DEEPVIEW

X400 Vision Processor

User Guide

AI-powered image analysis with neural network support.

Supports GigE and USB cameras for industrial automation.

Train, test, and deploy AI models remotely via browser.

Fanless, rugged design for industrial environments.

This guide provides detailed instructions on setting up, configuring, and using the X400 Vision Processor.

It covers system requirements, installation steps, user interface navigation, and key operational functions.

Additionally, it includes troubleshooting guidance and configuration options for networking and training datasets.

For a video walkthrough, scan the QR code or visit
https://www.youtube.com/watch?v=D46z0q3oqs0

Revision Date
3/9/2025

Tech Support
(586) 854-7475
support@deepviewai.com

Table of Contents

Contents

X400 Vision Processor Setup

The X400 Vision Processor includes a versatile I/O interface designed for industrial automation applications. This section provides an overview of the available input and output options.

The I/O connector includes:

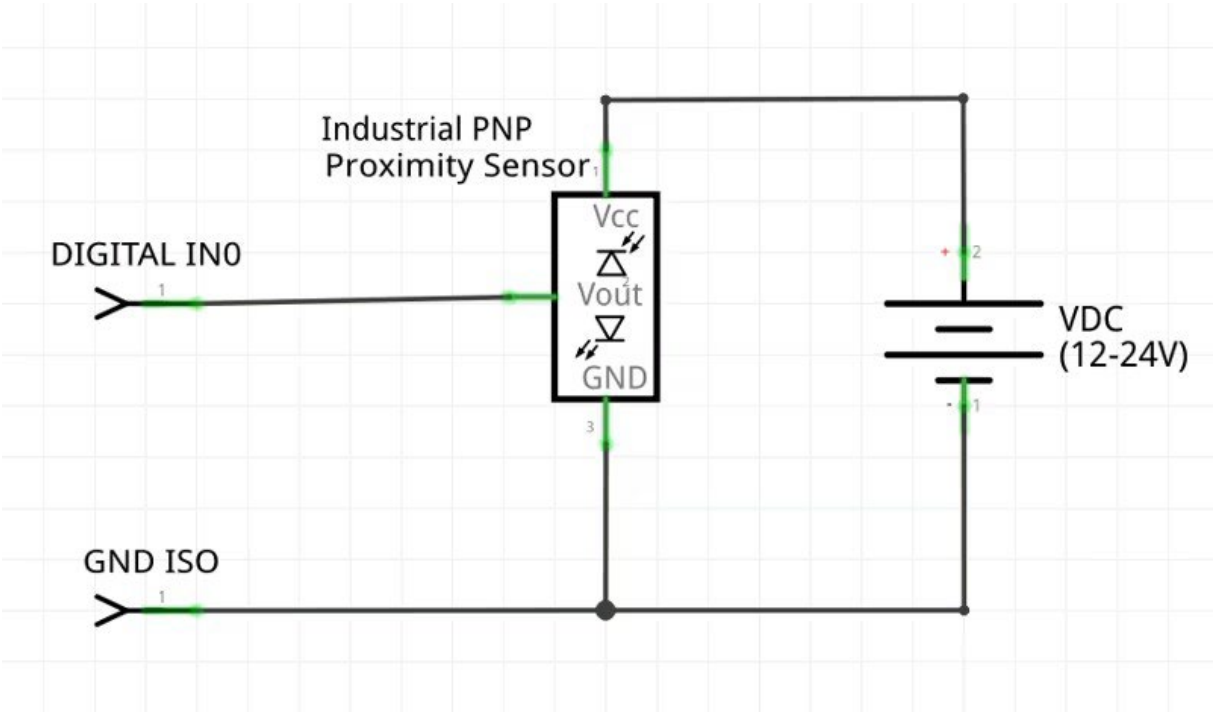
- **Isolated Inputs: 3**
- **Outputs: 2**
- **Ground Connections: 3**

Configuring and Reading Inputs

The digital input side accepts 12-24V signals and is rated for up to 2.25mA. These inputs can be used for various trigger and sensing applications, such as proximity sensors or other industrial signaling devices.

Connecting an Input Device

To connect a digital input device, ensure that it is properly wired to the input pin and ground. Below is a generic wiring example of an industrial PNP proximity sensor:

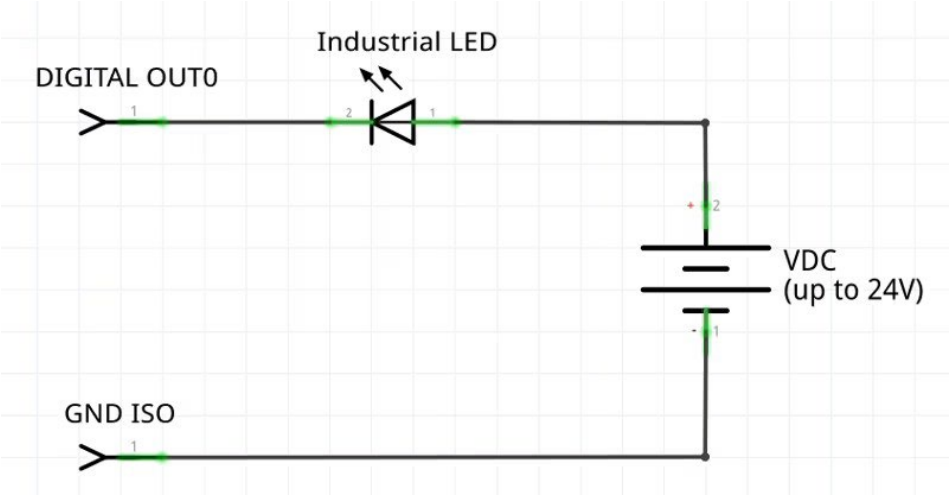


Setting Digital Outputs

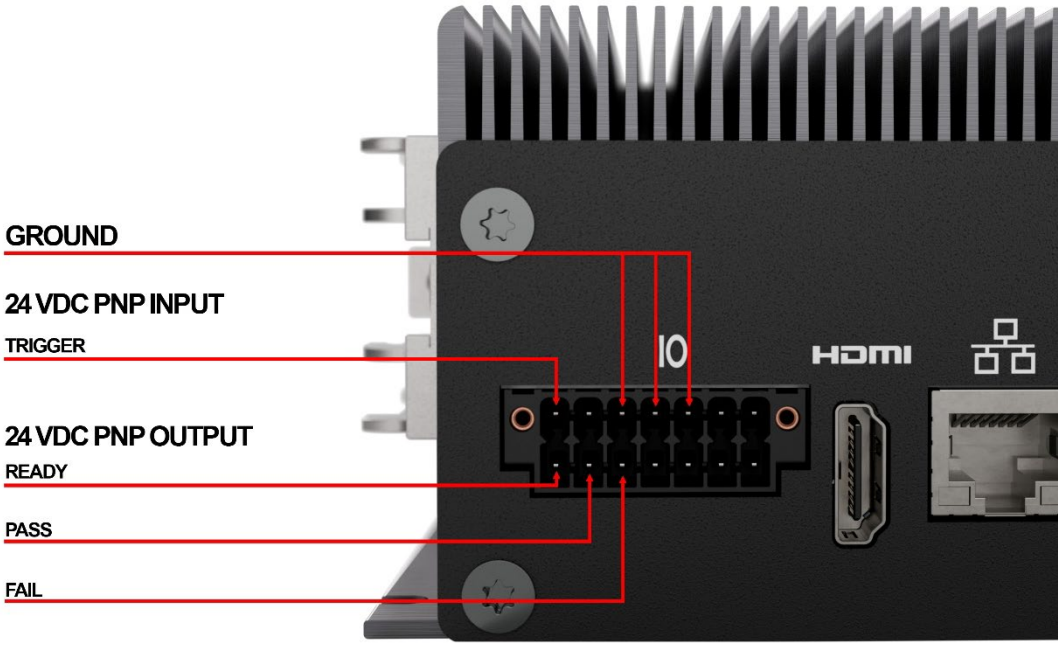
The digital output side supports **up to 24V** with a current limit of **1A per channel**. Outputs function as **low-side switches**, meaning they toggle between ground and an active state when triggered.

Connecting an Output Device

For output control, ensure the device (e.g., indicator light, relay, or other signaling components) is wired correctly. Below is a generic wiring example of an industrial LED indicator:



X400 Vision Processor Pinout



Connecting to a PC

Before connecting to the X400 Vision Processor for the first time, ensure that:

✅ The **RJ45 cable** is plugged into your **Ethernet port**.

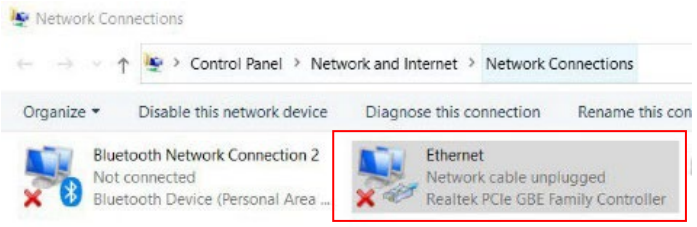
Follow the steps below to configure a **static IP address** on your PC (Windows) to match the processor's subnet.

1. Identify the Server's IP Address

- The IP address can be found on the processor's label.

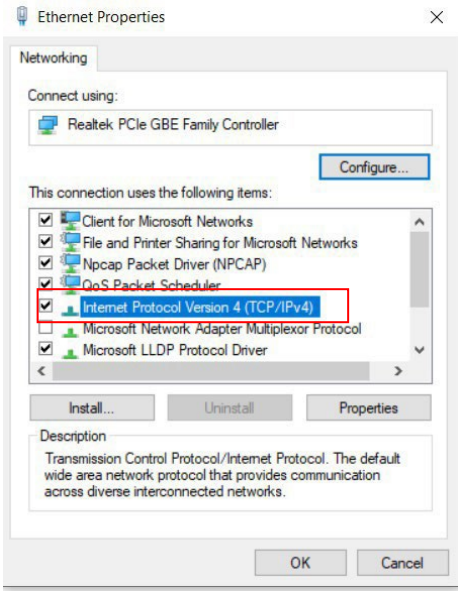
2. Open Network Connections

- In the **Windows search bar**, type **"network connections"** and select the top result.
- Locate and **double-click on your active Ethernet connection**.



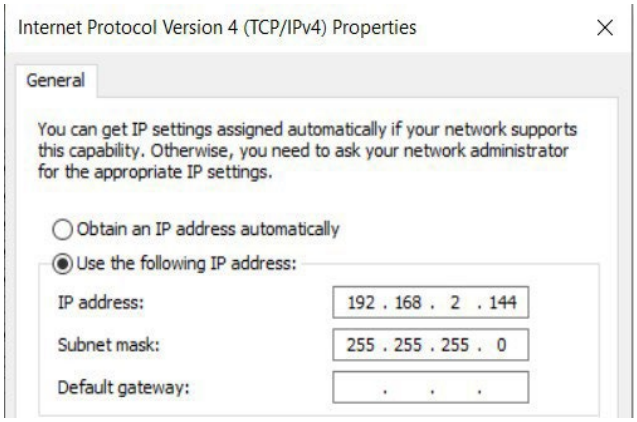
3. Access IPv4 Settings

- In the **Ethernet Properties** window, find and **double-click on "Internet Protocol Version 4 (TCP/IPv4)"**.



4. Assign a Static IP Address

- Select **"Use the following IP address"** and enter an IP similar to the example below:
    - **IP Address:** 192.168.2.xxx (Replace xxx with any number between **1-255**, ensuring it does not match the processor's last section.)
    - **Subnet Mask:** 255.255.255.0
    - **Default Gateway:** *(Leave blank or match your network settings if required.)*



💡 **Note:** If your PC and the processor are on different subnets, refer to **Page 21** for additional subnet matching details.

Logging in for the First Time

To access the X400 Vision Processor interface, follow these steps:

1. **Open an internet browser** (Chrome, Edge, etc.).

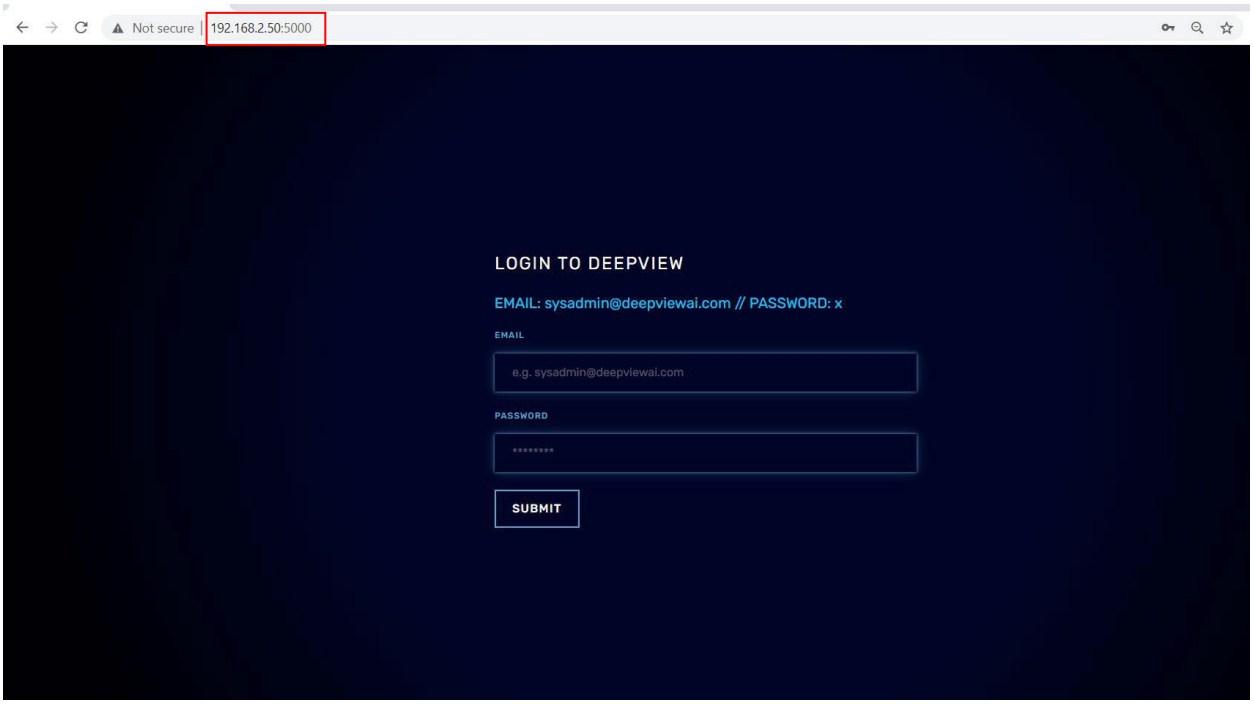
2. In the address bar, enter the processor's IP address in the following format: **192.168.2.xx:5000**

   Example: If your processor's IP address is 192.168.2.50, type: **192.168.2.50:5000**

3. Press Enter to load the login page.

4. Use the default credentials to log in:
   - Username: **sysadmin@deepviewai.com**
   - Password: **x**

5. Click Submit to access the Deepview dashboard.



💡 **Tip:** If you cannot connect, verify that your PC's **static IP is correctly configured** (see "Setting a Static IP Address" section).

User Interface Overview

Once logged in, you will be presented with the **Deepview AI Camera interface**, which is divided into two main sections:

1. Training Mode
Used for **creating applications**, **uploading and labeling images**, and **training neural networks**.

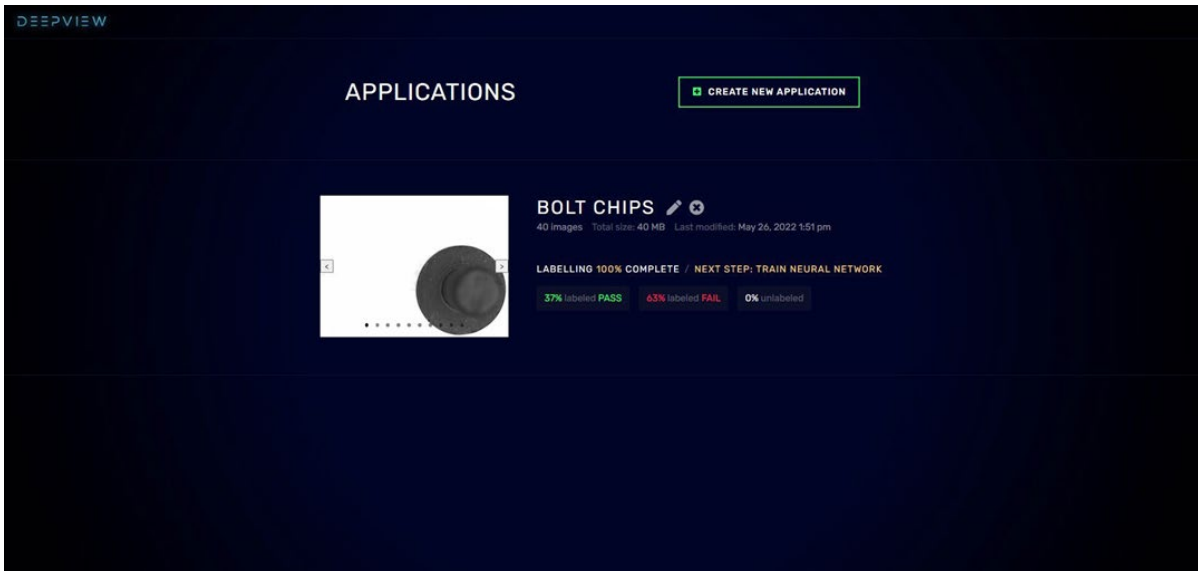
2. Camera Mode
Used for **capturing images**, **viewing history**, and **running production tasks**.

You can easily switch between Training and Camera modes by clicking the Deepview logo located in the top left corner of the interface.

💡 **Tip:** Return to this page at any time by clicking the Deepview logo at the top left corner of the screen.

Training Page

The **Training Page** is where you create applications, upload and label images, and train neural networks for object detection and classification.



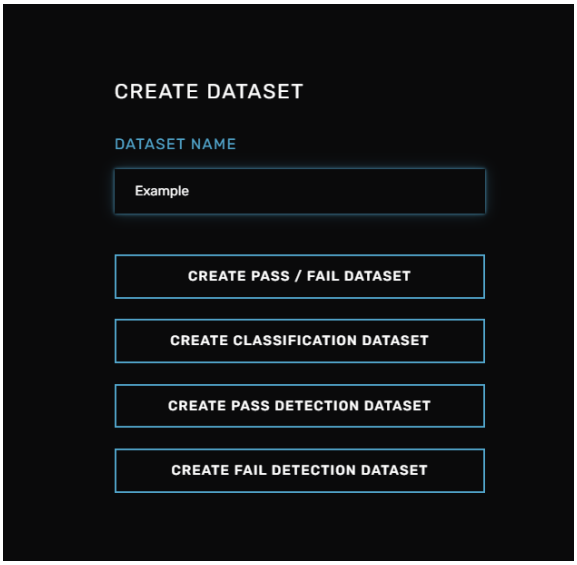
Creating a New Dataset

To create a new dataset:

1. Click **Create New Application**.
2. Enter a **dataset name**.
3. Select a **dataset type** from the following options:

Dataset Types

1. **Pass/Fail:** A simple dataset where each image is either a **pass** or **fail** (no detection boxes).
2. **Classification:** Used to sort multiple **object classes** within images (**no bounding boxes**).
3. **Pass Detection:** Detects the **presence of an object**, using a **GREEN** bounding box to highlight valid regions.
4. **Fail Detection:** Identifies **defects or anomalies**, using a **RED** bounding box to highlight failing areas.



Uploading and Labeling Images

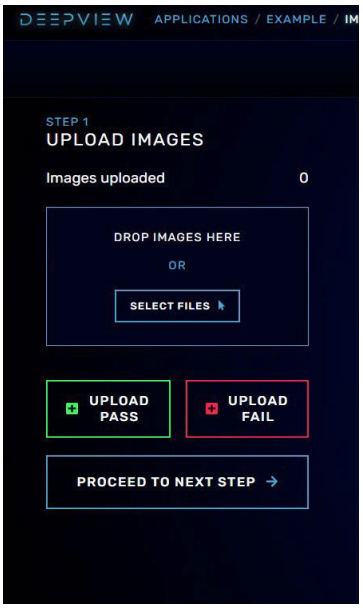
Uploading Images

You can add images in three ways:

5. **File Upload:** Click **Select Files** to upload images from your computer.

6. **Pre-Labeled Upload:** If your images are already sorted, use **Upload Pass** or **Upload Fail** buttons.

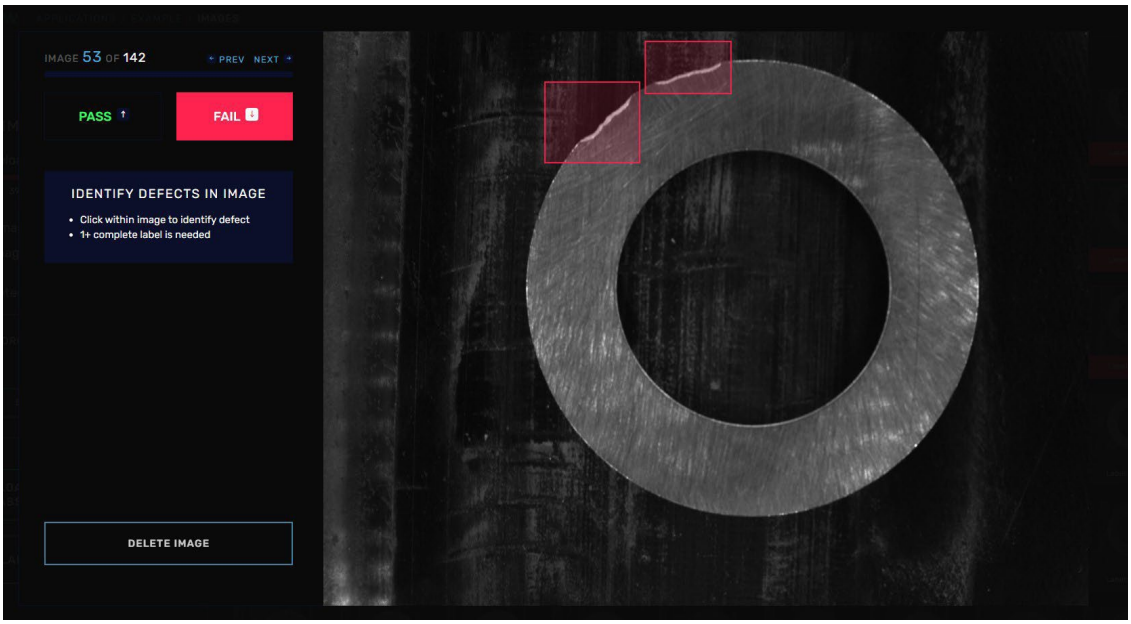
7. **Camera History:** Import images directly from the **camera's history** (see page 10 for details).



Labeling Images

If images are not pre-labeled, you must manually **review and label** them

- Select an image, then click **Pass** or **Fail** (or use the **Up/Down arrow keys**).
- For **Fail Detection datasets, draw a bounding box** around the defect areas (highlighted in red).



Training a Network

Uploading Images

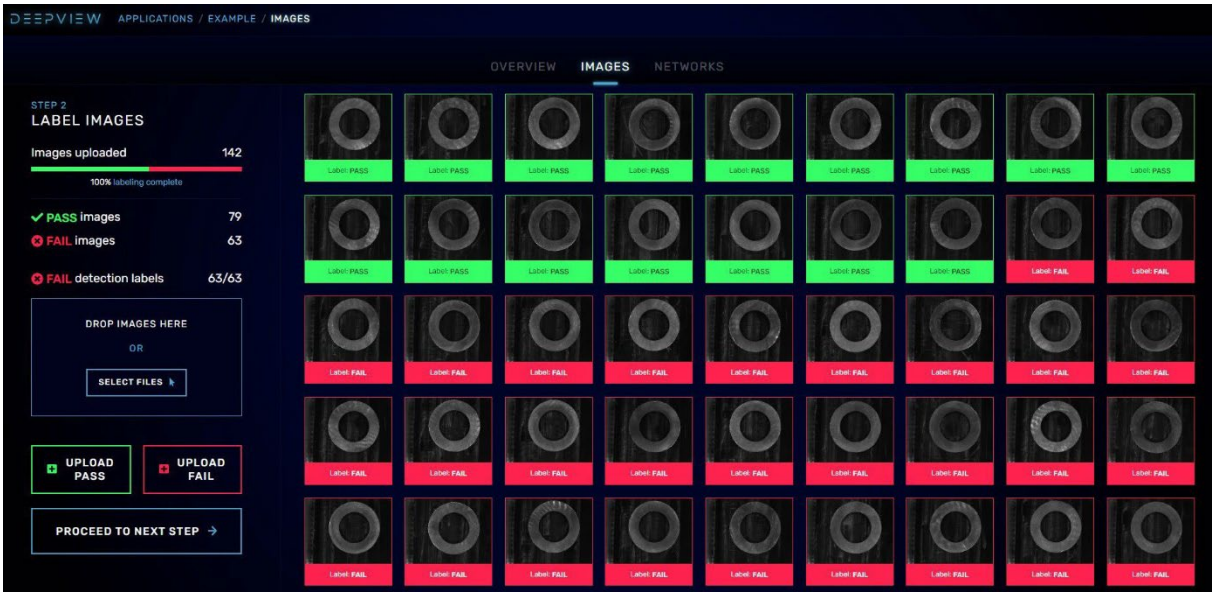
Once all images are **labeled**, you can **train a network** to recognize patterns and make predictions.

Finalizing Image Labels

1. After uploading, ensure **each image is labeled correctly** as either **Pass** (green) or **Fail** (red).
2. If using a **Fail Detection** dataset, verify **that bounding boxes** are accurately placed around defective areas.

Starting the Training Process

1. Click **Proceed to Next Step** in the bottom right.
2. Adjust training settings if needed (default settings are recommended for most cases).
3. Click **Start Training** in the top right to begin the process.



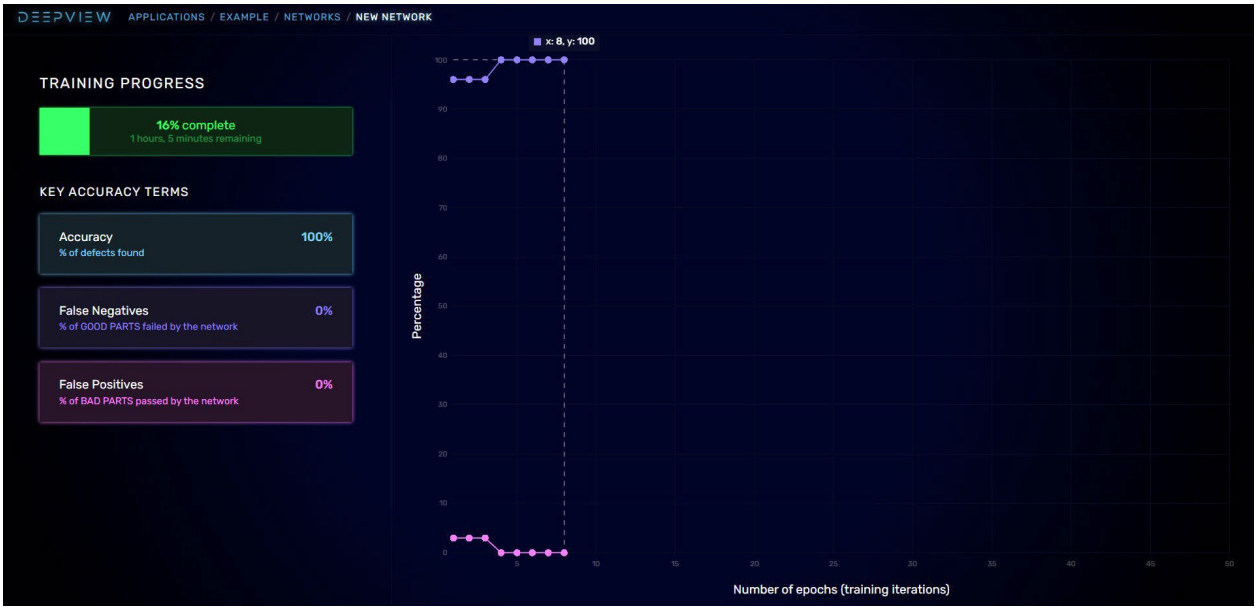
Training Time and Progress

Training can take anywhere from **5 minutes to 2+ hours**, depending on the **number of images and complexity** of the dataset.

The **Training Progress screen** will display:

- **Accuracy**: Percentage of correctly classified images.
- **False Negatives**: % of defects missed.
- **False Positives**: % of good parts incorrectly flagged as defective.

Once training is complete, the network is ready for **deployment and testing**.



Deploying the Network

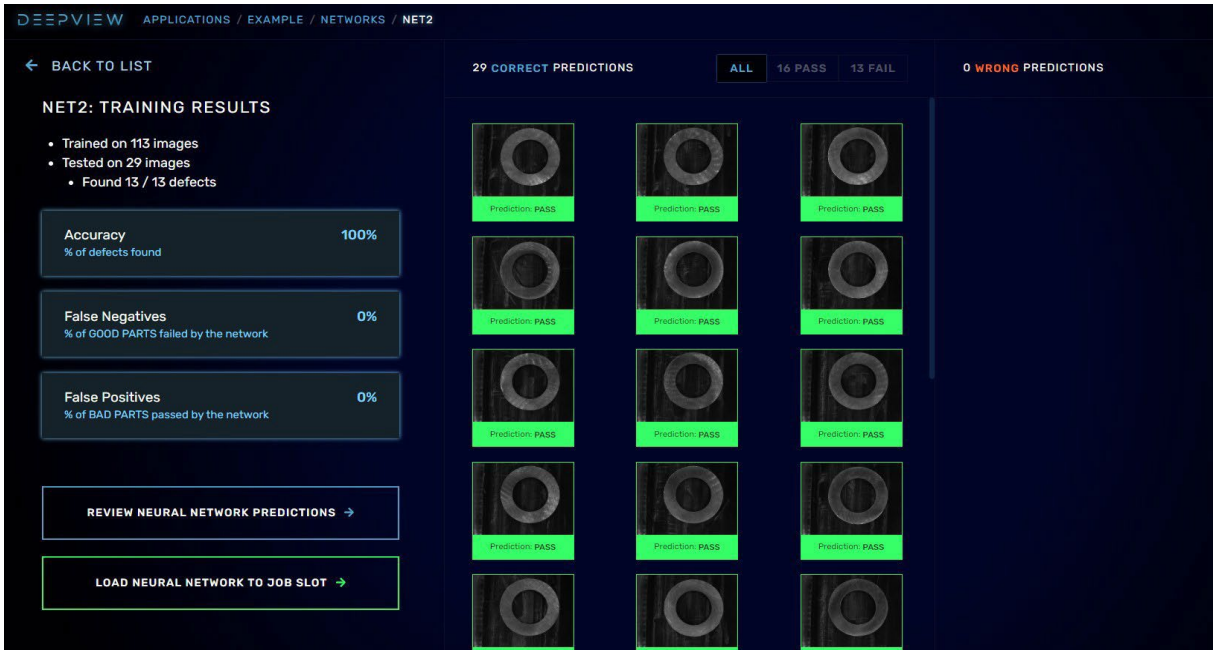
Once training is complete, the system will automatically display the Networks Screen, where you can review the camera's predictions and training accuracy.

Reviewing Training Results

The network will display key metrics such as:

- **Accuracy:** The percentage of defects correctly identified.
- **False Negatives:** The percentage of defects **missed** by the network.
- **False Positives:** The percentage of **good parts** incorrectly classified as defects.

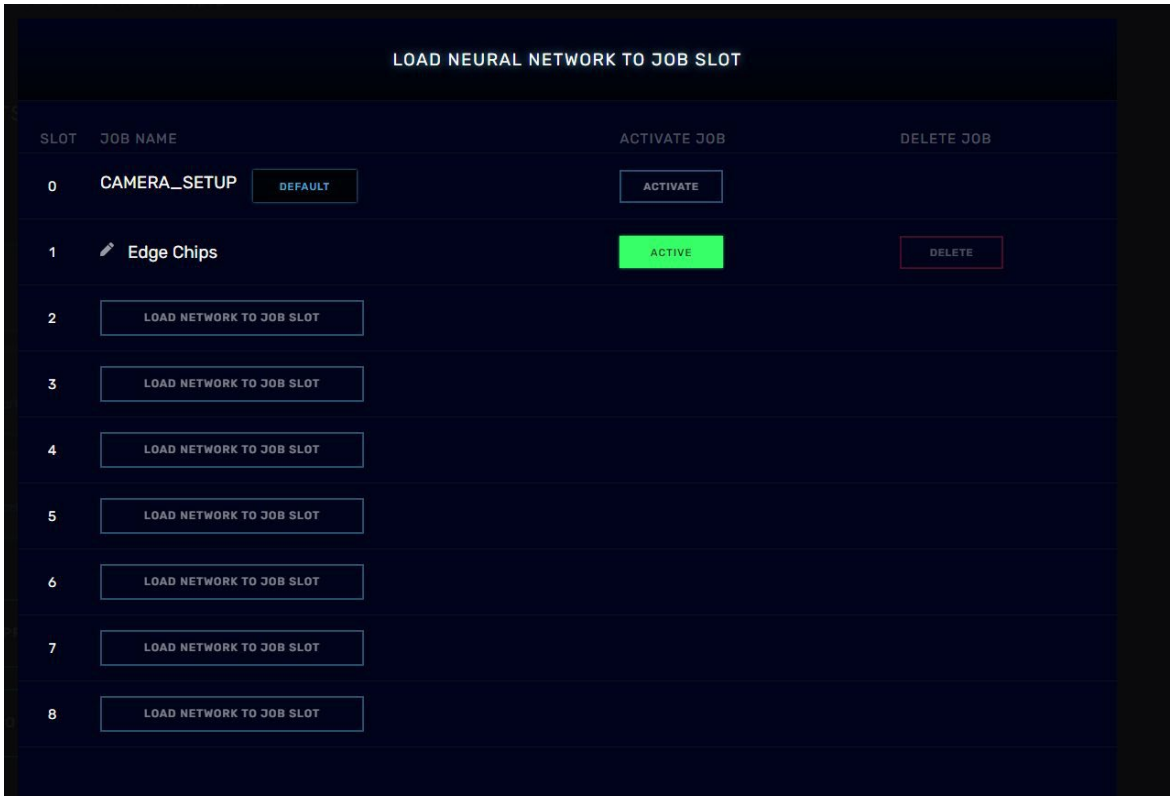
If the results meet your expectations, proceed to deploy the network.



Deploying the Network to a Job Slot

1. Click **Load Neural Network to Job Slot** (bottom left of the screen).
2. Select one of the **eight available job slots** and click **Load Network to Job Slot.**
3. Once loaded, click **Activate** to deploy the model for live production.
4. (Optional) Rename the job slot for easier identification.

Once activated, the **neural network will now be running in production mode** on the camera screen, enabling real-time inference.



Camera Screen Overview

The Camera Screen is where you can capture images, view history, and run production.

Navigation Bar Icons

The navigation bar contains four icons, each corresponding to a different screen:

Camera Icon (Main Camera Screen)

- Displays the live feed and allows image capture.
- This is the default screen when the camera starts.

History Icon (Image History)

- Opens a screen displaying previously captured images.
- Images are organized by network and timestamp for easy reference.

Gear Icon (Settings)

- Opens the Settings Screen where you can adjust:
  - Image exposure settings
  - Communication & I/O configurations
  - Job selection and changes

Person Icon (Info Screen)

- Displays camera system information, including:
  - Storage details
  - IP address and network settings

Additional Camera Features

- **Trigger Modes**: Switch between **Manual Trigger** and **Auto Trigger.**
- **Live Video Toggle:** Enables or disables live video feed for real-time monitoring.
- **Current Job Display:** Shows the **active job slot** running on the camera.

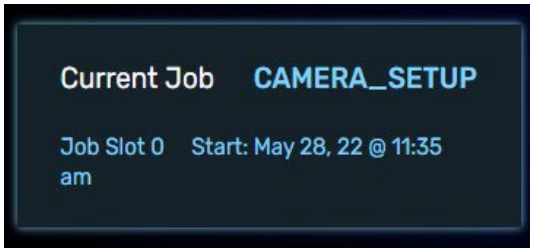
The **Main Camera Screen** allows you to capture and view images live as the camera runs in production mode.

Trigger Modes

The camera has **two trigger modes**, located in the top left corner:

1. **Manual Trigger**
    - Requires the user to click "**Capture Image**" or press the **space bar** for each capture.
2. **Auto Trigger**
    - The camera captures images **automatically** based on settings.
    - It can be triggered at a **set interval** or via a **PLC or other I/O device**.
    - **Important**: The camera must be in **Auto Trigger mode** to receive external triggers.

Current Job Display



The **Current Job** is displayed in the bottom left corner.

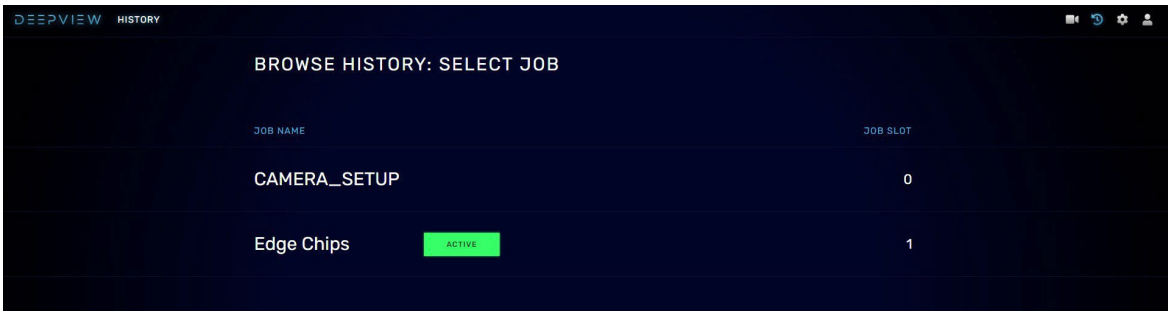
- By default, the job is **CAMERA_SETUP**, which is used for capturing images before training.
- The **Current Job can be change**d in the **Settings Screen** (refer to **Page 18**).
- If following the manual step by step, the **job you trained earlier will now be active** instead of CAMERA_SETUP.

Recent Image History

The **right-hand side** of the screen displays a column of **recently captured images and predictions.**

- These images are automatically **saved to the camera's history**, where they can be accessed anytime.

History Screen Overview



The **History Screen** allows you to view previously captured images along with their prediction results.

Organizing and Accessing Images

- Images are **automatically organized** in folders **by job** and **by date**.
- Select a **job folder** to view its associated images.
- Click on an image to **open it in full screen**, where you can see:
  - **Timestamp**
  - **Prediction confidence**

Downloading and Reusing Images

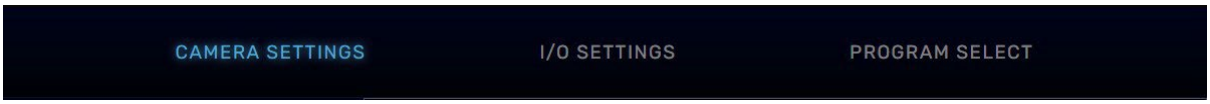
- You can **download images** to your PC for further analysis.
- Images can be **copied back into a Training Dataset** for retraining.

This screen helps in **monitoring network performance**, identifying **errors**, and improving **dataset quality**.

Settings Screen Overview

The **Settings Screen** is divided into **three sections**, accessible via the navigation bar at the top:

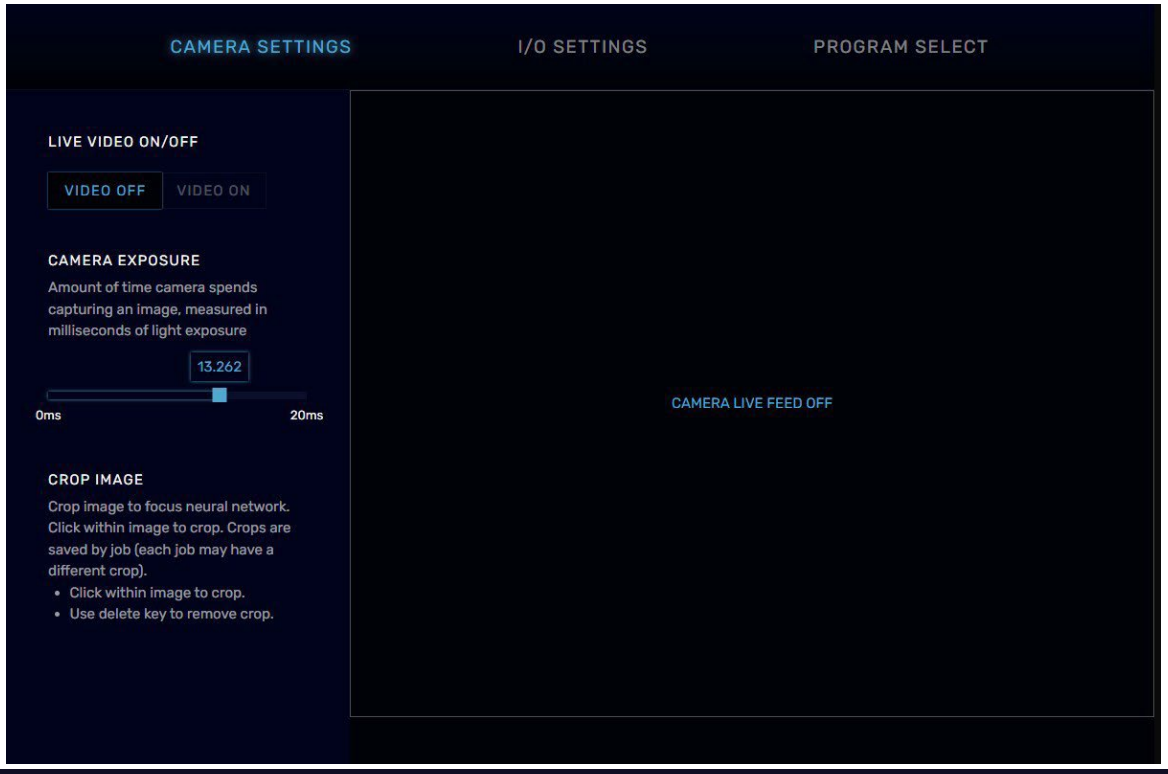
1. **Camera Settings** – Adjust exposure and enable/disable live video.
2. **I/O Settings** – Configure input/output communication.
3. **Program Select** – Manage and switch between job configurations.



Camera Settings

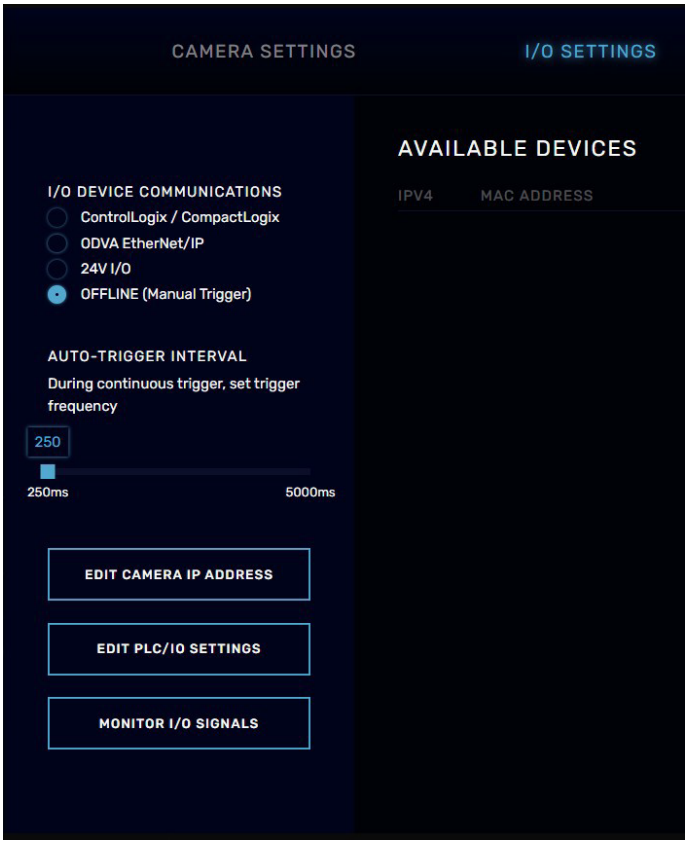
- **Live Video Toggle**: Enable or disable live video to assist with framing and setup.
- **Camera Exposure**: Adjust the exposure time (in milliseconds) using the slider bar.
- **Crop Image**: Crop the image to focus the neural network on a specific area.
    - **Click** within the image to create a crop.
    - **Press Delete** to remove a crop.

  Live video and exposure adjustments help ensure **optimal image capture conditions** for training and production.



I/O Settings

The **I/O Settings** section allows you to configure input/output communication for the Deepview camera.



Selecting an I/O Type

You can choose from **four I/O communication options**, located in the **top left**:

1. **Control/CompactLogix** – Use this option if you are connecting to a **Rockwell PLC**.
2. **ODVA Ethernet/IP** – A **generic Ethernet/IP option** for non-Rockwell devices.
3. **24V I/O** – A **hard-wired I/O** option with **4 inputs and 6 outputs** (see pinout on **Page 3**).
4. **Offline Mode** – The default setting on a new camera. When in **Auto Trigger mode**, the camera will **continuously capture** at the interval set by the slider.

Configuring Communication Settings

- Click **"Edit Camera IP Address"** to modify the camera's network settings.
- Click **"Edit PLC/I/O Settings"** to configure communication with a PLC or other devices.
- Click **"Monitor I/O Signals"** to view real-time I/O activity.

💡 **Note:** If you **change the camera's IP address**, you will **immediately lose connection**. To reconnect, **enter the new IP address** into your browser's address bar.

Program Select

The **Program Select** screen allows you to manually change the **active job** running on the camera.

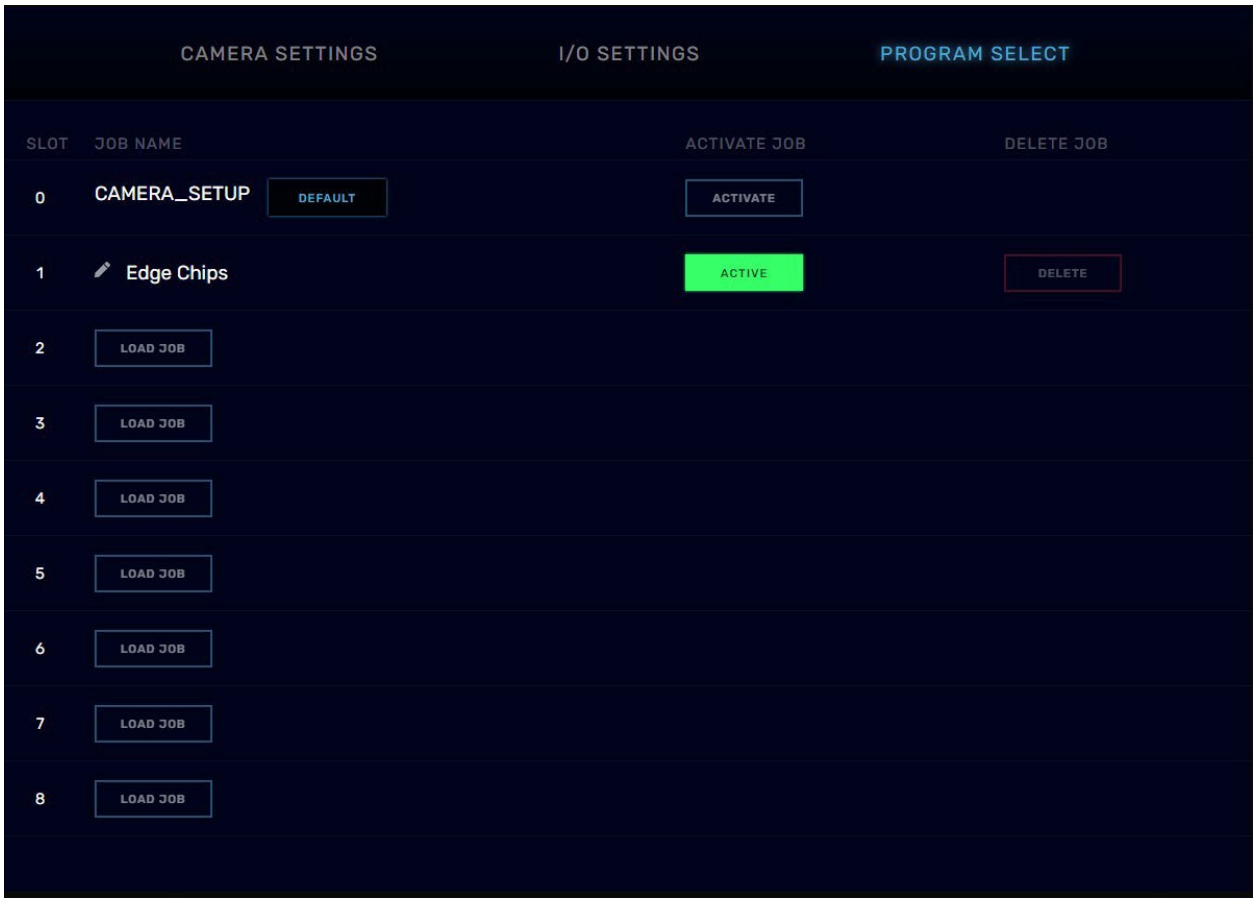
Changing the Active Job

- Click **Activate** on the desired job to set it as the active job.
- You can also **rename jobs** from this screen for better organization.

Loading Job Files

- Jobs can be loaded from **.job files**, which can be downloaded from the **Networks Screen** within Training (see **Page 13**).

Using **Program Select**, you can efficiently switch between different inspection tasks without retraining the network.



Rockwell Ethernet/IP Setup

To establish communication between the **Deepview Camera** and a **PLC**, follow these steps:

Verify the PLC and Camera IP Addresses

- Determine the **IP address of the PLC**.
- The **first three sections** of the camera and PLC IP addresses **must match** for proper communication.

Example of Matching Subnets:

✅ **Camera IP:** 192.168.2.50

✅ **PLC IP:** 192.168.2.10 (Matching subnet: 192.168.2.xxx)

Example of Mismatched Subnets:

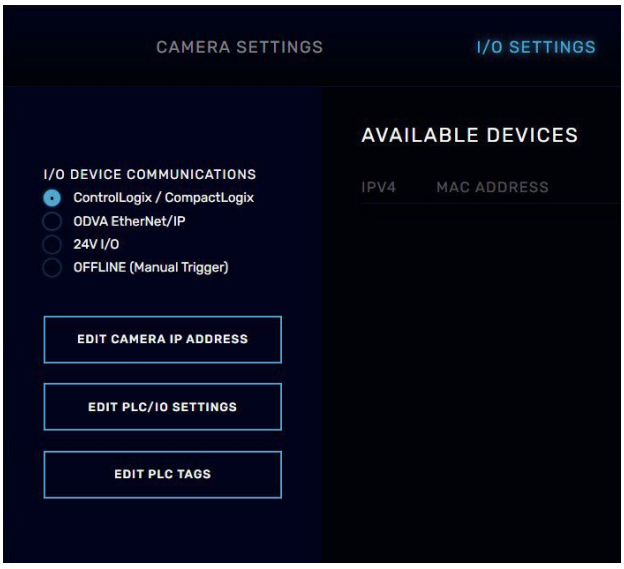
❌ **Camera IP:** 192.168.2.50

❌ **PLC IP:** 192.168.1.10 (Different subnet: 192.168.1.xxx)

If the **subnets do not match**, you must **change the camera's IP address**.

Changing the Camera's IP Address

1. Click the **Settings Icon** (⚙️) in the **top right corner** of the UI.
2. Select **I/O Settings**.
3. Click **Edit Camera IP Address**.
4. Enter the **new static IP and subnet**, ensuring it aligns with the PLC's network.



Connect to the PLC

1. In the **I/O Settings** menu, click **Edit PLC/I/O Settings**.
2. Enter the **PLC's IP address**.

The **camera** will now establish **communication with the PLC**.

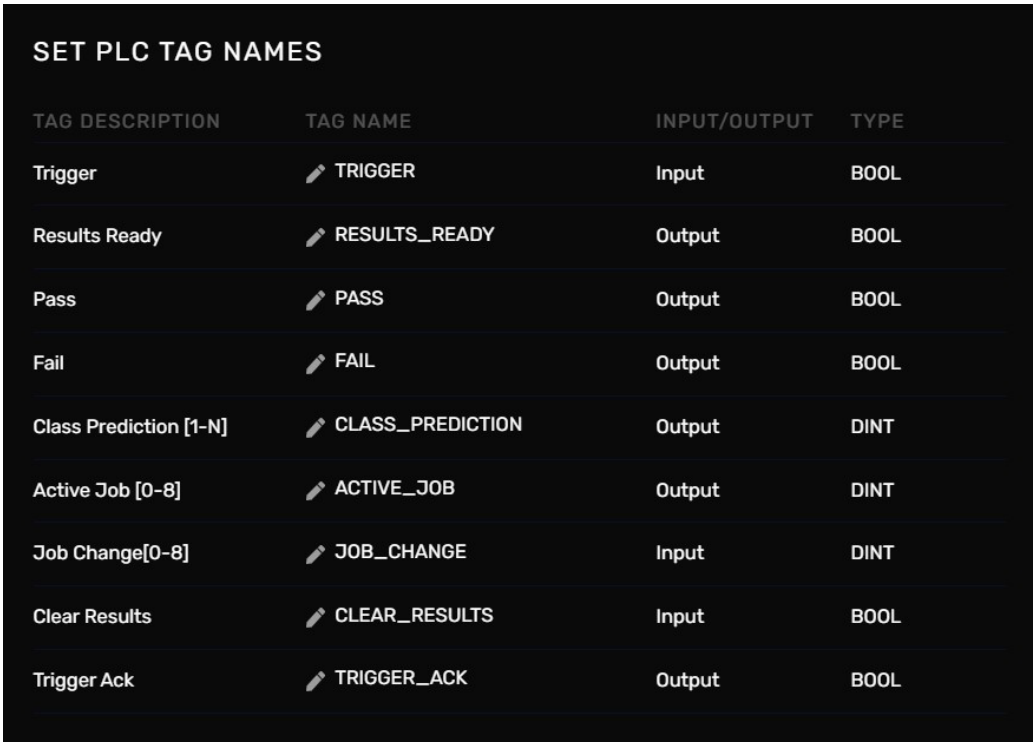
💡 **Note: Write down the new IP address** before applying changes. Changing the subnet will **disconnect the camera**. You must **update your PC's subnet** to match the camera's new network (see **Page 5** for instructions)

Configuring PLC Integration

Once the **PLC IP address** is set, you can write programs using the **PLC tag names** in the table below.

- **Inputs** are signals **sent to the camera** from the PLC.
- **Outputs** are signals **sent from the camera** to the PLC.
- Tags must be configured within the **controller scope** in your PLC program.

You can **view and edit these tags** by clicking **EDIT PLC TAGS** in the I/O settings menu.



SET PLC TAG NAMES

TAG DESCRIPTION	TAG NAME	INPUT/OUTPUT	TYPE
Trigger	TRIGGER	Input	BOOL
Results Ready	RESULTS_READY	Output	BOOL
Pass	PASS	Output	BOOL
Fail	FAIL	Output	BOOL
Class Prediction [1-N]	CLASS_PREDICTION	Output	DINT
Active Job [0-8]	ACTIVE_JOB	Output	DINT
Job Change[0-8]	JOB_CHANGE	Input	DINT
Clear Results	CLEAR_RESULTS	Input	BOOL
Trigger Ack	TRIGGER_ACK	Output	BOOL

💡 **Note: Tag names in the PLC program must match** the table above for proper functionality. The camera **must be in Auto Trigger mode** to receive a trigger from the PLC (see **Page 15**).

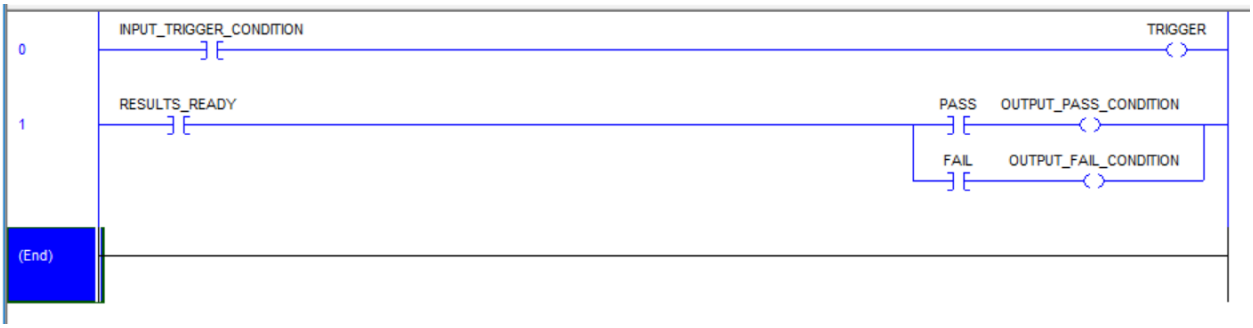
PLC Integration Example: Pass/Fail Job Configuration

Basic Pass/Fail Workflow

1. The **PLC sets INPUT_TRIGGER_CONDITION**, sending a **TRIGGER** signal to the camera.
2. The camera **captures an image** upon receiving the trigger.
3. The camera **analyzes the image** and determines whether it **passes or fails**.
4. The camera sets **RESULTS_READY** high, followed by either:
    - **PASS** output high (if the part is good).
    - **FAIL** output high (if the part is defective).

Handling Outputs

- The **OUTPUT_PASS** and **OUTPUT_FAIL_CONDITION** signals represent the camera's decision.
- If a part **fails**, OUTPUT_FAIL_CONDITION can be used to **trigger an alarm** or notify an operator.



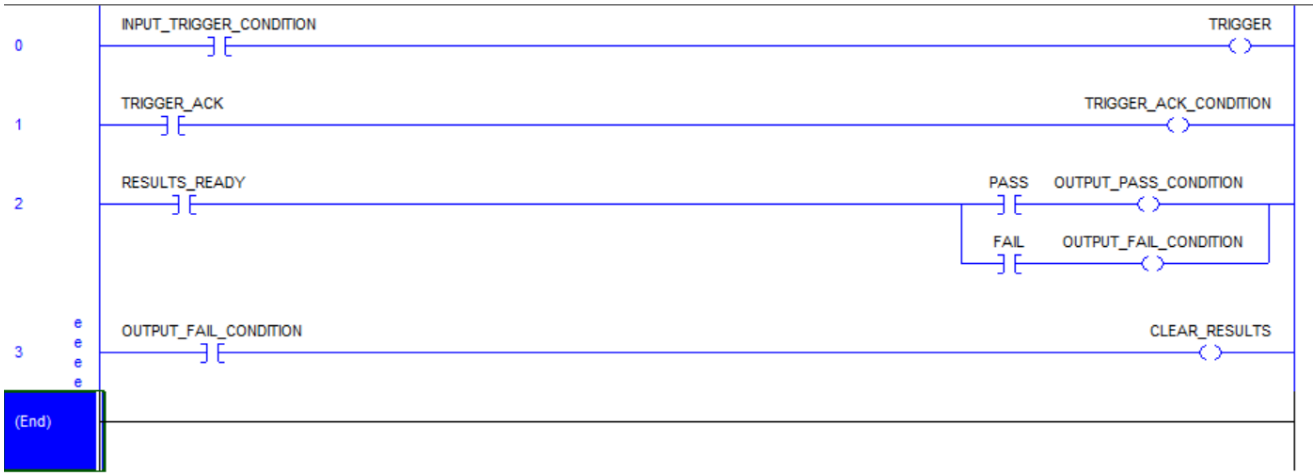
💡 **Note:** Each time **TRIGGER** is set high, **RESULTS_READY**, **PASS**, and **FAIL** are reset to low.

Advanced Example: Using TRIGGER_ACK and CLEAR_RESULTS

The advanced example includes **two additional optional tags**:

1. **TRIGGER_ACK** (Output)
    - Sent from the camera to confirm that it has received a **trigger signal** from the PLC.
    - It is **automatically set low** once the determination is made (**RESULTS_READY goes high**).
2. **CLEAR_RESULTS** (Input)
    - Used to **reset results** before the next trigger cycle.

These additional signals **enhance synchronization** between the PLC and the camera, ensuring smooth automation.



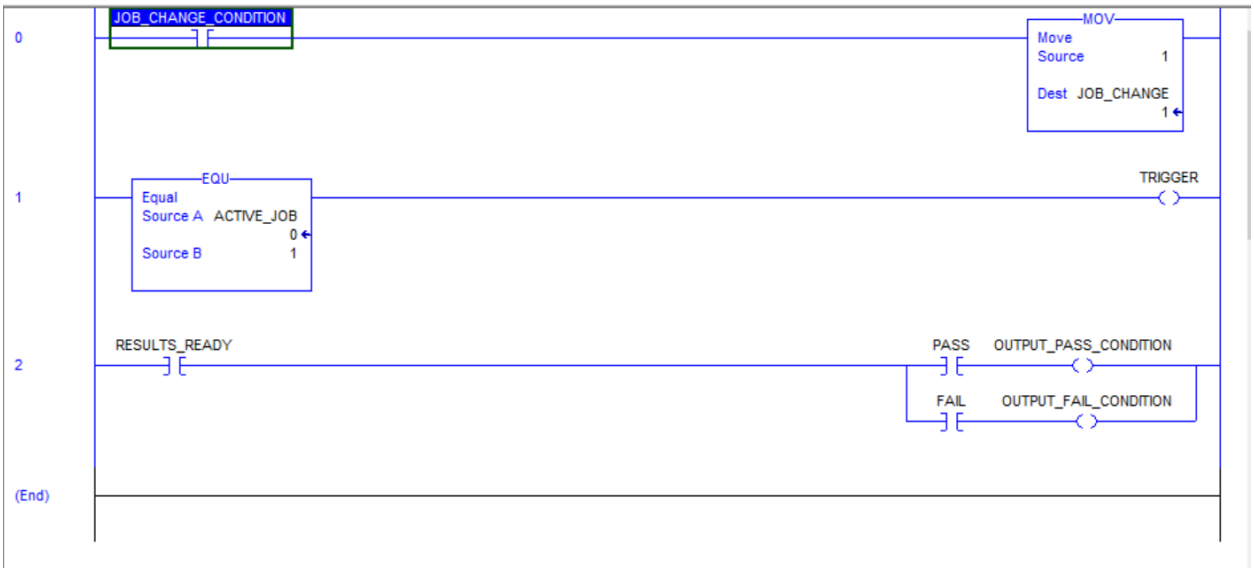
Changing Jobs over Ethernet/IP

The diagram below illustrates how to perform a job change using a PLC, allowing the system to:

- **Quickly switch between jobs** for different inspections.
- **Run multiple jobs consecutively** without manual intervention.

How It Works

1. When **JOB_CHANGE_CONDITION** is set high, the **MOV instruction updates JOB_CHANGE** to the specified job slot.
2. The camera updates the **ACTIVE_JOB** value.
3. The **EQU instruction** sets **TRIGGER** high when **ACTIVE_JOB = 1.**
4. The camera captures an image and evaluates the part based on the active job settings.



💡 **Note: This process can be repeated for up to 8 different job slots**, allowing flexible automation for different inspection requirements.

PLC Program Example

The diagram below illustrates a **PLC program designed to run a classification job**.

How It Works

- **CLASS_PREDICTION** is a **DINT (Double Integer)** set by the camera to indicate the predicted class.
- When **RESULTS_READY** is high and the corresponding **EQU instruction** is satisfied:
    - **OUTPUT_CLASS_CONDITION** is set high for the predicted class.

This allows the PLC to respond dynamically based on the camera's classification output, enabling automation for **sorting, verification, or process control** based on detected object classes.