Question 1: Elevator Pitch

This project develops an efficient facial expression recognition algorithm using a hierarchical deep neural network (DNN) structure for real-time analysis. It processes facial expressions from live video feeds using convolutional neural networks (CNNs), integrating techniques like transfer learning and adaptive learning rates for accuracy and robustness. Designed as a web-based application, the system supports applications in emotion analysis, healthcare monitoring, and interactive AI.

Question 2: Dataset Details

1. **Collector(s):** Noam Segal
2. **Year:** Not explicitly mentioned (Dataset derived from AffectNet, originally created in 2017)
3. **Title of Dataset:** *AffectNet Training Data*
4. **Version Number:** Not specified
5. **Publisher:** Kaggle
6. **DOI or URL:** AffectNet Training Data - Kaggle
7. **Study/Paper/Reason:**
    - The dataset was collected for research on facial expression recognition in the wild.
    - AffectNet is one of the largest facial expression datasets, containing over 1 million labeled images spanning different emotions.
    - It is widely used to train deep learning models for emotion detection in various applications, including healthcare, human-computer interaction, and sentiment analysis.

Question 3: Language and Libraries

- **Language:** Python 3.x
- **Libraries:**
    - **Machine Learning & Deep Learning:** TensorFlow, Keras, PyTorch
    - **Computer Vision:** OpenCV, Dlib
    - **Data Handling:** Pandas, NumPy
    - **Visualization:** Matplotlib, Seaborn
    - **Web Framework:** Flask or Django

Question 4: Code Development

The following parts of the code will be written from scratch:

- **Dataset Preprocessing:**
    - Face detection, alignment, and augmentation techniques (rotation, flipping, noise addition).
- **Model Development:**
    - Custom hierarchical deep neural network (DNN) combining CNN layers.
    - Implementation of transfer learning using pretrained models like ResNet, VGG, or MobileNet.
- **Training and Optimization:**
    - Training models with batch normalization, dropout, and adaptive learning rates.
    - Hyperparameter tuning for better accuracy.
- **Real-time Processing & Inference:**
    - Live video feed integration using OpenCV.
    - Real-time expression recognition and tracking.
- **Web Application Integration:**
    - Developing backend APIs in Flask/Django.
    - Designing frontend using HTML, CSS, and JavaScript.

Question 5: Best Choice of Model(s) and Justification

The best model choices include:

- **Hierarchical CNN Model:**
    - Captures both local micro-expressions and global facial changes.
    - Provides efficient feature extraction and classification.
- **Transfer Learning with Pretrained CNNs (ResNet50, VGG19, MobileNetV2):**
    - Speeds up training and improves accuracy by leveraging pre-learned features.
- **Hybrid Approach (CNN + LSTM or Attention Mechanisms):**
    - Captures temporal dependencies in video-based emotion recognition.
    - Improves robustness in real-time expression tracking.

Question 6: Hyperparameters and Optimization Strategy

Key hyperparameters:

- **Learning Rate:** Initially set at 0.001 with decay strategy.
- **Batch Size:** 32 or 64, tuned based on memory constraints.
- **Number of CNN Layers:** Experimenting with 4–6 layers for feature extraction.
- **Dropout Rate:** 0.3–0.5 to prevent overfitting.
- **Optimizer:** Adam or RMSprop for efficient gradient updates.
- **Loss Function:** Categorical Cross-Entropy (since it’s a multi-class classification problem).

- **Early Stopping & Regularization:** Used to prevent overfitting and improve generalization.

  Optimization Strategy:
- Grid Search and Random Search for hyperparameter tuning.
- Cross-validation to evaluate model generalization.

Question 7: Model Evaluation Metrics

- **Accuracy:** Measures overall correct predictions.
- **Precision & Recall:** Evaluates model balance between false positives and false negatives.
- **F1-score:** Provides a harmonic mean of precision and recall for balanced assessment.
- **Confusion Matrix:** Visualizes classification performance for each emotion.
- **Inference Latency:** Ensures real-time performance for video-based applications.