Semantic Segmentation for SLAM Augmentation

7TH SEMESTER MINI PROJECT REPORT

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD

Submitted to: Dr. Rahul Kala,

Assistant Professor,

IIIT Allahabad

Submitted by:

(IEC2016012) Nikhil Mundra

(IEC2016027) Bhanu Bhandari

Index

Problem Statement

Simultaneous Localization and Mapping have been thoroughly investigated, especially with the usage of Velodyne, LiDARs and stereo cameras to detect surroundings. However, the methods of feature detection by SLAM approaches are ad-hoc to date, i.e. there is no special emphasis on the relative movement of objects inside the frame. This means that certain objects in the image which may be deterring the SLAM framework from localization and mapping may also be considered, and helpful image features may not be emphasized. This project aims at **developing a model to augment SLAM so as to remove dynamic obstacles that are irrelevant to the SLAM task**. To integrate specialized feature detection within SLAM, the Mask R-CNN semantic segmentation framework has been adapted to locally collected road scene data which has been segmented and thus fed to the framework for training and inference.

Motivation

Existing SLAM methods rely on feature descriptors such as SIFT and SURF to determine the location as well as map the surroundings. However, with the advent of semantic segmentation models using neural networks, dynamic and static objects can be detected individually with much higher accuracy than is possible for SLAM frameworks. A number of such models such as Fast R-CNN, Faster R-CNN as well as YOLO have shown promising results of segmenting road scenes over a variety of related datasets such as CARLA, Cityscapes, and TUSimple. This serves as the primary motivation for creating a locally collected dataset and employing semantic segmentation methods. These detection results can then be relayed to the SLAM techniques to aid the removal of irrelevant features and emphasizing useful features.

Related Work

SLAM has been extensively studied over time, and there are many states of the art approaches for the same. Two of the most popular and reliable SLAM methods in recent times are S-PTAM and RTAB-MAP.

S-PTAM (Stereo Parallel Tracking and Mapping) heavily exploits the parallel nature of the SLAM problem, separating the time-constrained pose estimation from less pressing matters such as map building and refinement tasks. On the other hand, the stereo setting allows reconstructing a metric 3D map for each frame of stereo images, improving the accuracy of the mapping process with respect to monocular SLAM and avoiding the well-known bootstrapping problem. [1]

RTAB-Map (Real-Time Appearance-Based Mapping) is an RGB-D, Stereo, and Lidar Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. The loop closure detector uses a bag-of-words approach to determinate how likely a new image comes from a previous location or a new location. When a loop closure hypothesis is accepted, a new constraint is added to the map's graph, then a graph optimizer minimizes the errors in the map. A memory management approach is used to limit the number of locations used for loop closure detection and graph optimization so that real-time constraints on large-scale environments are always respected. [2]

For dynamic obstacle detection, which is a semantic segmentation problem over a variety of images and objects, multiple approaches have been devised. These include the Region-based Convolutional Neural Networks (or R-CNN) [3] and two improvements (Fast R-CNN and Faster R-CNN) [4], YOLO (You Only Look Once) [5] and its subsequent versions, SSD (Single-Shot Detection) [6], etc.

For lane detection, a number of classical computer vision methods such as edge detection and Hough transform have been employed. Deep Learning methods have also been investigated recently, which include Self Attention Distillation, Cascaded CNNs, and LineNet.

Terminology

- **Localization**: Localization of a robot is its ability to identify the position and orientation of the robot within the built map. Different ways of localization include wheel odometry, inertial measurement unit (IMU), laser range finder (Velodyne), visual cameras including monocular, stereo or RGB-D cameras.
- **Mapping:** The process of making a set of distinguishable landmarks for use in feature matching. Various ways of mapping include *Feature Maps* and *Occupancy Grids*.

- **Simultaneous Localisation And Mapping:** It is a process where a robot builds a map representing its spatial environment while keeping track of its position within the built map.
- **Feature Extraction:** It processes useful information in pictures. Features that are of interest range from simple point features such as corners to more elaborate features such as edges and blobs and even complex objects such as doorways and windows. The region around each detected feature is converted into a compact descriptor that can be matched against other descriptors. The simplest descriptor of a feature is its appearance or the intensity of the pixels in a patch around the feature point. Some of the popular techniques include SIFT and SURF.
- **Feature Matching:** The process of individually extracting features (descriptors) and matching them over multiple frames. Feature matching is particularly useful when significant changes in the appearance of the features occur after observing them over long sequences. The simplest way to match features between two images is to compare all feature descriptors in the first image to all other feature descriptors in the second image using a similarity measure.

Datasets

For the purposes of this project, we have worked on two self-collected and prepared datasets with the first dataset being monochrome images collected within the IIIT Allahabad campus with a resolution of 1280x720. These images contain all the classes being considered besides traffic lights.

The second dataset consists of images collected in the Civil Lines area of Prayagraj, with traffic and road scenes collected. These images are color images with resolution 672 x 376. All classes which the network can detect are present in this dataset.

Data Preprocessing

The first dataset has over 800 images, while the second dataset has 100 images. These images have been manually segmented using the **supervise.ly** online tool.

They were then imported into a network-readable format using the *Data Transformation Language* (DTL). To preserve the state of images, no other augmentations have been done to the images.

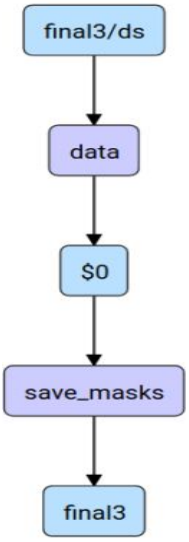

Fig 1. This flowchart depicts the fetching of segmented data, and hence, the application of a variety of colors to each class in both the machine-readable and human-understandable formats, prior to downloading.

Architecture

The network architecture used to train the collected images is the Mask R-CNN model.

Mask R-CNN is basically an extension of Faster R-CNN. Faster R-CNN is widely used for object detection tasks. For a given image, it returns the class label and bounding box coordinates for each object in the image. The Mask R-CNN framework is built on top of Faster R-CNN. So, for a given image, Mask R-CNN, in addition to the class label and bounding box coordinates for each object, will also return the object mask.

Working procedure of Mask R-CNN

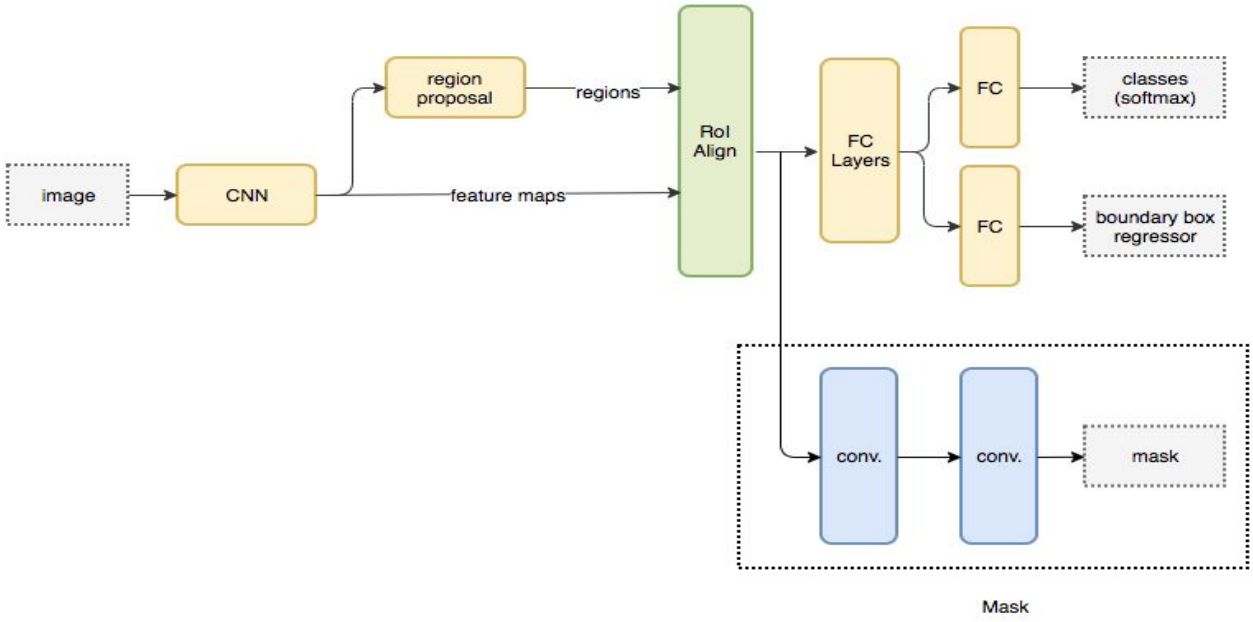

Fig 2. Architecture of the Mask R-CNN

The first step is to take an image and **extract features using the ResNet 101** architecture. These features act as an input for the next layer.

The feature maps thus obtained are taken and then applied with a **region proposal network (RPN)**. This basically predicts if an object is present in that region (or not). In this step, we get those regions or feature maps that the model predicts contain some object.

The regions obtained from the RPN might be of different shapes. Hence, a **pooling layer** is applied to convert all the regions to the same shape. Next, these regions are passed through a fully connected network so that the class label and bounding boxes are predicted. This procedure is exactly the same as Faster R-CNN.

However, *Mask R-CNN* achieves pixel-level segmentation, and thus the major difference between *Mask R-CNN* and *Faster R-CNN* is that **the former also generates the segmentation mask.**

For finding segmentation masks, first, the region of interest is computed so that the computation time can be reduced. For all the predicted regions, the *Intersection over Union* (IoU) is computed with the ground truth boxes.

IoU = Area of the intersection / Area of the union

If the *IoU* is greater than or equal to 0.5, only then is the region considered as the *region of interest*. Otherwise, we neglect that particular region. This is done for all the regions.

For the task of pixel-level obstacle and road detection, we used the pre-trained weights collected by training the Mask R-CNN model over the Microsoft COCO (Common Objects in Context) dataset [6]. This is used as a starting point for the training process. A few salient features of the COCO dataset:

- A total of 2.5 million labeled instances in 328k images
- 1.5 million object instances
- 80 object categories, 91 stuff categories
- 5 captions per image, 250,000 people with keypoints

We are particularly focussed on detecting **traffic signals, stop signs, streetlights, signboards** and moving objects such as **vehicles** and **pedestrians** since they are the most important obstacles which need to be highlighted for SLAM. Dynamic objects do not add to the information which SLAM methods need to compute locations and maps of the world, and hence removing them can considerably improve the detection quality of SLAM.

Results

Dataset 1: IIITA Dataset

Fig. 3 (a) Ground truth image

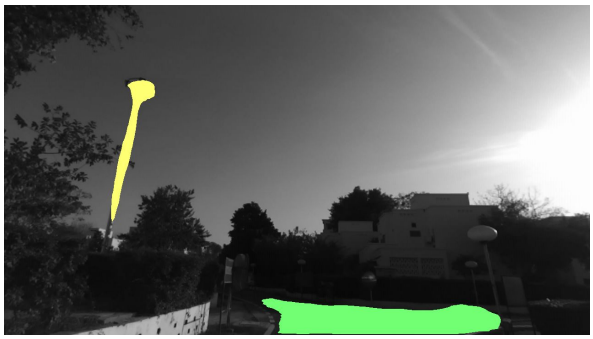

Fig. 3 (b) Segmented image

Fig. 4 (a) Ground truth image

Fig. 4 (b) Segmented result

Average Class IOU scores

Class	IoU score
Road	86.2%
Pedestrian	1.23% (absent)
Vehicle	6.14%
Signboards	5.36%
Street Light	9.06%

Average IOU score = 84%

Dataset 2: Prayagraj Dataset

Ground Truth

Predicted

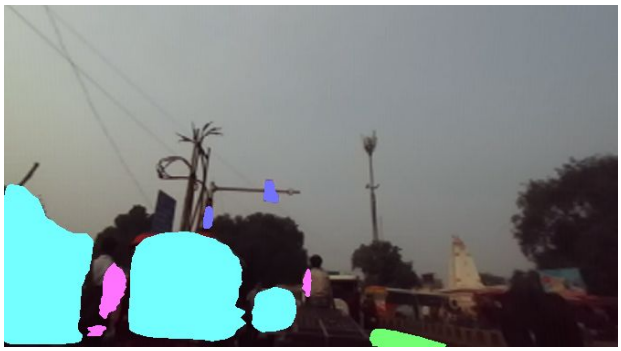

Fig. 5 (a) Ground truth image

Fig. 5 (b) Segmented image

Fig. 6 (a) Ground truth image

Fig. 6 (b) Segmented image

Class	IoU score
Road	86.2%
Pedestrian	43.82%
Vehicle	82.15%
Signboards	7.05%
Street Light	18.22%
Traffic Light	36.16%

Average IoU score = 73%

Conclusion

The above work presents a theoretical review of Simultaneous Localisation and Mapping, Semantic Segmentation and related terminology. Then, the task of detecting dynamic and static objects for enhancing SLAM efficiency was outlined, including the method used for semantic segmentation (Mask R-CNN), datasets used and preprocessed for training as well as the training process. Finally, the results were presented with class-wise Intersection over Union scores.

Future Scope

- Integrate the above architecture into a SLAM module such as RTAB-MAP or S-PTAM and evaluate the effectiveness of pre-detecting static and dynamic objects on SLAM
- Extend framework to detect objects in stereo images, in order to extrapolate the depth and hence the geographical coordinates of a location

References

[1] Taihú Pire,Thomas Fischer, Gastón Castro, Pablo De Cristóforis, Javier Civera and Julio Jacobo Berlles. **S-PTAM: Stereo Parallel Tracking and Mapping** Robotics and Autonomous Systems, 2017.

[2] M. Labbé and F. Michaud, "RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation," in *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019. (Wiley)

[3] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587.

[4] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 2980-2988.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 779-788.

[6] Lin TY. et al. (2014) Microsoft COCO: Common Objects in Context. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham