Berger Scientific Press Inc.
Innovation & Technology Advances
Volume 1 Issue 1, June, 2023
www.bergersci.com
DOI: https://doi.org/10.61187/ita.v1i1.2

Smart warehouse track identification based on Res2Net-YOLACT+HSV

Xiujuan Zhao*, Lei Zhang, Zhonghua Hu

School of Automation and Information Engineering, Tianjin University, Tianjin, China
**Correspondence Author, xiujuan2018@yeah.net*

Abstract: ***In recent years, the rapid development of the courier industry, in order to solve the problem of the accumulation of goods in courier warehouses, low efficiency of manual sorting and transportation, many courier companies have introduced intelligent unmanned warehouse logistics systems for unmanned transport vehicles or unmanned inspection The study proposes a recognition method based on the combination of Res2Net-YOLACT and HSV to achieve accurate recognition of the route trajectory of unmanned vehicles in the logistics warehouse environment . After the detection of the YOLACT network with improved backbone network, the area of the track is extracted by HSV, and the extracted area is used as the object of secondary processing, i.e., the extracted track is then subjected to centerline extraction to form the actual track. By comparing the recognition effect of this algorithm on the test set under different direction shapes of tracks, the correct rate of Res2Net-YOLACT in the same experimental environment is 97.37%, and the speed of detecting a single image is 30.26 ms, and then the centerline extraction algorithm is ported to the Res2Net-YOLACT network, and the speed of detecting a single image is 37.26 ms. Compared with the 93.28% accuracy of the original YOLACT network, the speed of single image detection is improved by 3.6%. In addition, the centerline extraction algorithm designed using this study is less computationally intensive and less code engineering than the prevailing algorithm, increasing the memory footprint by only 3.2%. To verify the performance of this centerline extraction algorithm, a comparison between this algorithm and the commonly available algorithms was performed, which showed a 1.31% improvement in correctness and an 8.73% improvement in the speed of detecting videos compared to the general algorithm, indicating that the centerline extraction algorithm processed by this study has higher accuracy and real-time performance without significantly consuming more memory. In addition, to test the practicality of the algorithm, the time used by the algorithm to detect the same video on the embedded device jetson nano was counted, and the average frame rate was calculated to be 28FPS and the maximum frame rate was 33FPS, which can be achieved in real transportation applications.***

Keywords: Dividing; Identification; Smart Warehouse; Track; YOLACT; HSV; Centerline**.**

1. Introduction

Logistics[1] is a key and important part of today's express, cargo and other transportation, with the development of artificial intelligence technology, many courier companies led the way and began to introduce intelligent warehouse logistics[2-3] transportation. Compared with the traditional artificial logistics warehouse, the intelligent warehouse replaces manual labor with robots, and its fast and convenient transportation and orderly formation to complete the transportation of goods sorting work. In transportation[4-5], unmanned vehicles, inspection vehicles, robots, etc.[6] are the mainstay of the smart warehouse link, and these unmanned devices operate with instructions from a central dispatching system that can independently collect images ahead, identify segmented tracks and derive a centerline, and then return the information to the central dispatching system for control of the equipment.

At present, the recognition methods for roads and tracks are mainly divided into two categories, one is based on traditional vision detection methods, such as Wang Xiaojuan et al.[7] used HSI spatial conversion method and two-dimensional Otsu threshold segmentation method to extract road area features. Nie Sen et al [8] used HSV color model and maximum interclass variance method and Hough transform to fit and merge the fruit tree tree rows finally obtained the midline of two adjacent tree rows as the navigation path. Another type of detection method is based on deep learning, such as Qi Hao et al [9] used an improved lightweight semantic segmentation model (MBv2-DPPM), which is more accurate and more effective for complex traffic scenes compared with the traditional semantic segmentation model, while satisfying the segmentation speed condition. Thuan-Leung Ruan et al[10]used a fast road detection method based on bilateral segmentation optimization network (DAM-BiSeNetV2) to lighten the feature pyramid and optimize the attention mechanism.

In this paper, we propose an instance segmentation network based on Res2Net-YOLACT+HSV using a combination of deep learning algorithm and traditional vision. Firstly, the image is input to Res2Net backbone, the segmented image is output through YOLACT network, and then the HSV color space conversion is done, and finally each midpoint is derived by fast point finding algorithm to obtain the The centerline trajectory is obtained by fast point finding algorithm. The experiments show that this algorithm improves the segmentation accuracy and efficiency.

2. Network Structure

2.1 ResNet-YOLACT network structure

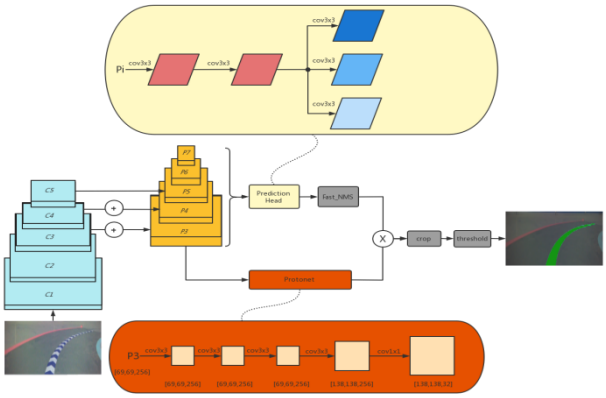


Fig.1: YOLACT network structure

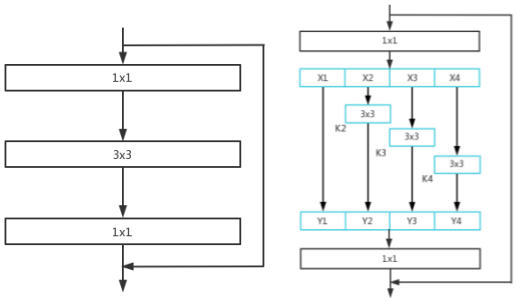
The conventional YOLACT [11] network uses ResNet101 as the backbone network, as shown in Fig. 1.

The detection process of YOLACT: the image is first fed into the feature extraction network with ResNet101 as the backbone to get five feature branch sub-tasks from C1 to C5, and P3 will do segmentation prediction of different foreground backgrounds at each location by branch one, i.e., the Protonet branch structure of Fig. x. The k 138*138 proto prototype results are obtained to generate different masks (mask). Branch 2, the Prediction Head, is added to P3 to P7, and each pixel is predicted by the Prediction Head to generate an anchor frame with each pixel point as the anchor point. Then the two sub-tasks are operated independently for Mask Coefficient Vector prediction, Anchor Box Class, and Bounding Box Regression. Finally, after Non-Maximum Suppression (NMS) to obtain the instance anchor box and each instance, the final instance segmentation detection results are obtained by combining the k 138*138 proto prototypes output by Protonet (superposition, cropping, threshold segmentation).

2.2 Res2Net based improved backbone network

Considering that the complex environment of the smart warehouse has an impact on the detection accuracy, in order to improve the network's ability to extract multi-scale features, increase the perceptual field of the network, and extract trajectory instances more accurately, this study replaces the original Bottleneck's ResNet with the Res2Net module.

Fig 2a shows the ResNet [12] module and Fig 2b shows the Res2Net [13] module, and it can be seen that the latter obviously inserts more hierarchical residual-like connections into the residual block.



a. ResNet b. Res2Net

Fig.2: Backbone network

Res2Net replaces a set of 3×3 convolution kernels with a smaller set of filters, while connecting different sets of filters in a hierarchical residual-like manner. Each 3x3 convolution operation can potentially accept all the feature information on its left side, and each output can increase the receptive field, so each Res2Net can acquire a different number of feature combinations with different receptive field sizes. kn as the scale dimension control parameter indicates that there are n sets of filters, and the larger the n filter, the more filters allow more features with richer receptive field sizes to be In the Res2Net block, the hierarchical residual connectivity in a single residual block allows the variation of the receptive field at a finer granularity level to capture details and global properties. For the implementation of Fig 2b, the feature vector is split by torch-split, and the first branch is output directly without 3x3 convolution, and each subsequent branch is cat with the next branch after 3x3 convolution, and then part of the output goes down and part of the output is cat with the next branch. The number of channels is the width*scale, and finally a 1x1 convolution is added for feature fusion again. The core structure of Res2Net is completed, and there are two hyperparameters, width and scale, even though this is a similar way to FPN, replacing the single 3x3 convolution of ResNet with such a relatively finer-grained convolution, but the number of actual parameters does not increase because of the splitting of channels.

3. Centerline extraction algorithm

3.1 Polynomial-based trajectory fitting

At the beginning of the study, an attempt was made to fit the extracted left and right trajectories based on the quadratic polynomial method with sliding windows [14-15], but this method was abandoned considering that the neural network already takes up part of the resources and there may be some impact on the real-time performance if a more computationally intensive algorithm is used to extract the centerline.

Inspired by this using contour detection and scatter plot three times curve fitting, the process is shown in Fig 3 below.

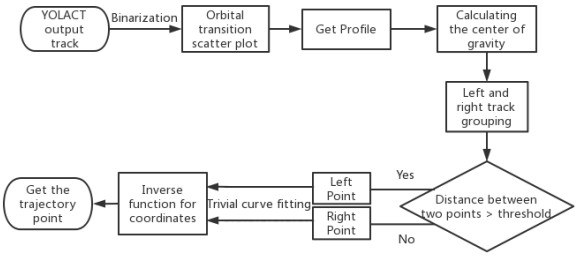
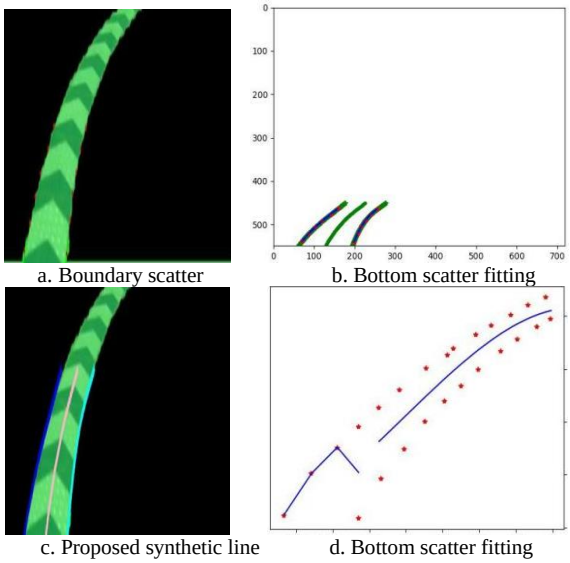


Fig.3: Scatter fitting

The detection effect is shown in Fig. 4. Although the fit is better for the lower half of the image field of view as in Fig. 4b and 4c, the tracks near the distant overlap points cannot be fitted well as in Fig. 4d.



a. Boundary scatter b. Bottom scatter fitting

c. Proposed synthetic line d. Bottom scatter fitting

Fig.4: Scatter fitting effect

3.2 Fast point finding algorithm based on HSV + Canny

HSV (Hue, Saturation, Value) is a color space created by A. R. Smith in 1978 based on the intuitive properties of color, also known as the Hexcone Model. In the HSV model, color is composed of Hue, Saturation, and Value. The algorithm converts the RGB color space into HSV color space by segmenting the track instances. Let (r,g,b) be a color coordinate with value $\in (0,1)$, max denotes the maximum value in r, g, b, and min denotes the minimum value in r, g, b. The conversion formula is as follows:

$$H = \begin{cases} 0^\circ, & \max = \min \\ 60^\circ \times \frac{g-b}{\max - \min} + 360^\circ, & \max = r \ g \geqslant b \\ 60^\circ \times \frac{g-b}{\max - \min} + 360^\circ, & \max = r \ g < b \\ 60^\circ \times \frac{g-b}{\max - \min} + 360^\circ, & \max = g \\ 60^\circ \times \frac{g-b}{\max - \min} + 360^\circ, & \max = b \end{cases}$$

$$S = \begin{cases} 0, & \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & otherwise \end{cases}$$

$$V = \max$$

First, the output instance object from the network is passed into the algorithm, and the target object after eliminating the background is obtained by color space conversion and binarization, then the edges are extracted from the target using the Canny operator, then each coordinate point is quickly found by the function np.where() and stored in the list, and finally each midpoint is obtained by using the midpoint coordinate formula to obtain the Centerline. The detection effect and flow chart are shown as follows:

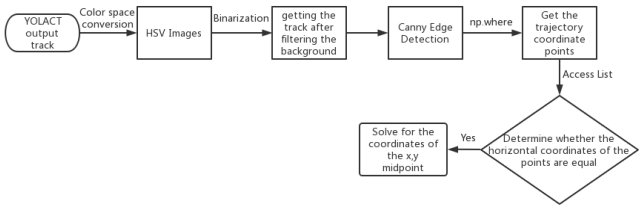


Fig.5: Middle line extraction algorithm flow

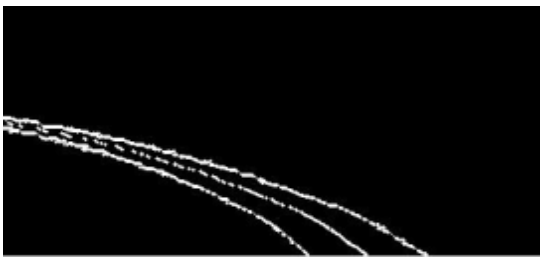


Fig.6: Effect of midline extraction algorithm

4. Experiment and performance analysis

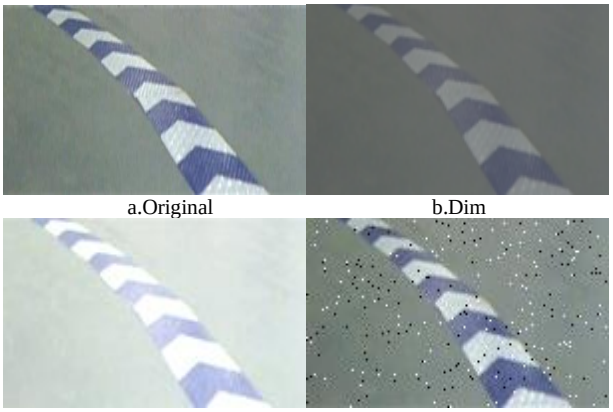
4.1 Res2Net-YOLACT + HSV split extraction track experiment

4.1.1 Experimental platform and data

The experimental platform for this study is conFigd with Win10, Intel(R) Core(TM) i5-10300H CPU, NVIDIA GeForce GTX 1650 Ti, CUDA11.6 and cuDNN8.3.2 for GPU acceleration, and Python3.8 and torch1.12.0 for the experimental runtime environment.

The experimental data were automatically collected by the smart car on-board camera, and 500 shots were taken on the straight and curved roads according to the fixed trajectory, and the data were expanded as shown in Fig 7. The unclear images were removed and finally 2117 images were divided 9:1 as the data set, of which 1905 were used as the training set and 212 as the test set, and then the images were resized to 640x480 size by OpenCV.

The images were then manually labeled by Labelme to obtain the dataset in VOC format.



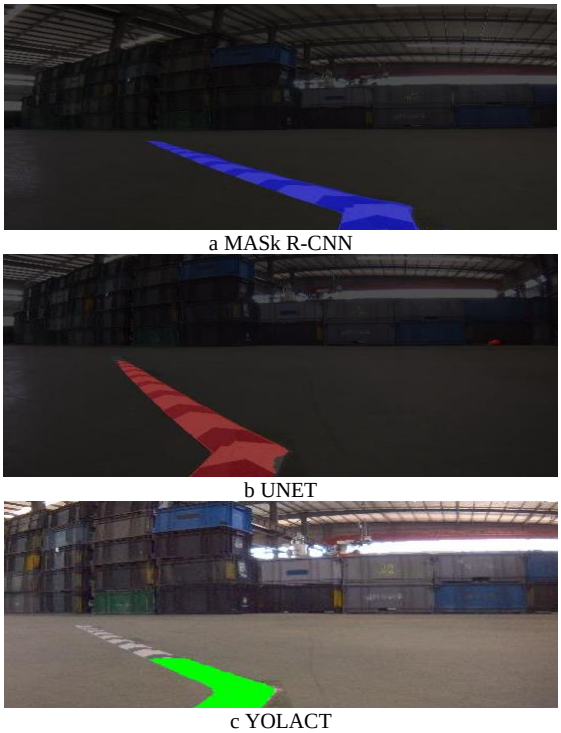
a.Original b.Dim

c.Bright d.Salt and pepper noise

Fig.7: Data enhancement

4.1.2 Network model and algorithm testing

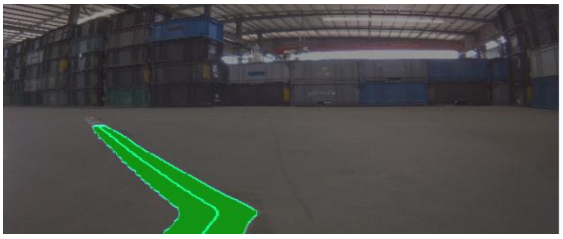
To test the performance, the research YOLACT network in this paper is compared with the original network and Unet [16-17] and Mask R-CNN [18-19] networks, and the sample detection effect is shown in Fig. 8. The computational results are shown in Table 1.



a MASk R-CNN

b UNET

c YOLACT



d Res2Net-YOLACT + HSV

Fig.8: Four segmentation algorithms to detect the effect

Table 1: Comparison of different algorithm metrics

algorithm	*MIOU/%*	*F1-Sorce/%*	*Image Inspection(ms)*
Mask R-CNN	97.85	81.38	50.32
Unet	96.07	52.24	44.15
YOLACT	95.12	78.44	35.64
Res2Net YOLACT HSV	97.23	80.73	37.26

Combining the detection effect and performance index, the same image Mask R-CNN can segment all trajectories very accurately, but due to its complex network structure, the detection speed is slow and cannot achieve real-time effect. Compared with the YOLACT network with Resnet101 as the backbone, the correct segmentation rate of YOLACT recognition improved to Res2Net101 in this study improved by 4.09%, and the speed of detecting a single image improved by 5.38ms, which is more lightweight. Compared with Unet network and Mask R-CNN, the detection speed and accuracy of Res2net-YOLACT network are better than the former three.

4.1.3 Real-time detection

This experimental platform relies on the Ackermann model intelligent vehicle, through the vehicle camera real-time acquisition and recognition of the captured image directly, running on its processing equipment Jetson Nano, multiple experiments to calculate the average time and average frame rate used to identify different videos and cameras in real-time detection under different conditions, the results are shown in Table 2, all meet the practical application requirements and detection accuracy.

Table 2: Comparison of different algorithm metrics

Test sample	*Test condition*	*Contains tracks*	*Accurate frames*	*False frames*	*Average FPS*
Video1	General	300	295	5	28.21
Video2	Strong light	300	287	13	27.77
Video3	Low light	300	296	4	28.62
Capture	General	300	292	8	27.46
Capture	Strong light	300	283	17	27.51
Capture	Weak light	300	294	6	27.93

5. Conclusion

For similar smart trajectory scenarios such as unmanned smart warehouse and handling dock, this research proposes a YOLACT-based trajectory segmentation method, which can achieve real-time extraction of trajectory centerline by improving the network backbone and adding fast point finding algorithm. The study has higher detection frame rate, higher detection accuracy and lower miss detection rate compared with the network before improvement. It can reach about 30fps on the embedded platform jetson nano, which meets the practical requirements. In subsequent work, we combine it with the control system to achieve efficient applications on fewer resources.

References

[1] Liu Zhaoling,Tang Hongyu. Research on the development of intelligent logistics storage experiment system based on Internet of Things technology [J]. China Aviation Weekly. 2022, (19).

[2] Hu Xiaoxu. Research on the job substitution effect of artificial intelligence on logistics industry[J]. Journal of Economic Research. 2022, (25).

[3] Liu, Chunhua. Research on the impact of "artificial intelligence+" on the economic development of logistics industry[J]. China Storage and Transportation. 2022, (07).

[4] Huang Yu, Bo Na. Research on the application of intelligent logistics distribution in automobile production[J]. Automotive Technician. 2022, (06).

[5] Xi Ruizi, Zhang Feng. Analysis of logistics and transportation scheduling system based on multi-intelligent body technology[J]. Logistics Engineering and Management. 2017, 39(05).

[6] Sun Pu-Xi. Jingdong logistics:delivery robot and drone technology in intelligent logistics system[J]. Robot industry. 2022, (05).

[7] Wang Xiaojuan, Li Yunwu Liu Dexiong. Sun HW Huang XUEYAN. Virtual median extraction method of field roads in hilly mountainous areas based on machine vision[J]. Journal of Southwestern University (Natural Science Edition) . 2018, 40(04).

[8] Nie Sen, Wang B-Long, Hao Huanhuan, Chen Jun. Research on line extraction algorithm for orchard navigation based on machine vision line extraction algorithm based on machine vision[J]. Agricultural mechanization research. 2016, 38(12).

[9] QI Hao, LI Yongting, QI Yongsheng, LI Qiang Liu. A novel lightweight semantic segmentation network for track and obstacle detection[C]. Proceedings of 32nd Chinese Process Control Conference.

[10] Ruan Shun-Ling, Jiao Xin, Jing Ying, Lu Cai-Wu, Gu Qing-Hua. An unstructured road segmentation detection method for open pit mining areas [J]. Survey and Mapping Science. 2022, 47(06).

[11] Jing X.H., Sun G.D., He S.B., Liao Y. A time-varying channel estimation method based on sliding window filtering and polynomial fitting[J]. Computer Applications. 2021, 41(09).

[12] Liu JC, Wang TQ, Chen ZH, Zhang P, Zhu TD, Zhang SQ, Wang Y. Residual anomaly separation in gravity potential field region using polynomial fitting algorithm with sliding window[J]. Journal of Wuhan University (Information Science Edition) . 2018, 43(10).

[13] Li Jiaojiao, Liu Zhiqiang, Song Rui, Li Yunsong. A remote sensing image segmentation algorithm with improved Unet network[J]. Journal of Xi'an University

of Electronic Science and Technology, 2022-09-29 1:17.24
[14] Li Xiaoling, Liu Guangzhong, Qiao Dare. Application of improved Mask RCNN in sea surface ship instance segmentation[J]. Ship Engineering, 2021, 43(12).
[15] Xie S.-P., Li B., Zhang D.. Semantic segmentation of distribution lines based on improved Mask- RCNN[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition), 2021, 41(06).