RESEARCH ARTICLE

Efficient Human-Robot Interaction using Deep Learning with Mask R-CNN: Detection, Recognition, Tracking and Segmentation

Than Le[1], Dang Huynh[2]

[1] IEEE member, University of Bordeaux, French

[2] Axon Enterprise, French

Abstract: We address social human-robot interaction problem by proposing an integration of deep neural network with mechanical robotic system to make it robust for human-robot interactive activities. Mask R-CNN, a neural network for object detection, can effectively help localize human faces which can be manipulated to instruct movement of the robot head. Our approach is not only suitable for detection and segmentation tasks but able to integrate as well with the mechanism of parallel mini-manipulator representing the 3D dimensions, in position and orientation of workspace. It can also solve the object segmentation problem which appears to be one of the most challenging issues in computer vision nowadays.

Keywords: Human robot interaction; deep neural network; tracking robotics; detection; mini-parallel kinematic

1. Introduction

In the past decade, indoor and outdoor problems of autonomous mobile manipulator and mobile robotics [21], [22], [27] in uncertain environments with probabilistic robotics serve a crucial role in this field of study. Researchers often focus on either software or mechanical solutions. However, an interdisciplinary system may take advantage of both worlds where advancement of algorithms and mechanics can mutually help each other. Such a perspective can be applied to build an efficient interaction system between human and robot, called Human-Robot Interaction (HRI). This field of study consists in designing a robotic system for use by or with humans. On the other hand, safety awareness in human-robot interaction [26] is one of the topics attracting the most interest of researchers in industry context where potential and readiness to learn and transfer technology are essential.

Deep neural networks have shown an ability to outperform traditional feature-based approaches in computer vision tasks, for instance detection, recognition and segmentation [9], [10], [25], [32]. As an attempt to improve the human robot interaction efficiency, we propose a use of Mask R-CNN [11] as a module that can detect and segment human face and body. In a previous work [12], we described how to build a stable dynamic system with multiple-angle views for camera calibration in human-robot interaction, based on real-time localization and tracking. In this paper, we continue the work by integrating the Mask R-CNN to meliorate the system. We show that by exploiting deep neural network, the robotic system works accurately when interacting with human via detection, tracking and recognition.

The paper is organized as follows. In Section II, we give an overview of neural network object detectors and introduction of face detection. Section III consists of robotics control system architecture description, Robot Operating System (ROS) set-up and a summary on how to train and evaluate the performance of Mask R-CNN. Finally Section IV concludes the paper.

doi: 10.18063/phci.v1i2.783

2. Deep neural network approach for object detection

In this section, we discuss two recent techniques, Faster R-CNN and Mask R-CNN, used for object detection. Object in general can be things, face, human, etc., depending on what type of dataset is fed to train the neural network model.

2.1 Faster R-CNN

Faster R-CNN is a multi-class object detection system in- troduced by Ren *et al*. [10]. It consists of two main modules: a regional propose network followed by the Fast R-CNN [9]. The regional proposal module is a fully convolutional network that takes an image as input and returns locations at which objects probably are. These areas then will be deeply 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 L is a combined loss of classification L_{cls} and regression L_{box}:

$$L = L_{cls} + L_{box} \tag{1}$$

Compared to Fast R-CNN, this approach is faster because the convolutional feature map is shared in both stages: first in candidate object proposal, then in classification and regression. Therefore, it requires less computational effort.

2.2 Mask R-CNN

Mask R-CNN [11] 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 L considers the mask loss L_{mask}:

$$L = L_{cls} + L_{box} + L_{mask} \tag{2}$$

Additionally, the Mask R-CNN can achieve a high pixel-level accuracy by replacing RoIPool [9] 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 [11] for further detail. To train the detector, we reuse a Mask R-CNN implementation available at [30].

2.3 Face detection

Face detection is crucial for many face applications such as face recognition and video analysis. It also plays an important role in human-robot interaction.

Face detection problem has attracted a lot of interest over years. Studies have been done extensively and many algorithms have been proposed. Traditional method exploit features like HOG [1], [2] and Haar [3], [4] while deep neural network approaches [5], [6], [7] can learn the features from a sufficiently large dataset. In other words, one is feature-based and the other is featured-learned. It has been experimentally shown that neural network significantly improves accuracy of detection tasks, not only for face but for any object type. In this paper if not otherwise mentioned, we exploit Mask R-CNN to detect faces and general objects which serve as instructional information in the robotics system.

3. Experiment

3.1 Control System Architecture

We set up a simple Parallel Kinematic Machine (KPM) for simulation purpose, described in **Figure 1.** A parallel manipulator is a mechanical system using several chains to support a single end-effector [28], [29]. This is implemented in the robot neck so that it has three degrees of freedom, including row-pitch-yaw axes. Thus it is more flexible than normal design. The system can be separated into two frames, one is fixed and the other is the end-effector. The points A, B, C, D are linked with ball joints and O E is a universal joint, while O F is fixed. Note that the length DC and AB can be adjusted whereas the length O F O E is constant.

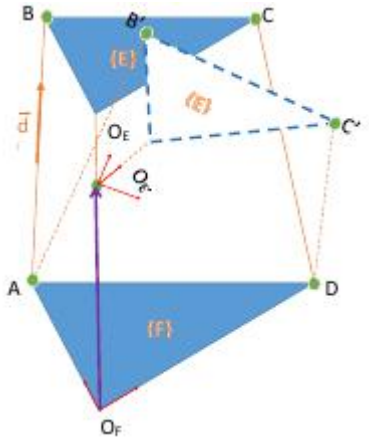

Figure 1. Simple Parallel Kinematic: Base Frame AO F D and End-Effector Frame BO E C.

In order to control the system, we need a representation of forward and inverse kinematics problems by defining the formalization equation in Cartesian coordinate system:

$$s_i^2 = d_i^T d_i = \left(p - a_i + Rb_i\right)^T \left(p - a_i + Rb_i\right) \tag{3}$$

as forward kinematics based on s_i indicating length of the leg AB or CD, where $\vec{d}_i = \vec{DC}$ or $\vec{AB}$; $\mathrm{p} = \mathrm{O_F O_E}, R$ is rotation matrix with respect to angles $\phi, \theta, \omega; \vec{a}_i$ is equal to $\vec{O_F D}$ or $\vec{O_F A}$; $\vec{b}_i$ is equal to $\vec{O_E B}$ or $\vec{O_E C}$ and set

$$J_e^{-1} = \begin{bmatrix} -w_1 \times q_1 & -w_2 \times q_2 \\ \omega_1 & w_2 \end{bmatrix}^T \tag{4}$$

as Jacobian of inverse problem, where q_i is the constant joined vectors concerning each link of frame.

3.2 Datasets

For training person detection module, we use Inria person dataset [13] and Standford 40 actions dataset [14]. The Mask R-CNN is a supervised learning method so it requires both input data and corresponding output label. To generate pixel-level mask labels, we use VIA, VGG Image Annotator [15], an open-source software providing a manual polygon drawing tool to quickly annotate different types of object. We have manually annotated 300 images containing 500 objects of two classes: human face and body. These samples are not sufficient for training the detection model from scratch, but can be leveraged when combining with a technique called Transfer Learning described in Section III-D.

On the other hand, we also prepare a dataset for testing segmentation (i.e., the mask) of Mask R-CNN as mentioned in Section II-B. This dataset contains approximately 5000 sushi samples in which every single shape (mask) is manually annotated using VIA tool. The interest of developing an accurate object segmentation appears not only in HRI but undoubtedly in robotics grasping and manipulation as well.

3.3 Training Mask R-CNN

Training is an attempt to force input dataset, when being fed to the model, to produce desired output labels. During this process, the neural network characterizes and memorizes input-output relation; model weights are gradually adjusted so that it can make correct decision when facing similar data next time. Generally, the training process can be challenging due to convergence-related issues which depend on nature of datasets and hyper-parameters of the model.

As a pre-processing step, all images in the datasets are resized to 1000x1000 pixels. To evaluate accuracy, it is natural to split the dataset into three subsets: training, validation and test set such that they do not overlap each other. The training set is used for training purpose; validation set is for keeping track of overfitting problem and test set is for

precision estimation. In the literature, a suggested ratio of training, validation and test set is 60:20:20 respectively.

The Mask-RCNN supports ResNet50 and ResNet100 as feature extractors, and both are evaluated on the datasets. As what we observed, there is not much difference between them as long as the loss value is sufficiently small.

3.4 Transfer Learning

Collecting data samples for training is not easy in terms of volume and quality. Transfer learning or inductive transfer [17], [18], [19] is a popular solution for the lack of data where only a small volume of samples is available for training. The main idea is we do not train the model from scratch but continue training on a pre-trained model which was previously trained on large high-quality datasets. The intuition behind is since the pre-trained model is capable of learning and extracting general object features, we only need a small dataset to train and to adapt it to our specific objects.

On the other hand, training from scratch on a large dataset may take weeks on powerful machines. Transfer learning also helps to boost the training speed as well as to guarantee a better convergence.

One of the most popular dataset used in transfer learning is ImageNet [20]. It contains more than 14 millions images classified into more than 20000 object classes. A model trained on the ImageNet is said to be able to learn and extract features of objects in general. In the experiment, we take advantage of the Mask R-CNN pre-trained on ImageNet to build a face detector and object segmentation module for robot.

3.5 Robot Operating System (ROS) set-up

ROS is a framework for writing robot software. It is capable of managing robot system thanks to its maintainability and scalability. The following set-up of ROS can be efficiently implemented on real robot system.

The openni2_launch package is used as a driver for extracting raw RBG image and depth from Kinect XBOX ONE version 2 of Microsoft's camera. The package consists of three main functional cluster nodes. One provides information transformation between links of camera frames, one feeds raw RBG image with metadata and the other one feeds depth image. Transformation link handling node provides transformed data to message routing node called /camera nodelet nnanager via /tf topic for dynamic links and /tf static for fixed links of the camera. Both Raw RGB

image and depth image are also fed to message routing node by /camera driver node.

The package named robotvision has two nodes. A node named CameraHandle captures frames taken by camera and conveys them to the second node named FaceVision. This node is a combination of face detection and face recognition and database update. The update database process is called after the node FaceVision terminated. **Figure 3** indicates how FaceVision node is structured.

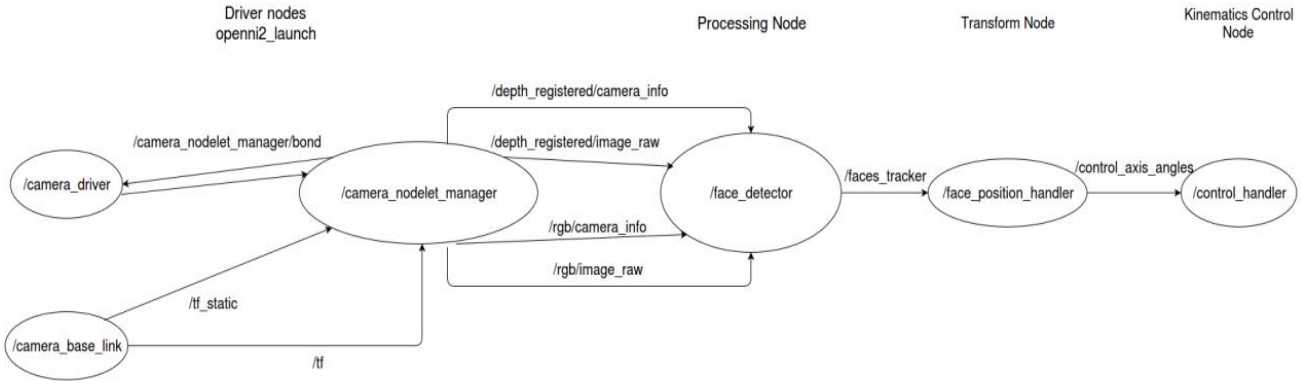

Figure 2. Diagram of Real-time Multi-modal localization system on ROS, described as nodes communicated via topics and services

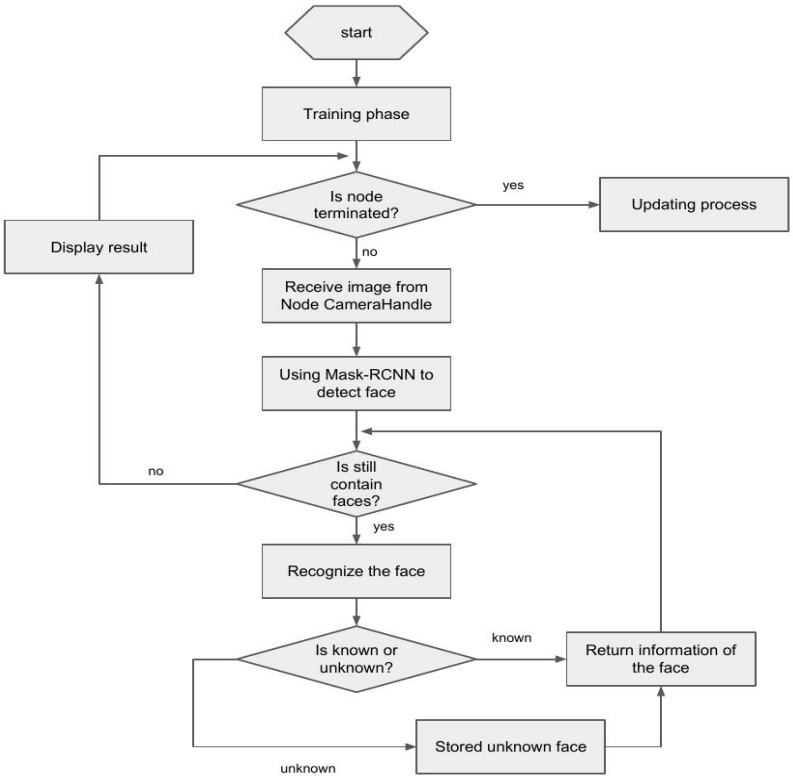

Figure 3. Flowchart for FaceVision node implemented in ROS. The integration of Mask-RCNN helps increase detection and segmentation accuracy.

Message routing node broadcasts the data to Processing node to analyze data implemented EigenFaces for face recognition and Mask R-CNN for face detection. It first uses these algorithms to obtain a set of initial detections, then prunes false positive using stereo depth information from depth image. The depth information is used to predict real size of detected faces, hence remove a majority of false positive given by the initial detection. This node then uses depth information together with the two algorithms for calculating 3D float-type coordinates relative to the reference frame of the camera and publishes these messages to the Transform node via topic /faces tracker.

The Transform node named /face positon handler converts the 3D coordinates of the detected face to set angles of the three axes in order to deviate the robotic frame such that it always follows the movement of the tracked face. This node then publishes control messages to the Control node. We use this information to calculate desired motors or servos angles and to communicate with a motors or servos controller board when we consider the potential to integrate our controller into other robot control systems. Overview of the ROS system architecture is illustrated in **Figure 2.**

The architecture of ROS provides many inter-processes operating in parallel. It provides services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message passing interface and package management. We are planning to use a laptop with dedicated GPU (NVIDIA GTX 1070) running ROS as central processing unit for our robot. It is responsible for inverse kinematics calculation, motion planning as well as expensive deep learning computation. The forward and inverse kinematics control processes have roll and pitch angle input while output is the status of servos.

3.6 Results

1) Real-Time Robot Tracking System and Evaluation: For localization and tracking purpose, 2-D square workspace is set up with a total size 40x40 comprising four non overlapping sub-areas 20x20 as illustrated in **Figure 4.** In this workspace, moving distance $D_{distance}$ D distance is calculated:

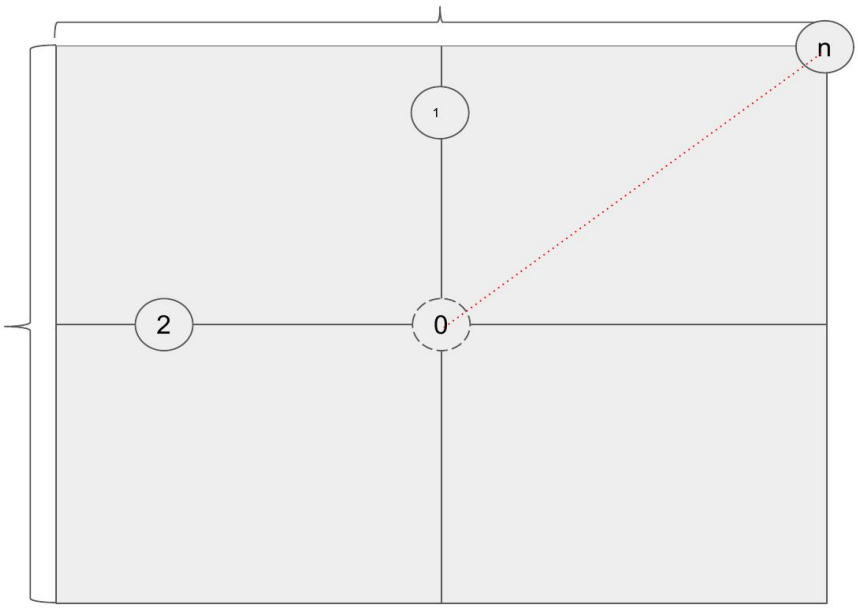

Figure 4. Example of four evaluation points: 0,1,2 and n

$$D_{distance} = \begin{cases} \sqrt{2a^2} = \sqrt{2}a \\ 0 \\ 0 \end{cases} \quad (5)$$

where a = 20 is base of the square sub-area. It is obvious that in this workspace the moving distance cannot exceed $20\sqrt{2}$ (length of the dotted red line).

• Point ⓪ : original point in base frame or target point in reference frame. In this case $D_{distance} = 0$, the robot head does not make any move.

• Point ①: no horizontal move is made. The robot only moves up.

• Point ②: at this point neither up nor down is expected, the robot head only turns left. Note that the two points 1 and 2 result in a single-direction movement.

• Point ⓝ: extreme point where $D_{distance} = D_{\max} = 20\sqrt{2}$ Robot head moves up and to the right simultaneously.

In **Figure 5**, we show some simple cases to demonstrate how the system works. The yellow circle is camera calibration center and the red one is human face.

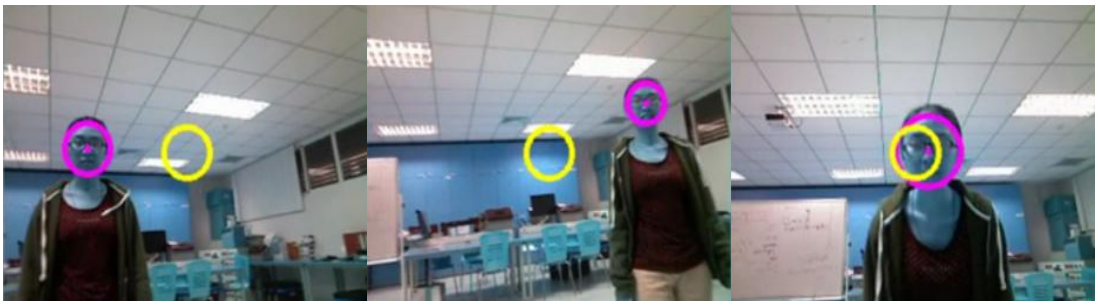

Figure 5. Demonstration of Real-Time Localization and Tracking system.

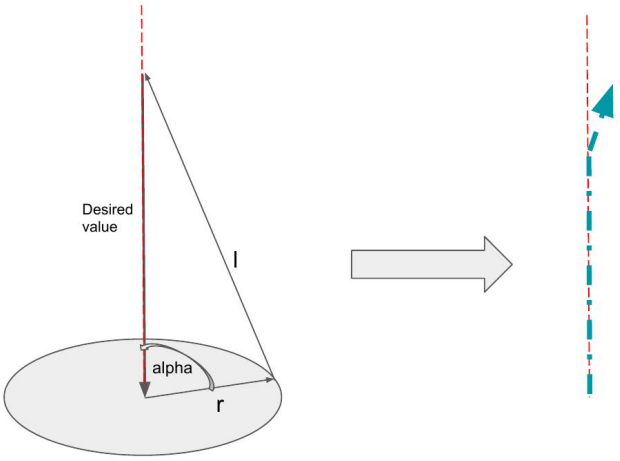

Figure 6. Estimation and evaluation of robot vision mechanism based on representative joint adaptation.

Table 1 demonstrates the three orientations of the system.

	Roll (degree)	Pitch (degree)	Yaw (degree)
$a_{\max}$	20	58	30
$a_{\min}$	-20	-23	-30

Table 1. The robot's three orientations.

2) Mask R-CNN for detection and segmentation: We train and test on 5000 sushi images, then on 500 face/person samples. The result is shown in **Figure 7**, 8, 9, 10 and 11. As we can see, even though Transfer Learning is applied in both cases, the dataset volume has a significant impact on the segmentation quality. The sushi segmentation (**Figure 11**) is more apparent than the face/person (**Figure 7**). To enrich the data samples, one may think of applying data augmentation techniques such as rotation, noise adding, etc.

We plot in the **Figure 8** the relation between Precision and Recall in case of sushi detection. AP@50 is the average precision when Intersection-over-Union (IoU) is 50%. This IoU indicates the ratio between overlapping area and union area of predicted object bounding box (P) with its ground-truth (G) so that IoU(P,G) = (P ∩G)/(P ∪ G). With Mask R-CNN, we achieve an 89.5% accuracy compared to 80% in our previous work [12].

When training Mask R-CNN, we always keep track on how training and validation accuracy vary. The behavior is visualized in **Figure 12.** The vertical red line shows a stopping condition and beyond that is overfitting zone where validation error starts fluctuating.

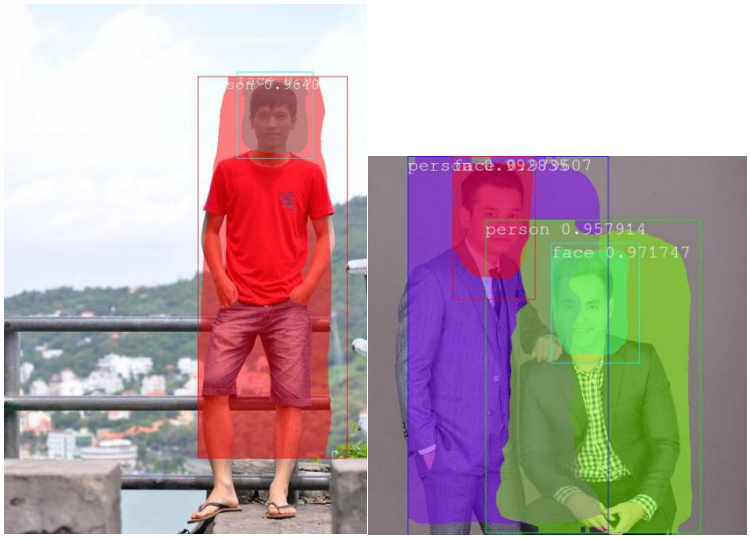

Figure 7. Human detection and segmentation using Mask R-CNN.

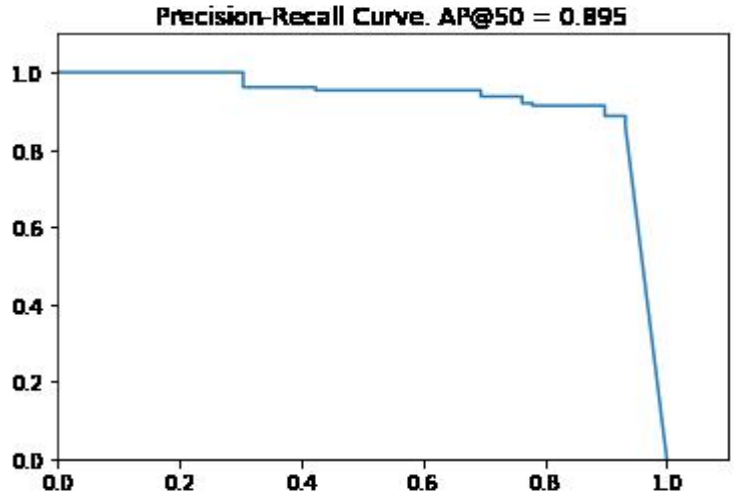

Figure 8. Precision-Recall visualization with 5000 sushi images.

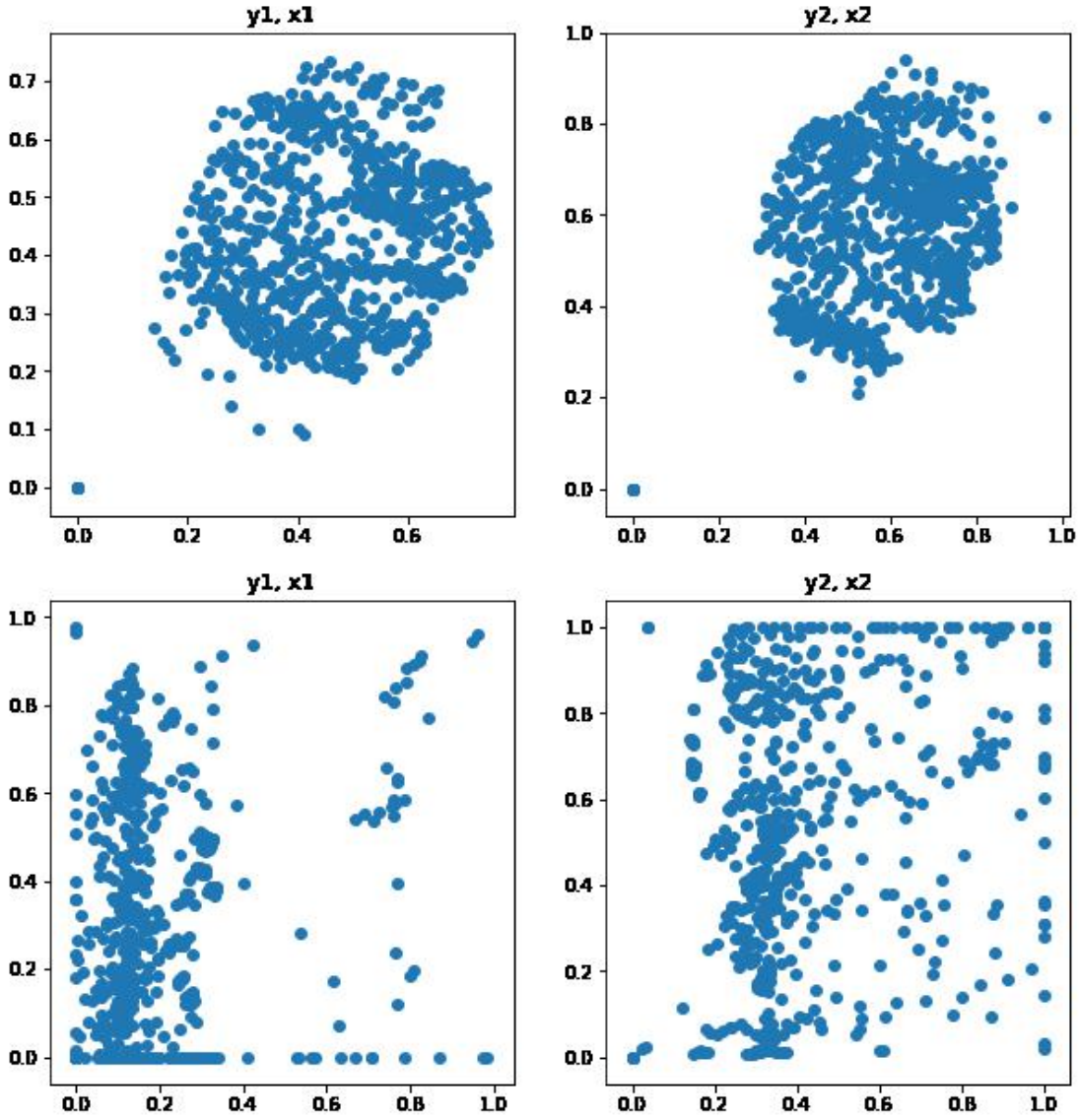

Figure 9. Distribution of normalized coordinates (y1, x1) and width-height (y2, x2) of data samples. Top is 5000 sushi images and bottom is 500 human images.

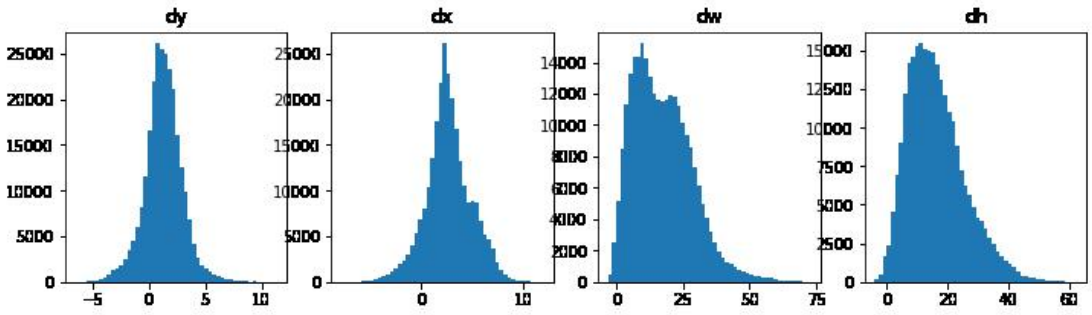

Figure 10. Histograms of Regional Proposal Network (RPN) bounding boxes in Mask R-CNN: coordinates (dy,dx) and width-height (dw,dh). Top is 5000 sushi images and bottom is 500 human images.

Figure 11. Object detection and segmentation using Mask R-CNN. Segmentation is more accurate with 5000 training images.

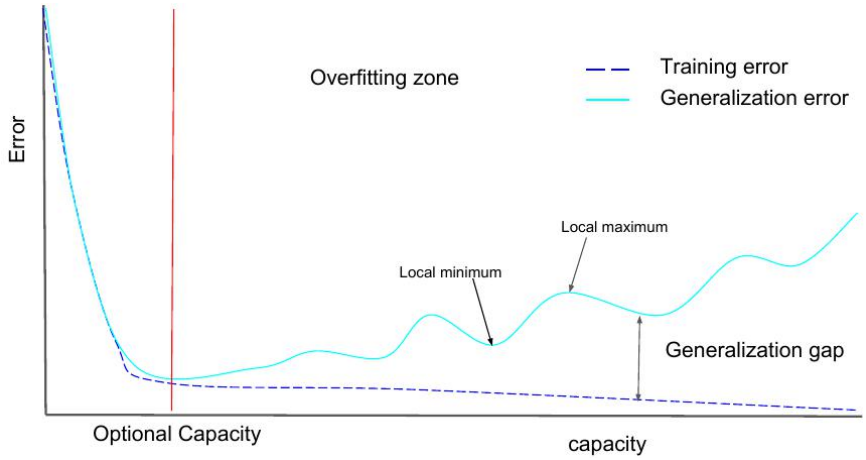

Figure 12. Convergence of Mask R-CNN. Overfitting is when the model behaves well on training data but poorly on validation set. The vertical red line can be used as stopping condition without loosing any generalization capability.

4. Conclusion

In conclusion, we integrated Mask R-CNN deep learning module to assist human-robot interaction with detection and segmentation tasks. The result has shown an accuracy improvement even with small volume of training samples. Furthermore, it is also potential to be used in robotics grasping and manipulation where precise segmentation plays a crucial role. In our study due to inadequate training samples, the precision of the detection and segmentation is limited. However in comparison with traditional featured- based approach, we show that neural networks can bring a significant improvement to the field of robotics vision. Of course, it is not straightforward to deploy these powerful tools to mobile platform and is subject to our future work.

References

1. Rekha N, M.Z.Kurian. Face Detection in Real Time Based on HOG. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2014.
2. L. R. Cerna, G. Cámara-Chávez, D. Menotti. Face Detection: Histogram of Oriented Gradients and Bag of Feature Method. Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), 2013.
3. J. Barreto, P. Menezes, J. Dias. Human-robot interaction based on haar-like features and eigenfaces. International Conference on Robotics and Automation (ICRA), 2004.
4. P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition (CVPR), 2001.
5. R. Ranjan, V.M. Patel, R. Chellappa. HyperFace, A Deep Multitask Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2016.
6. K. Zhang, Z. Zhang, Z. Li, Y. Qiao. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. IEEE Signal Processing Letters (SPL) 2016.
7. S, Yang, P. Luo, C. Loy, X. Tang. From Facial Parts Responses to Face Detection: A Deep Learning Approach. The IEEE International Conference on Computer Vision (ICCV), 2015.
8. F. Schroff, D. Kalenichenko, J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. Computer Vision and Pattern Recognition (CVPR), 2015.
9. R. Girshick. Fast R-CNN. The IEEE International Conference on Computer Vision (ICCV), 2015.
10. S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Conference on Neural Information Processing Systems (NIPS), 2015.
11. K. He, G. Gkioxari, P. Dollár, R. Girshick. Mask R-CNN. The IEEE International Conference on Computer Vision (ICCV), 2017.
12. Quan H. Nguyen; Trinh N. P. Tran; Dung D. Huynh ; An T. Le ; Than D. Le. Real-Time Localization and Tracking System with Multiple-Angle Views for Human Robot Interaction. The IEEE International Conference on Robotic Computing (IRC), 2017.
13. INRIA Person. http://pascal.inrialpes.fr/data/human/, INRIA.
14. 40 Actions. http://vision.stanford.edu/Datasets/40actions.html, Stanford.
15. VGG Image Annotator. http://www.robots.ox.ac.uk/ vgg/, Oxford.
16. Luis Perez, Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv, 2017.
17. Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson. How transferable are features in deep neural networks? Advances in Neural Information Processing Systems, 2014.
18. CNN Features off-the-shelf: an Astounding Baseline for Recognition Ali S. Razavian, Hossein Azizpour, Josephine Sullivan Stefan Carlsson. CNN Features off-the-shelf: an Astounding Baseline for Recognition. arXiv, 2017.
19. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. The International Conference on International Conference on Machine Learning, 2017.
20. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition, 2009.
21. Alessandro De Luca, Alin A. SchafferSami HaddadinGerd Hirzinger. Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. IEEE/RSJ International Conference on Conference: Intelligent Robots and Systems, 2006.
22. Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics.
23. Hoi V. Nguyen, Than D. Le, Dung D. Huynh, Peter Nauth. Forward kinematics of a human-arm system and

inverse kinematics using vector calculus. International Conference on Control, Automation, Robotics and Vision (ICARCV), 2016.
24. Miao Li, Hang Yin, Kenji Tahara, Aude Billard. Learning Object level Impedance Control for Robust Grasping and Dexterous Manipulation. IEEE International Conference on Robotics and Automation (ICRA), 2014.
25. Joseph Redmon, Ali Farhadi. YOLO9000: Better, Faster, Stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
26. Emanuele Magrini, Fabrizio Flacco, Alessandro De Luca. Control of generalized contact motion and force in physical human-robot interaction. IEEE International Conference on Robotics and Automation (ICRA), 2015.
27. An T. Le, Than D. Le. Search-based Planning and Replanning on Robotics and Autonomous Systems. Advanced Path Planning for Mobile Entities, IntechOpen, 2018.
28. John J. Craig. Introduction to Robotics: Mechanics and Control. PEARSON, 2009.
29. Lung-Wen Tsai. Robot analysis: The mechanics of serial and parallel manipulators 1st edition. pages 118129. John Wiley and Sons, Inc, 1999.
30. Ian Goodfellow and Yoshua Bengio and Aaron CourvilleMask R-CNN implementation, https://github.com/matterport/Mask RCNN.
31. Ian Goodfellow and Yoshua Bengio and Aaron Courville. Deep Learning (Adaptive Computation and Machine Learning). MIT Press, 2016.
32. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmen-tation. Techical report, 2014.