighting to handle uneven emotion distribution. While the proposal mentioned saving the trained model, in practice, we saved it using the model.keras format to ensure better compatibility for future use. For evaluation, we initially proposed using basic metrics like accuracy and classification reports, but we extended this by analyzing confusion matrices and detailed precision, recall, and F1-scores across all classes, providing a much deeper performance understanding.

Performance Metrics & Results:

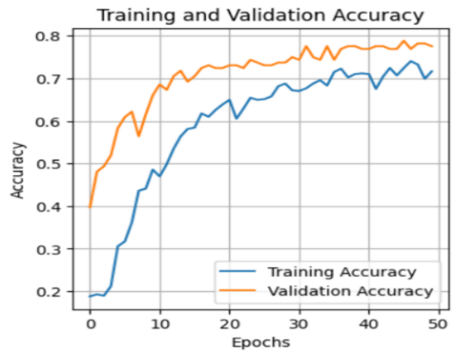
- **Test Accuracy**: ~84% on unseen test set.

Final Test Set Evaluation:

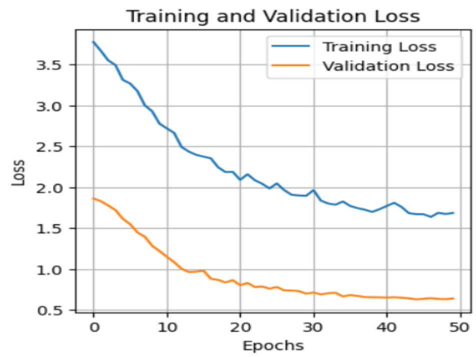
	precision	recall	f1-score	support
anger	0.86	0.44	0.59	27
contempt	0.83	0.45	0.59	11
disgust	0.94	0.86	0.90	35
fear	0.67	0.93	0.78	15
happy	0.93	0.95	0.94	42
sadness	0.50	0.94	0.65	17
surprise	1.00	0.98	0.99	50
accuracy			0.84	197
macro avg	0.82	0.79	0.78	197
weighted avg	0.88	0.84	0.84	197

- **Macro Average**:
  Precision: 0.82
  Recall: 0.79
  F1-Score: 0.78
- **Weighted Average**:
  Precision: 0.88
  Recall: 0.84
  F1-Score: 0.84

- **Best performing classes:** Surprise (F1-Score 0.99), Happy (F1-Score 0.94), Disgust F1-Score (0.9)



(a)Accuracy

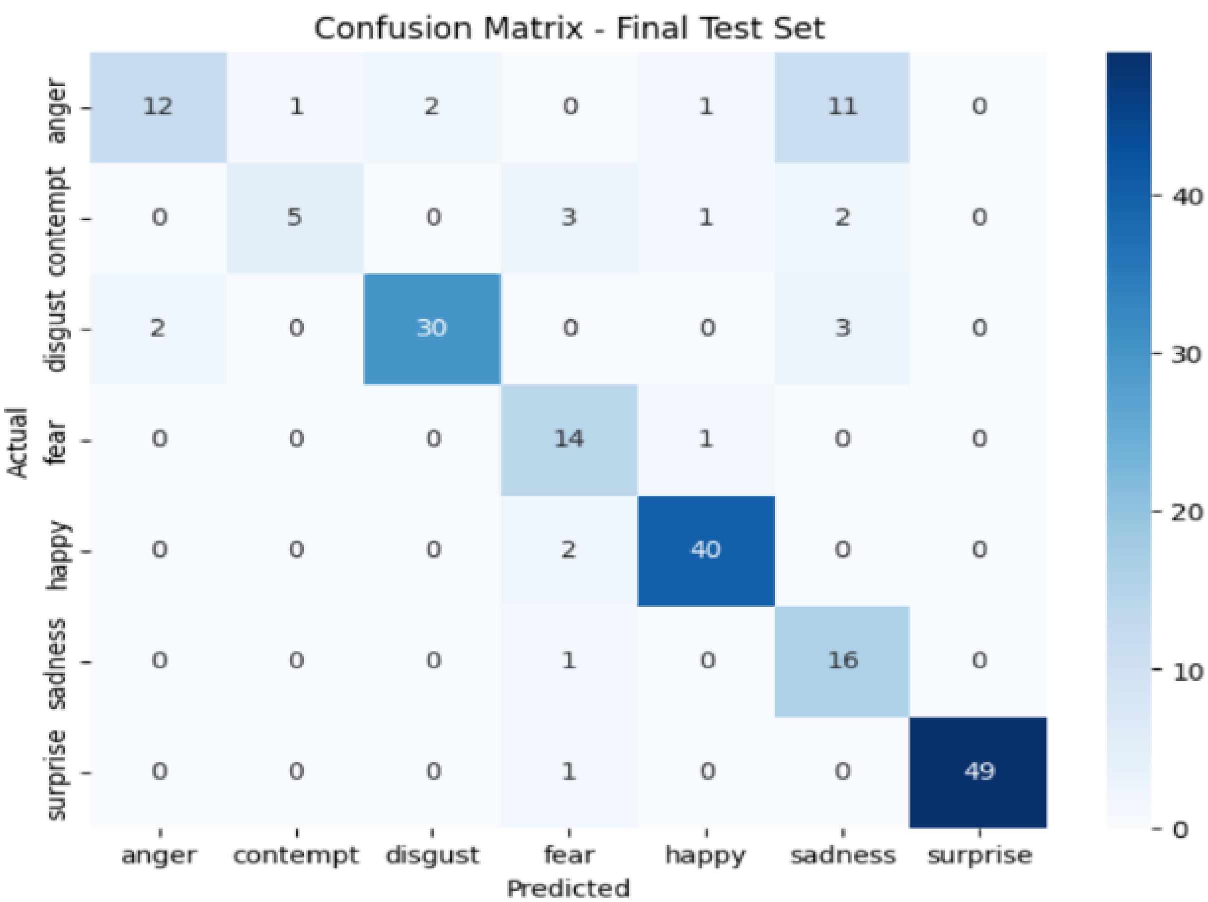


(b)Loss

Graph(a): The Accuracy graph shows a steady improvement in both training and validation accuracy. Validation accuracy reaches a peak of around 78–80%, slightly outperforming training accuracy in the later epochs. This suggests that the model generalizes well and is not severely overfitting.

Graph(b): The loss plot indicates a **consistent decrease** for both training and validation losses. Validation loss converges earlier than training loss and remains lower, indicating good generalization. The absence of divergence between the two curves suggests that the model does not suffer from overfitting or underfitting.

Confusion Matrix Insights: Minor misclassifications occurred between similar emotions such as Anger and Sadness, while major classes like Surprise showed strong separation.



Challenges Faced:

- **Long training times** due to the heavy VGG16 model and repeated 5-Fold Cross-Validation.
- **Dataset imbalance** affecting minority classes like Contempt.
- **Risk of overfitting**, managed by EarlyStopping, ReduceLROnPlateau, and data augmentation.
- **Memory limitations** encountered during K-Fold training on Google Colab, requiring session clearing and batch size adjustments.

Were They Similar to Proposal Expectations?

- Yes, long runtimes, class imbalance, and overfitting risks were expected and properly addressed.
- Memory issues were slightly more severe than anticipated but were effectively managed.

Conclusion:

In this project, we successfully developed a facial expression recognition system using transfer learning with the VGG16 convolutional neural network on the CK+ dataset. Through careful data preprocessing, model fine-tuning, and robust validation strategies such as 80/20 stratified splitting combined with 5-Fold Cross-Validation, we achieved strong generalization performance with a test accuracy of approximately 84%. Our enhancements over the initial proposal including stronger data augmentation, class weighting, and advanced callbacks like ReduceLROnPlateau significantly improved the model's ability to distinguish subtle emotional expressions. Despite challenges like long training times and memory constraints, all expected risks were effectively managed, and our results demonstrated consistent accuracy across multiple emotion classes. Overall, the project objectives were met successfully, and the implemented system shows promising potential for practical facial emotion recognition applications.

References:

[1] I. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests," in *International Conference on Neural Information Processing*, 2013, pp. 117-124.

[2] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 94-101.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815-823.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[7] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1195-1215, 2022.

[8] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1-43, 2011.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.
[10] CK+ Dataset: https://www.kaggle.com/datasets/shawon10/ckplus
[11] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 2003, pp. 53-53.
[12] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in Proc. 3rd IEEE Int. Conf. Autom. Face Gesture Recognit., 1998, pp. 200-205.
[13] G. Levi and T. Hassner, "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns," in Proc. ACM Int. Conf. Multimodal Interact., 2015, pp. 503-510.
[14] B. Martinez, M. F. Valstar, B. Jiang, and M. Pantic, "Automatic analysis of facial actions: A survey," IEEE Trans. Affect. Comput., vol. 10, no. 3, pp. 325-347, 2019.
[15] D. H. Kim, W. J. Baddar, J. Jang, and Y. M. Ro, "Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition," IEEE Trans. Affect. Comput., vol. 10, no. 2, pp. 223-236, 2019.