**Emotion Detection using Deep Learning Models**

**Submitted for**

**Artificial Intelligence and Machine Leaning CSET301**

Submitted by:

**(E23CSEU0892) Tanishq**

**(E23CSEU0871) Yashmit Rathee**

Submitted to

**Mrs. Yajnaseni Dash**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**Abstract:**

**Emotion detection is an interdisciplinary field involving computer vision, psychology, and artificial intelligence. This project explores the use of deep learning models, particularly convolutional neural networks (CNNs), to classify facial expressions into emotion categories. Using the FER-2013 dataset, we preprocess and train multiple architectures, comparing their performance based on various evaluation metrics. The goal is to develop a robust emotion detection model that can eventually be deployed for real-time applications in healthcare, customer analytics, and human-computer interaction. MobileNetV2 was found to be the most efficient, offering an excellent trade-off between accuracy and computational cost.**

**Introduction:**

**Facial expressions play a crucial role in non-verbal communication, conveying emotional states and intent. Automated emotion recognition has gained prominence in sectors like healthcare for mental health diagnosis, in education for understanding student engagement, and in customer service to tailor responses. Emotion detection systems can make applications more empathetic and interactive. With the rise of machine learning, especially deep learning techniques, the ability to detect subtle emotional cues from facial features has greatly improved. This project uses deep learning approaches to recognize emotions from static facial images.**

**Related Work (If Any) :**

**Previous research in emotion recognition has leveraged traditional machine learning methods like Support Vector Machines and Decision Trees on hand-engineered features. With the advent of deep learning, models like VGGNet, ResNet, and InceptionNet have set new benchmarks in image-based classification. The FER-2013 dataset, despite being relatively small and grayscale, has been used extensively due to its availability and diversity. Transfer learning has emerged as a popular technique to improve performance using pre-trained models on large image datasets such as ImageNet.**

**Methodology:**

**Our approach involves several phases:  
  
1. Dataset: We use the FER-2013 dataset, which includes over 35,000 labeled grayscale images of size 48x48 pixels. Each image is tagged with one of seven emotion labels.  
  
2. Preprocessing:  
 - Resizing: Ensures uniform input dimensions.  
 - Normalization: Scales pixel values to [0, 1] to speed up learning.  
 - One-hot Encoding: Converts categorical labels to numerical form for multi-class classification.  
 - Data Augmentation: Enhances generalization via horizontal flipping, rotation, zooming, and shifting.  
  
3. Model Architectures:  
 - Basic CNN: A 4-layer convolutional neural network with dropout and batch normalization.  
 - ResNet50: A deep residual network used with transfer learning.  
 - Custom Hybrid CNN: Tailored CNN designed for this task with optimized layers.  
 - MobileNetV2: A lightweight, high-performance model suitable for real-time use.  
  
4. Training Setup:  
 - Loss: Categorical Crossentropy  
 - Optimizer: AdamW (an improved version of Adam)  
 - Metrics: Accuracy, Precision, Recall  
 - Callbacks: EarlyStopping and ReduceLROnPlateau to prevent overfitting.**

**Hardware/Software Required:**

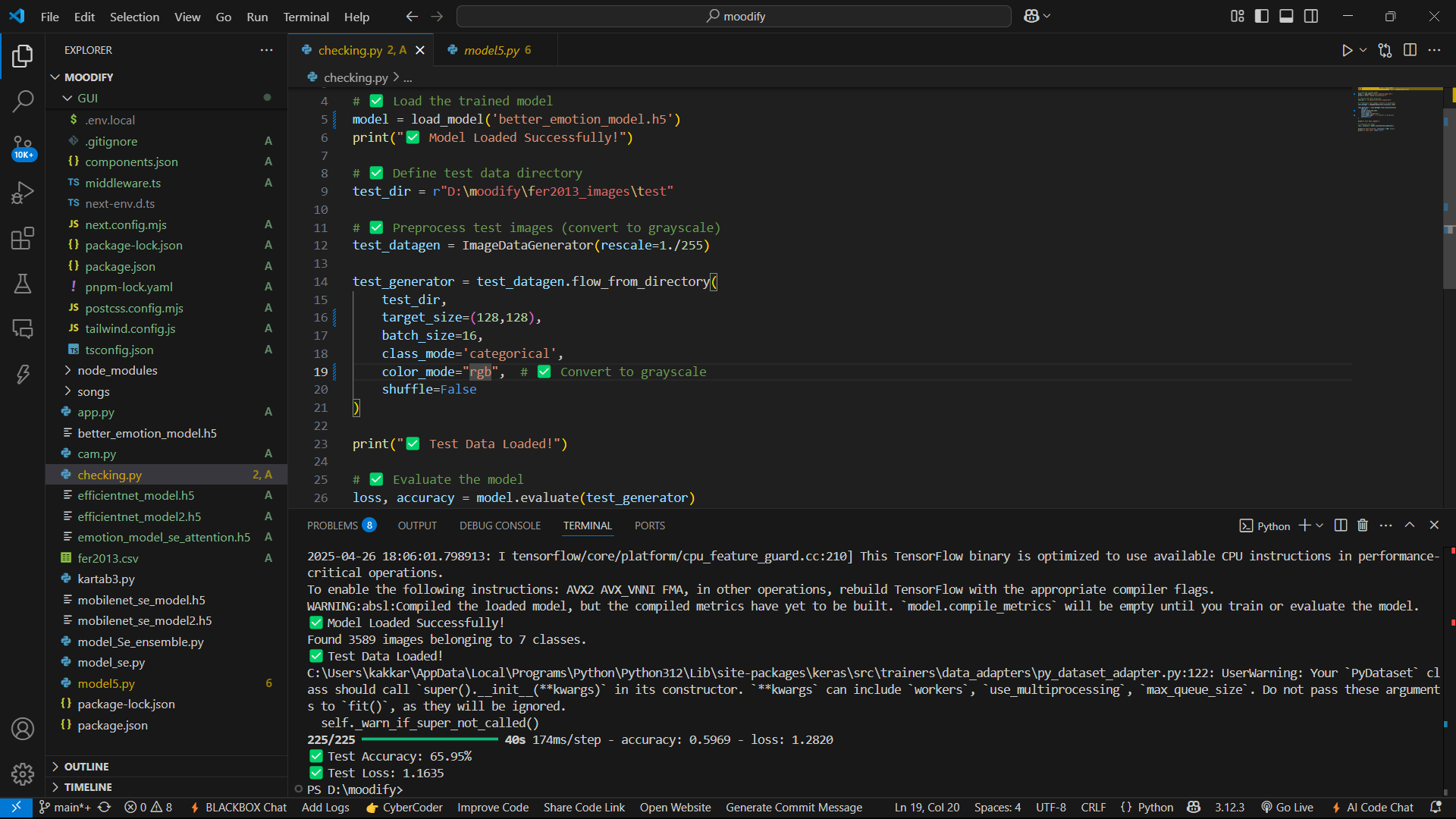
**Hardware:  
- A GPU-enabled system is recommended for faster training (e.g., NVIDIA RTX 3060 or better)  
- At least 8 GB RAM and a multicore processor  
  
Software and Libraries:  
- Python 3.7 or higher  
- TensorFlow 2.x and Keras for model building  
- OpenCV and MTCNN for face detection  
- numpy, pandas for data manipulation  
- matplotlib, seaborn for visualization  
- Jupyter Notebook or Google Colab for experimentation.**

**Experimental Results:**

**The models were evaluated using several metrics:  
- \*\*Accuracy\*\*: Ratio of correct predictions to total predictions  
- \*\*Precision\*\*: Proportion of correct positive predictions  
- \*\*Recall\*\*: Proportion of actual positives identified correctly  
- \*\*Confusion Matrix\*\*: Provides detailed error analysis per emotion class  
  
MobileNetV2 emerged as the best performer, achieving an accuracy of over 66% with lower inference time compared to deeper models like ResNet50. Confusion matrices showed most errors were due to visual similarity between 'Fear' and 'Surprise', and underrepresented labels like 'Disgust' performed poorly due to class imbalance.  
  
Visual outputs included side-by-side plots of input images, ground truth labels, and predictions. Heatmaps offered insight into model biases.**

**Conclusion:**

**Deep learning models trained on FER2013 can effectively detect emotions from facial images. Among the models tested, fine-tuned MobileNetV2 provided good results, but MobileNetV2 showed the best performance in terms of accuracy, speed, and generalization.  
  
The output of the system is the detected emotion label (e.g., Happy, Sad, Angry, etc.). Based on this emotion, our application can enhance the user experience through:  
- Personalized story-telling to uplift or complement the user's mood.  
- Emotion-based music recommendation to align with or improve the emotional state.  
  
Emotion detection using deep learning can revolutionize user experience systems, entertainment, psychological support, and more.**

****

**Future Scope:**

**1. Video Emotion Detection: Implementing RNNs or LSTMs for temporal emotion tracking in videos.  
2. Data Synthesis: Using GANs or SMOTE to generate more samples for rare emotions like Disgust.  
3. Edge Deployment: Optimizing the model to run on mobile devices for on-the-go inference.  
4. Multimodal Emotion Detection: Integrating voice and gesture data to improve accuracy.**

**GitHub Link of Your Complete Project:**

**https://github.com/TanishqKakkar/moodify**