CVPDL hw1: Object Detection for Occupational Injury Prevention

R13946003 李承彥

1. Abstract

This study aims to apply the RT-DETR model for Object Detection in Occupational Injury Prevention. The RT-DETR model achieves fast inference speed while maintaining high accuracy. The final experimental results show that the model demonstrates high accuracy in detecting various objects, with an overall mAP@50 of 0.683, mAP@75 of 0.493 and mAP@50-95 of 0.456. The inference speed is approximately 21.2ms, indicating the model's good detection capability in most cases. For example, the mAP@50 for categories such as "Person" and "Glasses" exceeded 0.9. However, the model still exhibits lower performance in detecting certain objects, which requires further improvement in the future.

2. Model achitecture

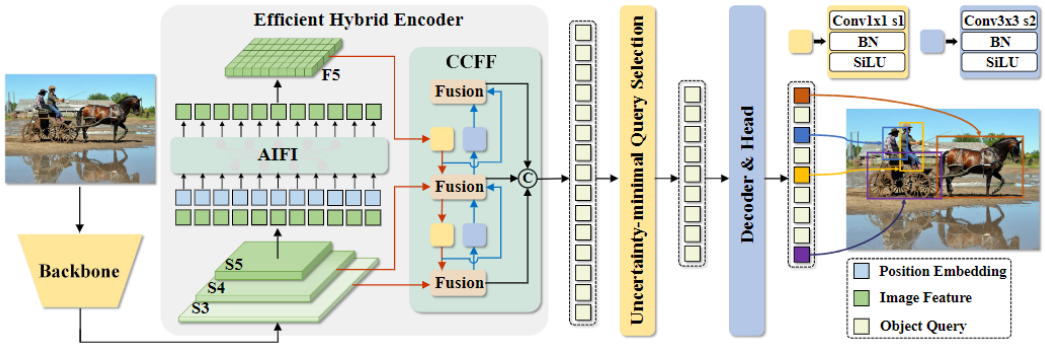

Figure 1: RT-DETR

The RT-DETR model used in this study (Figure 1) is mainly composed of several modules: Backbone, Efficient Hybrid Encoder, Uncertainty-minimal Query Selection, and Transformer Decoder.

a. Backbone

The Backbone uses the ResNet architecture (ResNet-50 or ResNet-101) and is responsible for extracting multi-scale features from the input image. It outputs the features from the last three stages (S3, S4, S5), which contain feature maps at different scales.

b. Efficient Hybrid Encoder

This component aims to balance accuracy and inference speed through efficient feature fusion. The encoder consists of two main modules:

- AIFI (Attention-based Intra-scale Feature Interaction): This module performs intra-scale interaction only on S5 using a single-scale Transformer encoder. By leveraging self-attention, it captures semantically rich high-level features while reducing redundant computations on low-level features. This design allows the encoder to effectively utilize high-level semantic features for object localization.
- CCFF (CNN-based Cross-scale Feature Fusion): This module handles cross-scale fusion, integrating features from different scales using convolutional networks to generate new feature representations.

c. Uncertainty-minimal Query Selection

Traditional Transformer-based detection models typically select the K most representative features generated by the encoder to initialize object queries (or position queries) using the confidence score. The confidence score represents the likelihood that the feature contains foreground objects. However, the detector needs to model both the object's class and location, and these two factors jointly determine the quality of the feature. The current query selection mechanism introduces a significant level of uncertainty in the selected features. This uncertainty affects the overall performance of the detector. To address this issue, the authors propose the Uncertainty-minimal Query Selection (Equation 1 and Equation 2), which explicitly build and refine epistemic uncertainty to represent the joint latent variable of the encoder's features, thereby generating high-quality queries for the decoder. $\widehat{X}$ is encoder feature. $\mathcal{P}(\widehat{X})$ and $\mathcal{C}(\widehat{X})$ are distributions of localization and classification respectively. $\hat{y}$ and y denote the prediction and ground truth, and $\hat{c}$ 、$\hat{b}$ represent the category and bounding box.

$$\mathcal{U}(\widehat{X}) = \left\| \mathcal{P}(\widehat{X}) - \mathcal{C}(\widehat{X}) \right\|, \quad \widehat{X} \in R^{D} \tag{1}$$

$$\mathcal{L}(\widehat{X}, \hat{\mathbf{y}}, \mathbf{y}) = \mathcal{L}_{box}(\hat{\mathbf{b}}, \mathbf{b}) + \mathcal{L}_{cls}(\mathcal{U}(\widehat{X}), \hat{\mathbf{c}}, \mathbf{c}) \tag{2}$$

3. Experiment

In this experiment, the pretrained RT-DETR large model was used for fine-tuning. The GPU utilized was the Nvidia Geforce RTX 3060. The hyperparameters settings during the experiment are shown in Table 1, while the data augmentation parameters are configured as shown in Table 2.

Hperparameter	Setting
epochs	50
batch	16
lr0	0.01
lrf	0.01
momentum	0.937
weight_decay	0.0005
warmup_epochs	3.0
warmup_momentum	0.8
warmup_bias_lr	0.1

Table 1: Hyperparameters settings

Augmentation	Setting	Description
hsv_h	0.015	Adjusts image hue by a fraction, adding color variability.
hsv_s	0.7	Alters image saturation, adjusting color intensity
hsv_v	0.4	Modifies the brightness of the image by a fraction
translate	0.1	Translates the image horizontally and vertically
scale	0.5	Scales the image by a gain factor
fliplr	0.5	Flips the image left to right
mosaic	1.0	Combines four training images into one
erasing	0.4	Randomly erase a portion of the image
crop_fraction	1.0	Crops the classification image to a fraction of its size

Table 2: Data augmentation settings

4. Experimental result

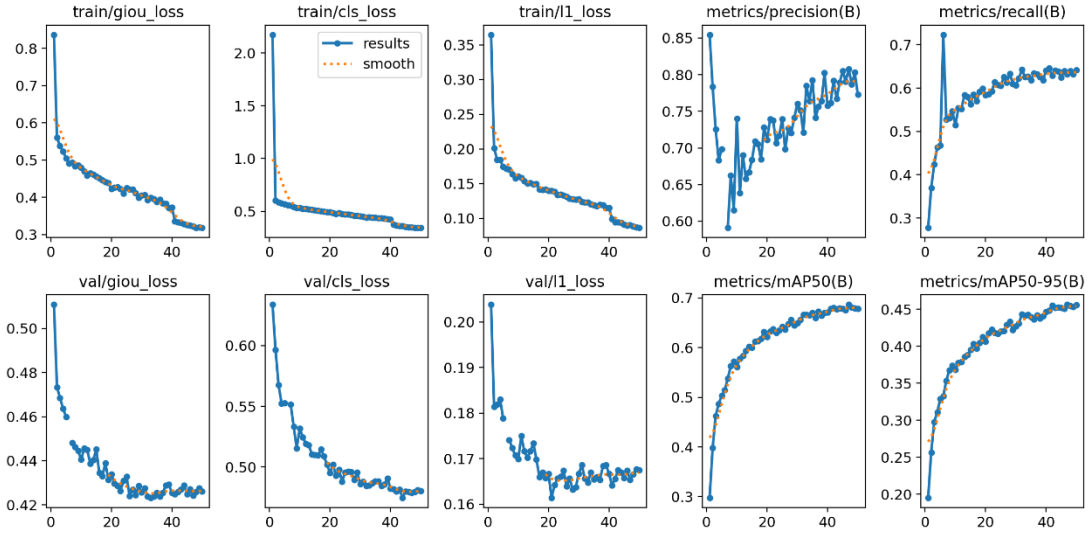

Figure 2: Changes in loss and performance metrics

Figure 2 illustrates the changes in various losses and performance metrics during the training and validation process of the object detection model, including GIoU Loss, Cls Loss, L1 Loss, Precision, Recall, mAP@50, and mAP@50-95. Both in the training and validation sets, the loss values show a consistent decreasing trend, while precision, recall, and mAP steadily improve. This indicates that the model's ability in classification and bounding box regression is gradually strengthening.

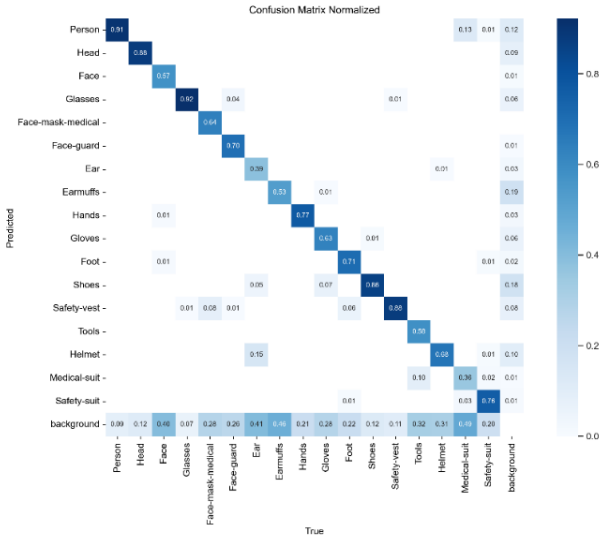

Figure 3: Confusion matrix

Next, the confusion matrix (Figure 3) effectively illustrates the prediction errors and accuracy across different classes. A few examples observed from the confusion matrix include: the Person class achieved a prediction accuracy of 0.91, indicating the model has high accuracy in identifying humans. Additionally, the Glasses class reached the highest prediction accuracy of 0.92. On the other hand, the classification accuracy for the Earmuffs and Helmet classes was lower, at 0.53 and 0.68, respectively. The Medical-suit class has the lowest accuracy, with only 0.36.

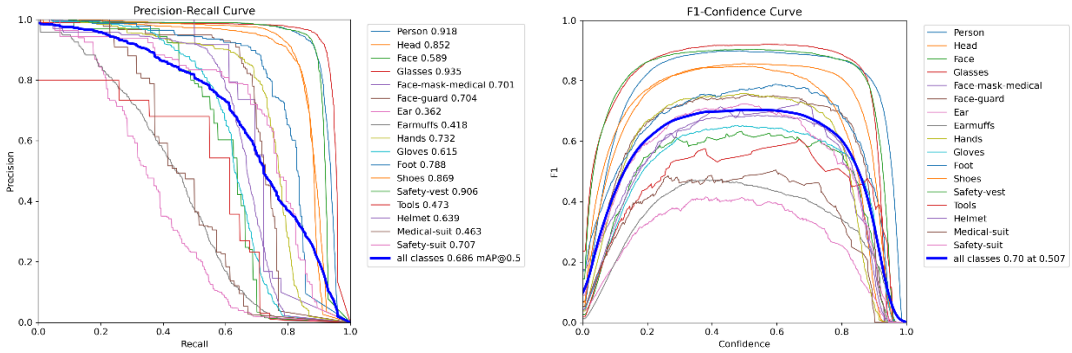

Figure 4: Precision-Recall curve and F1-Confidence curve

The results in the Precision-Recall Curve in Figure 4 indicate that while the model performs well on certain classes like Person and Glasses, it underperforms on others such as Tools and Medical-suit. The overall mAP@50 is 0.686, suggesting that the model is capable of detecting objects effectively in most cases, but there is still room for improvement in certain object classes. The F1-Confidence curve clearly shows the performance variation of different object classes at various confidence thresholds. Across all classes, the optimal confidence threshold is 0.507, where the model achieved an F1 score of 0.70, demonstrating overall strong detection capability. However, for some classes like Safety-suit and Medical-suit, there remains significant room for improvement, indicating the need for further model optimization or dataset refinement.

Class	**Images**	**mAP@50**	**mAP@75**	**mAP@50-95**	**mAP@50-95 (Provided by TAs)**
All	2160	0.683	0.493	0.456	0.6979

Table 3: Overall validation result

Class	**Images**	**mAP@50**	**mAP@50-95**

Person	2045	0.918	0.747
Head	1320	0.852	0.540
Face	57	0.592	0.365
Glasses	1511	0.935	0.694
Face-mask-medical	35	0.647	0.393
Face-guard	98	0.704	0.445
Ear	93	0.364	0.200
Earmuffs	610	0.415	0.233
Hands	426	0.731	0.428
Gloves	362	0.615	0.396
Foot	133	0.778	0.555
Shoes	1713	0.869	0.599
Safety-vest	1736	0.905	0.700
Tools	26	0.467	0.288
Helmet	440	0.644	0.371
Medical-suit	43	0.465	0.339
Safety-suit	64	0.707	0.464

Table 4: Validation result for each class

Table 3 and Table 4 present the evaluation results of the model's detection performance on the validation dataset. The model provides detailed metrics such as mAP@50, mAP@75 and mAP@50-95 for different object categories. The results show that the model performs exceptionally well in detecting categories like Person, Head, and Safety-vest, with the mAP@50 for the Person category reaching as high as 0.918, and mAP@50-95 at 0.747, indicating high accuracy and stability for these common objects. However, the model performs poorly in detecting certain categories such as Tools and Medical-suit, with mAP@50-95 values of only 0.288 and 0.339, respectively, possibly due to data imbalance or indistinct object features. Overall, the model demonstrates good detection performance across most categories, with an overall mAP@50 of 0.683, mAP@75 of 0.493 and mAP@50-95 of 0.456, indicating a certain level of generalization ability. Additionally, the model achieves fast inference speeds, with an inference time of approximately 21.2 ms per image.

5. Conclusion and future work

This study successfully demonstrates the potential of the RT-DETR model for object detection in industrial environments. The experimental results show that the model exhibits high detection capability for certain object categories (e.g., Person, Glasses), but there remains room for improvement in detecting less frequently occurring categories in the dataset (e.g., Ear, Earmuffs). Overall, the model provides fast and accurate detection, contributing to the automation of occupational injury prevention. However, future work should focus on improving the detection performance for rare objects, potentially by enhancing the dataset or adjusting the loss function to further improve the overall capability of the model.

6. Reference

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024). Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16965-16974).

Jocher, G., Qiu, J., & Chaurasia, A. (2023). *Ultralytics YOLO* (Version 8.0.0) [Computer software]. Ultralytics. https://github.com/ultralytics/ultralytics