# Micro-I: Technical Report for VISION Challenge Track1

Zhe Wang, Zhiming Qain and Fangzheng Lin
Micro-Intelligence Manufacturing Tech. Co., Ltd., Shanghai, China

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

## Introduction and Contribution

**Competition task:** Locate the position of industrial defects and segment the shape of the defects

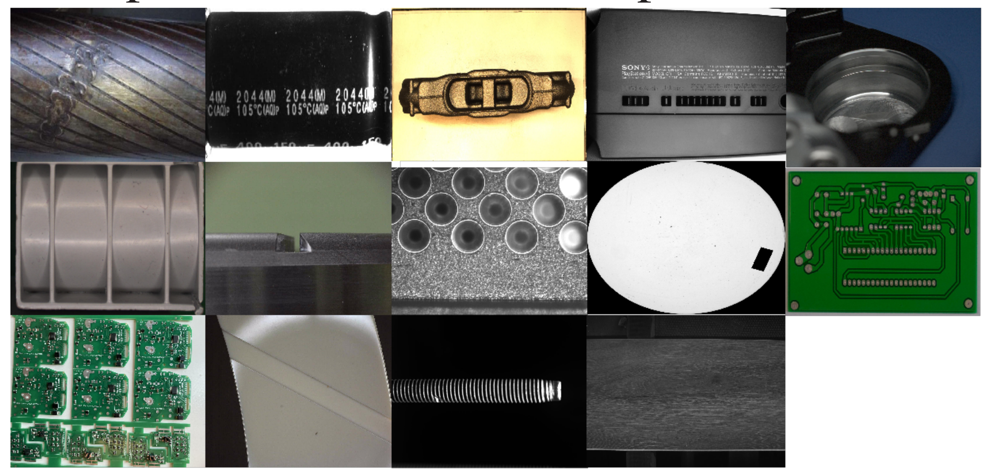
**Challenges:**
- Few-shot learning
- Small size of objects
- Class Imbalance

**Key Contributions:**
- Use online cropping methods to alleviate insufficient GPU memory during training and increase the detection accuracy for small objects by increasing the testing scale.
- By using Gaussian Receptive Field based Label Assignment (RFLA), we achieve balanced learning for small objects and improve the detection accuracy.
- Select appropriate data augmentation methods, such as mosaic, random erasing, and 90° rotation, based on the characteristics of the dataset.
- Inspired by the WBF (Weighted Boxes Fusion) ensemble algorithm for object detection, we have developed an ensemble algorithm based on masks voting. By segmenting the results based on score thresholds, different ensemble algorithms are applied to different segments. This approach allows us to improve mAP (mean Average Precision) while maintaining a high mAR (mean Average Recall).

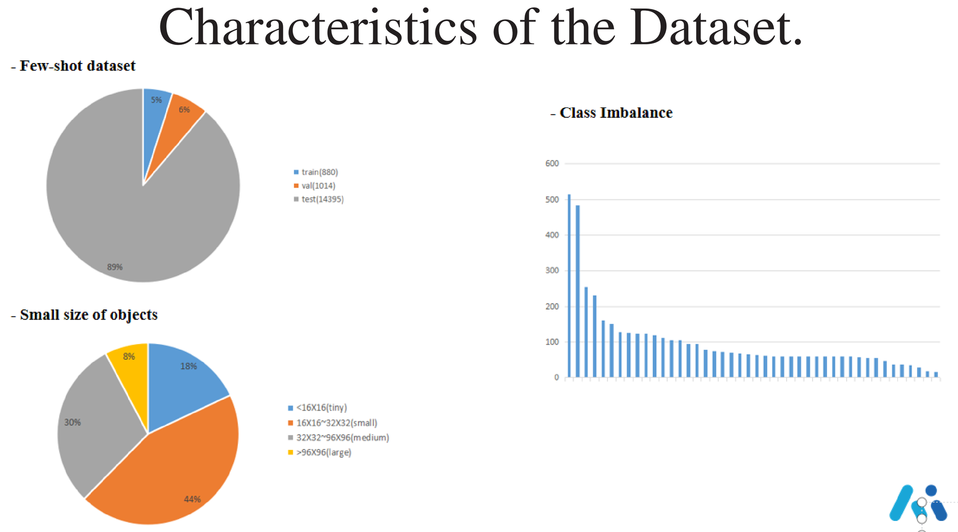
## Competition Dataset

- Containing 14 industrial inspection datasets sourced from roboflow.com, these datasets cover a wide range of manufacturing processes, materials, and industries.
- Includes 880 training samples, 1014 validation samples, 14395 testing samples, and 44 distinct categories.

Examples of 14 Industrial Inspection Datasets

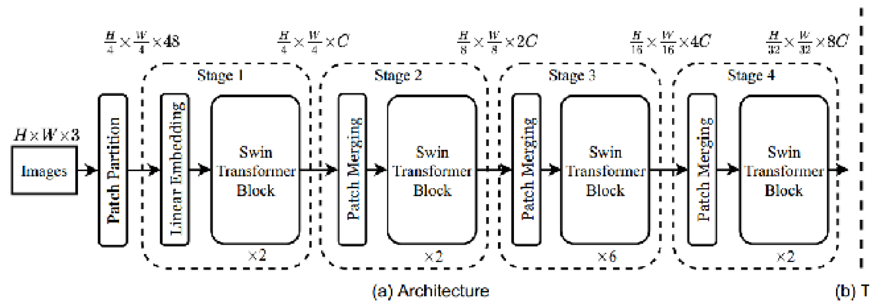


Characteristics of the Dataset.

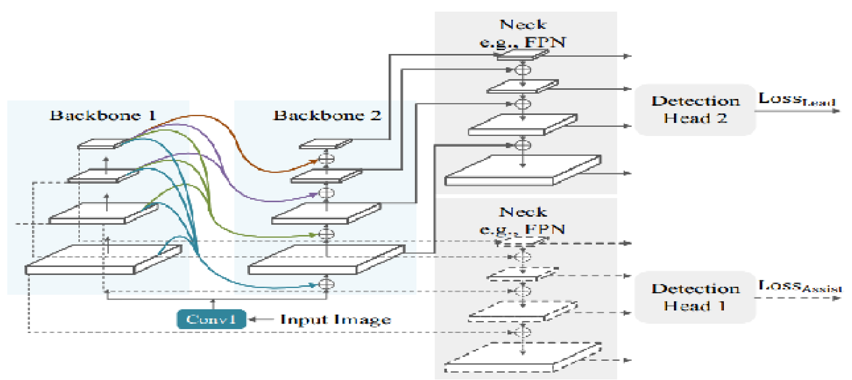



## Overall Pipeline

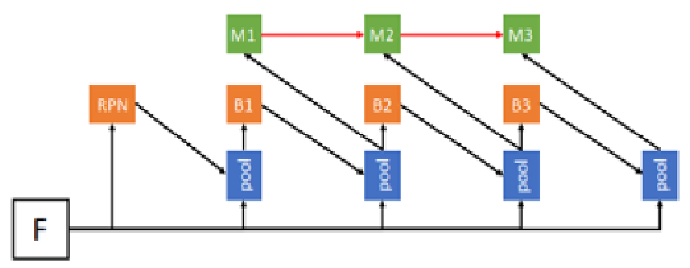
**Swin Transformer(S/B/L)**



**CBNetV2**



**Hybrid Task Cascade**

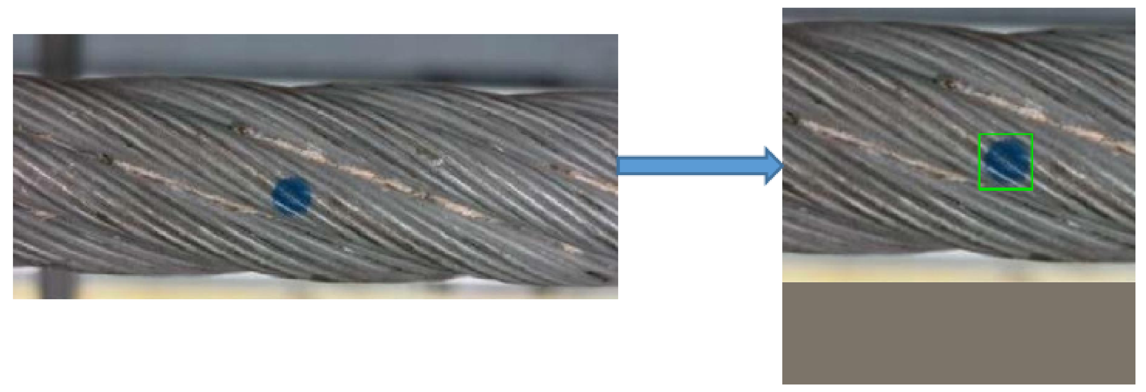


**Backbone:** We use Swin Transformer models of different sizes, namely small, base, and large, as the backbone networks for defect detection. Additionally, we employ the 'checkpoint' technique in PyTorch to reduce GPU memory consumption.
**CBNetV2:** We use CBNet as the detector. The CBNet architecture connects multiple identical backbone networks through composite connections. It integrates high-level and low-level features from multiple identical backbone networks and gradually expands the receptive field for more efficient object detection. This design enables the extraction of richer detailed features and demonstrates strong generalization ability and accuracy performance.
**Hybrid Task Cascade:** We use the Hybrid Task Cascade as the head for instance segmentation. The Hybrid Task Cascade intertwines segmentation and detection through the method of combining tasks, allowing for the progressive learning of more discriminative features and enhancing the ability to distinguish challenging foregrounds from complex backgrounds.

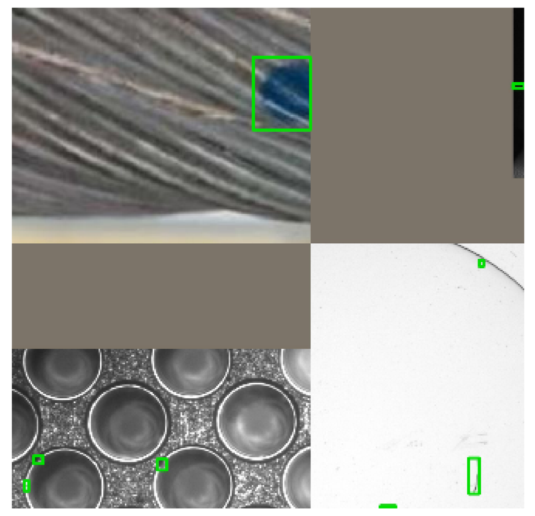
## Useful Modules

**Data augmentation:** We use data augmentation strategies such as Centercrop, Mosaic, Random Erasing, and Rot90°.

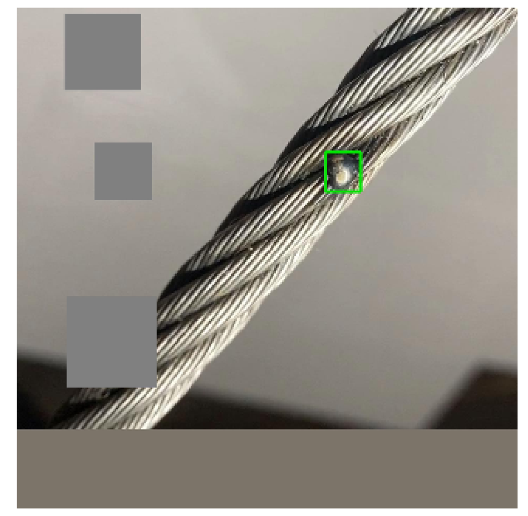
**RandomCenterCropPad**



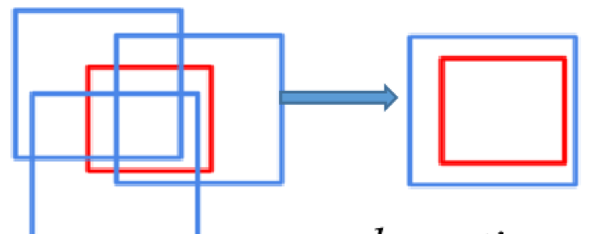
**Mosaic**



**Random Erasing**



**Label assignment:** RFLA is a label assignment strategy based on Gaussian receptive fields. It enables balanced learning for small objects and enhances the detection accuracy of these objects.
**Model Ensemble:** we use masks voting to improve the mAP for high-scoring segments and employ soft NMS to improve the mAR for low-scoring segments.



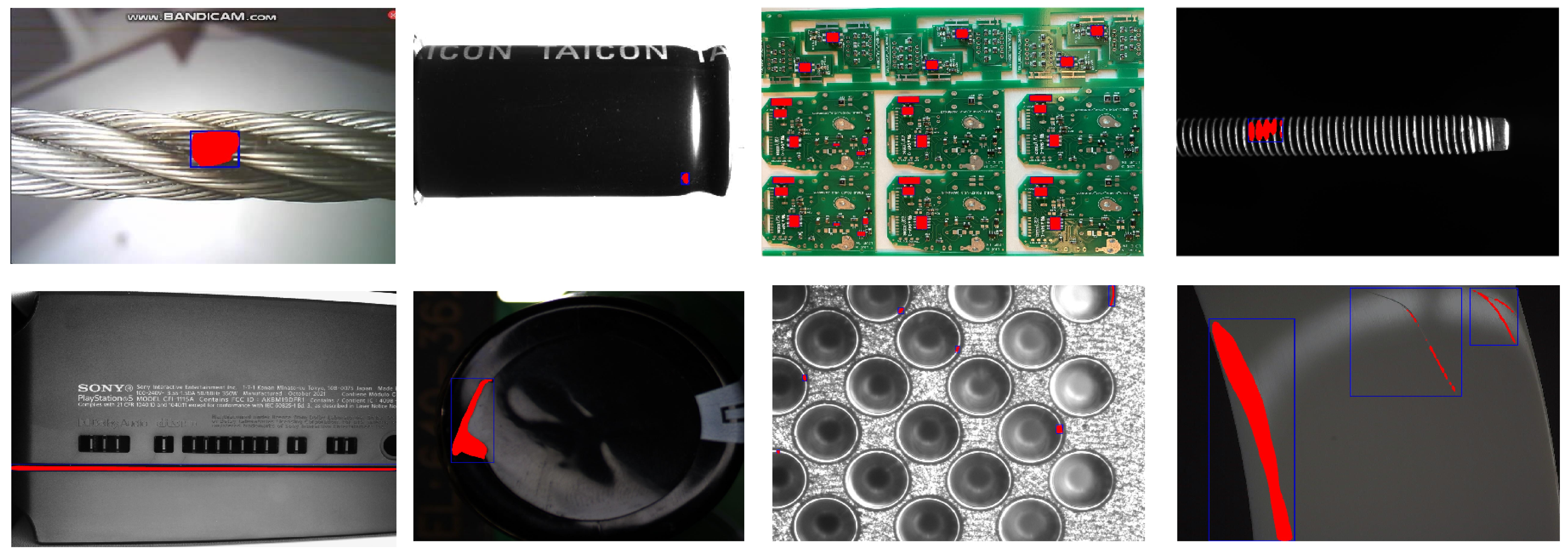
*masks voting*

$$m_{ensemble} = \begin{cases} 1, if\ avg(m_1, m_2, \ldots, m_n) \geq 0.5; \\ 0, elif\ avg(m_1, m_2, \ldots, m_n) < 0.5 \end{cases}$$

$$\begin{cases} masks\ voting(iou\_threshold = 0.8)\ , if\ score \geq 0.3; \\ masks\ voting(iou\_threshold = 0.7)\ , elif\ 0.1 \leq score < 0.3; \\ soft\ \ nms(iou\_threshold = 0.65), else\ socre\ < 0.1. \end{cases}$$

## Results and Experiment

**Visualization:**



**Results:**

| Model | RLFA | mAP | mAR |
|---|---|---|---|
| Cas R50 | - | 0.299 | 0.426 |
| Cas R50 | ✓ | 0.345 | 0.484 |
| CBNetV2 Swin-S | - | 0.464 | 0.616 |
| CBNetV2 Swin-S | ✓ | 0.476 | 0.636 |

| Model | Mosaic | RE | Rot90 | mAP | mAR |
|---|---|---|---|---|---|
| Cas R50 | | | | 0.345 | 0.484 |
| Cas R50 | ✓ | | | 0.350 | 0.501 |
| Cas R50 | | ✓ | | 0.349 | 0.495 |
| Cas R50 | | | ✓ | 0.348 | 0.503 |
| Cas R50 | ✓ | ✓ | ✓ | 0.375 | 0.521 |

| Model | ensemble | mAP | mAR |
|---|---|---|---|
| cbswins 1440 | | 0.447 | 0.586 |
| cbswinl 1440 | | 0.455 | 0.600 |
| cbswins 1800 | | 0.455 | 0.614 |
| cbswinl 1800 | | 0.462 | 0.606 |
| cbswinb 2400 | | 0.466 | 0.631 |
| 10models | soft nms | 0.468 | 0.644 |
| 26models | masks voting | 0.487 | 0.647 |
| 28models | masks voting2 | 0.493 | 0.651 |

| backbone | detector | mAP | mAR |
|---|---|---|---|
| ResNet-50 | Cascade | 0.345 | 0.489 |
| Swin-S | Cascade | 0.452 | 0.611 |
| Swin-S | CBNetV2 | 0.457 | 0.625 |
| Swin-B | CBNetV2 | 0.462 | 0.631 |
| Swin-L | CBNetV2 | 0.465 | 0.637 |

| Resize scale | Crop scale | mAP | mAR |
|---|---|---|---|
| (1800,600)~(1800,1200) | - | 0.299 | 0.426 |
| (2880,960)~(2880,1920) | (1440,1440) | 0.332 | 0.472 |
| (3600,720)~(3600,2880) | (1800,1800) | 0.345 | 0.489 |
| (4800,800)~(4800,4000) | (2400,2400) | 0.354 | 0.496 |