

Semantic Segmentation for Visual SLAM Augmentation

Project Supervisor: Dr. Rahul Kala

IEC2016012 - Nikhil Mundra

IEC2016027 - Bhanu Bhandari

Problem Statement

To develop a model which can achieve:

- ❖ dynamic obstacle detection (vehicles, pedestrians)
- ❖ static object detection (signboards, streetlights, traffic lights)
- ❖ road area detection for emphasising the drivable boundaries

in order to filter features for Visual SLAM methods.

Introduction: Simultaneous Localisation and Mapping

- ❖ Given some kind of data (inertial measurements using wheel odometry, visual odometry, LiDAR point data) can a robot simultaneously:
 - Determine its location in the world?
 - Determine where objects in the world are relative to its location?
- ❖ First solved in the 80s by Extended (non-linear) Kalman Filters till late 2000s
- ❖ Current SOTA approaches: all based on observed data (e.g. SeqSLAM, S-PTAM, RTAB-MAP)

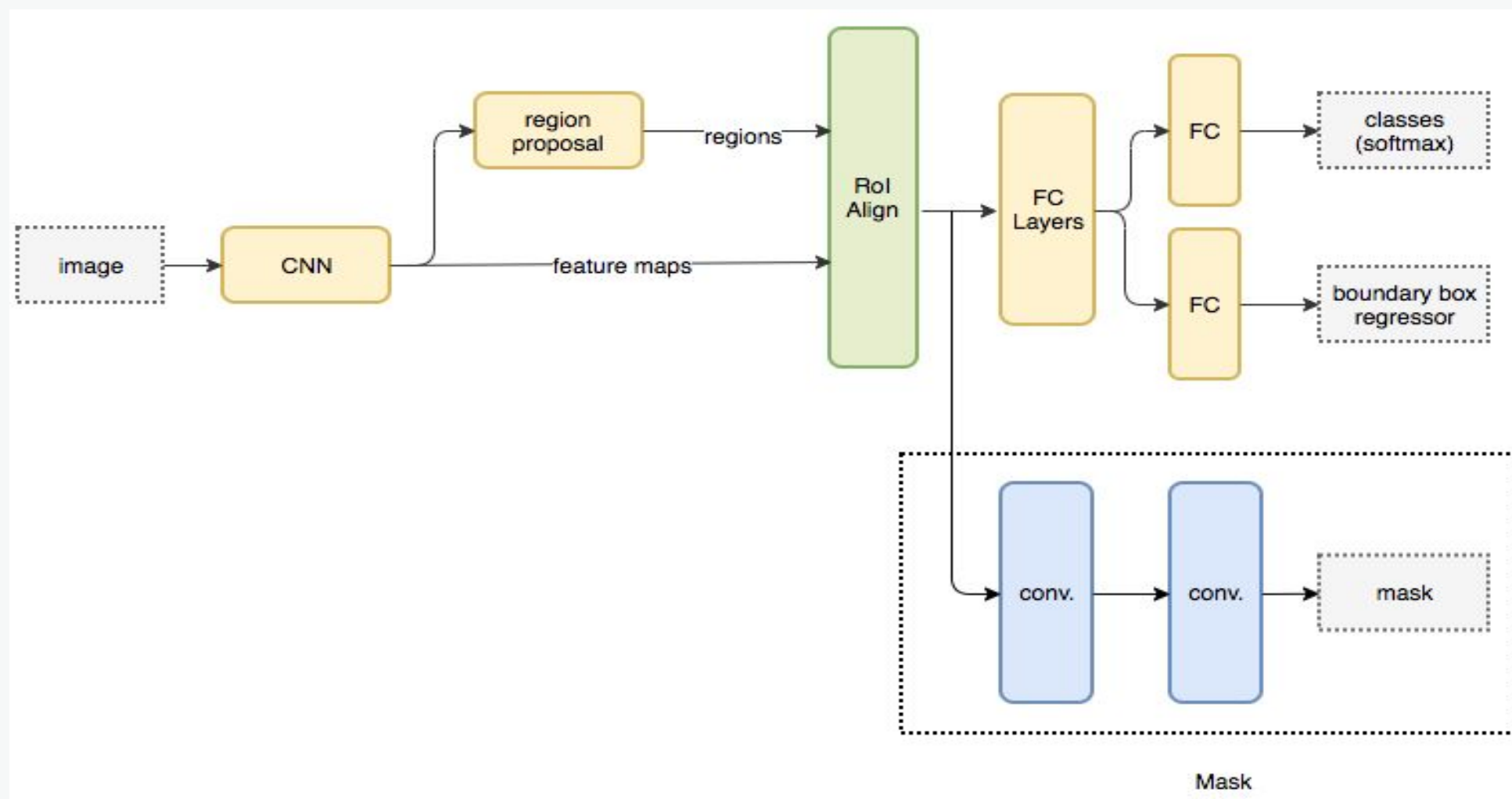
Introduction: Semantic Segmentation

- ❖ Process of labelling each pixel of an object in an image
- ❖ Also covers instance segmentation, where each instance is recognised as an individual object
- ❖ Architectures available include U-Net (for biomedical image segmentation) and Mask R-CNN (state of the art) based on R-CNN, Fast R-CNN and Faster R-CNN.

Architecture: Mask R-CNN

- ❖ Builds upon the Faster R-CNN object detection method
- ❖ Faster R-CNN first uses a ConvNet to extract feature maps from the images
- ❖ These feature maps are then passed through a Region Proposal Network (RPN) which returns the candidate bounding boxes
- ❖ Apply an RoI pooling layer on these candidate bounding boxes to bring all the candidates to the same size
- ❖ Finally, the proposals are passed to a fully connected layer to classify and output the bounding boxes for objects

Architecture: Mask R-CNN



Datasets

Microsoft Common Objects in Context (COCO)

- Object segmentation
- Mask-RCNN training started from weights pre-trained on COCO
- 1.5 million object instances with 330k images

IIITA Campus dataset

- 838 grayscale images
- Resolution: 1280 x 720
- Contains classes:
street light, road,
vehicle, pedestrian,
sign board

Prayagraj Dataset


- 100 RGB images
- Resolution: 640 x 480
- Contains classes:
streetlight, traffic light,
road, vehicle,
pedestrian, signboard

Semantic Segmentation Workflow: Preprocessing

The screenshot displays a semantic segmentation software interface. The main window shows a street scene with a green car, two purple pedestrians, and a red traffic light. The interface includes a toolbar on the left with various tools like pan, zoom, and segmentation. The top bar shows the file name 'left000787.png' and navigation controls. The right sidebar contains several panels:

- IMAGES 61**: A list of image files, including 'left000787.png' and 'left000790.png'.
- OBJECTS 7**: A list of detected objects with their classes and bounding boxes. The classes are 'Signboard', 'TrafficLight', and 'Pedestrian'.
- IMAGE PROPERTIES**: A panel with tabs for 'TAGS', 'DATA', and 'INFO'. It shows 'No tags defined in Project' and a button to 'Define a new Tag'.
- OBJECT PROPERTIES**: A panel with tabs for 'TAGS', 'DATA', and 'INFO'. It shows 'Select object first'.
- HISTORY 1**: A panel showing the 'Initial state'.

Semantic Segmentation Workflow: DTL



Explore

Data Commander ^β

S

Current Team

subtleseeker ▾

Workspaces

Members

Cluster

Plugins

Labeling Jobs

Current Workspace

First Workspace ▾

Projects

Import

DTL

Neural Networks

Python Notebooks

Tasks

What's new? ¹

Documentation

DTL ?

Saved Configs ▾

SuperviseLY A... ▾

START

Code ▾

```
1 {
2   "action": "data",
3   "src": [
4     "final3/ds"
5   ],
6   "dst": "$0",
7   "settings": {
8     "classes_mapping": "default"
9   }
10 },
11 {
12   "action": "save_masks",
13   "src": [
14     "$0"
15   ],
16   "dst": "final3",
17   "settings": {
18     "masks_machine": true,
19     "masks_human": true,
20     "gt_machine_color": {
21       "Lane": [
22         1,
23         1,
24         1
25       ],
26     },
27     "Street_light": [
28       1
29     ]
30   }
31 }
```

Ln: 1 Col: 1

final3/ds

data

\$0

save_masks

final3

support@supervise.ly

© 2019 Deep Systems LLC. Built with icons8.

[Terms of Service](#) · [Privacy Policy](#)

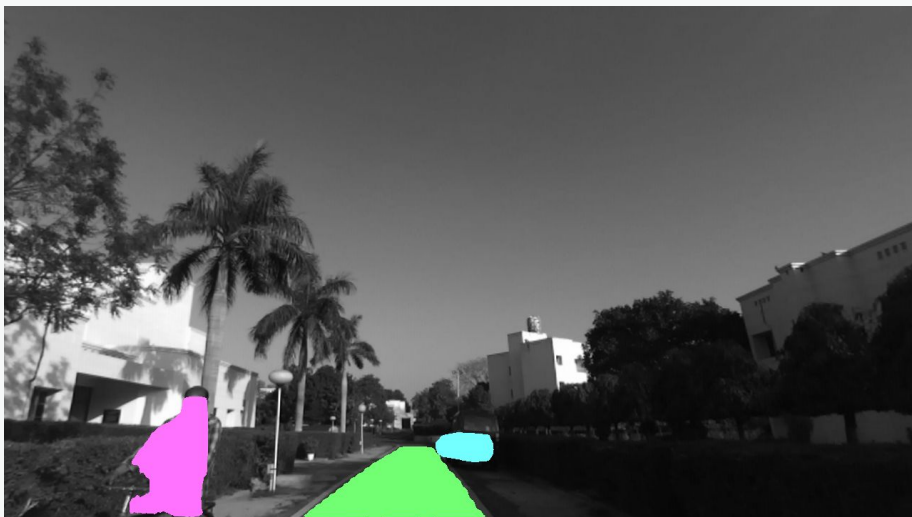
#

🔄

📺

M

Results: IIITA dataset

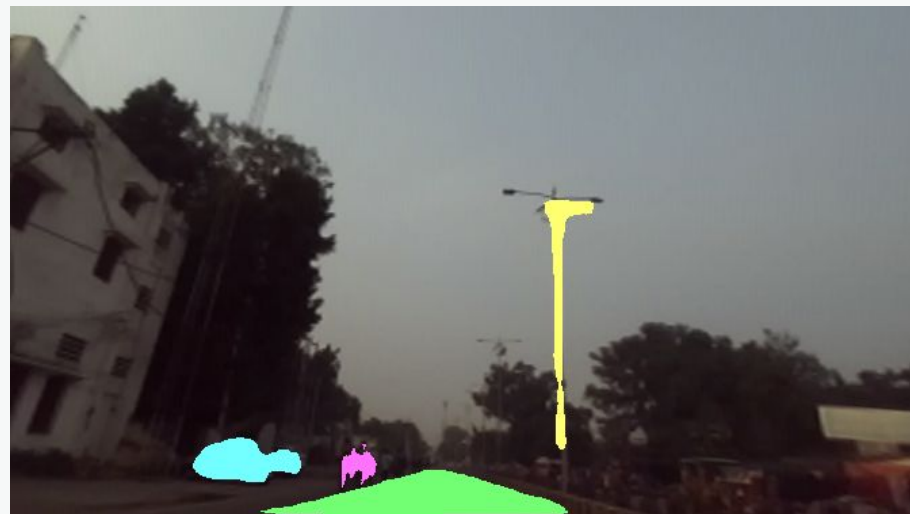


Results: IIITA dataset

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

Average IOU score = 84%

Results: Prayagraj dataset



Results: Prayagraj dataset

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%

Future Scope

- ❖ Integrate the above framework with an existing Visual SLAM method and evaluate performance improvements
- ❖ Use the network and extrapolate depth information from stereo cameras, thus leading to geographical coordinates of objects and the scene