Facial Emotion Recognition using VGG19 and Real-Time Detection

**Arnab Sutar**

**Asritha Suraparaju**

**Dhanraj Pallepati**

**Esha Yepuri**

**Greeshma Poli**

**Sanjay Nenavath**

# 1. Introduction

Facial emotion recognition is a significant application in the field of computer vision, enabling machines to interpret human emotional states from facial expressions. In this project, we develop a deep learning model using VGG19 and deploy it for real-time emotion detection using a webcam feed. The goal is to accurately classify emotions such as Happy, Sad, Angry, and more, from grayscale images.

# 2. Technologies Used

This project integrates various Python libraries for model development, data processing, and evaluation.  
  
- Python: The core programming language used for implementation.  
- TensorFlow and Keras: Used to build, train, and manage the deep learning model.  
- OpenCV: Facilitates image processing and live video capture from the webcam.  
- NumPy and Matplotlib: Used for numerical operations and data visualization.  
- Scikit-learn: Provides tools for evaluating the model using metrics such as confusion matrices and classification reports.

# 3. Dataset & Preprocessing

The model is trained on a structured facial emotion dataset, that is downloaded from Kaggle, divided into training and testing sets. Preprocessing is crucial for improving model performance.  
  
- **Dataset Structure:** Images are placed in train and test folders, each containing subfolders for emotion classes.  
- **Resizing:** All images are resized to 48x48 pixels to match model input requirements.  
- **Color Conversion:** Images are converted to grayscale for simplicity and computational efficiency.  
- **Data Augmentation:** Applied to training data to artificially expand the dataset using transformations like rotation, zoom, shear, and horizontal flip. This helps the model generalize better to unseen data.

[**Link to the Datasets-**](https://drive.google.com/drive/folders/12jvg5RvMAP-QoqA5Zb1Ej0ZblJvpzJFX)

**Name of the Dataset- archive.zip**

# 4. Model Architecture

We use transfer learning with the pre-trained VGG19 model, which is effective for image classification tasks. The base model is modified to suit the emotion recognition problem.  
  
**VGG19 Base:** Loaded without its top layer (include\_top=False) and pre-trained on ImageNet.  
- **Input Handling:** Since the model expects 3-channel input, a 1x1 convolution layer is added to convert grayscale input (1 channel) to 3 channels.  
  
**Custom Top Layers:**  
- GlobalAveragePooling2D: Reduces the spatial dimensions while retaining important information.  
- Dense Layers: Two fully connected layers with 512 and 256 neurons using ReLU activation.  
- Batch Normalization and Dropout: Regularization techniques to prevent overfitting.  
- Output Layer: A softmax layer with 7 neurons, representing the 7 emotion classes.

# 5. Training Configuration

The model is compiled and trained using optimal hyperparameters to ensure effective learning.  
  
- **Batch Size:** Set to 64, meaning the model processes 64 images at a time.  
- **Epochs:** The training runs for 50 full passes over the dataset.  
- **Optimizer:** Adam optimizer is used with a learning rate of 0.0001, offering adaptive learning.  
- **Loss Function:** Categorical Crossentropy, suitable for multi-class classification problems.  
- **Transfer Learning:** To speed up training and improve performance, all layers of VGG19 are frozen except the last four, which are fine-tuned.

# 6. Evaluation Results

After training, the model is evaluated on the test dataset to measure its performance.  
  
- **Test Accuracy:** Approximately 58.37%, indicating that the model correctly classifies emotions in more than half of the test images.  
- **Test Loss:** The loss value is 1.21, which corresponds to the error in predictions.  
  
**Evaluation Tools:**  
- **Confusion Matrix:** Provides a summary of correct and incorrect predictions for each class.  
- **Classification Report:** Shows precision, recall, and F1-score for each emotion category.

# 7. Real-Time Emotion Detection

The trained model is integrated with a live webcam feed to detect and display emotions in real time.  
  
- Face Detection: Uses OpenCV's Haar cascade classifier to locate faces in each video frame.  
- Image Preprocessing: Detected faces are converted to grayscale, resized to 48x48 pixels, and normalized.  
- Emotion Prediction: The processed face is passed through the model to predict the emotion label.  
- Display: The predicted emotion is shown on the screen with a bounding box around the face.  
- Exit Condition: The real-time detection can be stopped by pressing the 'q' key.

# 8. Saving and Reloading the Model

To reuse the trained model without retraining, it is saved and reloaded when needed.  
  
- Saved As: my\_model\_vgg19.h5, which includes model architecture and weights.  
- Loading Method: The model is reloaded using load\_model() from Keras.

# 9. Conclusion

This project demonstrates how transfer learning and computer vision techniques can be used to recognize emotions from facial expressions in real time.  
  
- It combines VGG19's pre-trained capabilities with custom layers for emotion recognition.  
- The model achieved moderate accuracy and can be improved further.  
- Potential enhancements include using larger datasets, balancing class distribution, or adopting more advanced architectures like ResNet or EfficientNet.

[**Link to the Zoom Video Recording**](https://drive.google.com/drive/folders/12jvg5RvMAP-QoqA5Zb1Ej0ZblJvpzJFX)

**Screenshots of the behavior of the Model in Real -time-**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A person and person smiling

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.