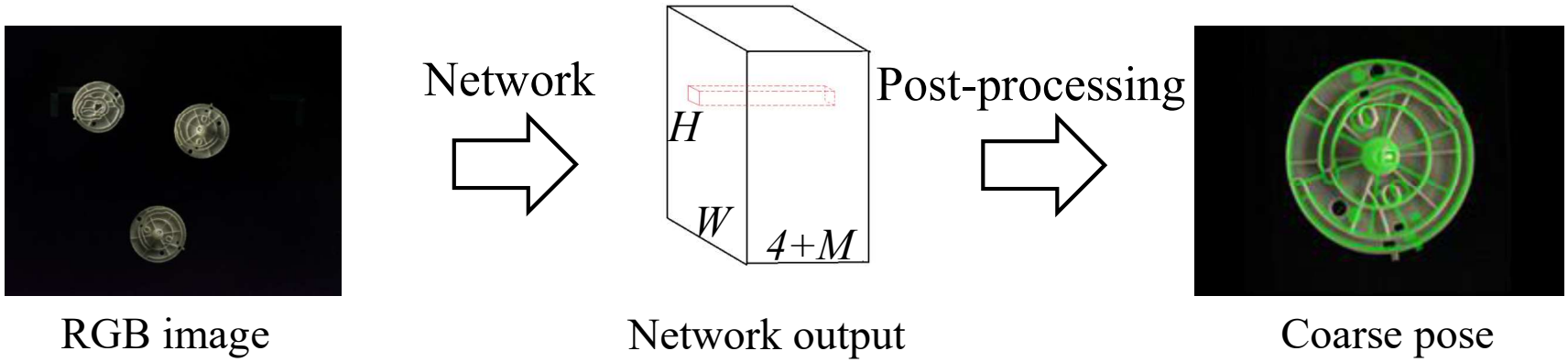
OUELINES

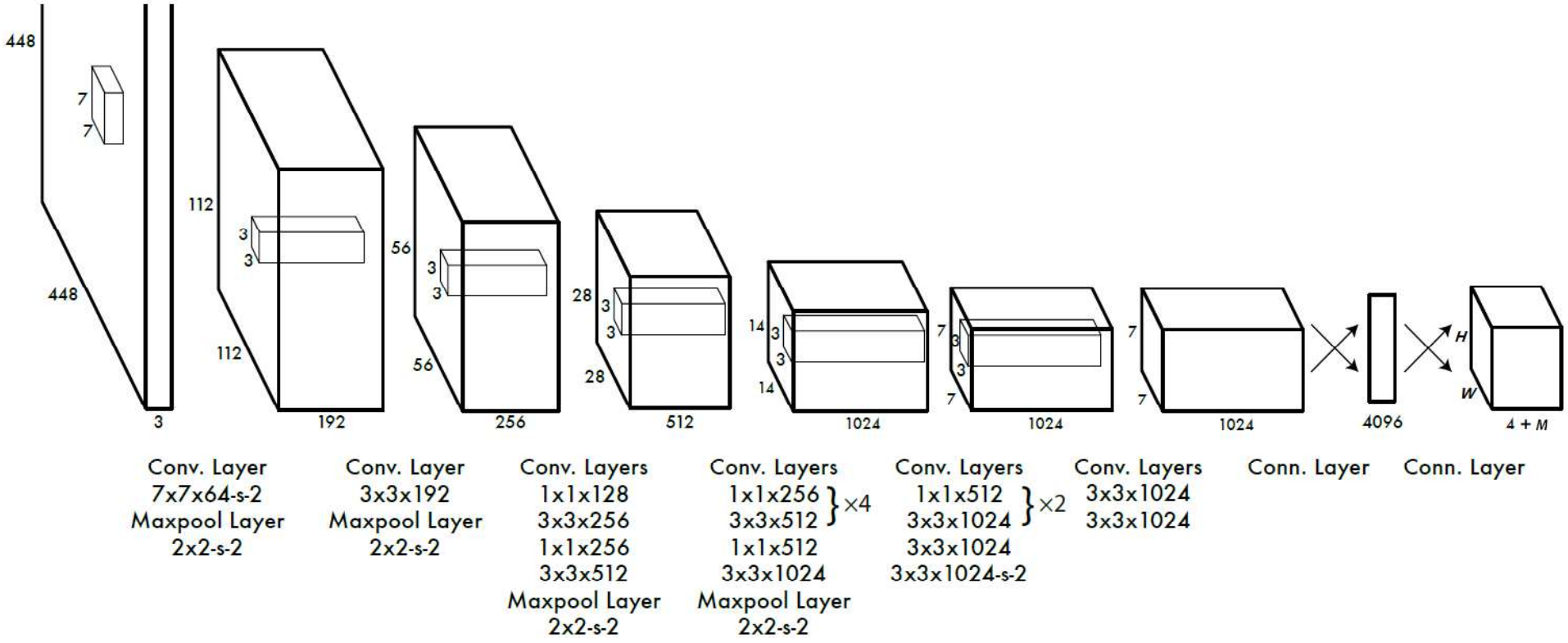
1. Research background
2. Research objectives
3. Related works
4. Grasp detection
5. Pose estimation
6. Conclusion and future work

5.1 Algorithm pipeline



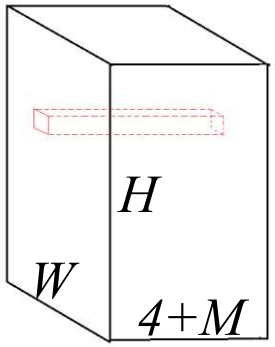
5.2 Network structure

- We use the network of YOLO [10] but redefine the output. YOLO is an end-to-end network used for 2D image detection by making grids on the image.

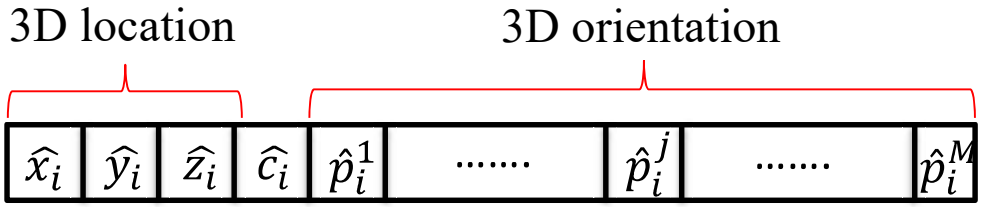


5.3 Definition of network output (1/3)

- Make $W \times H$ grids for the image.



Output of network



Output of i-th grid

- The output of i^{th} grid are:
  - $\hat{x}_i, \hat{y}_i$: Estimated planar position of object center in grid coordinate.
  - $\hat{z}_i$: Estimated depth of object center in mm.
  - $\hat{c}_i$: Estimated confidence of accuracy of 3D location
  - $\hat{p}_i^j$: Estimated discrete orientations

Example of grids and their confidence

5.3 Definition of network output (2/3)

In training data, there are two types of grids.

- Grid i doesn't contain center of any object:
  - $x_i = y_i = z_i = c_i = p_i^j = 0$
- Grid i contains center of an object:
  - $x_i = \frac{x_{obj} - x_{grid}}{L}, y_i = \frac{y_{obj} - y_{grid}}{L}, z_i = \frac{z_i - \min(z)}{\max(z) - \min(z)}$
    - x_{obj}, y_{obj}: Position of object center in image coordinate.
    - x_{grid}, y_{grid}: Position of grid center in image coordinate.
    - z: Depth of object center in mm.
    - L: Grid length in image coordinate.
  - $c_i = e^{-\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2}}$
    - $\hat{x}_i, \hat{y}_i, \hat{z}_i$: Estimated value of x_i, y_i, z_i.
    - c_i: Confidence of i^{th} grid.

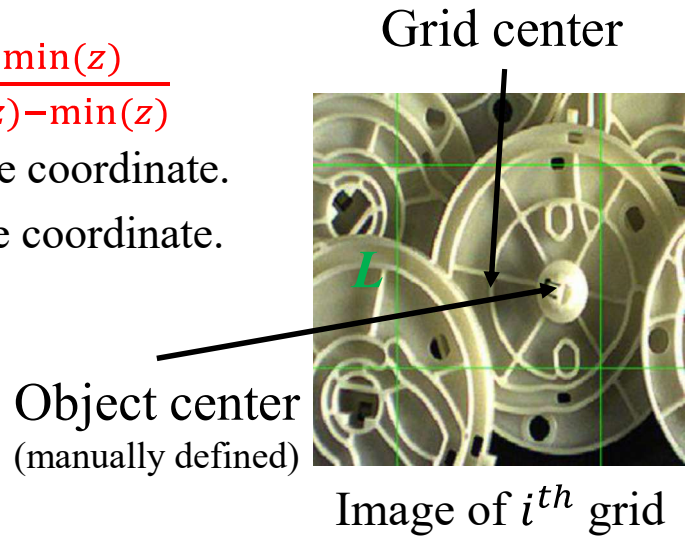


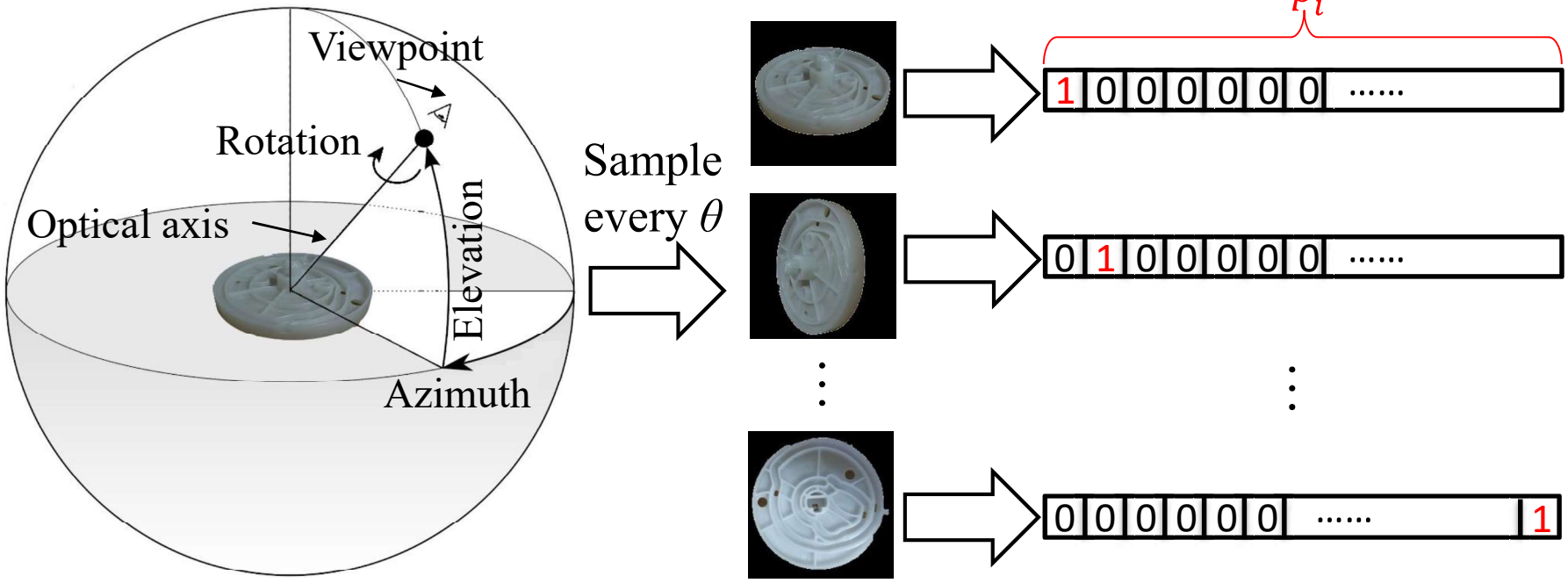
Image of i^{th} grid

- If the grid contains multi objects, we only compute the one with largest visible area.

5.3 Definition of network output (3/3)

- p_i^j is discrete orientations.

Sample 3D orientation from different viewpoints (decided by azimuth, elevation and rotation around optical axis), then transfer into one-hot code.



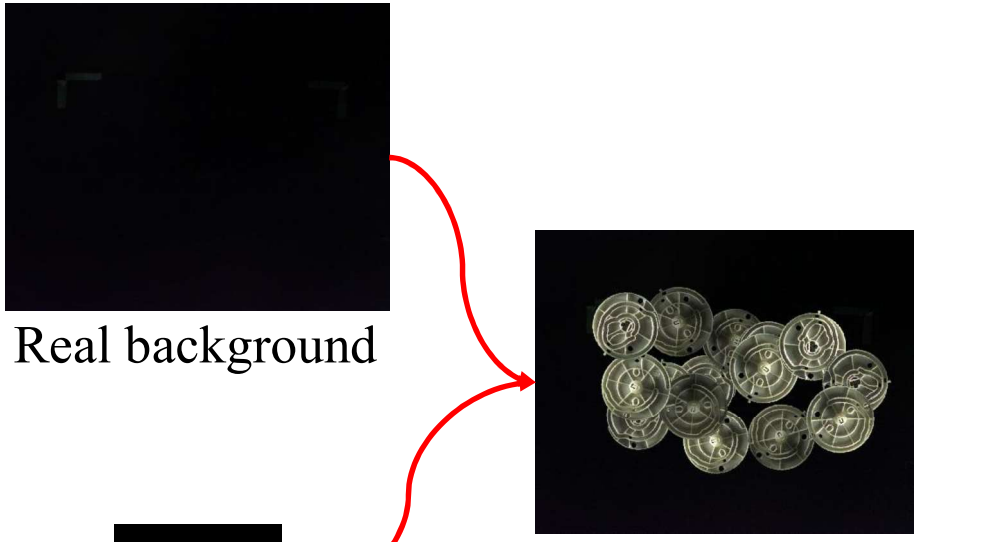
Sample 3D orientation by changing azimuth, elevation and rotation.

RGB image

One-hot code

5.4 Training process (1/2)

- Generate training data by simulation.



Hyper-parameters:

- Data size: 448 × 448 × 3
- Dataset size: 5000
- Batch size: 32
- Dropout: 0.5
- Regularization: 0.0002 L2
- Optimizers: GDO
- Learning rate: 0.001

5.4 Training process (2/2)

- Loss function is divided into three terms.

$$L = \lambda_1 \underbrace{\sum_{i=1}^{W\times H} \delta_i^{obj} \{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2\}}_{\text{Location}}$$

$$+\lambda_2 \underbrace{\sum_{i=1}^{W\times H} \delta_i^{obj} (c_i - \hat{c}_i)^2}_{\text{Confidence of accuracy of 3D location}} + \lambda_3 \underbrace{\sum_{i=1}^{W\times H} \left(1 - \delta_i^{obj}\right) (c_i - \hat{c}_i)^2}_{\text{Confidence of accuracy of 3D location}}$$

$$+\lambda_4 \underbrace{\sum_{i=1}^{W\times H} \delta_i^{obj} \sum_{j=1}^{o} (p_i^j - \widehat{p_i^j})^2}_{\text{Orientation}}$$

$$\delta_i^{obj} = \begin{cases} 1, \text{if grid } i \text{ contains object} \\ 0, \text{if grid } i \text{ doesn't contain} \end{cases}$$

λ_1: Weighting factor of location.

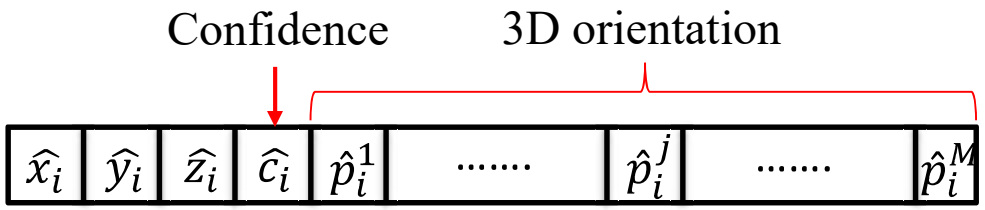
λ_2: Weighting factor of confidence of grid contains object.

λ_3: Weighting factor of confidence of grid doesn't contains object.

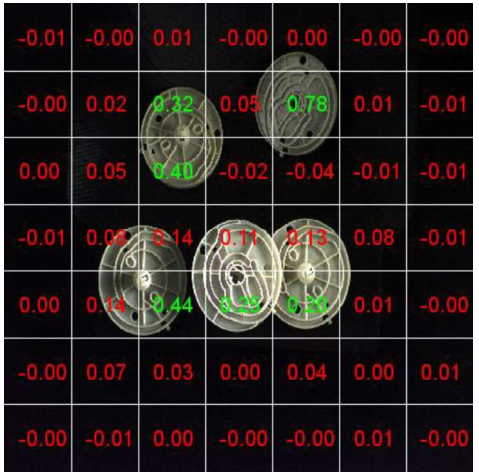
λ_4: Weighting factor of orientation.

5.5 Post-processing of network output

- Perform coarse estimation by using output of network.
  1. Compute the estimation scores s_i for every grid.



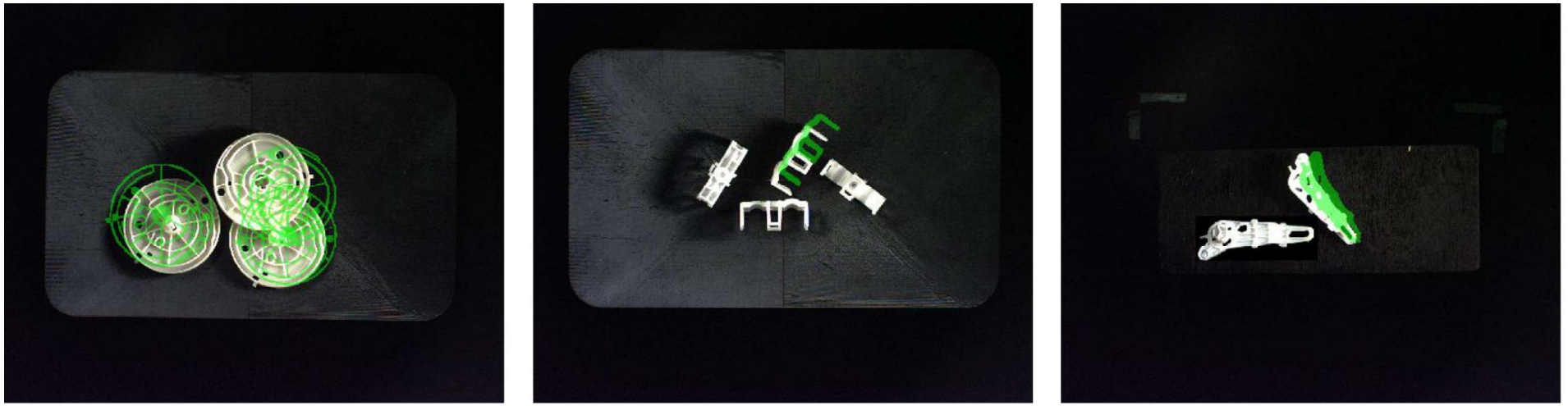
For i^{th} grid: $s_i = c_i \times \max_j \left(p_i^j\right)$



Estimation scores of grids
$threshold = 0.2$

  2. Use the estimation of grids whose scores are higher than threshold as the estimation result.

5.6 Experiment results



Example of pose estimation results.

Table: Experiment results of three mechanical parts

Object No.	Location Error mm	Orientation Error degree	Computation Time ms
A	10.2	18.5	59
B	9.3	19.3	61
C	9.8	14.8	59
Average	9.8	17.5	60