CodeScribed

Coding Assignment for Junior AI Engineer

Title: Facial Emotion–Driven Sentiment Analysis with RAG-based Insights

Duration: 8 - 10 Hours

Overview

This assignment evaluates your ability to build an end-to-end AI pipeline that combines Computer Vision, Natural Language Processing (NLP), and Retrieval-Augmented Generation (RAG) using LangChain. You'll complete the task in three stages:

1. Facial Expression Recognition
2. Synthetic Review Generation, Sentiment Analysis & RAG Integration
3. Design Thinking & Architecture

Stage 1: Facial Expression Recognition (40 Points)

Objective: Build a CNN or fine-tune a pretrained model to classify human facial expressions into emotion categories.

Tasks

- Dataset Preparation (5 pts): Use a small open dataset like FER-2013 or CK+. Split into training and test sets.
- Model Development (15 pts): Implement a CNN from scratch or fine-tune a pretrained model. Train to recognize at least five emotions.
- Evaluation & Output (10 pts): Evaluate accuracy, F1-score, and confusion matrix. Save predictions for sample test images.
- Code Quality & Documentation (10 pts): Clean, modular, well-commented code with clear explanation of dataset, preprocessing, and architecture.

Deliverables (Stage 1)

- Source code or Jupyter notebook
- Trained model / weights (if applicable)
- Confusion matrix and classification metrics screenshot
- CSV or JSON mapping image → predicted emotion

Stage 2: Synthetic Review Generation, Sentiment Analysis & RAG Integration (40 Points)

Objective: Use predicted facial emotions from Stage 1 to create synthetic text reviews or captions, perform sentiment analysis, and integrate a RAG (Retrieval-Augmented Generation) system using LangChain for intelligent querying and summarization.

Tasks

- Data Generation (10 pts): Generate text reviews for each predicted emotion using simple rule-based or templated logic.
- Embedding & Vector Store Setup (10 pts): Convert generated reviews into embeddings using sentence-transformers or OpenAI embeddings. Store them in FAISS, Chroma, or Pinecone.
- RAG Query System using LangChain (15 pts): Integrate LangChain to build a RAG pipeline that accepts user queries, retrieves relevant reviews, and optionally summarizes them using an open-source LLM.
- Sentiment Analysis Integration (5 pts): Apply sentiment classification to the retrieved text snippets to display sentiment context.

Deliverables (Stage 2)

- Source code or notebook implementing embeddings, vector database, and LangChain integration
- Sample queries and outputs with screenshots or CSV results
- Short explanation of RAG architecture and LangChain usage

Stage 3: Design Thinking & Architecture (20 Points)

Objective: Evaluate the your ability to reason about architecture, scalability, technology selection, and ethics.

Questions

- Technology Stack & Architecture (5 pts): Describe your overall architecture. How do facial recognition, review generation, embeddings, and LangChain fit together? Why did you choose specific libraries?
- Data Flow and Protocol Design (5 pts): Sketch or describe the data pipeline from image → text → vector → query → response. Which protocols or APIs (REST, gRPC, FastAPI) would you use for deployment?
- RAG System Scalability (5 pts): How would you scale your RAG pipeline to handle millions of reviews? Which vector database or embedding model would you choose in production?
- Ethics & Bias (5 pts): Discuss ethical implications of emotion-based sentiment analysis. How would you mitigate bias in both vision and NLP models?

Evaluation Summary

Component	Criteria	Points
Stage 1 – Facial Expression Recognition	Dataset prep, model, metrics, code quality	40
Stage 2 – Review Generation, Sentiment, RAG	Data pipeline, embeddings,	40

& LangChain Integration	retrieval, sentiment results	
Stage 3 – Design Thinking & Architecture	Stack selection, RAG design, scalability, ethics	20
Total		100

Suggested Tech Stack

Layer	Recommended Tools
Image Model	TensorFlow / PyTorch / Keras
NLP + Sentiment Model	scikit-learn / HuggingFace Transformers
Embeddings	sentence-transformers / OpenAIEmbeddings / InstructorEmbeddings
Vector Store	FAISS / ChromaDB / Pinecone
RAG Framework	LangChain
Serving / UI	Streamlit / FastAPI

Bonus (Optional, +10 Points)

Add a simple Streamlit app that integrates the entire pipeline:
- Upload image → detect emotion → generate review → embed and store → allow user query → retrieve and show sentiment.