Commentary:

Findings:

We implemented multiple image filters, including the red, green, and blue (RGB) channels, as well as greyscale conversion and face detection. Each filter offers unique insights into different image components. For instance, when applying a threshold to the red channel (`RChannelThreshold.js`), the result highlights the areas with higher red values. Similarly, the `GreyscaleImage.js` filter converts colored images to greyscale by averaging the RGB values, which proved useful for creating a cleaner output for subsequent image processing techniques like face detection. Especially, the RGBtoYCbCrThreshold filter which seems powerful in detecting main object and clearing all surrounded noise.

Challenges and Solutions:

1. **Flipping the Video Horizontally**: One of the main challenges was ensuring that the video feed appeared correctly, especially when applying transformations. This was solved by using a combination of the `push()` and `pop()` methods in p5.js to manage the transformations for the video (such as flipping and scaling) and text rendering separately. Here is the solution:

```
push();
translate(width - (width - (this.x + this.w)), 0);
scale(-1, 1);
this.drawImg();
pop();
this.drawText();
```

2. **Frame Organization**: To streamline the process of applying multiple filters across different frames, a `FilterControl` class was created. This allowed for efficient management of the frames and ensured that the filters could be applied consistently. This was crucial when handling multiple filters like RGB channels, thresholds, and face detection. by building predefined methods within the `FilterControl` Class which outlines the functionality for all frames.

3. **Face Detection Box Precision**: During face detection, the bounding box around the detected face occasionally produced decimal numbers, making pixel manipulation difficult. To overcome this, we used `Math.floor()` to round the dimensions to ensure integer values, allowing accurate pixel selection within the pixel array.

Project Target:

The project aimed to develop a modular filter control system capable of applying different filters to a video stream in real-time. This was achieved by creating the `FilterControl` class, which contains methods to apply filters and manage frames. Each filter was then built as a subclass of `FilterControl`, making it easy to extend and manage new filters.

Extension:

The unique extension added to the project was a real-time hand gesture detection feature. This feature allows the program to detect a hand gesture and trigger an action, such as displaying an alert when the hand is raised for more than five seconds. The uniqueness of this extension lies in its integration with the MediaPipe framework, which provided robust hand landmark detection. The decision to use MediaPipe over alternatives like OpenCV js and ml5.js was based on its superior performance, accuracy and simplicity as it provide us we just the methods and data required for implementing this gesture.

The logic for detecting whether a hand is raised was built around three criteria:

1. **All fingers must be open** (not collapsed).
2. **The fingers must be in order**, ensuring the hand is fully extended.
3. **The hand must be positioned at a near 90-degree angle** to the horizontal line.

This extension is relevant in modern video applications, allowing for intuitive gesture-based controls, such as triggering actions in video players or sending alerts based on gestures. The integration of this feature was smooth and made the project more interactive. It was implemented by adding a class called `Extension`, which extends the `FilterControl` class, maintaining the overall modularity and flexibility of the system. Unlike other frames in this project this gesture works only with VIDEO stream as it does nt make sense to get it working with static images.

Conclusion:

Our implementation for the project met its objectives by building a flexible framework for applying filters to video streams in real time. The inclusion of hand gesture detection extended the project's scope beyond basic filters, offering an interactive experience. We did overcome technical challenges through methodical problem-solving, ultimately delivering a robust and expandable solution.