Multimedia Finals Plan Book

Developing a Filter based on Image Recognition Methods

Group Members:

資訊二乙 D1231150 鄭健良

資訊二丁 D1229568 彭定康

資訊二乙 D1231400 張志成

List of Content:

1. Project Overview

1.1 Project Title

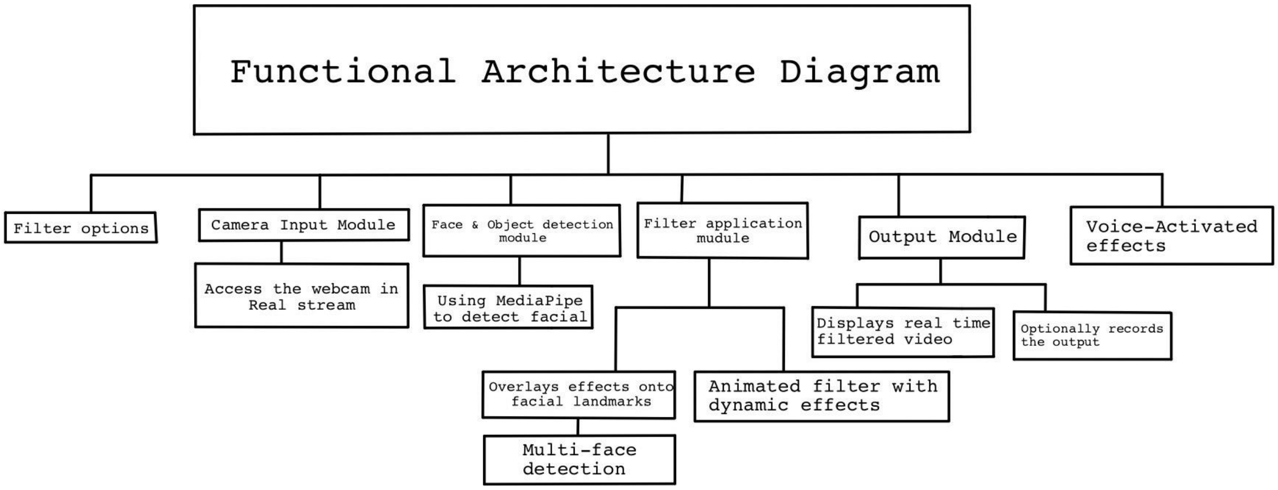
Developing a Filter based on Image Recognition Methods

1.2 Motivation

With the rise of short-form video platforms like TikTok, AI-powered filters have become a popular form of real-time digital interaction. This project aims to explore how image recognition techniques can be used to create filters that respond to facial features, expressions, or objects in the camera frame. By combining facial detection with overlay effects, we provide users with an engaging multimedia experience using OpenCV and Python.

2. Functional Architecture

2.1 Functional Architecture Diagram


Functional Architecture Diagram
Filter options
Camera Input Module
Access the webcam in Real stream
Face & Object detection module
Using MediaPipe to detect facial
Filter application mudule
Overlays effects onto facial landmarks
Multi-face detection
Animated filter with dynamic effects
Output Module
Displays real time filtered video
Optionally records the output
Voice-Activated effects

2.2 Module Descriptions

2.2.1 Filter Options

- This is the starting point of the system.
- Provides the user interface to select visual filters such as hats, mustaches, or glasses.
- The selected filter is passed to the Filter Application Module for processing.

2.2.2 Camera Input Module

- Captures live video frames using the device's webcam.
- Uses OpenCV's cv2.VideoCapture() to continuously read frames.
- Refreshes every 10 milliseconds to maintain real-time performance.
- Sends each captured frame to the Face & Object Detection Module for analysis.

2.2.3 Face & Object Detection Module

- Uses MediaPipe Face Mesh to detect facial landmarks with high precision.
- Each frame is converted to RGB, as required by the MediaPipe detection model.

- Detects 468 facial landmark coordinates, including key points like eyes, nose, mouth, and chin.
- These coordinates are used to accurately position filters (e.g., glasses, hats, masks) on the face in real-time.

2.2.4 Filter Application Module

- Takes the selected filter and applies it according to the detected face position.
- Dynamically resizes and positions the filter
- Ensures the filter stays within frame boundaries and does not distort.
- When multiple faces appear in the frame, the system can detect and apply different filters to each face individually.
- By integrating keypoint tracking using MediaPipe, the system ensures that filters remain accurately aligned with facial movements and rotations in real-time.
- Filters can change in response to facial expressions such as hearts appear when the user smiles, or lightning effects trigger when the mouth opens.

2.2.5 Output Module

- Combines the original frame and the applied filter.
- Displays the processed frame in a real-time window using cv2.imshow().
- Continues to update as long as the camera feed is active.
- Optionally records the output by writing each frame to a video file using OpenCV's cv2.VideoWriter, allowing the user to save filtered video clips for later viewing or sharing.

2.2.6 Voice-activated effects

- Special effects animations can be triggered when the user says specific voice commands (e.g., "Transform," "Launch").
- Uses *speech_recognition* to capture and interpret user voice input from the microphone.