Requirements

Process up to 400 caps/minute – <150ms processing time for top + bottom images (2 cameras)

Detect any defect that would be visible to the human eye -> **Tested and verified Resnet + Autoencoder is the best solution**

-Scrapes

-Major dents

-Misprinted logos

-Misaligned logos

-Any other major defects (poorly cut caps etc.)

Low cost

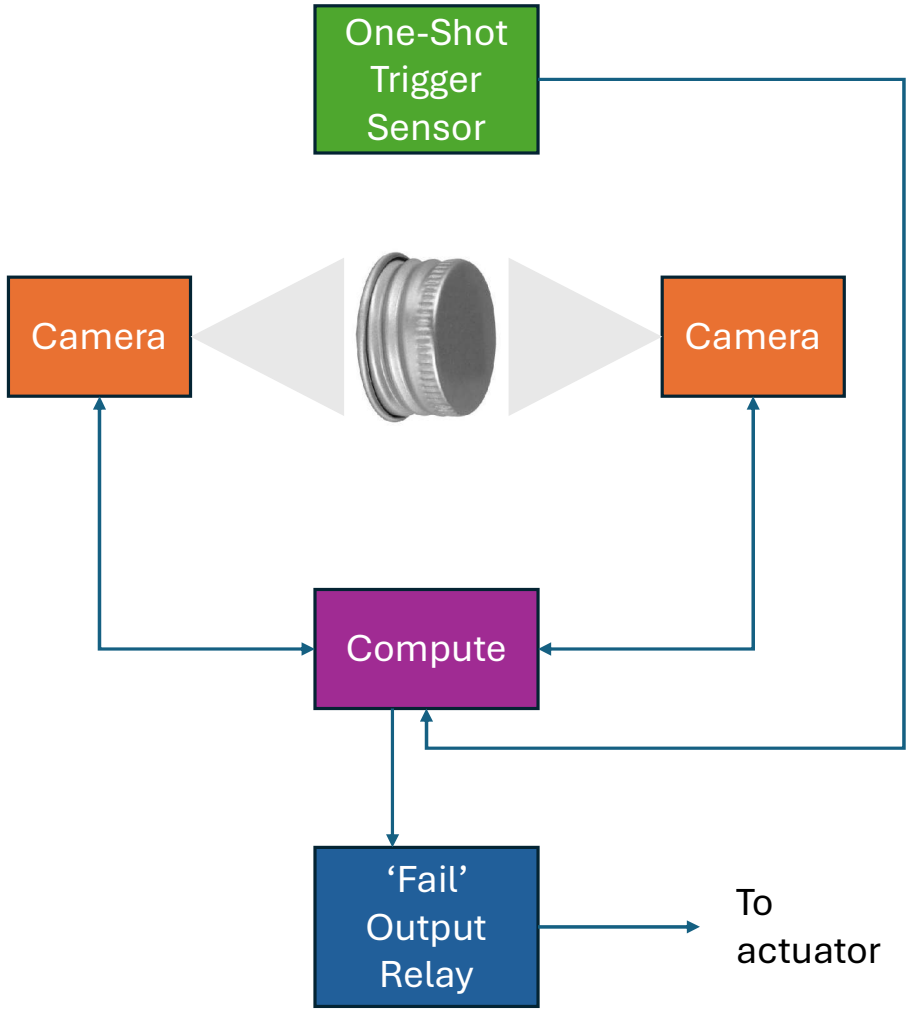
Easy to add new cap designs to the software as-needed

System Overview

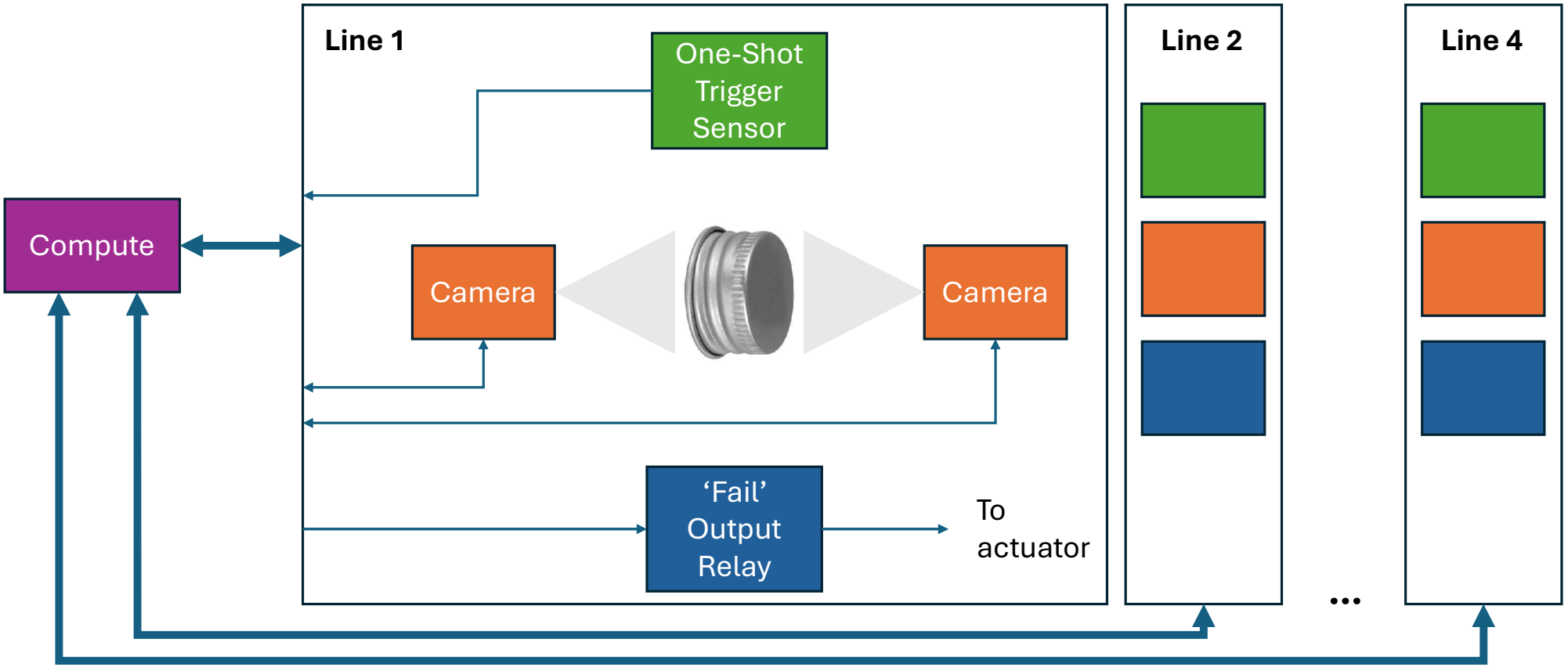
One-shot trigger:
Sensor to determine when the cap is stopped and ready for a frame capture. Computer measures sensor output and triggers a single frame capture on both cameras.

Fail Output Relay:
Relay to allow the computer to control an actuator that pushes defect caps off the line.

Cameras:
Hardware triggered frame output is a must. Polling frames from the camera would require an extremely high frame rate to maintain an acceptable worst-case delay.



Compute - Centralized

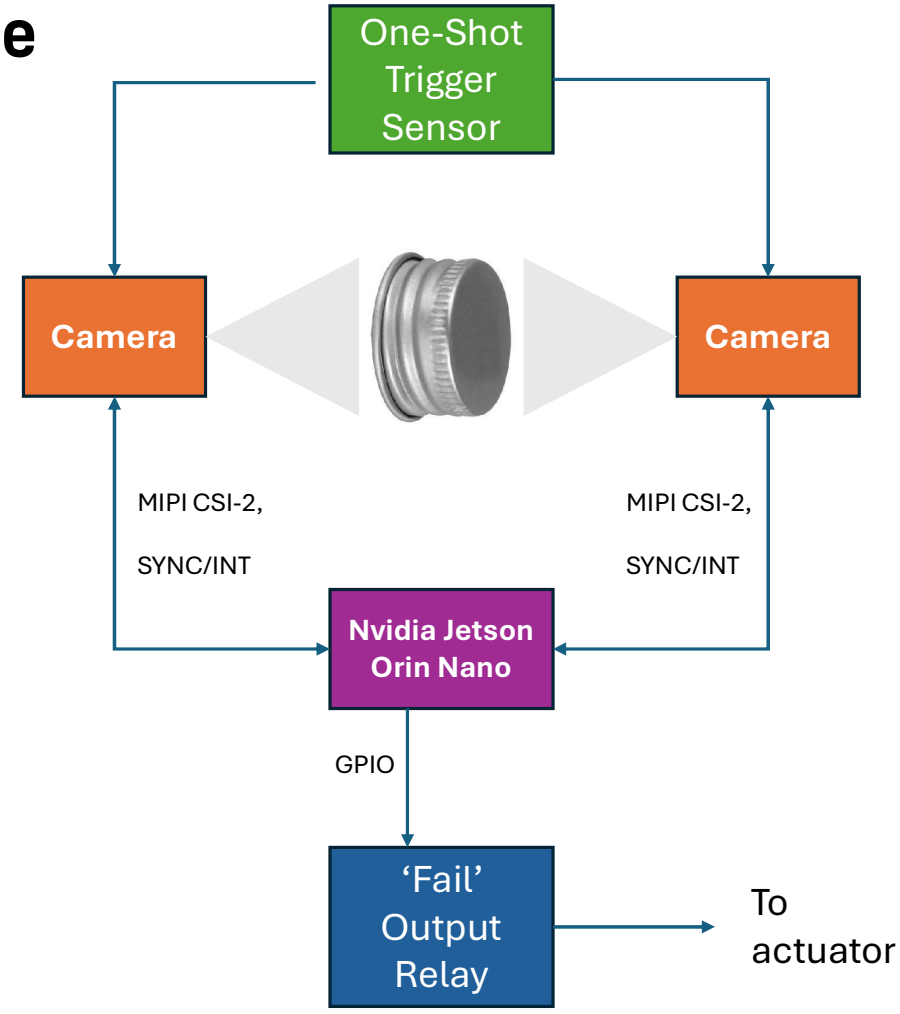


Compute - Centralized

- Expensive
  - $2-3k on a high-end desktop PC with GPU
  - multi-port ethernet NIC,
  - ethernet/USB GPIO expander for relay control
  - Gigabit Ethernet cameras – tend to be higher cost
- Power hungry – 500-1000W
- Too much coupled software and hardware makes development and testing difficult
- Camera trigger and frame latency will be high, reducing maximum potential throughput
- **Bad idea.**

Per-Line Embedded Compute

- ~$300 for embedded compute on each assembly line
- Capable of low latency capture and analysis (10's of ms)
- Each assembly line functions independently with its own system
- Software is relatively easy – very little CPU involvement (mostly AI accelerator workload)
- With the right trigger circuit, cameras can be triggered directly for ultra-low latency. After completing frame readout, a SYNC/INT signal can trigger DMA interrupt for frame buffering. No CPU involvement.
- **Good solution**



Compute Comparison					
Feature	**Jetson Orin Nano**	**Jetson Nano**	**BeagleBone Black**	**Raspberry Pi 4 Model B**	**Google Coral Dev Board**
Processor	6-core ARM Cortex-A78AE	Quad-core ARM Cortex-A57	1-core ARM Cortex-A8	Quad-core ARM Cortex-A72	Quad-core ARM Cortex-A53
GPU	1024-core NVIDIA Ampere (6.0 TFLOPs)	128-core Maxwell (472 GFLOPs)	None	Broadcom VideoCore VI (500 MHz)	Integrated Edge TPU (4 TOPS)
AI Accelerator	**NVIDIA Tensor Cores (32 TOPS AI)**	None (GPU-only inference)	None	None	Edge TPU (Tensor Processing Unit)
Memory (RAM)	8 GB LPDDR5	4 GB LPDDR4	512 MB DDR3	2/4/8 GB LPDDR4	1 GB LPDDR4
Camera Interface	2x MIPI CSI-2	2x MIPI CSI-2	None (via USB only)	2x MIPI CSI-2 (Raspberry Pi CSI)	1x MIPI-CSI and 1x MIPI-DSI
AI/ML Frameworks	TensorFlow, PyTorch, ONNX, NVIDIA SDK	TensorFlow, PyTorch, TensorRT	Limited AI support	TensorFlow Lite, PyTorch	TensorFlow Lite
Power Consumption	7–15W	5–10W	0.5W–2W	3–7W	2–4W
Cost	$249	$99 (4 GB)	$55	$35–$75	$130

- Coral dev board limited to one camera per board – only one CSI interface and limited RAM for frame buffering
- Beaglebone is a non-starter – will take seconds to process a single frame using the CPU for inference
- Pi has no AI hardware acceleration. Cannot use GPU for AI inference, it only supports OpenGL. Will take seconds to process a frame.
- Jetson Nano will struggle to process 2 simultaneous image streams, but may work with a single cam stream with some optimization.
- **Orin Nano can easily perform control and inference of a two-camera system at a frame rate of 100+ fps.**

Camera Selection

Feature	MIPI CSI-2	USB Cameras	GigE Cameras
Interface	Direct SoC connection (CSI lanes).	External USB 3.0/3.2.	Ethernet (1 Gbps, PoE).
Bandwidth	4-12 Gbps.	5–10 Gbps.	1 Gbps per camera.
Latency	Low (hardware interface).	Moderate (USB polling).	Higher (network overhead).
Trigger Support	Hardware-level GPIO.	Software-based commands.	Hardware/software support.
Synchronization	Precise (hardware sync).	Limited (software jitter).	Good (PTP hardware sync).
Real-Time Performance	Excellent.	Good but less precise.	Moderate (buffered).
Cable Length	Short (<30 cm).	Short (3–5 m).	Long (up to 100 m).
Cost	Low–Moderate ($50–$200).	Moderate ($100–$500).	High ($300–$1,000+).
Applications	AI, robotics, vision.	Prototyping, imaging.	Industrial, multi-camera.

If you can accommodate a short 15-30cm flex cable in the system, CSI-2 is by far the most reliable and highest performing interface.

Camera Requirements

- Proof-of-concept done on 224x224 images from laptop webcam (cropped region of interest)
- 900x900 should be more than enough to detect even small defects
- 2MP camera should be sufficient – must have a hardware trigger input and ideally should have a VSYNC/HSYNC output to interrupt CPU
- Purchase some cheap M12 lenses with varying FOV to allow flexibility in camera position and maximize pixel density in the region of interest
- Use a global shutter if you plan on capturing the caps as they're actively moving across the camera field of view. Local shutter is fine if the caps are stopping in front of the camera and remaining static for a short time.

Cameras

https://www.e-consystems.com/nvidia-jetson-orin-nx-orin-nano-cameras.asp

Based in Chennai

FHD AR0234 Global Shutter Camera for NVIDIA® Jetson Orin NX / Orin Nano

Seems like an OK camera – should get datasheet for more info. Website download link is broken.