

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

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

<b>Concatenation</b>	Merges the input images either horizontally or vertically.	Adjust the percentage of <b>Overlap</b> with the slider.
<b>Transitions</b>	Classic WebGL transitions between two inputs.	Select <b>Transition Type</b> (e.g. Fade, Wipe, Radial, Dissolve) and adjust the <b>Transition Duration</b> .
<b>Connected Components</b>	Finds superpixels in the input image.	Adjust the <b>Threshold Value</b> for the <b>Segmentation</b> .
<b>Background Subtraction</b>	Removes non-moving objects (background) from the input video.	No controls.
<b>Optical Flow</b>	Visualizes the optical flow with arrows.	Adjust the <b>Block Size</b> .
<b>Skeleton</b>	Applies the Distance Transform to the RGB channels.	No controls.

### 3. Rendering Nodes

They render 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 light's 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 the background.
- **Character Animation:** pose-driven movement of an uploaded **VRM humanoid model**.

### 5. Output Node

This node does not have an Output Port.

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