Aim

To implement a real-time face mesh detection application using TensorFlow.js and React.

Objective

- Integrate a face mesh detection model into a React application.
- Use the webcam to capture video and overlay a face mesh on detected faces.
- Compare performance between the old and new face mesh models.

Summary

This project involves creating a React application that utilizes TensorFlow.js to perform real-time face mesh detection. The application captures video from a webcam and overlays a mesh on detected facial features. Two face mesh models from TensorFlow.js are compared: the older @tensorflow-models/facemesh and the newer @tensorflow-models/face-landmarks-detection. The app dynamically updates the face mesh on the canvas as the video feed changes.

Tools and Libraries Used

- **React**: For building the user interface.
- **TensorFlow.js**: For running the face mesh detection model.
- **@tensorflow-models/facemesh**: The older face mesh detection model.
- **@tensorflow-models/face-landmarks-detection**: The newer face mesh detection model (commented out in code).
- **Webcam**: To capture video feed.
- **Canvas**: To draw the detected face mesh.
- **JavaScript**: For implementing the logic.

Procedure

1. **Setup React App**:

CODE:

npm init react-app facemesh

2. **Install Dependencies**: Install TensorFlow.js and the face mesh models:

CODE:

npm install @tensorflow/tfjs @tensorflow-models/facemesh react-webcam

3. **Code Explanation**:
    - **App.js**:

        - Imports necessary libraries and components.
        - Defines runFacemesh to load the face mesh model and start detection in intervals.
        - detect function captures the video frame, performs face mesh detection, and draws the results on a canvas.
        - The useEffect hook ensures that the runFacemesh function is called when the component mounts.
        - The Webcam and canvas components are used to display video and the face mesh overlay.
    - **utilities.js**:
        - Contains the TRIANGULATION array, which defines the mesh structure for rendering the detected face mesh.

Highlights

- **Real-Time Detection**: The application performs face mesh detection in real-time, providing immediate feedback on facial features.
- **Model Comparison**: Code supports switching between two versions of the face mesh model, showcasing flexibility in adapting to newer models.
- **Modular Design**: Separation of concerns with detection logic in App.js and mesh triangulation data in utilities.js.

Conclusion

The project successfully integrates real-time face mesh detection into a React application using TensorFlow.js. The ability to compare different models provides insights into performance and accuracy differences. The application effectively overlays a face mesh on detected faces, demonstrating practical use of machine learning models in web development. Future improvements could include adding user controls for selecting models and optimizing performance for different devices.