

## II. METHODS

### B. Machine Vision program development

The machine vision system serves as the primary sensor for the collaborative robot, enabling it to detect incoming sheet metal workpieces on the conveyor belt and extract their precise position and orientation. The software pipeline is structured to process raw camera frames dynamically and isolate reliable coordinates in real-time.

#### 1) Edge detection

The first objective of the vision pipeline is to isolate the workpiece from the background conveyor belt. A robust preprocessing sequence ( `src/vision/preprocess.py` ) is applied to each acquired frame to produce a clean binary mask:

- **Grayscale Conversion:** The 3-channel BGR image is converted to a single-channel grayscale image to reduce computational overhead.
- **Gaussian Blurring:** A Gaussian filter (e.g., 5x5 kernel) is applied to smooth out high-frequency noise and imperfections on the metal sheet's surface, ensuring stable edge extraction later.
- **Binary Thresholding:** A binary threshold is utilized to separate the lighter sheet metal from the darker background conveyor belt. Pixels above the given intensity threshold are set to maximum value (white, representing the object), while the background is set to zero (black).
- **Morphological Operations:** A morphological 'close' operation (dilation followed by erosion) fills in any small gaps, holes, or lighting reflections within the object's mask. This guarantees the workpiece is represented as a single contiguous block.

Following preprocessing, edge extraction is finalized by applying a contour-finding algorithm to the binary mask ( `src/vision/detection.py` ). This algorithm traces the external boundaries of the white regions in the mask, effectively extracting the continuous edges of the workpiece.

#### 2) Centroid identification

Once the contours are extracted, the system identifies the workpiece by intelligently filtering out noise. The detected contours are evaluated based on their enclosed area. The largest contour that exceeds a predefined minimum area threshold (e.g., 5000 square pixels) is selected as the primary workpiece. Any smaller contours, typically caused by lighting artifacts, scratches, or dust on the belt, are systematically discarded.

To pinpoint the exact center of the workpiece, a minimum-area bounding rectangle algorithm fits a rotated box tightly around the selected contour. This mathematical fitting returns the precise centroid coordinates  $(x, y)$  of the rectangle within the image plane. These centroid coordinates represent the exact center of the sheet metal in the camera's field of view, serving as the foundational reference point for the robot's inverse kinematics and path planning.

#### 3) Workpiece orientation calibration

In addition to the centroid, the minimum-area bounding rectangle inherently provides the rotational angle  $(\theta)$  of the workpiece. Depending on how the sheet metal was placed or whether it shifted while traveling on the conveyor belt, its orientation may deviate from the ideal, "approved" standard alignment.

The detection algorithm calculates the angle of the rectangle relative to the camera's image frame. To standardize the data for the robot controller, the angle is normalized such that a perfectly aligned sheet returns  $0^\circ$ , and deviations are expressed as relative degrees. By determining this angular deviation in

real-time, the vision system informs the collaborative robot of the exact orientation of the workpiece. This allows the cobot to dynamically rotate its end-effector to match the workpiece's physical posture before engaging the magnetic gripper, ensuring a secure and accurate pick without requiring the conveyor to halt.

#### 4) Camera calibration and coordinate mapping

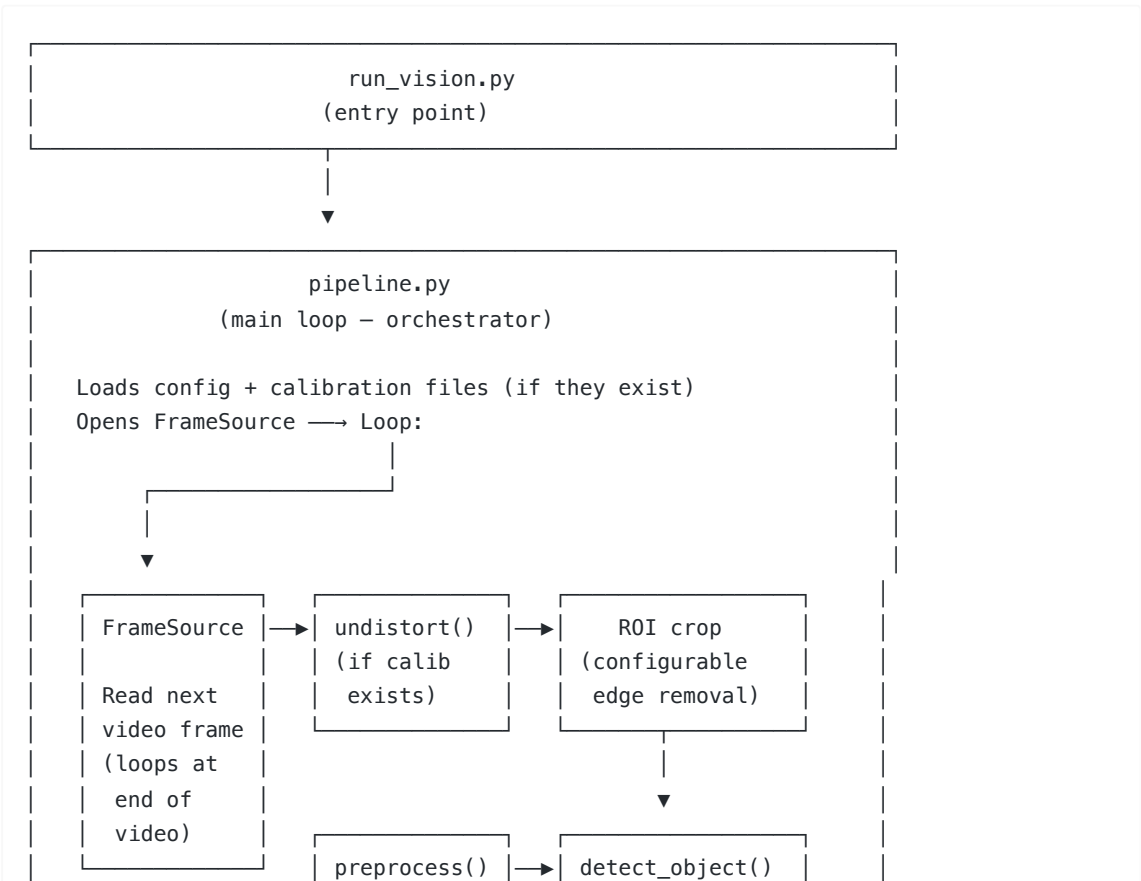
While edge detection and centroid identification operate in the pixel domain of the camera's image sensor, the robot controller requires coordinates in real-world dimensions (millimeters) to execute path planning. To bridge this gap, two critical transformations are integrated into the vision pipeline:

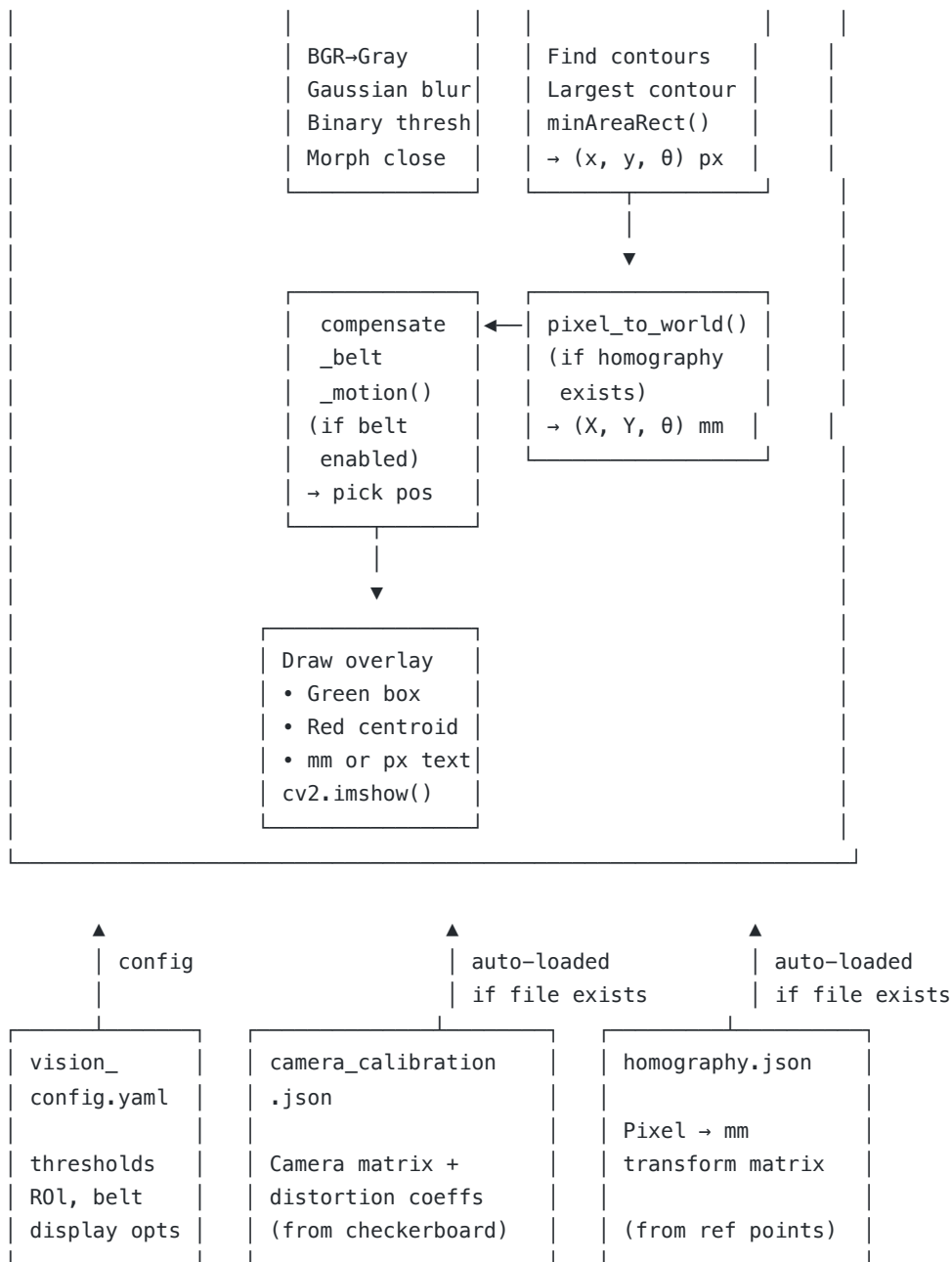
First, **intrinsic camera calibration** is performed to eliminate lens distortion. The system's wide-angle lens inherently introduces barrel or pincushion distortion, leading to inaccurate pixel coordinates near the edges of the frame. By capturing multiple images of a known checkerboard pattern, the system computes the camera matrix and distortion coefficients. The algorithm applies these parameters ( `cv2.undistort` ) to mathematically flatten every incoming frame before object detection begins.

Second, a **homography transformation** is employed to map the undistorted 2D pixel coordinates to real-world 2D coordinates on the surface of the conveyor belt. By placing markers at known physical distances on the belt and recording their corresponding pixel coordinates, a 3x3 transformation matrix is generated. During real-time operation, the pipeline multiplies the detected workpiece centroid  $(x, y)$  in pixels by this homography matrix, outputting the precise  $(X, Y)$  position in millimeters relative to a defined origin point on the conveyor. This ensures the vision system and the cobot share a unified coordinate space.

#### Vision Pipeline Flow

To visually summarize the execution flow described above, Figure X illustrates the sequential operations performed on every incoming camera frame.





### ### Vision Pipeline Output Visualization

The side-by-side image below illustrates the result of the vision pipeline operating on a camera frame. On the right is the binary mask generated after grayscale conversion, Gaussian blur, thresholding, and morphological closing. On the left is the final output layer demonstrating the continuous edge detection and bounding box fitting algorithm.

The green rectangle represents the computed minimum-area rotated bounding box. At its center lies the calculated red centroid alongside the normalized rotation angle, actively predicting exactly how the collaborative robot must approach and rotate to retrieve the targeted metallic sheet.

```
![Vision Pipeline Detection and Mask]  
(/Users/zain/.gemini/antigravity/brain/cf73f009-3a05-4132-a74b-  
c8de63cdc12f/vision_result_demo.jpg)
```