OLIVE (OpenCV **LIVE)** is a web-based, visual programming tool for real-time image, video, and 3D processing. It uses **nodes** to represent functions and **wires** (connections) to define the flow of media data between them, leveraging **OpenCV.js, Three.js** and **MediaPipe**.

I. System Requirements and Setup

- **Browser:** A modern web browser that supports **WebGL** and **JavaScript**.
- **Dependencies:** The application loads external libraries including opencv.js and three.js.
- **Status Check:** When the editor loads, the **OpenCV status** will be displayed in the interface, turning green once the library is ready.

II. Core Editor Concepts

Component	Description	Interaction
Node	A block representing a specific function.	**Add:** Drop the Node into the **Graph Editor.** **Move:** Drag the **Node Header** (title bar). **Remove:** Select the Node and press **Delete.**
Port	Connections for data flow. **Inputs** (left) receive data; **Output** (right) sends data.	**Create Connection:** Click and drag from an **Output Port** to an **Input Port**.
Wire	Defines the flow of data from an output to an input.	**Delete Connection: Right-click** on the **Port** (deletes all wires connected to that port).
Fullscreen	View media output (image, video, canvas) in full screen.	**Double-click** on the image/video/canvas element *inside* a node.
File	Load and save graphs in **JSON** format.	**Click** the **Save** icon to convert your project in a JSON string and save it locally. **Click** the **Open** icon to load a project.

III. Node Catalog and Functionality

Nodes are organized into categories based on their role in the workflow.

1. Input Nodes

These nodes provide the starting media source for your graph.

- **Image:** Upload a static image file or provide a link.
- **Video:** Upload a video file or provide a link.
- **Camera:** Capture a live video feed from your device's camera.

2. Processing Nodes

These nodes perform image and video manipulation using **OpenCV.js**.

Node Title	Primary Functionality	Key Controls & Parameters
Gamma Correction	Changes pixels' intensities.	Adjust **γ** value.
Histogram Equalization	Equalizes the histogram, globally or locally.	Adjust the **Grid Size** for **CLAHE** (Contrast Limited Adaptive Histogram Equalization)
Color Adjustment	Manipulates the HSV color space.	Use the **Color Picker** to define the color (or the target range) and the sliders for new **Hue** and **Saturation** values. Includes **Full Range** and **Invert** options.
Convolution	Applies spatial filtering.	Select **Filter Type** (e.g., Gaussian Blur, Sobel Edges) and adjust **Kernel Size** or define a **Custom Kernel**.
Morphology Rank	Applies morphological operations and ranking filters.	Select **Filter Type** (e.g. Erosion, Dilation, Median), **Kernel Size**, and **Kernel Shape** (Rectangle, Ellipse, Cross).
Polar Transformation	Warps the image using coordinate transformations.	Select **Effect Type** (Fish Eye, Cone, Swirl) and **Effect Strength**.
Glitch Effects	Applies dynamic visual distortions.	Select **Effect Type** (Shaking, Aberration, Fade, Glass) and **Effect Strength**.
Thresholding	Binarizes the color channels, using a global or local threshold.	Adjust the **Threshold Value** and select the **Type** (Binary, Otsu, Adaptive).
Matrix Operations	Adds, subtracts, multiplies, divides two images or applies min/max operations.	Select the **Operation Type** and the **Weights** for the inputs.
Channel Mixer	Remixes the RGB channels.	Adjust the **Percentage** of **Red, Green and Blue** at every color channel.
Color Blending	Replaces the color (e.g. Hue, Saturation) of an image with the color of another.	Select the **HSV** channels (Hue, Saturation, Value) to be replaced.

Concatenation	Merges the input images either horizontally or vertically.	Adjust the percentage of **Overlap** with the slider.
Connected Components	Finds superpixels in the input image.	Adjust the **Threshold Value** for the **Segmentation**.
Background Subtraction	Removes non-moving objects (background) from the input video.	No controls.
Optical Flow	Visualizes the optical flow with arrows.	Adjust the **Block Size**.
Skeleton	Applies the Distance Transform to the RGB channels.	No controls.

3. Rendering Nodes

They **r**ender the input texture onto a **3D geometry** using **Three.js**.

- **Projection:** A specialized node that uses the input as a **projected texture** onto the scene. Use the mouse to **orbit and zoom** the camera. A spherical indicator represents the **Projector** which can be dragged using **Drag Controls**.

- **Mapping / Lighting:** UV mapping of the input image/video on a **GLTF Model** or a Plane, Cube, Sphere etc. The spherical indicator represents a **Point Light**. Adjust the color and intensity with the color picker and the sliders.

4. Pose-Estimation Nodes

They track the human body using the **MediaPipe** library.

- **Human Pose:** Applies segmentation to the person (if any) and removes background.
- **Character Animation:** pose-driven movement of an uploaded **VRM humanoid model**.

5. Output Node

<u>This node does not have an Output Port.</u>

- **Canvas Viewer:** This is the final step in the graph. It displays the result of the connected node.