AI Reliability in Industrial Safety

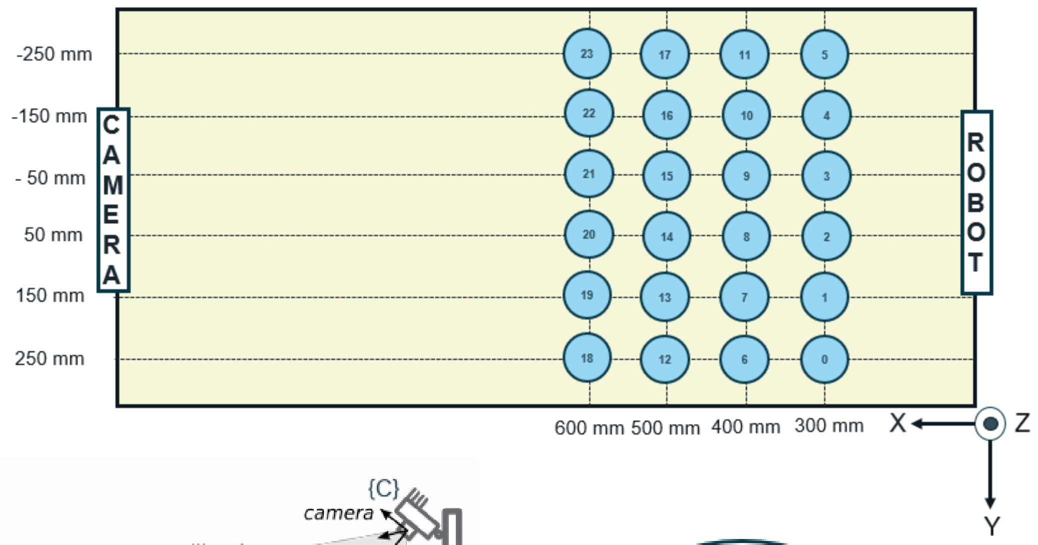
A Case Study with ABB IRB 1200 and Intel RealSense D415

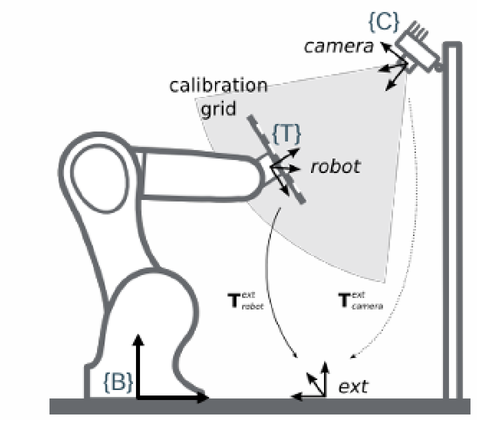
Supervisor: *Mathias Verbeke*

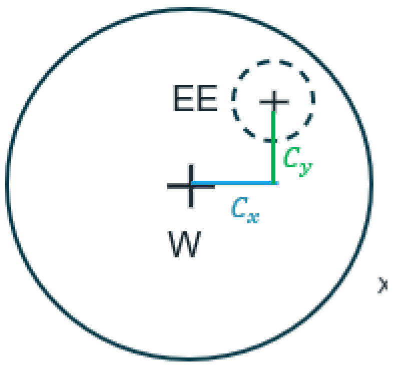
Supervisor: Matthias De Ryck

Iñigo Aduna Alonso
Academic year 2023 – 2024
Master of AI in Business and Industry

State of the Art [3]

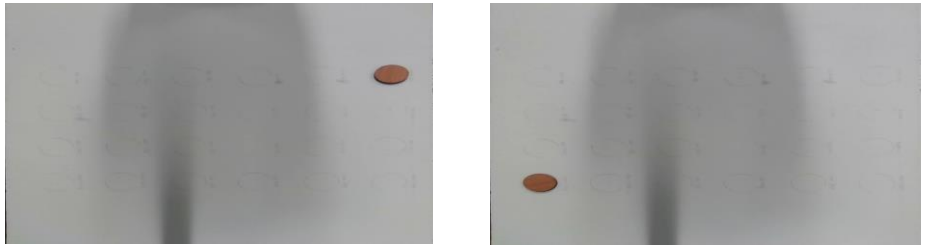






1.- Raw Images:

2.- Warped Images (Distortion correction + M):



3.- Masks

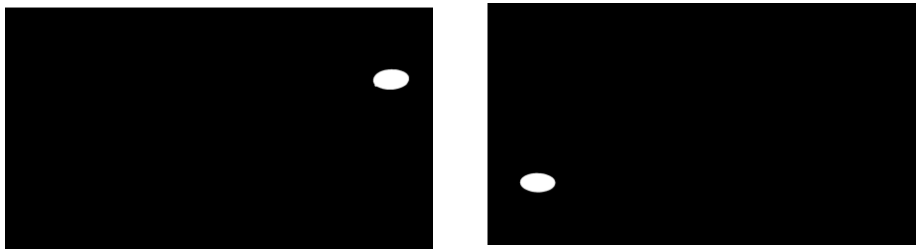
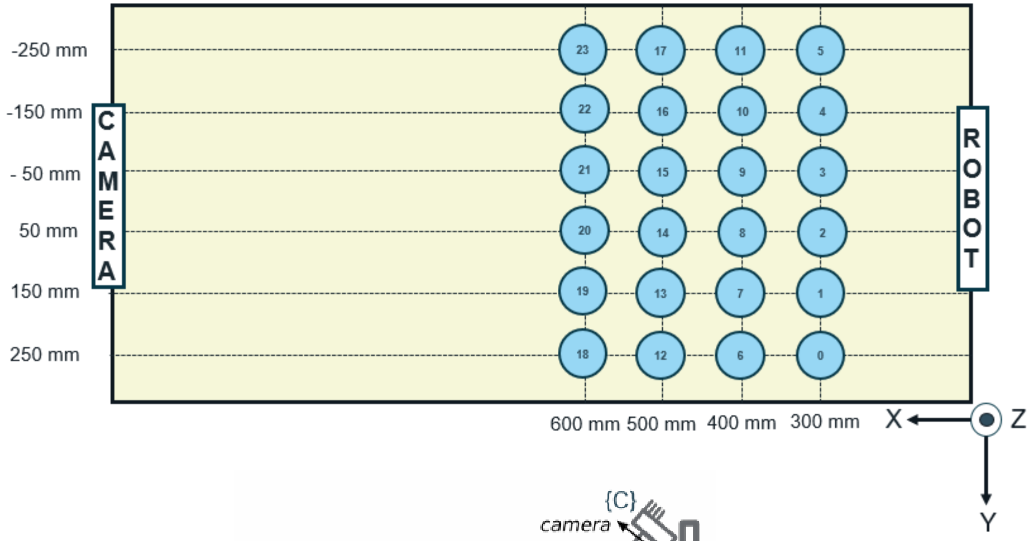
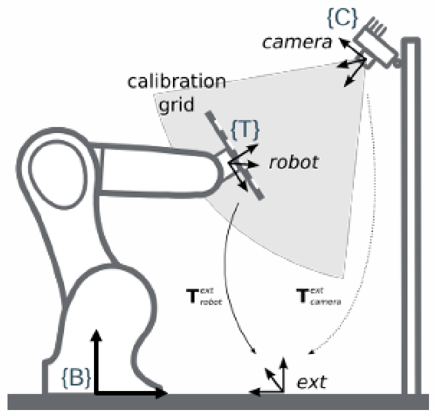


Image_0.jpg Image_23.jpg

State of the Art





3.- Masks

Image_0.jpg

Image_23.jpg

4.- Get object Pixel

$\{(x_i, y_i), r_i\}_{i=1,\ldots,N} \quad N \equiv \#pieces$

5.- 2D → 3D

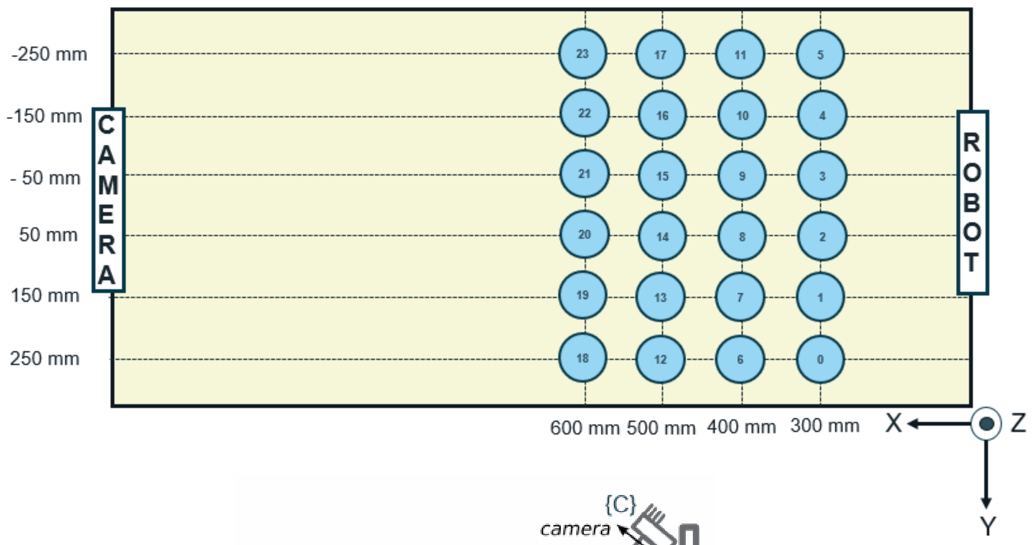
$\{(x_i, y_i, z_i)\}_{i=1,\ldots,N}$

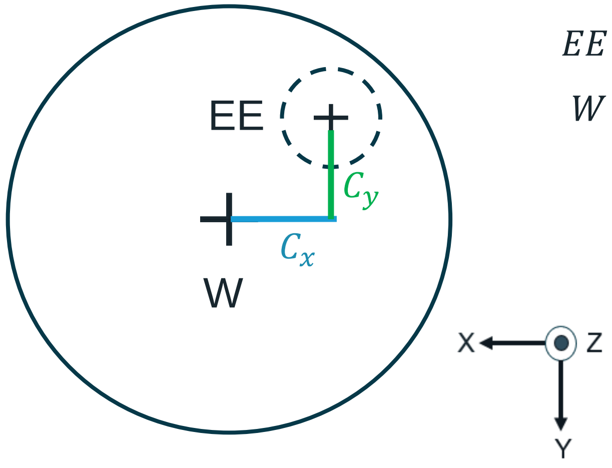
6.-Transformation camera coordinates to robot base coordinates T_{BC}

$\{(\hat{x}_i, \hat{y}_i, \hat{z}_i)\}_{i=1,\ldots,N}$

State of the Art

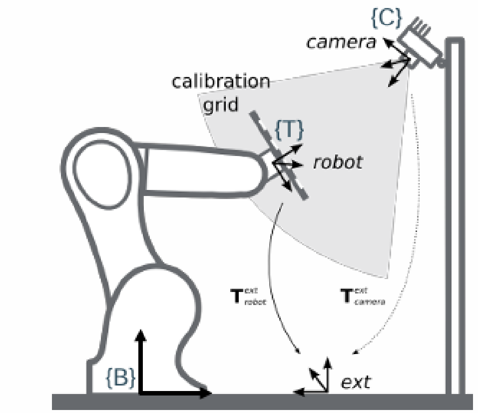
1.- **Tool Correction**: $(C_x, C_y) = (sqrt(200), sqrt(200))\ mm$

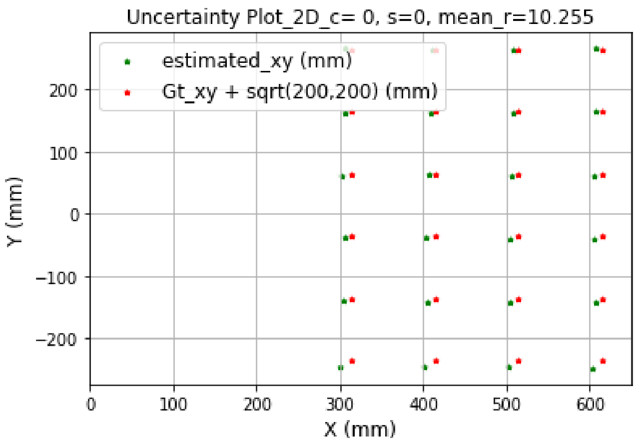




$EE \equiv End\ Effector$

$W \equiv Wrist\ (Suction\ device)$



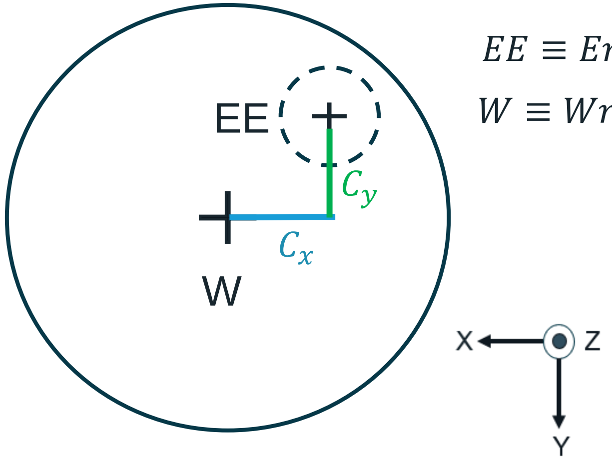


Uncertainty: $Euclidean\ Distance$

$$r = sqrt((x_2 - x_1)^2 + (y_2 - y_1)^2)$$

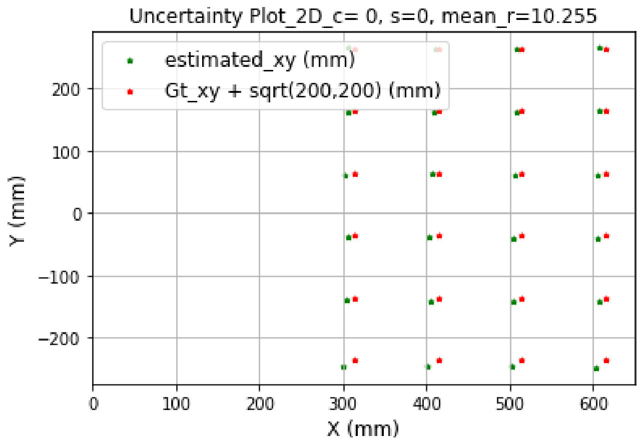
Modifications

1.- **Tool Correction**: $(C_x, C_y) = (sqrt(200), sqrt(200))$

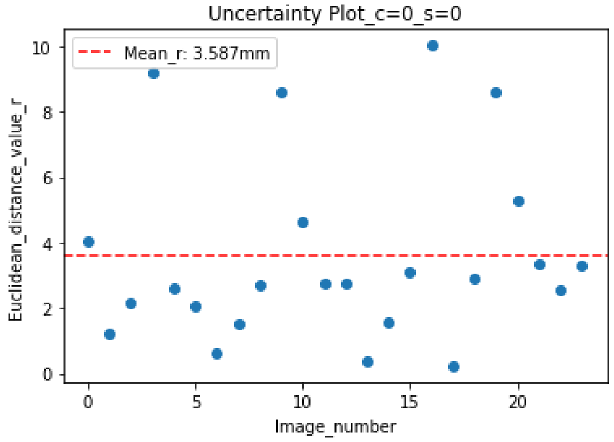


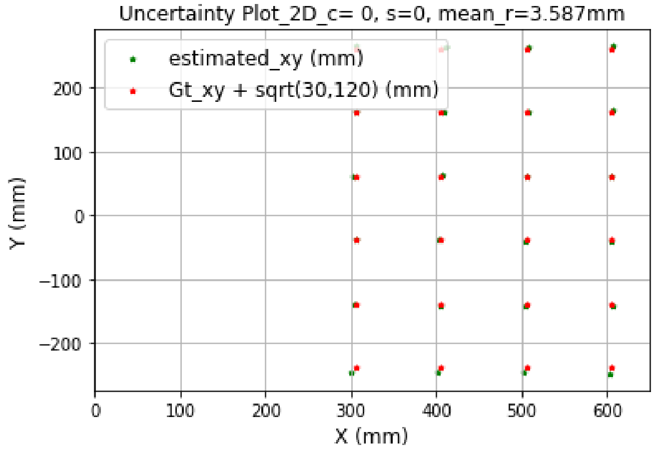
$EE \equiv End\ Effector$

$W \equiv Wrist\ (Suction\ device)$



2.- **Tool Correction + Camera Deviation**:
$(C_x, C_y) = (sqrt(30), sqrt(120))$





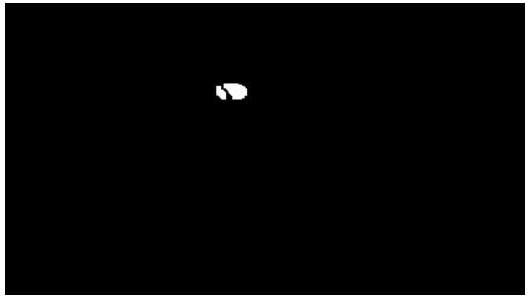
State of the art

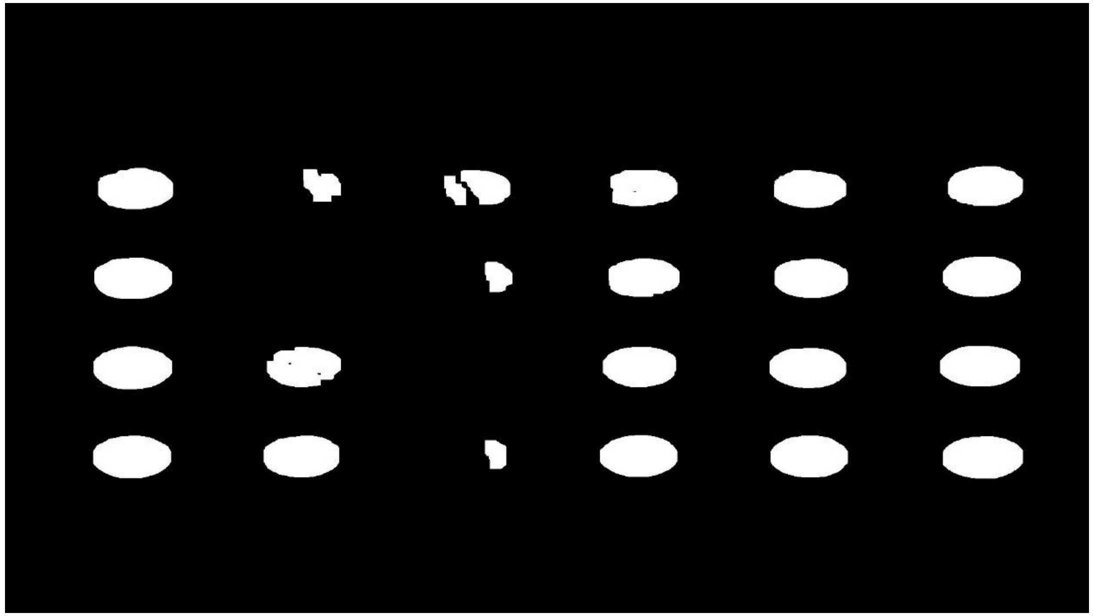
Previous Mask (H=Hue, S=Saturation, V=Value)

1.1 Load HSV calibration parameters.
1.2 Apply Gaussian Blur for image smoothing
1.3 Convert image to HSV for color-based segmentation.
1.4 Create mask using HSV calibration values
1.5 Apply opening (erosión+dilatation) to remove noise/small objects

Advantages.
Effective for color segmentation

Disadvantages
Sensitive to brightness changes (e.g., Sadows)





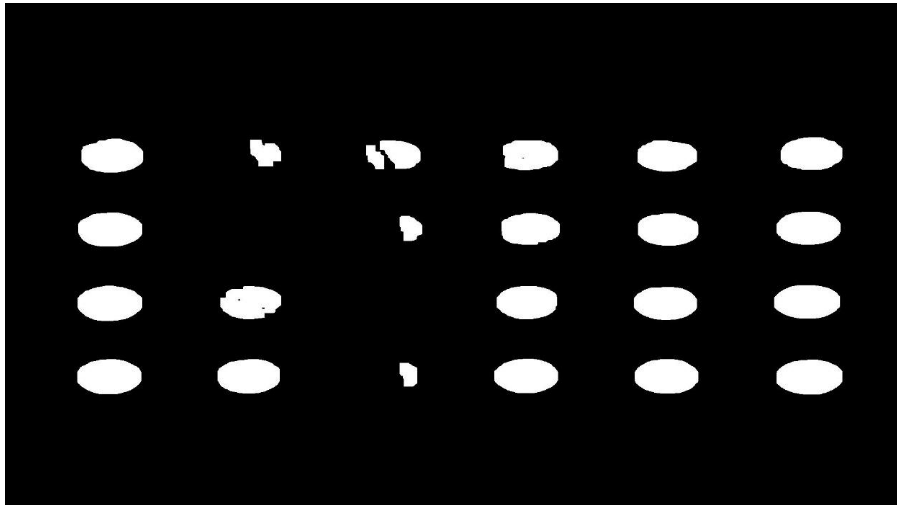
Modifications

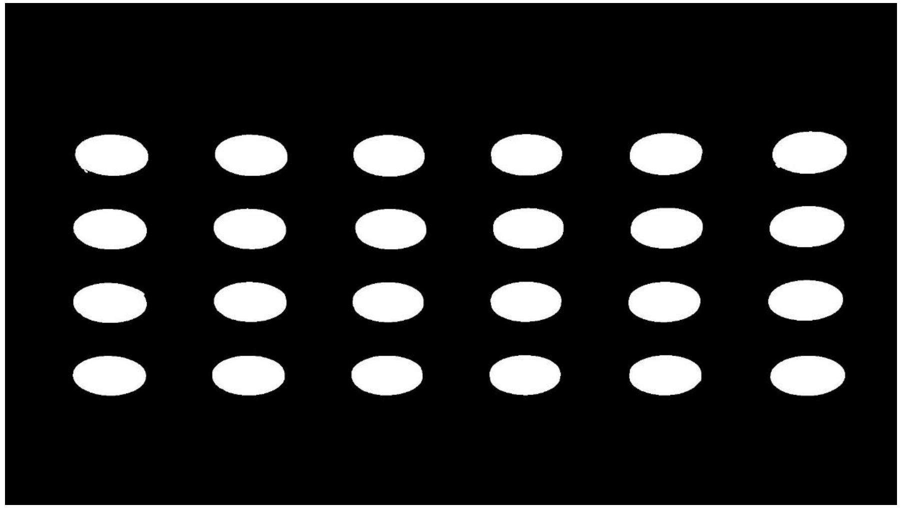
New Mask (Grayscale)

1.1 Grayscale Conversion to reduce complexity.
1.2 Apply Gaussian Blur for image smoothing
1.3 Edge Detection with Canny Algorithm
1.3.1 Identify edges by finding intensity gradients.
1.3.2 Filter out weak edges, keeping strong ones.
1.4 Contour Detection to find and outline distinct shapes.
1.5 Mask Creation

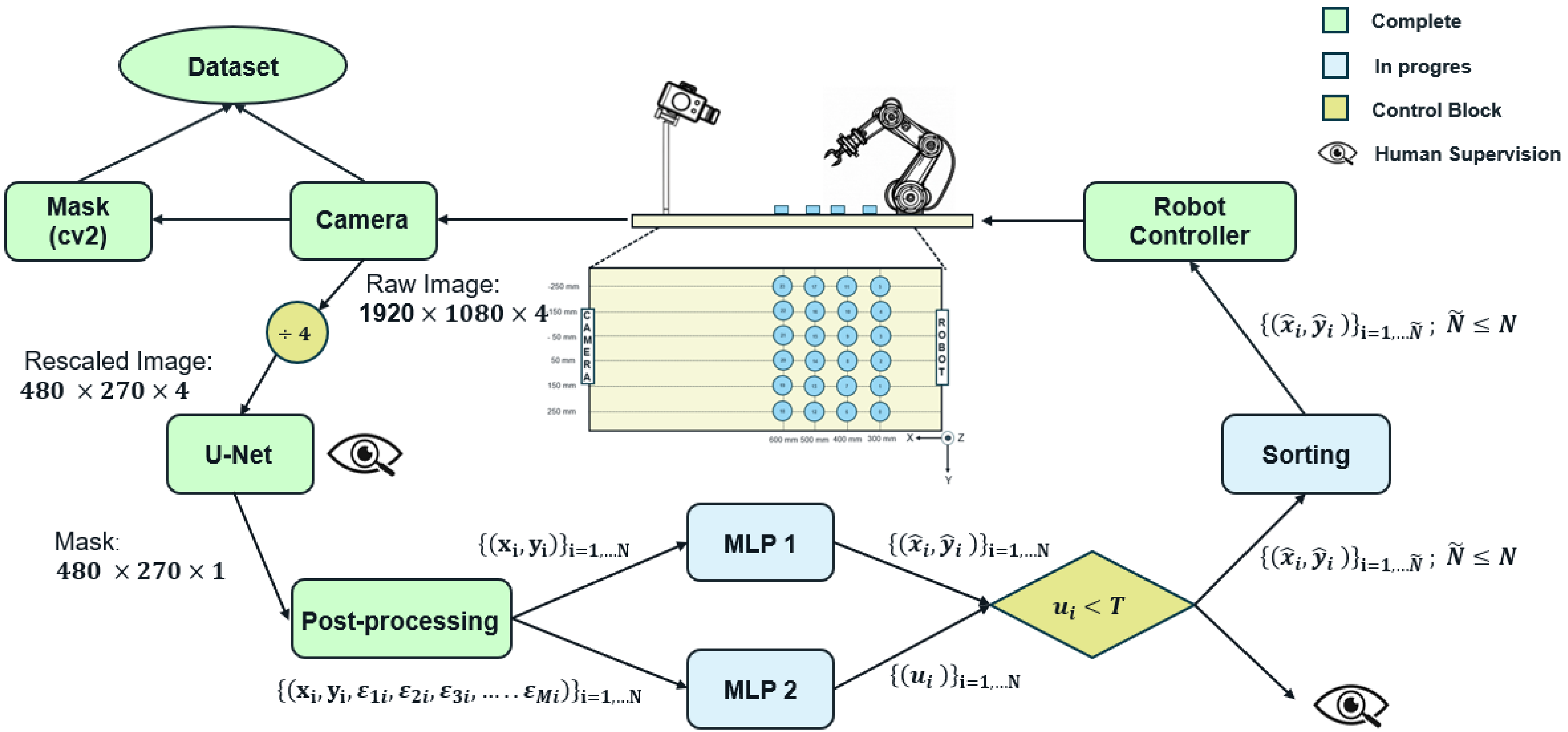
Advantages:
Objects where colour isn´t distinctive, but shape is.

Disadvantages:
Information loss when converting to grayscale.





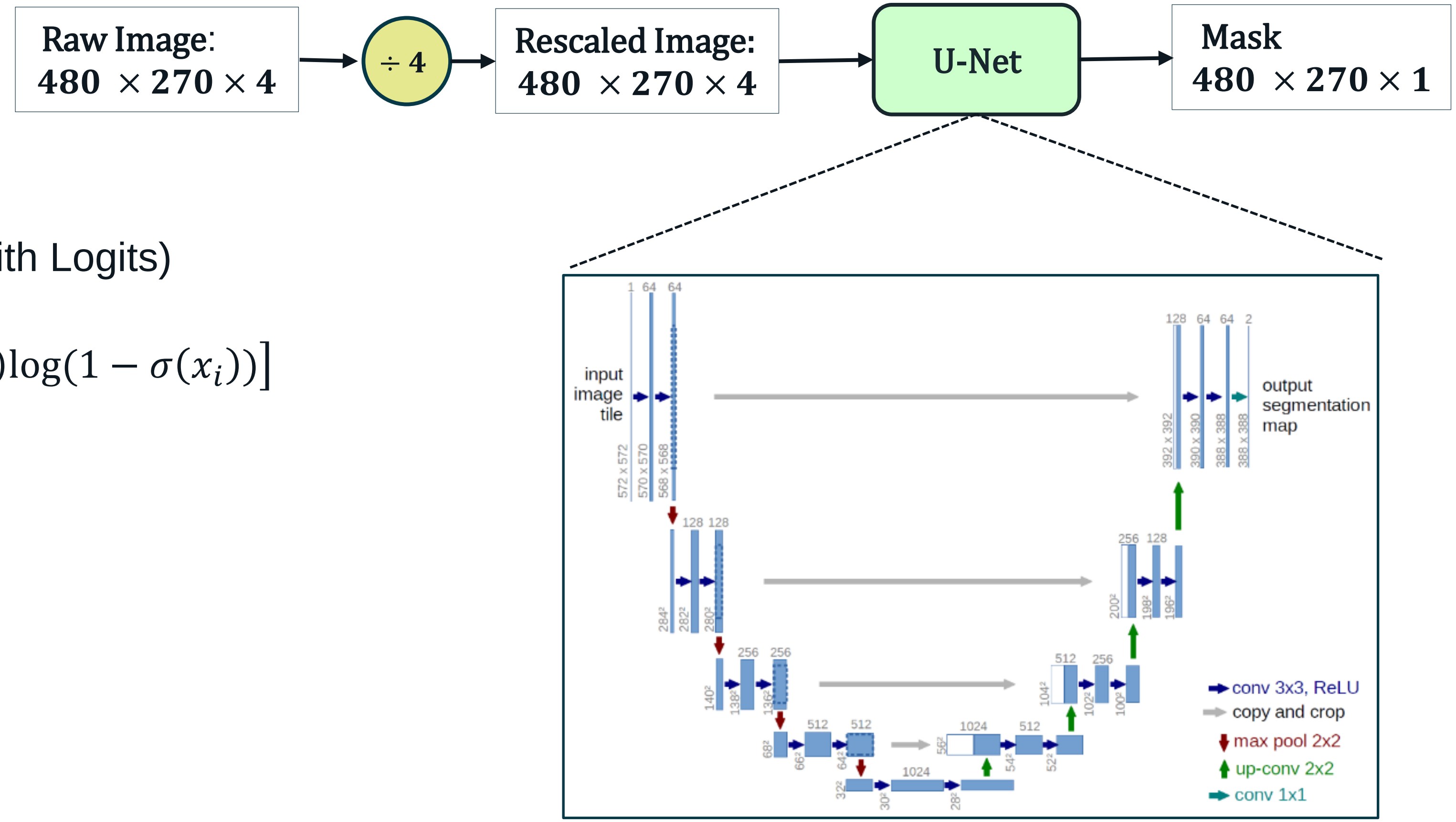
Workflow



Complete
In progres
Control Block
Human Supervision
Dataset
Mask (cv2)
Camera
Robot Controller
Raw Image: $1920 \times 1080 \times 4$
$\div 4$
Rescaled Image: $480 \times 270 \times 4$
U-Net
Mask: $480 \times 270 \times 1$
Post-processing
$\{(x_i, y_i)\}_{i=1,\ldots N}$
$\{(x_i, y_i, \varepsilon_{1i}, \varepsilon_{2i}, \varepsilon_{3i}, \ldots \varepsilon_{Mi})\}_{i=1,\ldots N}$
MLP 1
MLP 2
$\{(\hat{x}_i, \hat{y}_i)\}_{i=1,\ldots N}$
$\{(u_i)\}_{i=1,\ldots N}$
$u_i < T$
$\{(\hat{x}_i, \hat{y}_i)\}_{i=1,\ldots \tilde{N}}$; $\tilde{N} \leq N$
Sorting
$\{(\hat{x}_i, \hat{y}_i)\}_{i=1,\ldots \tilde{N}}$; $\tilde{N} \leq N$
CAMERA
ROBOT

Human Supervision

U-Net [1]



1.- Loss function (Binary Cross Entropy with Logits)

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\sigma(x_i)\right) + (1 - y_i)\log(1 - \sigma(x_i))\right]$$

2.- Evaluation Metric:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$dice\ score = \frac{2\,TP}{2\,TP+FP+FN}$$

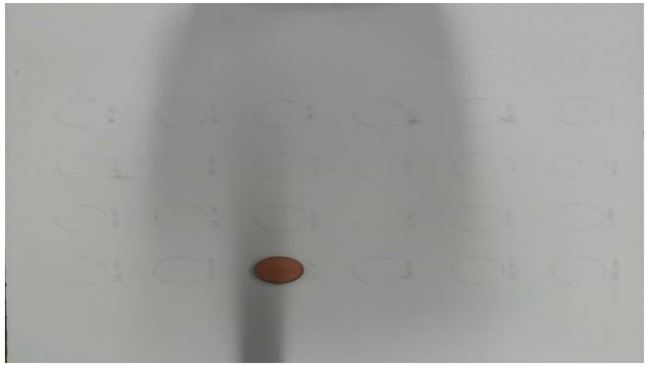
3.- Benchmark U-Net (ImageNet –C) [2]

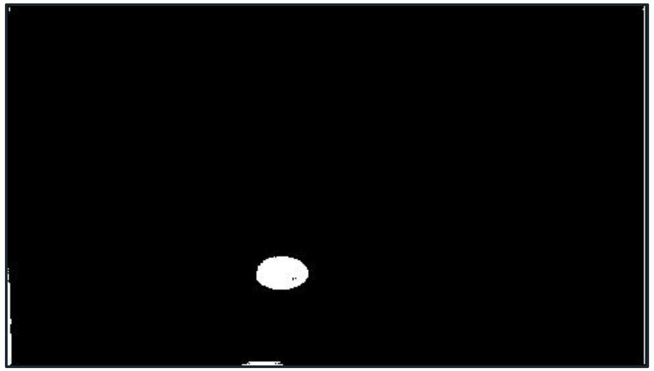
U-Net [1]

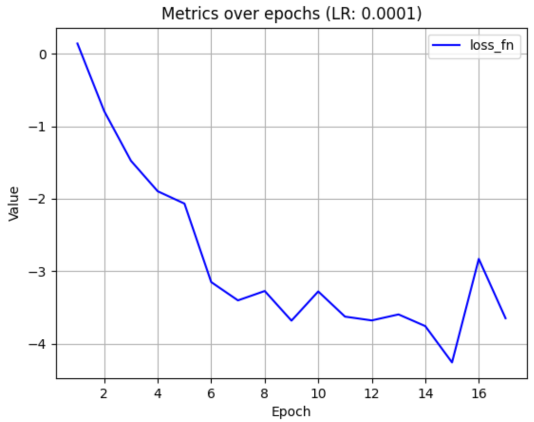
Accuracy = 99.75%
Dice Score = 155% (¿?)

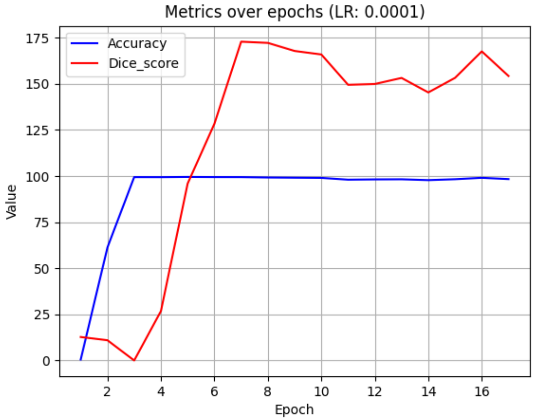
Future study:

1.- Dice Score Calculation
2.- Add Depth to Image (Currently RGB)
3.- Benchmark on Imagenet-C
 3.1 Evaluate Robustness to Corruption
 3.2 Compare performance Metrics
4.- Dataset Augmentation Effect (Imagenet-C)
5.- Hyperparameter Sweep Study

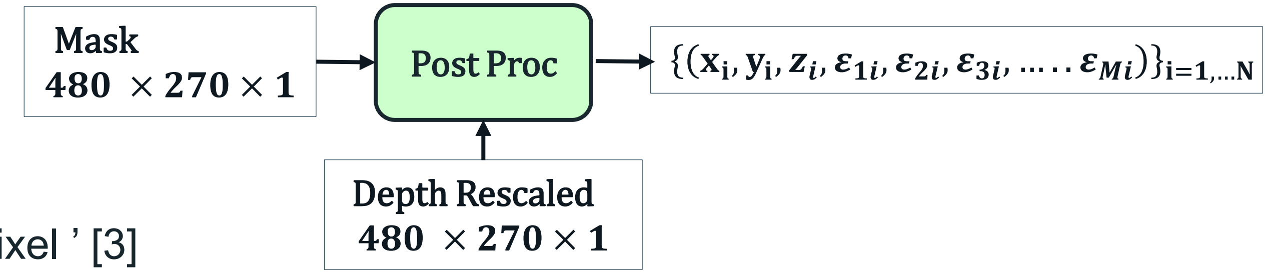






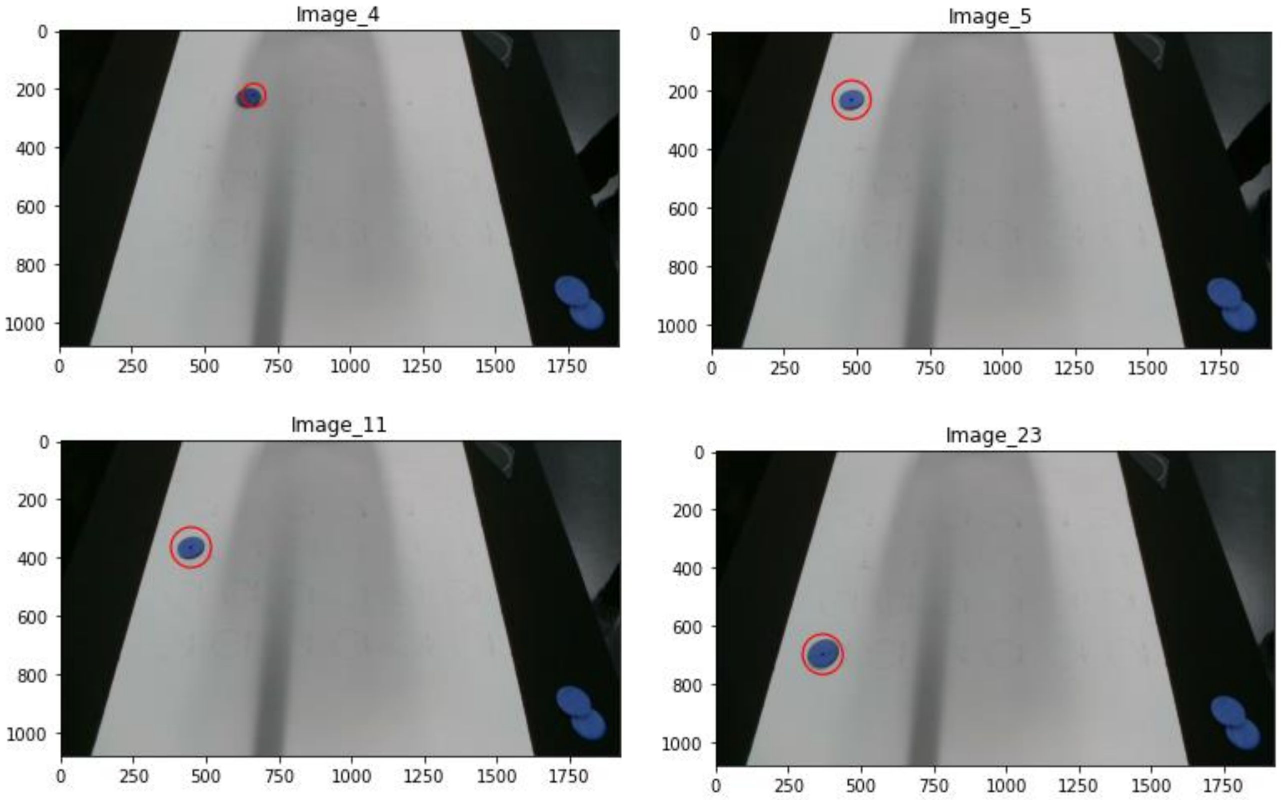


Post Processing



Use:
Mattias de Ryck ‘src/image_processing.py/get_object_pixel ’ [3]

Future study:
1.- Get more features $(\epsilon_{1i}, \ldots \ldots, \epsilon_{Mi})$
- 1.1 Piece Count
- 1.2 Geometric
  - 1.2.1 Area
  - 1.2.2 Perimeter
  - 1.2.3 Centroid
  - 1.2.4 Radius
  - 1.2.5 Aspect Ratio
  - 1.2.6 Orientation
- 1.3 Morphological
  - 1.3.1 Convexity
  - 1.3.2 Texture
  - 1.3.3 Skeletonization

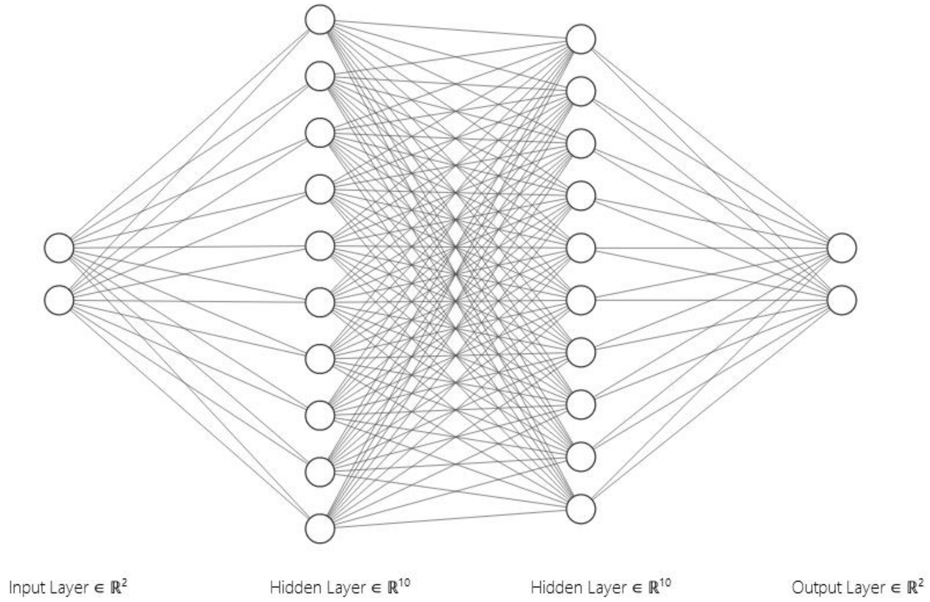


MLP 1

Objective:
Capture Transformation Matrix Complexity, Camera Deviation, Tool Correction.

Input:
$\{(x_i, y_i)\}_{i=1,\dots,N}$ Warped camera frame

Output:
$\{(\hat{x}_i, \hat{y}_i)\}_{i=1,\dots,N}$ Wrist ground truth position.



MLP 2

Objective: Capture Uncertainty in robot decision-making.

Process:

1.- Train a MLP

Objective: Given inputs, compute the end robots wrists predicted position.

Input: $\{(x_i, y_i, \epsilon_{i1}, \dots \epsilon_{iM})\}_{i=1,\dots.N}$ Warped camera frame

Output: $\{(\hat{x}_i, \hat{y}_i)\}_{i=1,\dots.N}$ Wrist ground truth position

2.- Fine tune Classifier with Previously Trained Weights

Purpose: Predict=1 for training data points

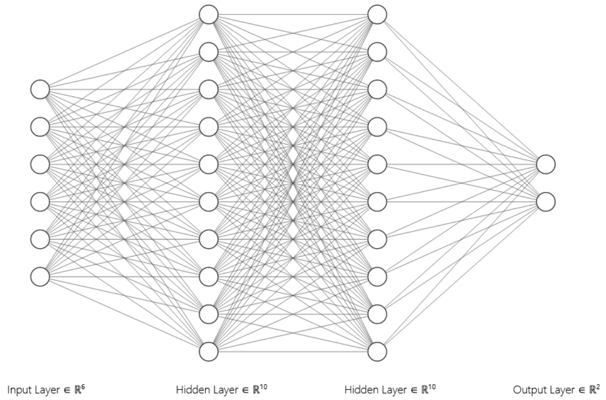
Key Features: Sigmoid activation function in the final layer.

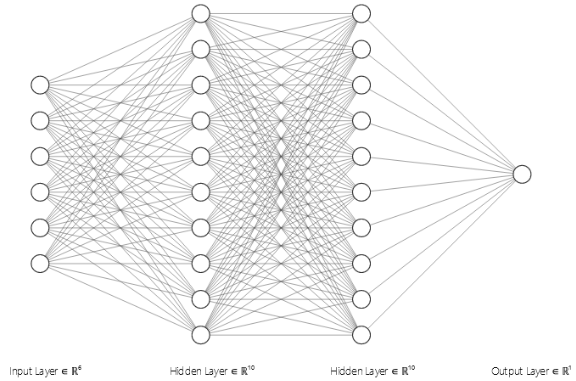
3.- Uncertainty:

3.1 First approach: Embeddings [5][6]

3.2 Second approach: Runtime Monitoring of Neuron Activation Patterns. [4][6]

Greedy Layer Wise Training:





Bibliography

[1] A. F. Frangi, J. Hornegger, N. Navab, and W. M. Wells, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Lecture Notes in Computer Science*, vol. 9351, pp. 234-241, Springer International Publishing AG, Switzerland, 2015, ISBN: 9783319245737.

[2] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," 2019.

[3] M. De Ryck, "Robot Demonstrator Code," 2023, GitHub repository. Available: https://github.com/MatthiasDR96/robot_demonstrator.git.

[4] Chih-Hong, Cheng, Nührenberg, Georg, and Yasuoka, Hirotoshi. "Runtime Monitoring Neuron Activation Patterns." arXiv.org, 2018. ISSN 2331-8422.

[5] Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2016). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. arXiv.org.

[6] Sahu, Amit; Vállez, Noelia; Rodríguez-Bobada, Rosana; Alhaddad, Mohamad; Moured, Omar; Neugschwandtner, Georg. "Application of the Neural Network Dependability Kit in Real-World Environments." arXiv (Cornell University), 2020. ISSN: 2331-8422.