TRCA-YOLOv8 — Modified YOLOv8 on Traffic Flow Detection

Xing Chen
Rice University
6100 Main St, Houston, TX 77005
Email: sc236@rice.edu

Xuemeng Zhao
Rice University
6100 Main St, Houston, TX 77005
Email: xz107@rice.edu

Sicong Chen
Rice University
6100 Main St, Houston, TX 77005
Email: xc58@rice.edu

Yufan Wang
Rice University
6100 Main St, Houston, TX 77005
Email: yw183@rice.edu

***Abstract*—In this project, we delve into the application of advanced object detection models to urban traffic analysis, focusing specifically on the latest iteration, YOLO v8. Our goal is to use this technology to not only compute but analyze traffic flows in complex urban environments. To achieve this goal, we employ the UA-DETRAC dataset, a powerful resource for training and evaluating our models. The core goal is to improve the accuracy of traffic flow detection by carefully refining the internal structure of YOLO v8.**
An important aspect of our research involves customized approaches to model optimization. We developed a specialized version of YOLO v8, called TRCA-YOLOv8, with localized modifications. This adapted model achieves significant advances in detecting and analyzing urban traffic flows. Our efforts ultimately improved traffic flow detection accuracy from an initial 54% to an impressive 59%.

1. Background

Traffic flow detection based on object detection techniques to identify and track different types of vehicles with a road environment in real-time. Object detection [1] is a fundamental task in the computer vision field and was greatly promoted by deep learning. The network detects different types of vehicles at the single-frame level by capturing appearance features.

In the short term, solving this problem could provide valuable insights into traffic management and transportation planning. In the long term, it could facilitate the development of smart cities. Previous studies have typically utilized old versions of the YOLO model, such as YOLOv3 [2], [3] or YOLOv4 [4]. However, we have optimized the state-of-the-art YOLOv8 model in this project, leading to improved accuracy of results. Fig1 provides an overview of our detection process, which involves using three colors of bounding boxes to tag cars, buses, and vans in the input video flow. The UA-DETRAC dataset, the dataset involved in this project, is described in detail in the Experiment section.

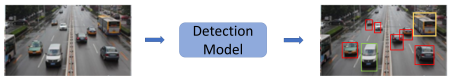


Figure 1: Overview of the process

2. Methodology

YOLOv8 can be divided into two main parts: YOLOv8 backbone and YOLOv8 head. Backbone part is the main module of the net for feature extraction, while the head part is used for multi scale feature fusion and then prediction. Our paper does modifications in both two parts to improvement the ability of feature extraction and fusion for YOLOv8.

2.1. C2fTR Block

In the original YOLOv8 model, C2f block uses Bottleneck [5] to lower the count of parameters in order to cut the calculation cost. From Fig2 we can see, Bottleneck is a kind of special residual structure.

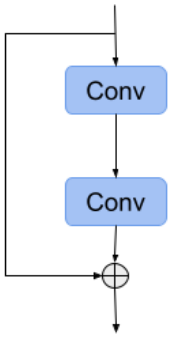


Figure 2: Bottleneck Block in YOLOv8

The backbone of YOLOv8 is to extract augmentation features for YOLOv8 head to detect. Based on those purposes, we replace the Bottleneck block with the TransformerBlock [6]. The TransformerBlock not only has the residual strcuture, but also its attention structure could help extract the spatio-temporal features in the video images. The main equation of dot-product scaled attention is as following,

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Beyond this modification, we come up a new C2fTR block as Fig3 shows.

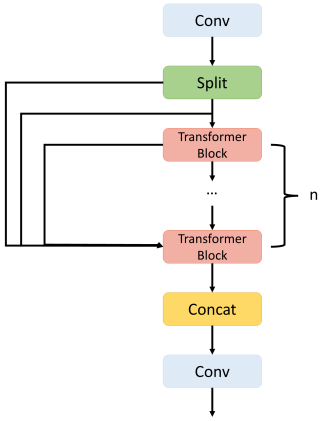


Figure 3: C2fTR Block

2.2. Coordinate Attention

The original YOLOv8 head just uses residual module and convolution for feature fusion, justing fusing features on the same dimension. So we add coordinate attention block [7] to this part. Coordinate attention contains two steps: coordinate information embedding and coordinate attention generation. It can not only takes direction-related positional information into consideration, but also it is more flexible and could be more easily added to other nets.

In order to avoid compressing all the spatial information into the channel, use avg pooling for decompose:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, i) \quad (2)$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j \le H} x_c(j, w) \quad (3)$$

After getting the feature map, use concat for feature fusion,

$$f = \delta(F_1([z^h, z^w])) \quad (4)$$

Do split operation along the space dimension, then we could get the attention vectors,

$$g^h = \sigma(F_h(f^h)) \quad (5)$$

$$g^w = \sigma(F_w(f^w)) \quad (6)$$

Finally the output of coordinate attention is like:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \quad (7)$$

The whole structure of coordinate attention is as Fig4 shows.

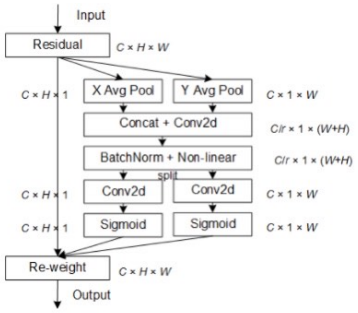


Figure 4: Coordinate Attention Structure

2.3. TRCA-YOLOv8

Generally based on those two modifications, we come up our own proposed model TRCA-YOLOv8 in Fig5.

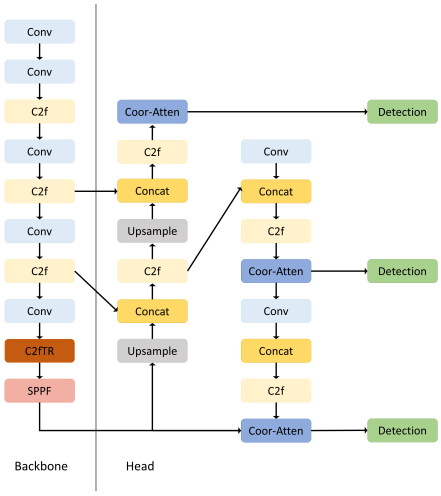


Figure 5: TRCA-YOLOv8 Structure

3. Experiment

3.1. Dataset

We use UA-DETRAC [8] dataset, which includes 100 challenging videos of real-world traffic scenes with various information in PASCAL VOC format, such as bounding box, trajectory, speed, weather, etc. Vehicles are classified into Car, Van, Bus and Others.

Figure 6: Dataset Samples

We choose 2116 images as training dataset and 1423 images as validation dataset. Before training, we need to do some transformation to change the PASCAL VOC format into YOLO supportive data format.

3.2. Experiment Settings

We primarily select the original YOLOv8 as the baseline model for comparsion. The batch size is set to 16, and the training epochs are set to 100. The classification branch uses VFL Loss as loss function.

$$VFL(p,q) = \begin{cases} -q(q(\log q) + (1-q)\log(1-q)) & q > 0 \\ -\alpha p^{\gamma}\log(1-p) & q = 0 \end{cases} \quad (8)$$

The regression branch uses Distribution Focal Loss and CIoU Loss. Then make weighted mean on those three loss function as the final loss function.

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (9)$$

Then we use confusion matrix and F1-curve as the evaluation metrics:

		Actual Class	
		Positive (1)	Negative (0)
Predicted Class	Positive (1)	TP	FN
	Negative (0)	FP	TN

TABLE 1: Confusion Matrix

$$precision = \frac{TP}{TP + FP} \quad (10)$$

$$recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1score = \frac{2 \times precision \times recall}{precision + recall} \quad (12)$$

3.3. Experiemnt Results

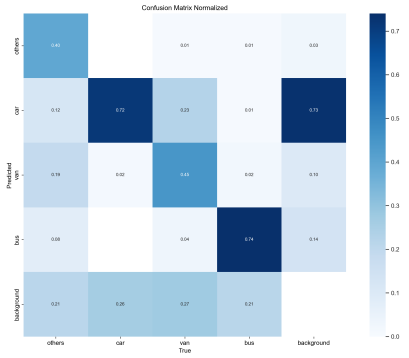
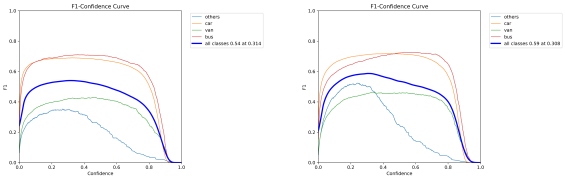


Figure 7: Normalized Confusion Matrix



(a) (b)

Figure 8: F1-curve Comparsion

From Fig8 we can see, the left curve is the result for YOLOv8 and the right is for our proposed model TRCA-YOLOv8. Our proposed model has a better performance of 59% detection accuracy at confidence

0.308, while YOLOv8 only has 54% accuracy at confidence 0.314.

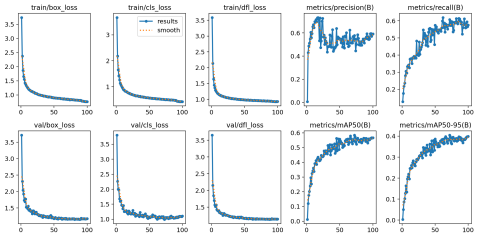
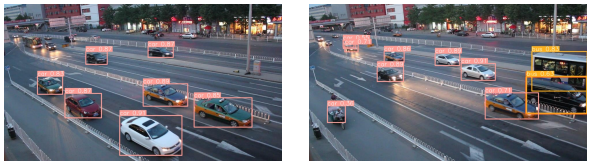


Figure 9: Experiment Loss

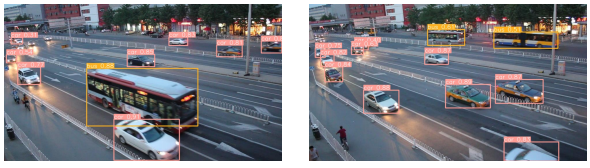
Also, grids on the diagonal of the confusion matrix are much deeper than others, which really reflect our classication accuracy.

(a) (b)

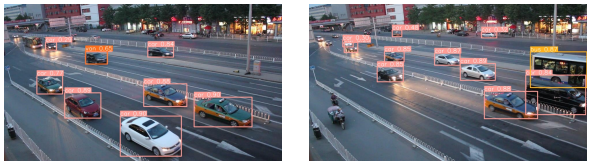


(c) (d)

Figure 10: Detection Results for YOLOv8



(a) (b)



(c) (d)

Figure 11: Detection Results for TRCA-YOLOv8

Compared Fig10 and Fig11, our proposed model has a better performance on detecting smaller and overlapping objects. YOLOv8 may miss some small objects in the videos or mistakes them as other wrong classification. But our proposed model slightly avoid these mistakes.

4. Conclusion

To improve the accuracy of current target detection model YOLOv8, we add some different attention modules to both YOLOv8 head and backbone parts to more accurately extract the spatio-temporal features. So we established a new detection model TRCA-YOLOv8. For YOLOv8 backbone, we change the last layer of C2f to C2fTR, which replace the bottleneck block as Transformer block. In addition, for YOLOv8 head, we add coordinate attention for the feature fusion. We do experiments on the real-world traffic dataset to compare the performance between YOLOv8 and TRCA-YOLOv8 and then come out that our proposed model improves the detection accuracy by 5%, from 54% to 59%. In the future, first we will use BoT-SORT for traffic statistics. Secondly, based on the inaccuracy reflected from the confusion matrix, we will try to add a two-stream motion capture module to more accurately extract spatio-temporal features for object detection.

5. Code Availability

Our code is available at https://github.com/SilasChen-US/MCSE_PROJECTS

6. Group Members Roles

Sicong Chen: group leader, code implementation, poster and report writing
Xuemeng Zhao: code implementation, poster and report writing
Xing Chen: code implementation, poster and report writing
Yufan Wang: code implementation, poster and report writing

References

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[2] A. Ćorović, V. Ilić, S. urić, M. Marijan, and B. Pavković, "The real-time detection of traffic participants using yolo algorithm," in *2018 26th Telecommunications Forum (TELFOR)*, pp. 1–4, 2018.

[3] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2021.

[4] C.-J. Lin and J.-Y. Jhang, "Intelligent traffic-monitoring system based on yolo and convolutional fuzzy neural networks," *IEEE Access*, vol. 10, pp. 14120–14133, 2022.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[7] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," 2021.

[8] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *Comput. Vis. Image Underst.*, vol. 193, apr 2020.