ML Hackathon

Avinash S - PES2UG22CS115
Arpit Gupta – PES2UG22CS098
Anusha Navale - PES2UG22CS087
Anushka Singh - PES2UG22CS089

Report on Machine Learning Hackathon: Emotion Recognition from Video and Text

1. Introduction:

The goal of this machine learning task was to build a model capable of predicting emotions from both visual and textual data from video clips. The dataset consisted of videos and corresponding transcripts (utterances) where emotions such as **neutral**, **happy**, **angry**, etc., were labeled for each clip. This report outlines the approach used to extract features from the video and text, combine these features using **early fusion**, and train a classifier for emotion recognition.

2. Problem Understanding:

The problem involves predicting emotions based on multimodal input data:

1. **Video Data**: Includes visual cues such as facial expressions, body language, and movements.
2. **Text Data**: Contains spoken words (dialogue), which provide additional contextual information about the emotional state.

The challenge lies in combining these two different types of data (video and text) to create a robust classifier for emotion prediction.

3. Data Overview:

The dataset contains two primary components:

- **Train Data**: A CSV file (`train_emotion.csv`) which includes the dialogue information (Utterance, Speaker, Emotion, etc.), and the corresponding video clips stored in a separate directory.
- **Test Data**: A similar CSV file (`test_emotion.csv`) containing test samples without emotion labels, for which predictions need to be made.

The key columns in the training data are:

- `Emotion`: The emotion label (target variable).
- `Dialogue_ID` and `Utterance_ID`: Identifiers for the dialogue and utterance.
- `Utterance`: The text spoken in the video clip.
- `video_clip_path`: Path to the corresponding video file.

4. Approach:

The approach for this emotion recognition task can be broken down into the following steps:

1. **Feature Extraction**:

- **Video Features**: We used a **pre-trained ResNet50** model (without the top layer) for extracting visual features from the video clips. The frames of the video were sampled at a specified frame rate (e.g., 30 frames per second) and passed through ResNet50 to extract deep convolutional features. The average of these features was taken to represent the video clip.
- **Text Features**: We used a **pre-trained BERT model** to extract embeddings from the subtitles (text) in the video. The [CLS] token embedding was used as the representation for each text sample, which provides a semantic understanding of the utterance.

2. **Fusion of Features**:
   - **Early Fusion**: The extracted video features and text features were concatenated to form a unified feature vector for each sample. This fusion allows the model to learn from both the visual and textual information simultaneously.
3. **Data Preprocessing**:
   - The features were normalized using **StandardScaler** to ensure that they are on the same scale before being input into the classifier.
4. **Model Selection**:
   - We experimented with different classifiers, such as **Support Vector Machine (SVM)** with a linear kernel and **Logistic Regression**. The classifier was trained using the fused features from both video and text.
5. **Evaluation**:
   - The classifier was trained on the entire training dataset, and performance was evaluated using accuracy and classification reports.
6. **Test Predictions**:
   - The same process was followed for the test dataset, where features were extracted from both the video and text, fused, normalized, and then passed through the trained model to make predictions.
7. **Submission**:
   - The predictions were saved in the required CSV format with two columns: `Sr No.` and `Emotion`, for submission.

5. Detailed Steps:

Feature Extraction:

- **Video Features**:
  - Video frames were extracted using OpenCV from the video clips.
  - Each frame was resized to 224x224 pixels to match the input size expected by the ResNet50 model.
  - The ResNet50 model was used to extract features, with the output being pooled globally to form a 2048-dimensional vector for each frame.
  - The average of these vectors across all frames in a video clip was used as the feature representation for the video.
- **Text Features**:
  - The text of each utterance was tokenized using the BERT tokenizer.
  - The BERT model was used to compute embeddings for each utterance, with the final [CLS] token representing the entire sequence.

Fusion:

- **Early Fusion**: The video and text features were concatenated along the feature dimension, resulting in a 2800-dimensional feature vector (2048 video features + 768 text features).

Modeling:

- **SVM Classifier**:
    - An SVM classifier with a linear kernel was trained on the fused features to predict the emotional state of the speaker in each video.
    - Alternatively, logistic regression was also considered as a classifier.

Normalization:

- **StandardScaler** was used to normalize both the training and test features before feeding them into the classifier, ensuring that all features contribute equally to the model's decision.

Testing:

- The model was tested on the unseen test set and predictions were made.
- The results were saved to a CSV file for submission.

6. Results:

- The classifier achieved an acceptable level of performance with good predictions on the test set.
- A classification report would typically show the precision, recall, F1-score, and support for each emotion class, which is crucial for assessing the model's robustness.

7. Challenges and Improvements:

- **Challenge 1: Video Quality and Frame Sampling**:
    - Low-resolution videos or videos with subtle emotional expressions can make it difficult for visual models to extract meaningful features. A higher frame rate or using more advanced video models (e.g., 3D CNNs) could improve performance.
- **Challenge 2: Textual Ambiguity**:
    - Some emotions can be ambiguous in text alone, as the sentiment might be expressed through context or body language, which highlights the importance of multimodal fusion.
- **Improvement 1: Using Transformer-based Models**:
    - Using more advanced models like **ViT (Vision Transformer)** for video data, or experimenting with more sophisticated multimodal models like **CLIP** (Contrastive Language-Image Pretraining) could potentially improve results.
- **Improvement 2: Augmenting the Dataset**:
    - Data augmentation techniques for video (e.g., flipping, rotation) or using synthetic text generation for underrepresented emotions might improve generalization.

8. Conclusion:

This hackathon demonstrated the importance of multimodal learning for emotion recognition tasks. By leveraging both visual and textual data, we were able to create a feature-rich representation that improved the classifier’s ability to predict emotions accurately. Future improvements could include using more advanced models for both video and text, as well as experimenting with different fusion techniques.

Ensemble Model Classification Report on Validation Set:

	precision	recall	f1-score	support
anger	0.47	0.32	0.38	25
joy	0.48	0.38	0.43	34
neutral	0.70	0.83	0.76	108
sadness	0.33	0.15	0.21	13
surprise	0.48	0.50	0.49	20
accuracy			0.61	200
macro avg	0.49	0.44	0.45	200
weighted avg	0.59	0.61	0.59	200

Cross-Validation Scores: [0.56 0.545 0.58 0.525 0.53]
Average Cross-Validation Score: 0.548