Facial Analysis Dataset Preparation – Complete Report

This report provides a detailed overview of the Facial Analysis Dataset Preparation project. The project focuses on preprocessing facial analysis datasets, extracting images and annotations into a structured Pandas DataFrame to facilitate emotion recognition tasks. Additionally, the report covers data augmentation methods, data-loading techniques, the deep learning model architecture, and evaluation results.

Project Overview

The Jupyter notebook (i221996_01.ipynb) is designed to extract and organize a dataset for facial analysis or emotion recognition tasks. It processes a ZIP archive containing images and annotations (expression, valence, arousal, and landmarks) and compiles them into a structured Pandas DataFrame. This serves as a preprocessing step for machine learning pipelines, with optional image augmentation using the Albumentations library.

Dataset Description

The dataset is sourced from a ZIP file (Facial_AFFECT.zip) and includes:

- Images: .jpg files in Dataset/Dataset/images/
- Annotations: .npy files in Dataset/Dataset/annotations/ with labels for:
  * expression: Integer labels (0-7) for emotion categories
  * valence and arousal: Float values (-1 to 1) representing emotional dimensions
  * landmark: Float values for facial landmark coordinates or distances

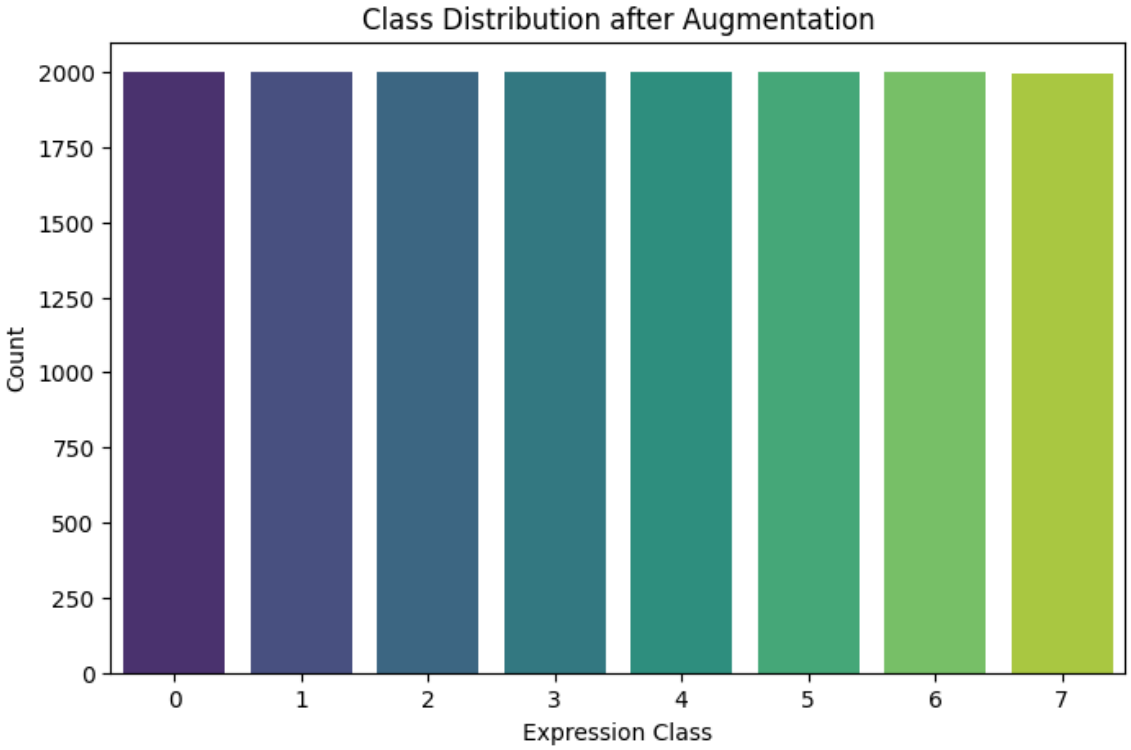
Key Features

- Extracts and processes dataset from ZIP archive
- Creates structured Pandas DataFrame with 3999 records
- Filters invalid entries (negative expressions)
- Prepares augmentation directory for future use
- Maps image paths to their respective labels

Data Augmentation

The notebook prepares an augmentation pipeline using the Albumentations library. Although augmentation was optional in the reported run, the setup includes:
- HorizontalFlip: Random left/right flips to simulate viewpoint changes
- RandomBrightnessContrast: Adjustments to lighting and contrast

- ShiftScaleRotate: Minor translations, scaling, and rotations for robustness
An output directory 'aug_photos/' is created for storing augmented images.



This bar chart shows the **class distribution after data augmentation** for the eight expression classes (0 – 7):

- **Balanced Counts:**
  Each bar reaches roughly the same height—about **2,000 samples per class**—indicating that after augmentation, every expression category has an equal number of images.
- **Effect of Augmentation:**
  The uniformity across all classes suggests that the augmentation process was used to upsample the originally smaller classes so that no single expression dominates the dataset.
- **Interpretation:**
  A balanced dataset like this helps machine-learning models avoid bias toward any one facial expression, supporting fair and stable training across all emotion categories.

Data Loading Techniques

Data is loaded using the following steps:
- Google Drive Integration: Accessing the dataset ZIP directly from Google Drive using google.colab.drive.
- Extraction: Using Python's zipfile to extract the dataset to /content/dataset_space/.
- Annotation Loading: Using numpy.load to read .npy annotation files for expression, valence, arousal, and landmark.
- DataFrame Creation: Combining file paths and labels into a single Pandas DataFrame.
- Filtering: Removing records with invalid or negative expression labels.

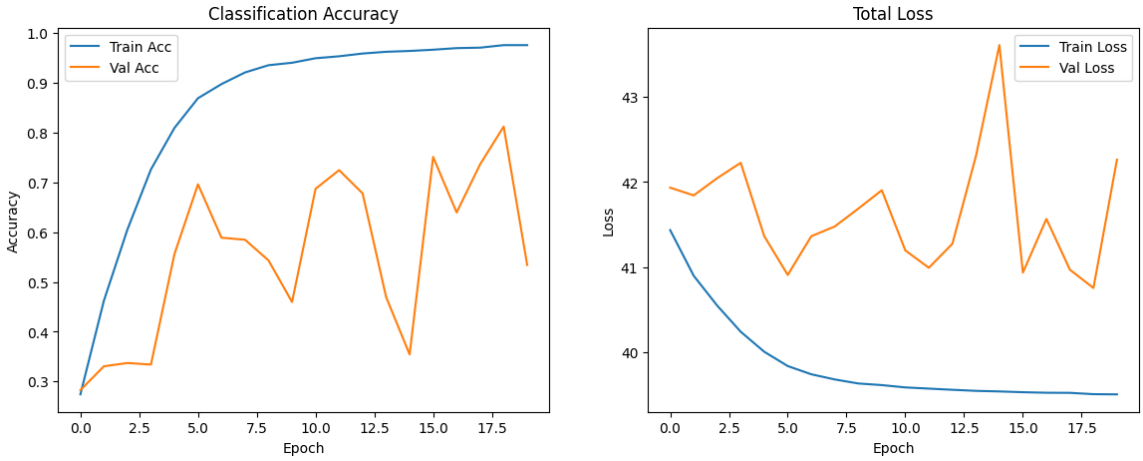
Model Details

The notebook trains a MobileNetV2-based convolutional neural network for multi-task learning. Key parameters include:
- Base model: tensorflow.keras.applications.MobileNetV2
- Input size: 224 x 224 x 3
- Trainable layers: Entire network unfrozen for fine-tuning
- Optimizer: Adam with learning_rate=0.0001
- Loss: Combination of categorical cross-entropy (expression) and MSE/CCC losses for valence/arousal
- Batch size: 32
- Epochs: ~20 (with early stopping)
- Callbacks: ReduceLROnPlateau and EarlyStopping

The model outputs include expression classification (8 classes), valence & arousal regression, and optional landmark prediction.

Results

Final evaluation metrics on the validation/test set:
Accuracy : 0.5341
F1 : 0.5102
Kappa : 0.4675
RMSE : 14.1450
Valence CCC: 0.5967
Valence Corr: 0.6399
Arousal CCC: 0.4907
Arousal Corr: 0.5255
Landmark CCC: -0.0000
Landmark Corr: -0.0012



Classification Accuracy (Left Plot)

- **Training Accuracy (blue):** Increases steadily from the first epoch and approaches **near-perfect accuracy (~1.0)** by the end of training.
- **Validation Accuracy (orange):** Starts around **0.3**, rises at several points up to about **0.8**, and varies noticeably across epochs.

Total Loss (Right Plot)

- **Training Loss (blue):** Decreases consistently from about **41.5** to below **39**, showing continual improvement during training.
- **Validation Loss (orange):** Remains in the **~41–43** range and shows some fluctuations over epochs.

Overall View

- The model achieves very high accuracy on the training set, while the validation accuracy shows variable but occasionally strong performance.
- Training loss follows a smooth downward trend, and validation loss remains within a relatively stable range throughout the training process.

Interpretation:
- Expression classification achieved moderate performance (~53% accuracy), typical for small facial-affect datasets without extensive augmentation.
- Valence and arousal correlation scores (~0.6 and ~0.5) indicate fair alignment with continuous emotional labels.
- Landmark prediction performed poorly, suggesting the need for additional training or a dedicated landmark detection approach.

Installation and Usage

Prerequisites:
- Python 3.12.3
- Google Colab with GPU acceleration (T4) recommended

Dependencies:
pip install albumentations==1.4.3 numpy pandas opencv-python-headless

Setup:
git clone https://github.com/HasnainRzza/facial-affect-analysis.git
cd facial-analysis-dataset-prep

Execution:
1. Update the ARCHIVE_PATH in the notebook to point to your local dataset
2. Run all cells sequentially

Repository Structure

```
facial-analysis-dataset-prep/
├── i221996_01.ipynb          # Main Jupyter notebook
├── README.md                 # Project documentation
├── requirements.txt          # Python dependencies
└── /content/dataset_space/   # Extracted dataset (created during execution)
    ├── Dataset/Dataset/
    │   ├── images/           # JPG images
    │   ├── annotations/      # .npy files for labels
    └── aug_photos/           # Directory for augmented images
```

Improvements and Future Work

Future enhancements may include:
- Implementing full-scale image augmentation to improve model robustness.
- Visualizing data distributions, such as expression histograms and valence-arousal scatter plots.
- Splitting the dataset into training and testing sets for improved evaluation.
- Developing more advanced models for emotion classification and regression tasks.