Project Title : Best Photo Identifier

Introduction:

This code utilizes various computer vision techniques to analyze a video input, detecting faces, poses, expressions, and image quality. It assigns scores to frames based on these detections, providing insights into the video content.

Libraries Used:

- `cv2`: OpenCV library for computer vision tasks.
- `mediapipe`: Mediapipe library for face and pose detection.
- `matplotlib.pyplot`: Matplotlib library for visualization.
- `deepface`: DeepFace library for facial expression analysis.
- `numpy`: NumPy library for numerical operations.

Functionality:

1. **Video Capture:**
    - Opens the specified video file for analysis.
    - Initializes an array to store scores associated with each frame.
2. **Face and Pose Detection:**
    - Utilizes Mediapipe's face detection model to detect faces in the frames.
    - Uses a pose detection model to identify human poses.
    - Increments or decrements the score based on the presence or absence of faces and poses.
3. **Blur Detection:**
    - Calculates the Laplacian variance of the grayscale frame to detect blurriness.
    - Adjusts the score based on the variance value.
4. **Contrast Score:**
    - Computes the mean and standard deviation of pixel intensities in the grayscale frame.
    - Calculates a contrast score based on the standard deviation relative to the mean.
    - Modifies the score according to the contrast score.
5. **Expression Detection:**
    - Utilizes DeepFace library to analyze facial expressions in the frame.

    - Adjusts the score based on detected emotions (e.g., happy, sad, angry, neutral).
6. **Sorting and Saving:**
    - Sorts the frames based on their scores in descending order.
    - Saves the top 15 frames with the highest scores as images.
7. **Cleanup and Output:**
    - Releases the video capture resources.
    - Prints the array of scores and their corresponding frame types.

Usage:

- Ensure the necessary libraries (`cv2`, `mediapipe`, `matplotlib`, `deepface`, `numpy`) are installed.
- Replace the file path in `cv2.VideoCapture()` with the path to the video file you want to analyze.
- Run the script to perform video analysis and obtain the top frames.

Conclusion:

The provided code offers a comprehensive approach to analyzing videos by integrating facial and pose detection, emotion recognition, and image quality assessment. By assigning scores to frames, it facilitates the identification of key moments in the video content. Users can further customize and extend this functionality based on specific requirements.

References:

1. **OpenCV:**
   - Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.
   - OpenCV. (n.d.). OpenCV Documentation. Retrieved from https://docs.opencv.org.
2. **Mediapipe:**
   - Mediapipe (n.d.). Mediapipe Documentation. Retrieved from https://google.github.io/mediapipe.
3. **Matplotlib:**
   - Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95.
   - Matplotlib. (n.d.). Matplotlib Documentation. Retrieved from https://matplotlib.org.
4. **DeepFace:**
   - Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
   - DeepFace. (n.d.). DeepFace Documentation. Retrieved from https://github.com/serengil/deepface.
5. **NumPy:**
   - Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362.
   - NumPy. (n.d.). NumPy Documentation. Retrieved from https://numpy.org/doc.