Facial Expression Recognition and Affective Computing Project Report

1. Introduction

In this project, we aim to develop a system that can recognize facial expressions and predict emotional states of valence and arousal from images. The system uses convolutional neural networks (CNNs) to learn features from facial images. Both **classification** (for discrete expressions) and **regression** (for valence and arousal) are performed simultaneously in a multi-task learning framework.

We explore multiple architectures, including **pre-trained models** (VGG16, ResNet50, EfficientNetB0) and a **custom CNN**, to determine which provides the best performance. The system is also designed to handle multiple dataset formats, including `.npy`, `.csv`, and image directories.

2. Dataset and Preprocessing

2.1 Dataset Loading

The dataset may come in different formats such as compressed ZIP files, CSV annotation files, or NumPy `.npy` arrays. A **DatasetLoader** class handles loading:

- Extracts files if zipped.
- Loads images and corresponding annotations for expressions, valence, and arousal.
- Handles missing files by generating dummy images and synthetic labels for testing.

2.2 Image Preprocessing

- Images are converted to **RGB** and resized to **224x224 pixels** to match CNN input requirements.
- Pixel values are **normalized** to the range `[0, 1]`.
- Data augmentation is applied during training to improve model generalization, including **flips, rotations, zooming, and brightness adjustments**.

2.3 Data Splitting

The data is split into **training, validation, and test sets**, maintaining a balanced distribution of emotion classes using stratified sampling.

3. Model Architectures

The project implements multiple CNN architectures for multi-task learning:

3.1 Pre-trained Models

- **VGG16**: Classic CNN with deep layers, good for feature extraction.
- **ResNet50**: Uses residual connections to improve gradient flow.
- **EfficientNetB0**: Optimized for parameter efficiency while maintaining high accuracy.

3.2 Custom CNN

The custom architecture includes:

- **Multi-scale convolutions** (3x3, 5x5, 7x7) to capture features at different spatial resolutions.
- **Residual blocks** to maintain information across layers.
- **Attention blocks** to focus on important features.
- **Global pooling** and dense layers for final outputs.

3.3 Multi-task Output

Each model predicts:

- **Expression** (classification into discrete emotion categories)
- **Valence** (continuous value, regression)
- **Arousal** (continuous value, regression)

4. Training and Optimization

4.1 Model Compilation

- **Optimizer**: Adam with learning rate 0.001
- **Loss functions**:
  - Sparse categorical cross-entropy for expression classification.
  - Mean squared error (MSE) for valence and arousal regression.

- **Metrics**: Accuracy for classification and MAE for regression.

4.2 Training Process

- Trained using **mini-batches** of images.
- **Early stopping** prevents overfitting by halting training if validation loss stops improving.
- **Learning rate reduction** on plateau adjusts learning if progress stalls.
- Best models are saved for evaluation.

4.3 Data Augmentation

Augmentation improves generalization by simulating variations such as flipping, zooming, and rotation. This reduces the chance of overfitting to training images.

5. Evaluation Metrics

The system evaluates both classification and regression tasks:

5.1 Classification Metrics

- **Accuracy**: Correct predictions over total predictions.
- **F1-score (macro)**: Harmonic mean of precision and recall across all classes.
- **Cohen's Kappa**: Measures agreement between predicted and true labels.
- **ROC-AUC**: Evaluates probability predictions.

5.2 Regression Metrics

- **RMSE**: Measures average error magnitude.
- **Pearson correlation coefficient**: Measures linear correlation between predicted and true values.
- **Sign Agreement Metric (SAGR)**: Measures directional accuracy of predictions.
- **Concordance Correlation Coefficient (CCC)**: Evaluates agreement between predicted and true continuous values.

6. Model Evaluation and Results

After training, models are evaluated on the test set:

1. **Predictions** for expressions, valence, and arousal are obtained.

2. **Classification metrics** (accuracy, F1, Cohen's Kappa) are computed for discrete emotions.
3. **Regression metrics** (RMSE, CCC, correlation) are computed for valence and arousal.
4. **Visualization**:
    - Training history plots (loss and accuracy over epochs)
    - Confusion matrix for classification performance
    - Valence-arousal scatter plots comparing predicted vs true values
    - Sample images with predicted labels for visual verification

7. Multi-Model Comparison

All models are trained and evaluated under the same conditions:

- Pre-trained models leverage transfer learning for faster convergence.
- Custom CNN provides flexibility and multi-scale feature learning.
- Metrics and visualizations allow side-by-side comparison to determine which architecture performs best for both classification and regression tasks.

8. Key Findings

- **Multi-task learning** improves efficiency by predicting classification and regression outputs simultaneously.
- **Pre-trained models** generally converge faster and achieve higher accuracy due to learned features.
- **Custom CNN** allows experimentation with multi-scale and attention mechanisms, which can improve feature extraction for subtle emotions.
- **Data augmentation** significantly improves generalization and reduces overfitting.
- Evaluation metrics provide a comprehensive picture of model performance, balancing accuracy, correlation, and agreement measures.

9. Strengths of the System

- Flexible dataset handling: Works with images, CSV, NPY, or dummy data.
- Multi-task learning: Simultaneously predicts emotion class and continuous affective states.
- Supports both **transfer learning** and **custom CNN architectures**.
- Robust evaluation: Includes classification and regression metrics.

- Visualization: Training trends, confusion matrices, and sample predictions enhance interpretability.

10. Conclusion

This project successfully implements a **facial expression recognition and affective computing pipeline** using deep learning. It supports multi-task learning, flexible datasets, and multiple CNN architectures. Pre-trained models show fast convergence, while custom CNNs allow deeper experimentation. The system provides reliable predictions for both discrete emotions and continuous valence-arousal values, with robust metrics and visualizations to assess performance.