Efficient Post-Contour Correctness in Object Detection and Segmentation

Than D. Le, IEEE Member
Faculty of Computer Science
University of Bordeaux
Bordeaux, France
than.ld@ieee.org

Faculty of Computer Science
Ton Duc Thang University
Ho chi Minh City, Vietnam

***Abstract*—In this paper, we propose the simple method to optimize the datasets noise under the uncertainty applied to many applications in industry. Specifically, we use firstly the deep learning module at transfer learning based on using the mask-rcnn to detect the objects and segmentation effectively, then return the contours only. After that we address the shortest path for reduce the noise in order to increasing the high-speed in industrial applications. We illustrate adaptive many applications web applications such as mobile application where power computer is limited a source.**

***Index Terms*—object segmentation, uncertainty, contour correctness, deep learning, transfer learning, Mask-RCCN.**

I. Introduction

Deep learning is currently a resolution in many applications such as YOLO3 [11] which is a module illustrates the very high speed in real-time tracking and object detection while Mask-RCNN [3] approach shows convenient way for representing the objects masking with the robust segmentation. It is recently one of the most challenge in computer vision and image processing nowadays. Most of autonomous robotic systems [10], motion planning [10], [5] , [18] and human-robot interaction [4], [1]) is need to clearly position and orientation (poses) to interact with unknown environment. However, there are many constraints to apply in robot applications.

Transfer learning is currently one of most approach applying in the industry. For instance, COCO dataset [6] is used to train most of experiments in efficient way. Hence, Mask-RCNN [3] is inherited to use as transfer training as supervised learning approaches based on COCO, ImageNet [14]. In this case, it will be token many time for pre-processing data such as labelling, data augrumentation [15]. In order to solve it, there are currently three solutions to improve active contour features and uncertainty problem. Firstly, we need to label carefully the object expectation based on representing the particular characteristic features. Secondly, we will need to change the network. For instance, we need to change the Resnet such as ResNet 50, 101, and 152, respectively. Finally, we can apply the shortest distance to predict quickly in order to improve the accuracy.

Bounding box 15did not any care from any researcher, most of them focusing on deep network design and data science to make efficient, and effectively it is impact precisely to most applications such as human-robot interaction, grasp and manipulation, autonomous self-driving []. Unfortunately, there is very limited on research to focus on contour feature extraction [12], and [19]. Hence, we are proposal a simplest way to correct the contour features based on predicting from bounding box and masking result from Mask-RCNN module. It is probably only applied to Mask-RCNN, where we can get the contour feature based on resulting from detection and segmentation that there are no module can be applied other deep learning modules like [11], [20], [22] (previous module of Mask-RCNN), etc.

In this paper, we extended our performance by representing the Contours Features attribution to improve the bounding box applying the deep learning in order to apply robot applications, (e.g. human-robot interaction [4], [1]). Firstly, we will explore our Mask-RCNN framework based on upgrading the network structure by using the Resnet 152 to improve our performance. Secondly, we use the shortest path by the correctness to modify the bounding box. Additionally, we also illustrate the Video Detection and Segmentation to improve our accuracy effetely.

In structure of paper, section I are already covered the introduction. Next, we will discuss about uncertainty based on representation. And then, we will discuss distance path based on describing the improvement both bounding box and contour feature in section 3. After that, it is addressed for experiments based on robot and mask-rcnn for detection and segmentation.

II. Uncertainty Representation

From raw data, the line extraction can create features. Firstly, Features are much more compact than raw data, and can reflect physical or abstract objects. Moreover, it is rich in information and can be able to assess accuracy of feature

A. Line Segment Extraction

While line extraction needs to estimate the line parameters by given points belong to lines, the segmentation problem is to be answering both many lines and data points in lines for solving the line extraction problems.

Let consider a problem by given describing in Figure 2 a given set of points by a measurement vector of tuples. In this case, it is a set of bounding box, and formalized by

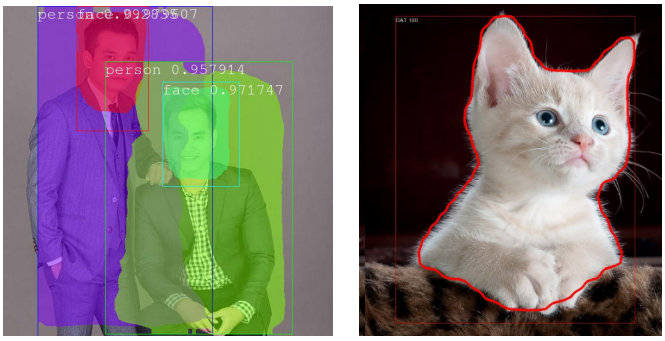


Fig. 1: Mask-RCNN Object Detection and Segmentation: (LEFT) Object Segmentation and Bounding Box Problems; (RIGHT) error localization and bounding box based on feature extraction.

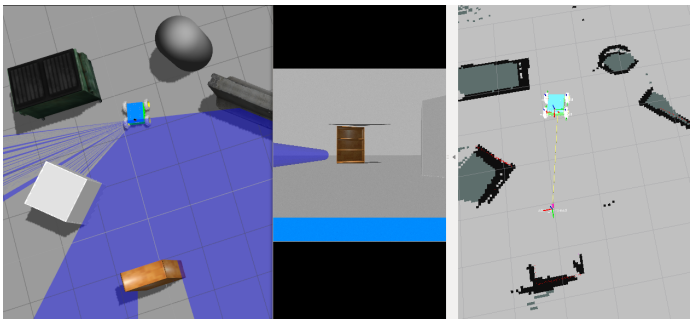


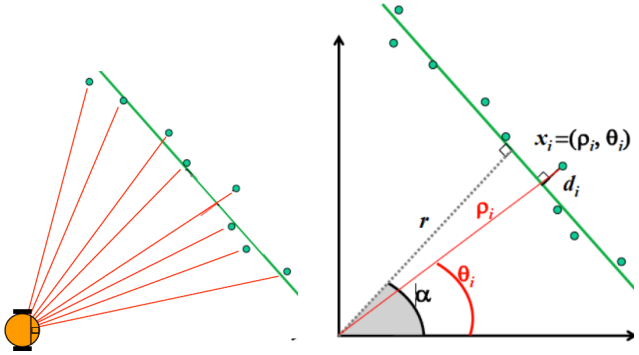
Fig. 2: Example of a raw data mapping from Sensor.

$x_i(\rho_i, \theta_i)$, with $i = 1 \cdots, N$. Others, we need to define two angles α and θ_i according to given Figure: 3

There is a constraint for applying the linear equation based on the measurements.

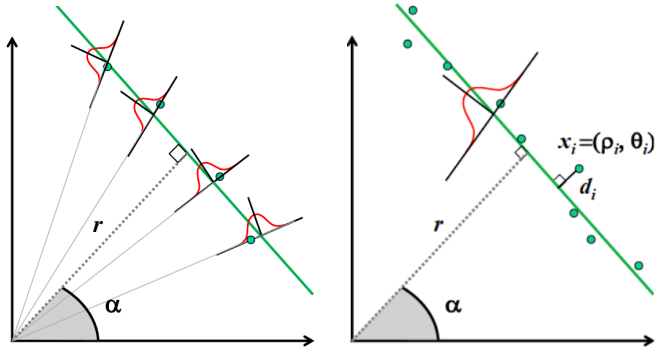
$$\rho_i cos(\theta_i - \alpha) r \tag{1}$$

But in the real world, there are many noisy by the measurements updating, thereby set of points will be simplify to the distance d_i given by the line:



(a) Problem (b) A Solution

Fig. 3: Line Segment Extraction [17]



(a) Problem (b) A Solution

Fig. 4: Line Segment Extraction [17]

$$\rho_i cos(\theta_i - \alpha) - r = d_i \tag{2}$$

It represents as a solution called Minimize Sum of Squared Errors (MSSE). Generally, we can find to minimize the error function by formalization:

$$S = \sum_i w_i d_i^2 = \sum_i (\rho_i cos(\theta_i - \alpha) - r)^2 \tag{3}$$

It shows the solution on Figure: **??** by the derivative of equation separately with respective to each parameter, including α and r. So, it will be equal:

$$\frac{\partial S}{\partial \alpha} = 0 \tag{4}$$

and

$$\frac{\partial S}{\partial r} = 0 \tag{5}$$

Another parameter will be given by:

$$w_i = \frac{1}{\delta_i^2} \tag{6}$$

There are two cases in this situation. It usually called as unweighted Least Square Error. On the other hand, according to each measurement, there is the way by using the associated error variance function illustrated better results. And the result of α and r is:

$$r = \frac{\sum w_i p_i cos(\theta_i - \alpha)}{\sum w_i} \tag{7}$$

and

$$\alpha = \frac{1}{2} atan \left(\frac{\sum w_i \rho_i^2 sin2\theta_i - \frac{2}{\sum_{w_i}} \sum \sum w_i w_j \rho_i, \rho_j cos\theta_i sin\theta_j}{\sum w_i \rho_i^2 cos2\theta_i - \frac{2}{\sum_{w_i}} \sum \sum w_i w_j \rho_i, \rho_j cos\left(\theta_i + \theta_j\right)} \right) \tag{8}$$

Let to consider two conditions below:

$$\rho_i \sim N\left(\widehat{\rho}_i, \delta_i^2\right) \tag{9}$$

and

$$\theta_i \sim N\left(\widehat{\theta}_i, \delta_{\theta_i}^2\right) \tag{10}$$

The covariance matrix shows the uncertainty based on the line by given two parameter above:

$$C_{\alpha r} = F_{\rho\theta} C_x F_{\rho\theta}^T \tag{11}$$

It must be to define the Jacobian as:

$$F_\rho = \begin{bmatrix} \frac{\partial \alpha}{\partial \rho_i} & \frac{\partial \alpha}{\partial \rho_{i+1}} & \cdots & \frac{\partial \alpha}{\theta_i} & \frac{\partial \alpha}{\theta_{i+1}} \\ \frac{\partial r}{\partial \rho_i} & \frac{\partial r}{\partial \rho_{i+1}} & \cdots & \frac{\partial r}{\theta_i} & \frac{\partial r}{\theta_{i+1}} \end{bmatrix} \tag{12}$$

$$C_{x_i} = \begin{bmatrix} \delta_{\rho_i}^2 & 0 \\ 0 & \delta_{\rho_i}^2 \end{bmatrix} \tag{13}$$

If there is independent between ρ_i and θ, we can define the function for c_x

$$C_x = \begin{bmatrix} diag(\theta_\rho^2) & 0 \\ 0 & diag(\theta_\rho^2) \end{bmatrix} = \begin{bmatrix} \delta_{\rho_i}^2 & 0 & \cdots & 0 & 0 \\ 0 & \delta_{\rho_{i+1}}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \delta_{\theta_i}^2 & 0 \\ 0 & 0 & \cdots & 0 & \delta_{\theta_{i+1}}^2 \end{bmatrix} \tag{14}$$

B. Shortest Distance Algorithms

There are currently we used to two approaches for solving the correctness contour problems given by Figure

1) Split-and-Merge Algorithms: It is used to the recursive procedure of fitting and splitting function. Detail about pseudocode of algorithms is described in Agl: 1

C. RANSAC

We can determine the number of iterations in RANSAC by formulizing equation:

$$k = \frac{log(1-p)}{log(1-w^2)} \tag{15}$$

Where: $w =$ is the percentages of inliers

Types	Commplexity	Speed	Accuracy
Spit-and-Merge	n log n	1500 (Hz)	90 %
RANSAC	s n k	30 (Hz)	10%

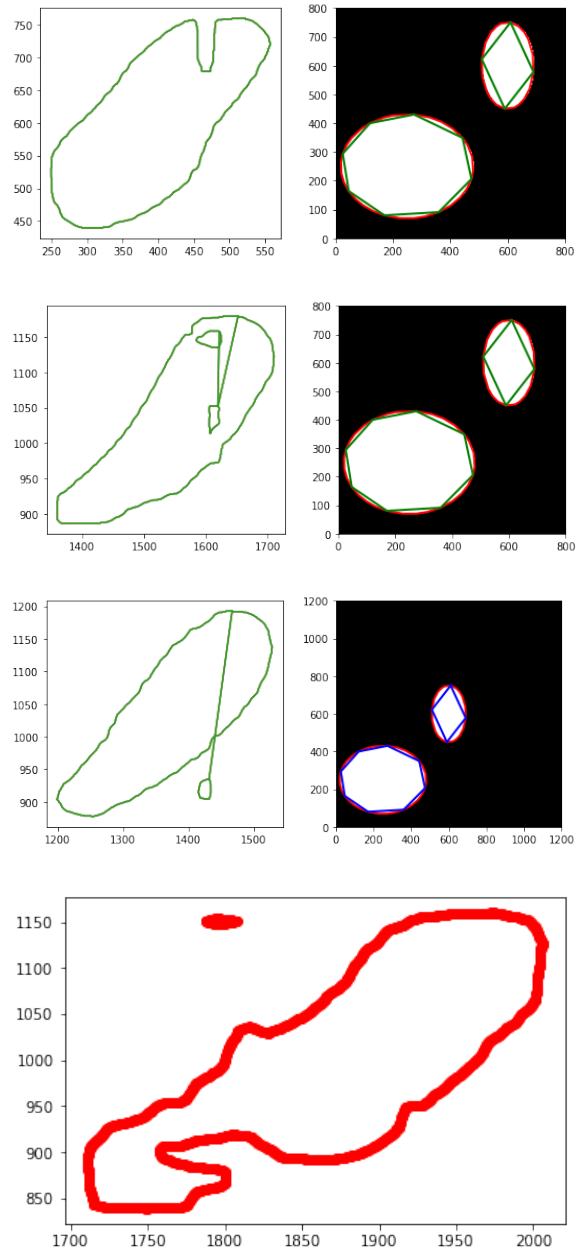


Fig. 5: Sample for Noise Datasets based on resulting from Mask-RCNN Object Detection and Segmentation

D. Faster R-CNN and Mask R-CNN

E. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. **??**", even at the beginning of a sentence.

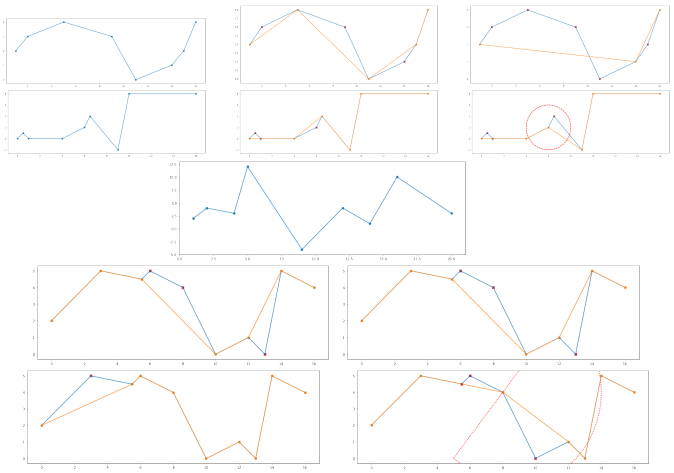


Fig. 6: Split-and-Merge: There are examples of iterative end point for solving contour feature

Algorithm 1: Splitting-and-Merging()

begin
  Initial set s_1 consists of N points;
  Fitting a line to the next set s_i in L ;
  Detect point P with maximum distance d_ρ to the line
    if $d_\rho < threshold$ **then**
      continue (go to step 2);
    else
      Splitting s_i at P into s_{i1} and s_{i2};
      Replace s_i in L by s_{i1} and s_{i2};
      Continue (go to 2);
  while $True$ **do**
    `/* Calculate the path distance */`
    while *all sets (segments) in L have been checked* **do**
      merge col-linear segments;

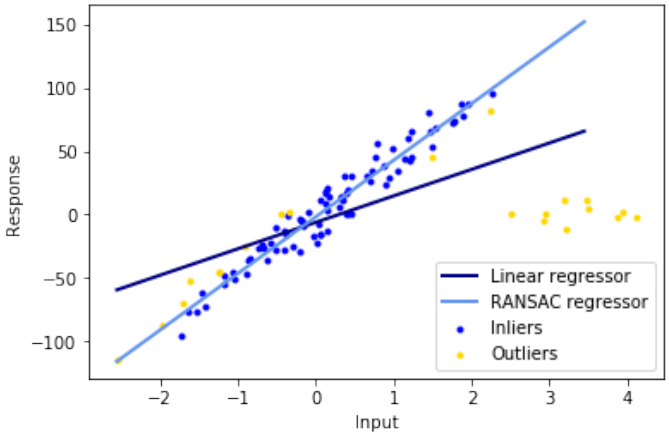


Fig. 7: RANSAC for Line Segment Extraction.

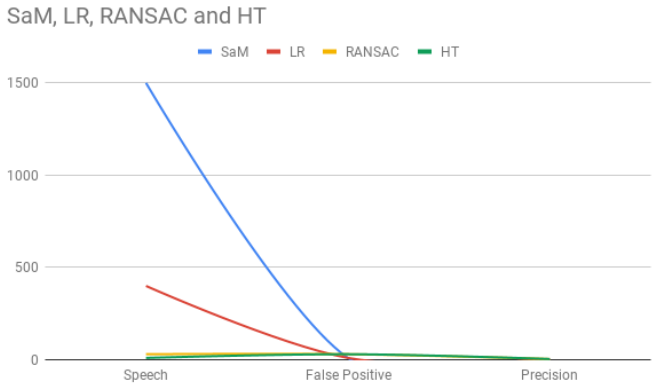


Fig. 8: Comparison of Line Segment Extraction.

III. Object segmentation with Mask-RCNN

In this section, we focus on describing Mask R-CNN architecture and how it performs object segmentation. Prior to that, we also discuss Faster R-CNN, a neural network object detector from which Mask R-CNN stems. Objects in general can be things in various shapes, depending on what type of dataset is fed to train the neural network model.

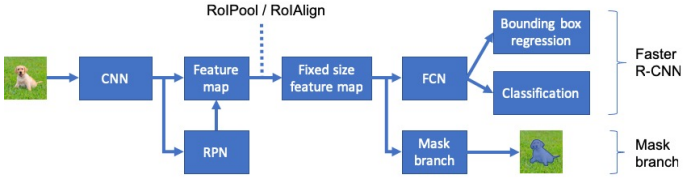


Fig. 9: Mask R-CNN architecture.

A. Faster R-CNN

Faster R-CNN is a deep neural network designed for multi-class object detection and introduced by Ren *et al.* [20]. It consists of two main modules: a Region Proposal Network (RPN) followed by the Fast R-CNN [22]. The region proposal module is a convolutional network fed with an image from which it extracts features and returns locations where the object lies. These areas will be further analyzed by the Fast R-CNN detector to determine object type (classification) and to adjust rectangular bounding boxes (regression) to better fit the object shape. The system loss function $\mathcal{L}$ is a combined loss of classification $\mathcal{L}_{cls}$ and regression $\mathcal{L}_{box}$:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} \tag{16}$$

Thanks to the share of convolutional feature map at classification, regression and RPN stage, the Faster R-CNN is $faster$ than Fast R-CNN and therefore it requires less computational effort.

B. Mask R-CNN

Mask R-CNN [2] is extended from Faster R-CNN. Besides the class label and the bounding box offset, the Mask R-CNN is able to detect shape of objects, called object $mask$.

This information is useful for designing high-precision robotic systems, specially autonomous robotics grasping and manipulation applications. The general loss function $\mathcal{L}$ considers the mask loss $\mathcal{L}_{mask}$:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} \tag{17}$$

Additionally, the Mask R-CNN can achieve a high pixel-level accuracy by replacing RoIPool [22] with RoIAlign. The RoIAlign is an operation for extracting a small feature map while aligning the extracted features with the input by using bi-linear interpolation. Reader may refer to the paper [3] for further detail. To train the detector, we reuse a Mask R-CNN implementation available at [2].

IV. Experiments

There are many approaches we are using to test our performance increasingly. To apply in robotics applications, it is explained more in [4], and [1] that based on given architecture describing in Figure:**??**.

A. Training Strategies

We intend to change our experiments by concentrating the pre-processing datasets, based on focusing supervised learning method. By the way, we using data argumentation for

B. Mask-RCNN for Video Detection and Segmentation

In this work we expend our experiments by demonstrating the Video based on Cat and Dog problems.

C. Visualization

Visualization for understanding deep learning is currently the challenge in research activities. One of the problems is the overfitting, and it is difficult to handle and manage. In this paper, we tryied to understand our dataset by using precision. Figure: 13 as an example which is visualized result by taking from 7200 image with respect to around 17 class

More visualization is explained by this paper [1].

For Contour feature extration, we have visualized in Figure:

In summary, we need to integrate currently three approaches proposed together because deep learning still have problems with segmentation due to noise datasets from prediction. For instance, we expect to detect and segment many tiny objects and quit similar features.

RESNET Types	Accurccy	Loss	Summary
RESNET 50	80-95 %	0.06	Not refered.
RESNET 101	94 - 99 %	0.05	Refer to used.
RESNET 152	94-99 %	0.04	Not refered.

Table above shows that there are not much different by using ResNet 101 and 215. Hence, we are referred to Resnet 101 used to training our model. Basically different here, ResNet 215 usually expended the size, and implied to slightly low when we implement in any API applications or Robotics system.

- ResNet 50: It is very poor result on object segmentation when we increase the complexity of the datasets.

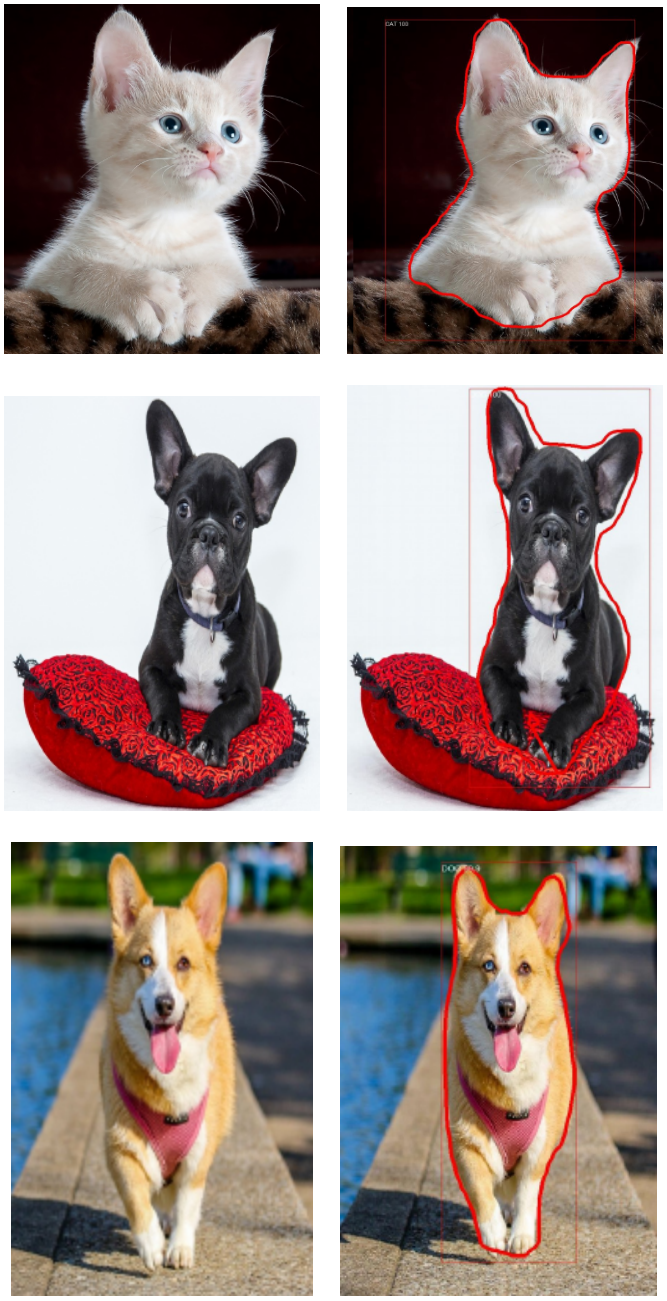


Fig. 10: Top and Middle: There are existed errors from masking. Bottom: Contour Feature is correct by an extraction using Flitting-and-Merging Algorithm

Algorithm 2: RANSAC()

begin
  Given: Point cloud P and model estiamation routine;
  Output:Model $\widehat{M}$ which was rated best amongst all interations ;
  Detect point P with maximum distance d_ρ to the line
  while $maxIterationsnotreached$ **do**
    sample k point;
    estimate a model M;
    compute model inliers;
    while *contraint c in contraints* **do**
      if $d_\rho < threshold$ **then**
        update maxInterations;
        continue;
    if *M is better than bestModel* **then**
      save M as bestModel;
      update maxInterations;
    update iterations;
  return bestModel;

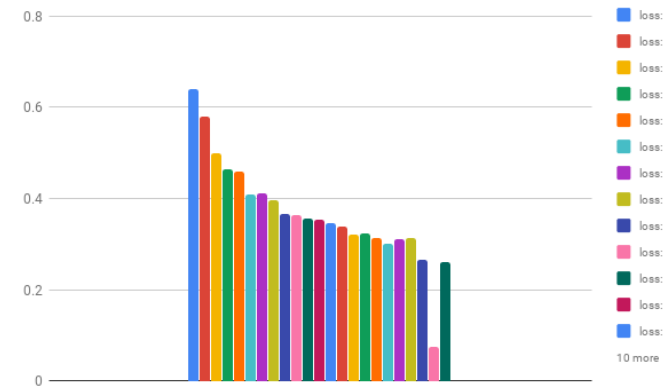


Fig. 11: Generative Training: Total loss is decreased by using epoch 1000. And we totally got minimum for our model

Fig. 12: Mask-RCNN Object Detection and Segmentation for Video Tracking: Object Segmentation and Bounding Box Solution .

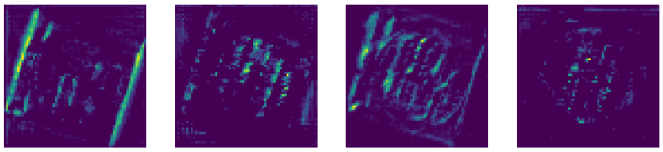


Fig. 13: Visualization for Activation Function based on 7200 images.

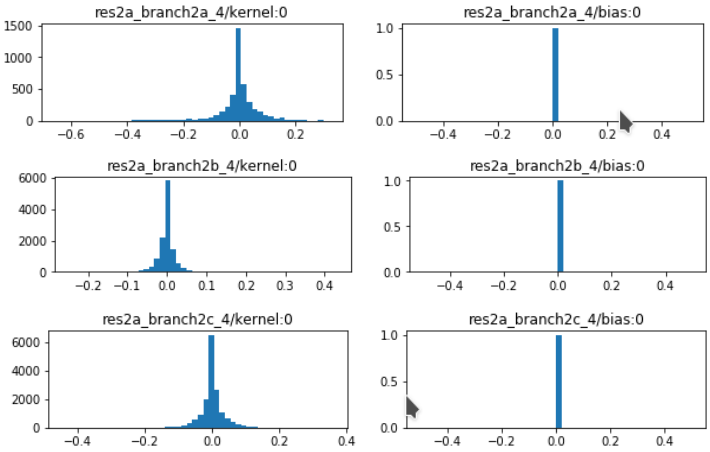


Fig. 14: Histograms for Object Segmentation .

- ResNet 101: It is good result in average by comparing with other remain ResNets, even the total loss is greater than ResNet 152.
- ResNet 150: The good thing here it contains smallest loss parameter. However, it is heavy the sizes since it take slow speed to integrate to robot applications.

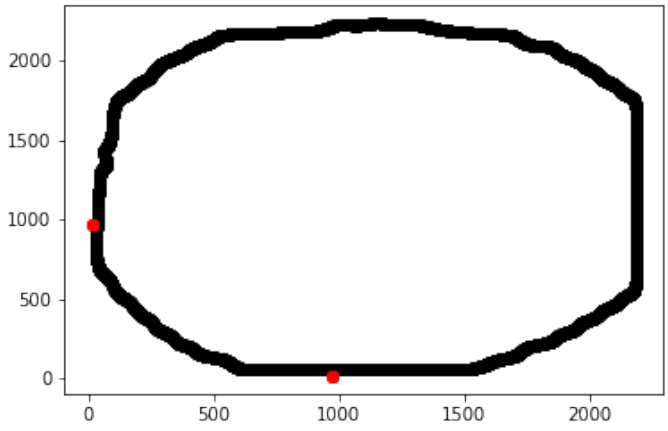


Fig. 15: Contour extraction after correctness. It can be corrected around 90-99 %.

V. Conclusion

In this paper, we address the uncertainty which is illustrated in deep learning approaches, and Mask-RCNN module is probably suitable solutions for human-robot interaction and grasp manipulator systems. It shows that integration between bounding box and masking can be able to increase the accuracy of bbox that in never applying to any deep learning modules without mask. We also conclude that RANSAC were not suitable our situation which is determine the distance path, and it must be replaced by

In the future work, we would to extend to our semi-supervised and unsupervised learning for improvement the object segmentation. It will be able to solve the moving objects and increase the segmentation.

References

[1] Than D. Le, Dang T. Huynh, Huy V. Pham, "Efficient Human-Robot Interaction using Deep Learning with Mask R-CNN: Detection, Recognition, Tracking and Segmentation," 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). London, vol. A247, pp. 529–551, April 2018.

[2] Waleed Abdulla, 'Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow',Github, GitHub repository,2017.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollr, Ross Girshick, 'Mask R-CNN', The IEEE Conference on Computer Vision, 2017.

[4] Quan H Nguyen, Trinh NP Tran, Dung D Huynh, An T Le, Than D Le, 'Real-Time Localization and Tracking System with Multiple-Angle Views for Human Robot Interaction', Robotic Computing (IRC), IEEE International Conference on, 2017, pp.316-319.

[5] Than D Le, Duy T Bui, VanHuy Pham, "Encoded Communication Based on Sonar and Ultrasonic Sensor in Motion Planning," IEEE Conference on Sensor, 2018, pp. 271–350.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollr, 'Microsoft COCO: Common Objects in Context,' The Conference on Computer Vision and Pattern Recognition, IEEE 2014.

[7] Than D. Le, An T. Le, anDuy T. Nguyen, "International Conference on Advanced Robotics (ICAR)," International Conference on Advanced Robotics (ICAR), IEEE 2017.

[8] An T. Le, Minh Quang Bui and Than D. Le, "D* Lite with Reset: Improved Version of D* Lite for Complex Environment," IEEE International Conference on Robotic Computing (IRC), IEEE 2017.

[9] "'Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography "

[10] An T. Le, Than D. Le, "'Search-Based Planning and Replanning in Robotics and Autonomous Systems"', Path Planning and Navigation, Chapter 4, IntechOpen, 2018. DOI: 10.5772/intechopen.71663.

[11] Redmon, Joseph and Farhadi, Ali, "'YOLOv3: An Incremental Improvement"', arXiv, 2018.

[12] Diego Marcos, Devis Tuia, Benjamin Kellenberger, Lisa Zhang, Min Bai, Renjie Liao, Raquel Urtasun, "'Learning Deep Structured Active Contours End-to-End"', The Conference on Computer Vision and Pattern Recognition, IEEE, 2018.

[13] Khiem N Doan, An T Le, Than D Le, Nauth Peter, 'Swarm Robots Communication and Cooperation in Motion Planning', Chapter 15, Mechatronics and Robotics Engineering for Advanced and Intelligent Manufacturing, Springer, 2017.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. 'ImageNet Large Scale Visual Recognition Challenge,' International Journal of Computer Vision, 2015.

[15] Luis Perez, Jason Wang, The Effectiveness of Data Augmentation in Image Classification using Deep Learning, ArXiv, 2017.

[16] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 'Understanding Neural Networks Through Deep Visualization', ICML Deep Learning Workshop, 2015.

[17] https://www.cs.princeton.edu/courses/archive/fall11/cos495/COS495-Lecture11-LineExtraction.pdf

[18] Khiem N. Doan, An Thai Le, Than D. Le and Nauth Peter, 'Swarm Robots Communication and Cooperation in Motion Planning,' Chapter 15, Mechatronics and Robotics Engineering for Advanced and Intelligent Manufacturing, Springer 2017.

[19] Yihui He, Xiangyu Zhang, Marios Savvides, Kris Kitani, 'Softer-NMS: Rethinking Bounding Box Regression for Accurate Object Detection,' arXiv, 2018.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,' Advances in Neural Information Processing Systems, 2015.

[21] John Hershberger and Jack Snoeyink, 'Speeding Up the DouglasPeucker Line-Simplification Algorithm', Proc 5th Symp on Data Handling, 134143 (1992). UBC Tech Report TR-92-07.

[22] Ross Girshick, 'Fast R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', International Conference on Computer Vision (ICCV), IEEE, 2015.