Global Scientific JOURNALS

GSJ: Volume 8, Issue 10, October 2020, Online: ISSN 2320-9186
www.globalscientificjournal.com

Deep Learning Approach for Autonomous Shopping Cart Robot

Md Matiqul Islam
Department of Information and Communication Engineering, University of Rajshahi, Rajshahi-6205, Bangladesh

Abstract. In the field of artificial intelligence deep learning based models are highly succeed in object detection and recognition task, but the sate of the art models has not yet been applied to autonomous navigation task. For autonomous robot navigation task, we face a great challenge for obstacle avoidance, shortest path searching and strong real-time. In our proposed method we use Mask-RCNN region based Convolutional Neural Network, one of the most recent best performing object detection task models for detecting different objects. We train this model with our own dataset with extra additional class obstacle free floor in indoor shopping mall. After detecting this floor, we have applied probabilistic roadmap planner(PRM) for getting roadmap and Dijkstra algorithm to calculate the shortest obstacle free path for autonomous shopping cart robot movement that outperforms better.

Keywords: Mask R-CNN, PRM, Dijkstra algorithm.

1. Introduction

In current era robots are developed rapidly. Autonomous and intelligent robots are the most red-hot topics in this field. They are required to work in many places such as shopping mall, factories, ware house automation, driverless cars, robotic surgery, offices and so on. Originally the problem was studied within the robotics community, but in recent years many new applications arise in such areas as animation, virtual environments, computer games, computer aided design and maintenance, and computational chemistry.

In this environment the collision-free path planning is one of the major problems to realize autonomous mobile robots. Since there are many stationary or moving obstacles in these environment, autonomous mobile robots should plan their own path that can avoid not only stationary obstacles but also moving ones such as human workers and other robots.The path planningproblem is described as:to find a shortest or optimized path between start point and goal in a spatial configuration consists of obstacles of various types. There are many fundamentallydifferent approaches suitable for different environmental configurations. The various methodsare Cell Decomposition, Sampling method, Probabilistic Roadmap methods, Generalized Voronoi diagrams etc. In this paper we propose a method by combining generalized Probabilistic Roadmap methods and Dijkstra shortest path algorithm in robot motion planning with deep neural networks(Mask R-CNN).

The paper organizes as follows: section 2 describes our proposed method for autonomous shopping

cart robot. Our experiment is illustrated in section 3. In section 4, the experimental results of the proposed method are presented.

2. Proposed Method

Fig.1 shows the block diagram of our proposed method for autonomous shopping cart robot. The functionalities of each block describe below one by one.

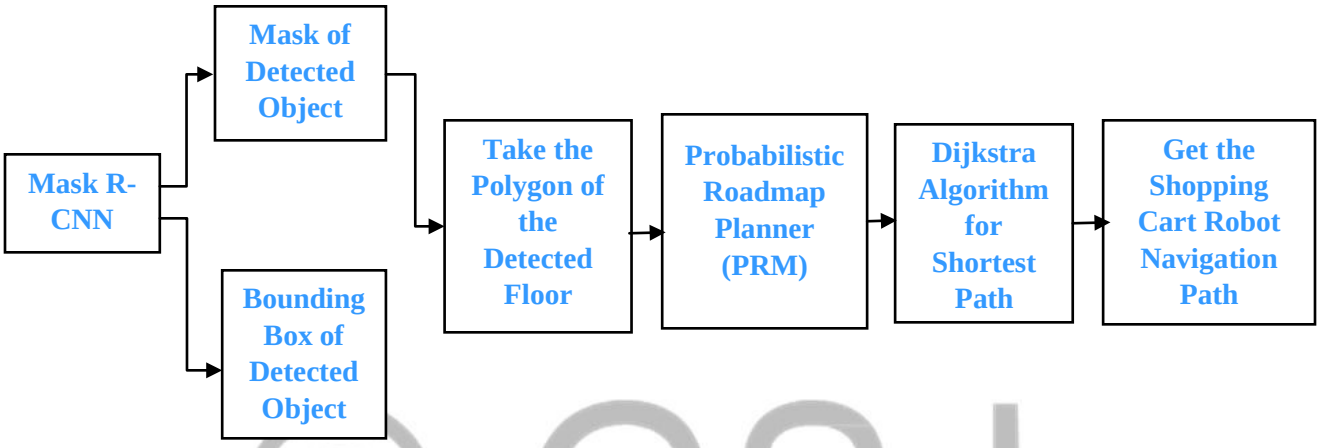

Fig.1 Our proposed method for autonomous shopping cart robot.

2.1. Mask R-CNN

Currently, a lot of CNN methods have been proposed in the detection task of the visual recognition, andthese methods have been greatly successful in turn. For example, an early study is R-CNN [2], which uses CNN for detection. R-CNN yields many region proposals from an input image by selective search, and then uses CNN foreach region proposal. Next, there Fast R-CNN [1] is theimproved version of R-CNN. In Fast R-CNN, region proposals are generated from a feature map of the whole image,so that CNN is used only once. Then, bounding boxes andclass scores for each category are estimated from a regionproposal finally. Thus, Fast R-CNN realizes high-speed andhighly accurate detection. However, these methods rely ona hand-crafted method of region proposal, and it is bottle-neck for the whole system. Faster R-CNN consists of two stages. The first stage, called a Region Proposal Network (RPN),proposes candidate object bounding boxes. The secondstage, which is Fast R-CNN [3], extracts features using RoI Pool from each candidate box and performs classification and bounding-box regression. The features used by both stages can be shared for faster inference. We refer readers to [4] for latest, comprehensive comparisons between Faster R-CNN and other frameworks.

For these methods, Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI. This contrast with most recent systems, where

classification depends on mask predictions (e.g. [7, 5, 6]). Our approach follows the spirit of Fast R-CNN [3] that applies bounding-box classification and regression in parallel (which turned out to largely simplify the multi-stage pipe line of original R-CNN [8]).

The network architecture of Mask R-CNN can be classified in two categories: (i) the convolutional backbone architecture used for feature extraction over an entire image, and (ii) the network head for bounding-box recognition (classification and regression) and mask prediction is applied to each RoI separately. Again, Mask R-CNN is designed for pixel-to-pixel alignment between networks inputs and outputs.

Mask head is used for instance segmentation task. This is fully convolution network, unlike the other heads which are FC layers. The output of the segmentation task should be a segmentation map big enough to represent an object of average size. The network architecture is taken from [11]and is shown in Fig.2 below.

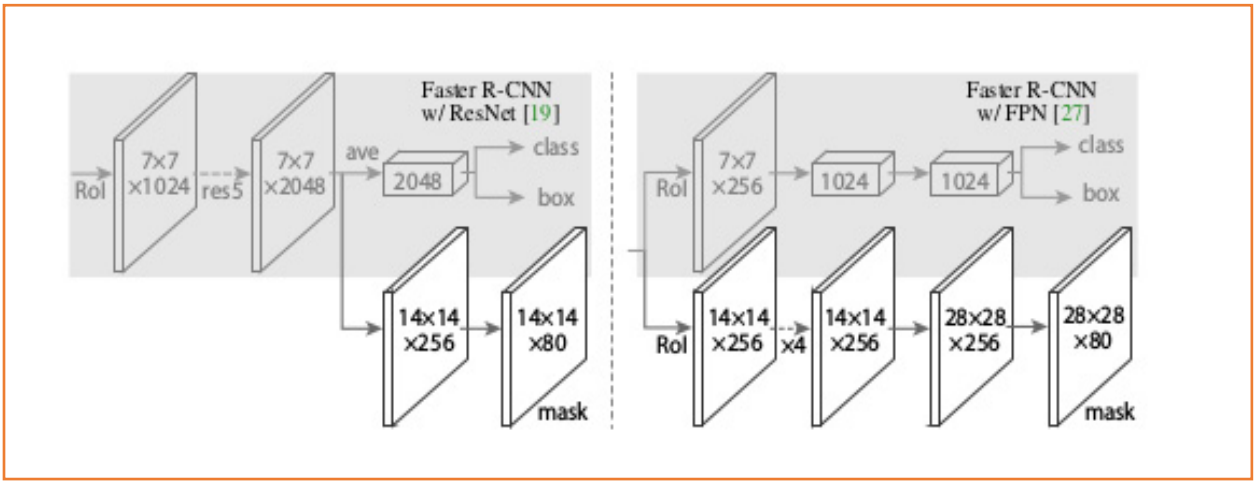

Fig.2 Head Architecture

Here they extend two existing Faster R-CNN heads. Left or right panels show the heads for the ResNet C4 and FPN backbones respectively, to which a mask branch is added. Numbers denote spatial resolution and channels. Arrows denote convolution, deconvolution, or fc layers. All convolutions are 3x3, except the output convolution which is 1x1, deconvolutions are 2x2 with stride 2, and use ReLU in hidden layers. Left 'res5' denotes ResNet's fifth stage, which is for the first convolution operates on 7x7 RoI with stride1.

One of the most important contributions of Mask R-CNN is the ROI align layer instead of ROI pool (in Faster R-CNN). RoIPool[3] is a standard operation for extracting a small feature map (e.g., 7×7) from each RoI. RoI Pool first quantizes a floating-number RoI to the discrete granularity of the feature map, this quantized RoI is then subdivided into spatial bins which are themselves quantized, andfinally feature values covered by each bin are aggregated (usually by max pooling).This basically doesn't round off your (x/spatial_scale) fraction to an integer (like it does in the case of ROIPool). Instead, it does bilinear interpolation to find out the pixels at those floating values [9, 3]. The same process is used to get floating point value instead of integers

(quantization) while assigning spatial portions into output bins in ROI pooling. RoI align improves mask accuracy by relative 10% to 50%.

Fig.3 shows the output of the detection result of Mask R-CNN with our own dataset.

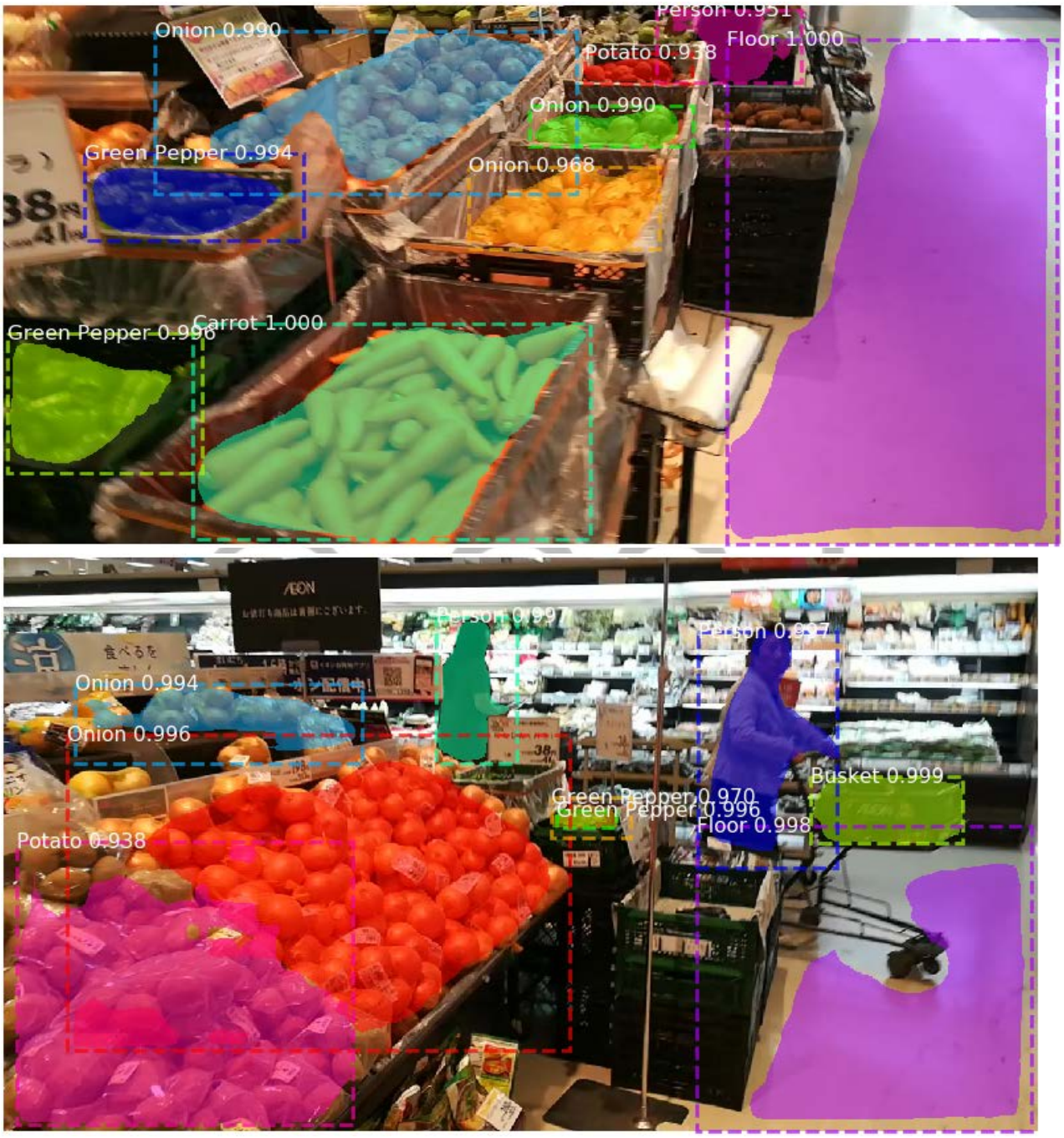

Fig.3 Detection results of Mask R-CNN

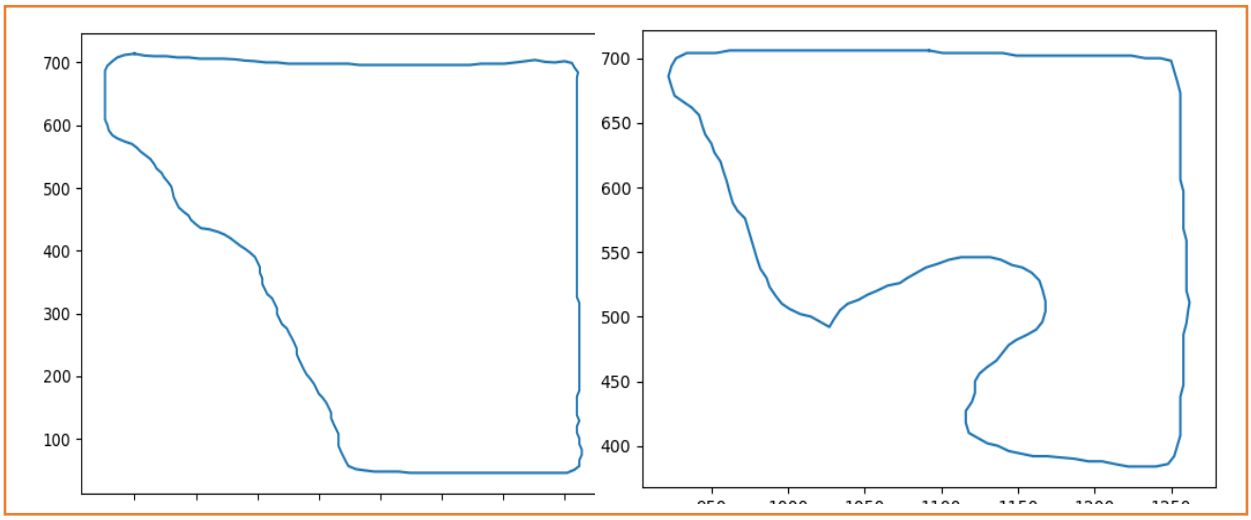

Fig.4 Crop the polygon of the detected floors in Fig.3.

2.2. Probabilistic Roadmap Planner(PRM)

Probabilistic roadmap planner is ordinarily used technique for Robot motion planning. In this research our goal is to generate roadmap procedurally using Voronoi diagrams. A Voronoi diagram is a set of sitesin the plane is a collection of regions that divide up the plane. Each region corresponds to one of the sites and all the points in one region are closer to the site representing the region than to any other site [10].

Once Mask R-CNN deep neural network generate floor mask polygon or graph it is applied to the PRM algorithm to generate the roadmap. We can use it to find the closest way between any starting location and goal.

One of the drawbacks of this roadmap is that if the starting location and goal location is not on map it can not find the path. But it is quite possible considering Voronoi diagram is a bunch of lines in the middle of all that free-zone. In this situation, we will simply choose the closest Voronoi vertex to starting position and goal, calculate path between them, then connect these nodes to given points.

In the lower algorithm shows the process of generating the roadmap using PRM method.

Algorithm (ConstructRoadmap)
Let$:V \leftarrow \emptyset;$
$E \leftarrow \emptyset;$
loop
$c \leftarrow$ *a (useful) configuration in* C_{free}
$V \leftarrow V \cup \{c\}$
$N_c \leftarrow$ *a set of (useful) nodes chosen from* V
for all $c' \in N_c$, *in order of increasing distance from* c ***do***
if c' *and* c *are not connected in* G ***then***
if *the local planner finds a path between* c' *and* c ***then***
add the edge $c'c$ *to* E

In Fig.5 black plots are representing the generated roadmap using probabilistic roadmap planner method (PRM) and the red plot shows the shortest path from starting location to goal location from five different locations.

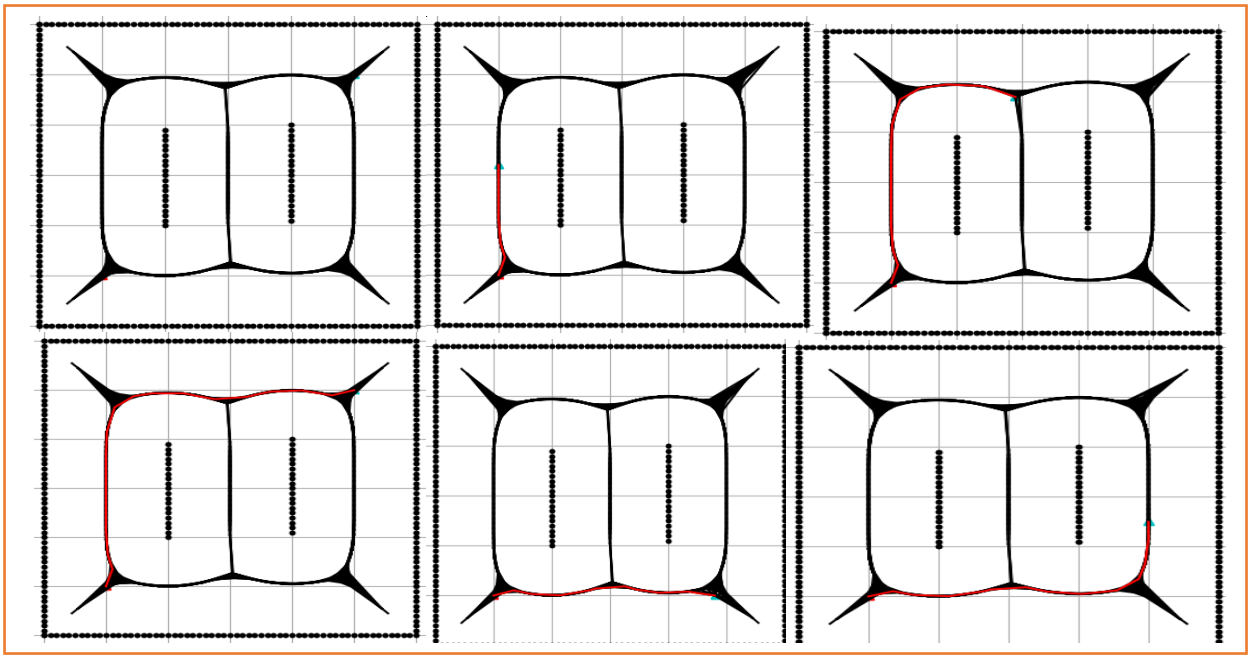

Fig.5 Bunch of samples with the shortest path solution

Fig.6 shows the generated roadmap using probabilistic roadmap planner (PRM) algorithm of our detected Mask R-CNN floor of Fig.4.

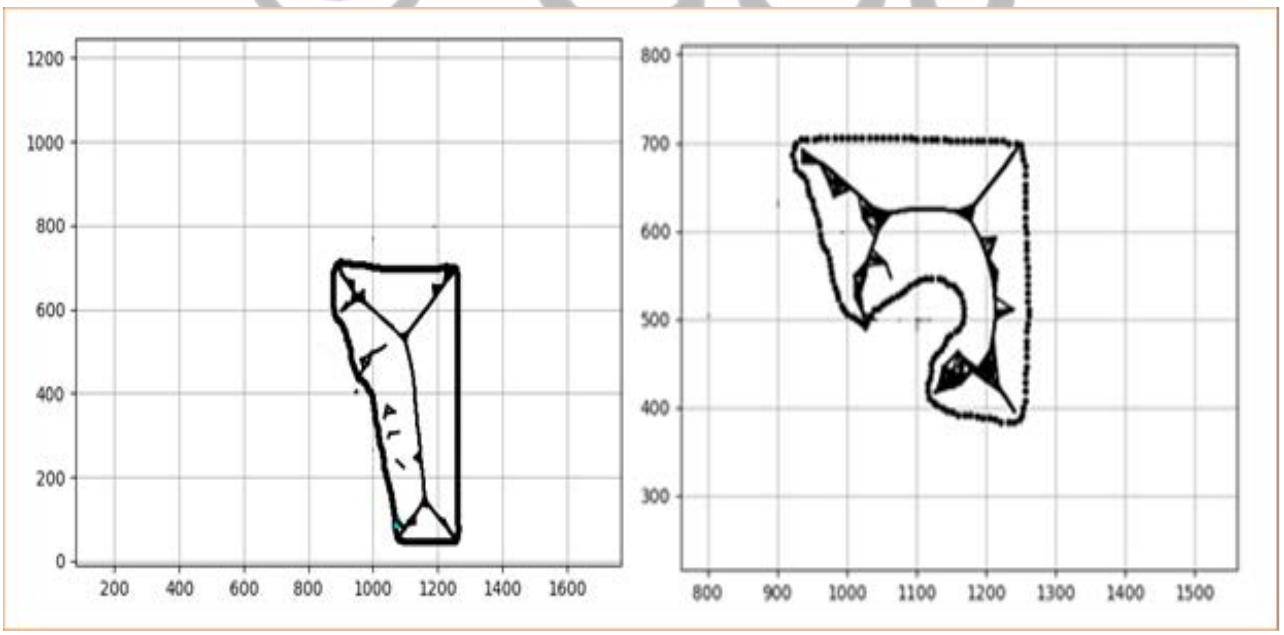

Fig.6 Roadmap of Mask R-CNN detected floor.

2.3. Dijkstra Algorithm for Shortest Path

Dijkstra algorithm measures the shortest path between nodes in a graph.Fig.7 shows the different shortest path of generated roadmap. The steps of Dijkstra shortest path algorithm are given below:

- Set the distance to the source to 0 and the distance to the remaining vertices to infinity.
- Set the **current** vertex to the source.
- Flag the **current** vertex as visited.
- For all vertices adjacent to the **current** vertex, set the distance from the source to the **adjacent** vertex equal to the minimum of its present distance and the **sum** of the **weight of the edge** from the current vertex to the adjacent vertex and the distance from the source to the **current** vertex.
- From the set of **unvisited vertices**, arbitrarily set one as the new **current** vertex, if there exists an edge to it such that it is the minimum of all edges from a vertex in the set of **visited vertices** to a vertex in the set of **unvisited vertices**. To reiterate: The new current vertex must be unvisited and have a minimum weight edges from a visited vertex to it. This can be done trivially by looping through all visited vertices and all adjacent unvisited vertices to those visited vertices, keeping the vertex with the minimum weight edge connecting it.
- Repeat steps 3-5 until all vertices are flagged as visited.

Algorithm (Dijkstra Shortest Path)

```
function Dijkstra(G,S,t)
for(each vertex v in V[G] do        //Initialize all distances to vertices(i.e., d[v] to ∞,
d[v]=infinity                       // except start which has distance 0
previous[v]=undefined               // also remembers how we got to this node
        d[s]==0
        Q=queue of all vertices     //Priority queue sorted by distances from s.
while(Q is not empty) do
  u=Extract-Min(Q)                  //Get the next closet unprocessed vertex. If it is the
  if(u==t) then return;             // destination, we are done.

for each edge(u,v) outgoing from u do     //Relax all edges from u to v, byupdating
if(d[v]>d[u] + dist(u,v)) then            //(reducing) the cost d[v] at each v if can
d[v]=d[u]+dist(u,v)                       // reached quicker from u

previous[v]=u
Q=Update(Q)
```

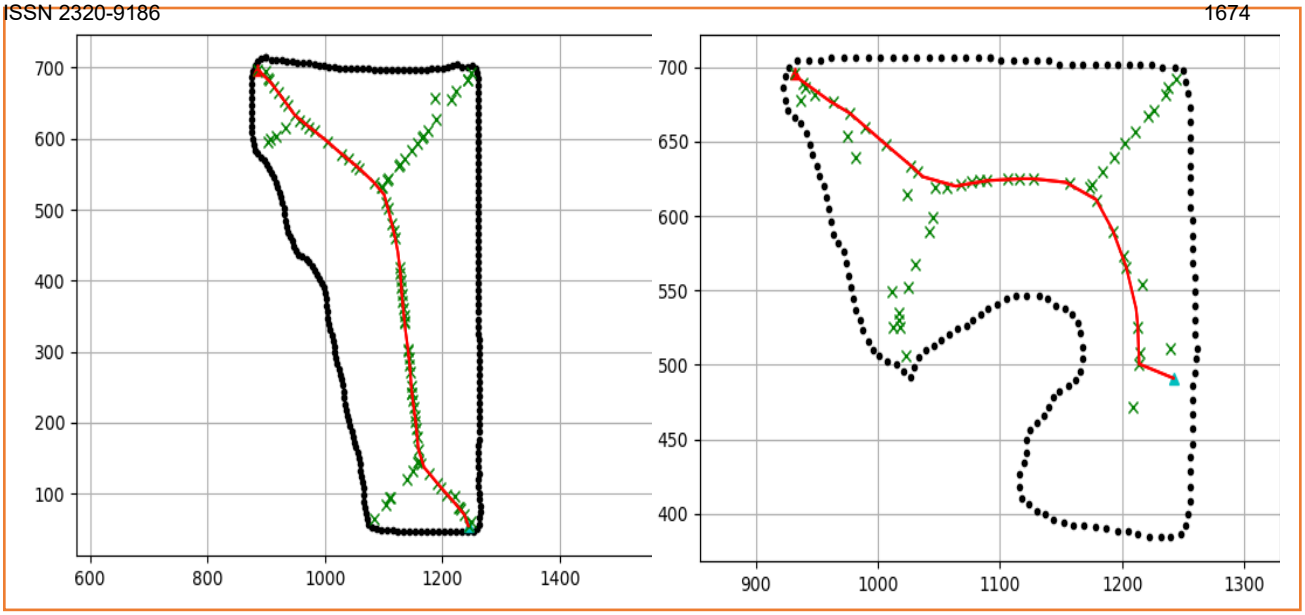

Fig.7 Dijkstra algorithm for generating shortest path

3. Experiments:

In our experiments we train the Mask R-CNN with our own dataset having 360 training images and 80 validation images with 37 classes (different vegetables and fruits) .The classes are person, floor, basket, cabbage, broccoli, green onion, corn, cauliflower, onion, garlic, carrot, eggplant, radish, tomato, mushroom, pumpkin, peas, potato, yellow pepper, red pepper, cucumber, lettuce, bok choy, bitter, sweet potato, ginger, burdock root, green pepper, lady's finger, banana, kiwi, apple, lemon, avocado, melon, water melon, orange. With this detected floor Mask R-CNN generate the mask like a polygon. We take this polygon coordinate and considered as Voronoi diagram to generate the roadmap. This roadmap is then passing to the Dijkstra shortest path algorithm to generate the shortest path to destination or goal location. Followed this path our shopping cart robot get the navigation path.

4. Results and Discussion:

Fig.8 shows the combined output of our whole proposed model. Black polygon is detected by Mask R-CNN and it is our obstacle free floor for shopping cart robot. After detecting the floor, we apply PRM algorithm to get the road map. The green plot shows the generated road map. After getting the road map we apply the Dijkstra shortest path algorithm to get the shortest path from start location to destination location. The red color plot shows the shortest path start location to destination location. Following this path our autonomous shopping cart robot automatically moves from one location to another location.

Fig.8 Final output of our proposed method with detecting different object and shopping cart robot navigation path.

5. Conclusion

In this paper we proposed an approach deep learning convolutional neural network based autonomous shopping cart robot. For this purpose, we train the network shopping related classes with additional class floor. At this stage we have just plan how the shopping cart robot navigate in the indoor shopping mall. In our next plan we will developed such a method that the client just chooses the product names, the shopping cart robot automatically moves and find that product thenthe seller put the product on shopping cart and then shopping cart return to the client position. For this reason, we also train the neural networks with different products (vegetables and fruits).

6. References

[1] R. Girshick. Fast R-CNN. In Proc. of IEEE International Conference on Computer Vision, 2015.2, 3

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hier-archies for accurate object detection and semantic segmentation. In Proc. of IEEE Computer Vision and Pattern Recognition, 2014.2

[3] R. Girshick. Fast R-CNN. In ICCV, 2015. 1, 2, 3, 4, 6

[4]J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In CVPR, 2017. 2, 3, 4, 6, 7

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In CVPR, 2016. 2, 3, 4, 5, 6

[6] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In CVPR , 2017. 2, 3, 5, 6

[7] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In NIPS , 2015. 2, 3

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semanticsegmentation. In CVPR , 2014. 2, 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV . 2014. 1, 2

[10] M.de Berg, M.vanKreveld, M.Overmars and O.Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer-Verlag, Berlin, 2000.

[11] He, Kaiming, Georgia Gkioxari, Piotr Dollár and Ross B. Girshick. “Mask R-CNN.” *2017 IEEE International Conference on Computer Vision (ICCV)* (2017): 2980-2988.