Autonomous Parking Documentation

November 2020

The module consists of 2 main stages as shown in Figure 1.
The first section gives some details of the Neural Network(ShelfNet) used for road segmentation.
The second section gives an overview of the Algorithm.

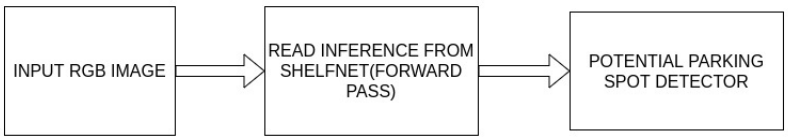

Figure 1: CodeFlow

1 Road Segmentation

- **Neural Network used for Road Segmentation:** ShelfNet
- **Input:** RGB Image
- **Output:** Segmentation map(result of road segmentation)
- **Training dataset:** 4270 images from IDD dataset + 1000 images of the IISC campus
- **IOU Obtained on the test dataset:** 91%
- **Inference Time:** 0.2 seconds for an image of size 1280 * 720.

2 Potential Parking Spot Detector

Algorithm 1: Auto Park Module Documentation

Input: RGB Image
Output: Midpoint of the potential parking spot
`// The camera intrinsics and the homography are assumed to be known`
Function `Parking_Spot_Estimator(`Seg_map`):`
  `//` Seg_map `is a binary image with the Road region shown in White and the background in Black`
  $l1(x)$ = Line equation in the image plane corresponding to a real world distance of 5m. (Computed using homography);
  $l2(x)$ = Line equation in the image plane corresponding to a real world distance of 10m. (Computed using homography);
  $flag\_top \leftarrow False$;
  $flag\_bottom \leftarrow False$;
  for $x \leftarrow 0$ **to** $num\_columns$ **do**
    if $Seg\_map[l1(x), x] == White$ *and* $flag\_top == False$ **then**
      $point\_topleft \leftarrow x, l1(x)$;
      $flag\_top \leftarrow True$;
    end
    if $Seg\_map[l2(x), x] == White$ *and* $flag\_bottom == False$ **then**
      $point\_bottomleft \leftarrow x, l2(x)$;
      $flag\_bottom \leftarrow True$;
    end
  end
  $px\_transvahan\_width \leftarrow W$;
  `// W = Width of the transvahan in pixels`
  $point\_topright = point\_topleft + px\_transvahan\_width$;
  $point\_bottomright = point\_bottomleft + px\_transvahan\_width$;
  $midpoint\_coordinate \leftarrow$
   FindMidpoint($point\_topleft$**,**$point\_topright$**,**$point\_bottomleft$**,**$point\_bottomright$**)**
   return $midpoint\_coordinate$
end

Function `Main`**:**
  while *frame* **do**
    $Segmentation\_map$ = **ShelfNetinference**(frame);
    $Midpoint\_coordinates =$
     `Parking_Spot_Estimator(`$Segmentation\_map$`)`;
  end
  return 0;
end