a. Explanation of How You Approached the Given Problem

For this project, the task was to implement a gesture recognition system using videos that capture various hand gestures. The goal was to extract key features from the videos and use machine learning techniques to recognize gestures.

Here's the general approach:

1. Problem Understanding: The task is to identify specific gestures from video data, which involves analyzing video frames and extracting key features from them. These extracted features can be passed into a pre-trained model to classify the gestures based on learned patterns.

2. Video Processing and Feature Extraction: For each gesture video, we need to extract the middle frame as a representative of the gesture. This is done using the `frameextractor.py` code, which captures the key frame. The middle frame provides a good approximation of the gesture without needing to process the entire video.

3. Hand Shape Feature Extraction: Using a pre-trained model (`cnn_model.h5`), features are extracted from the captured frame. This is handled by the `handshape_feature_extractor.py` file, where the model processes the image to identify key hand features.

4. Training and Testing Data: The training videos were used to create a dataset of gesture features. Each feature vector represents the key characteristics of a particular gesture. Test videos were also processed in the same way, extracting features from the middle frame.

5. Gesture Recognition: For gesture recognition, cosine similarity was used to compare the test video's feature vector to the stored feature vectors from the training data. This comparison helps identify the most similar gesture from the training set. The recognized gestures were saved in a results file (`Results.csv`).

B. Solution for the Problem

The solution implemented for this project involved several steps and technologies:

1. Technologies Used:

- Python: For implementing the entire solution, including video processing, feature extraction, and gesture recognition.
- TensorFlow/Keras: A pre-trained model was used for extracting handshape features from video frames.
- OpenCV: Used to handle video processing and frame extraction from gesture videos.
- Cosine Similarity: A method from the `scipy.spatial` module was used to compare the extracted feature vectors and identify the closest matching gesture.

2. Steps in the Solution:

1. Video Frame Extraction: A key frame (middle frame) from each video was extracted using OpenCV to reduce computational load while still capturing the key characteristics of each gesture.

2. Feature Extraction with CNN: A pre-trained model (`cnn_model.h5`) was used to extract meaningful features from the handshape present in the key frame. This model identifies patterns in the image, such as hand shapes or movement, which are essential for gesture recognition.

3. Training and Testing: Feature vectors were extracted from the training set and stored. These vectors represent the different gestures used in the training data. Similarly, feature vectors from the test videos were extracted to be compared against the training vectors.

4. Gesture Recognition Using Cosine Similarity: For each test video, the feature vector was compared to all training vectors using cosine similarity. This helped determine which gesture in the training set most closely matched the test gesture.

5. Output: The recognized gestures were stored in a CSV file (`Results.csv`), which lists the predicted gestures for each test video.

Final Summary: The approach involved frame extraction, feature extraction using cnn_model.h5 , and gesture recognition using cosine similarity. The problem was solved by leveraging machine learning techniques to extract features and then matching them using cosine similarity to recognize gestures from video data.