Module 5: Semantic Segmentation

Semantic segmentation is useful for a range of self-driving perception tasks such as identifying where the road boundaries are and tracking motion relative to lane markings.

自動運転以外：from tumor segmentation in CAT scans, to cavity segmentation in tooth x-ray images.

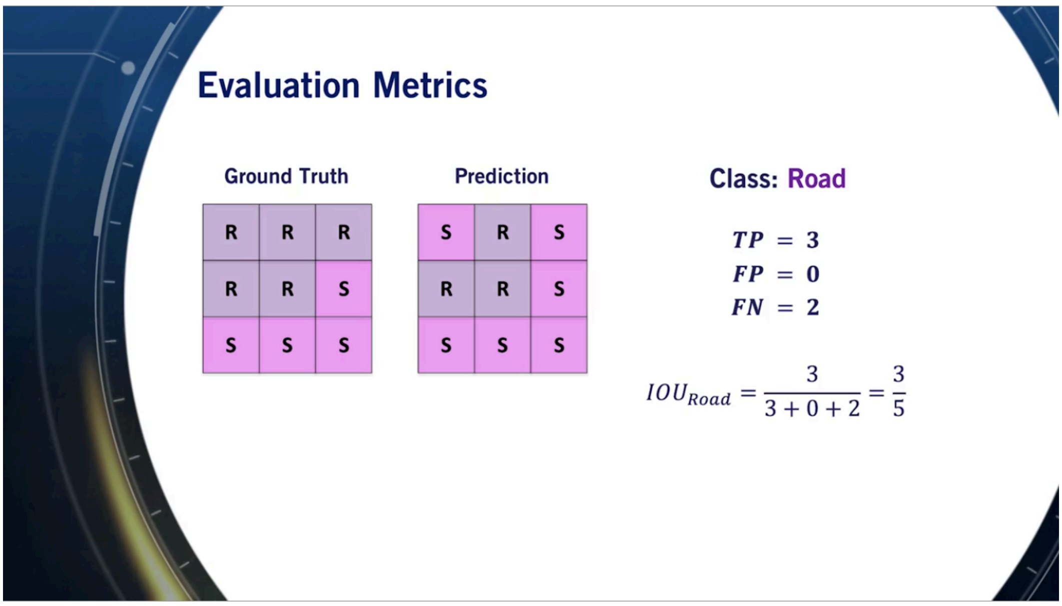
Lesson 1: The Semantic Segmentation Problem

単語
- pole: A pole is a long thin piece of wood or metal, used especially for supporting things.
- incidence: The incidence of something, especially something bad such as a disease, is the frequency with which it occurs, or the occasions when it occurs.

内容
- The semantic segmentation task problem formulation.
- Determine how well a semantic segmentation model is performing with task relevant performance measures.
- module 4の2D object detectionと同じように、先に問題記述や評価方法を説明する！

The Semantic Segmentation Problem
- Given an input image, we want to classify each pixel into a set of preset categories.
- Categories
  - static road elements: road, sidewalk, pole, traffic light, traffic sign.
  - dynamic obstacles: car, pedestrian, cyclist.
  - background class, vegetation, terrain, sky.
- input: every pixel.
- output: $f(x;\theta) = [s_{class_1}, \ldots, s_{class_k}]$.

Semantic Segmentation is Not Trivial!
- Occlusion, truncation, scale, and illumination changes.
- Smooth boundaries, an ambiguity of boundaries in image space.
  - thin objects, such as poles.
  - similar looking objects, such as a road and a sidewalk.
  - faraway objects.

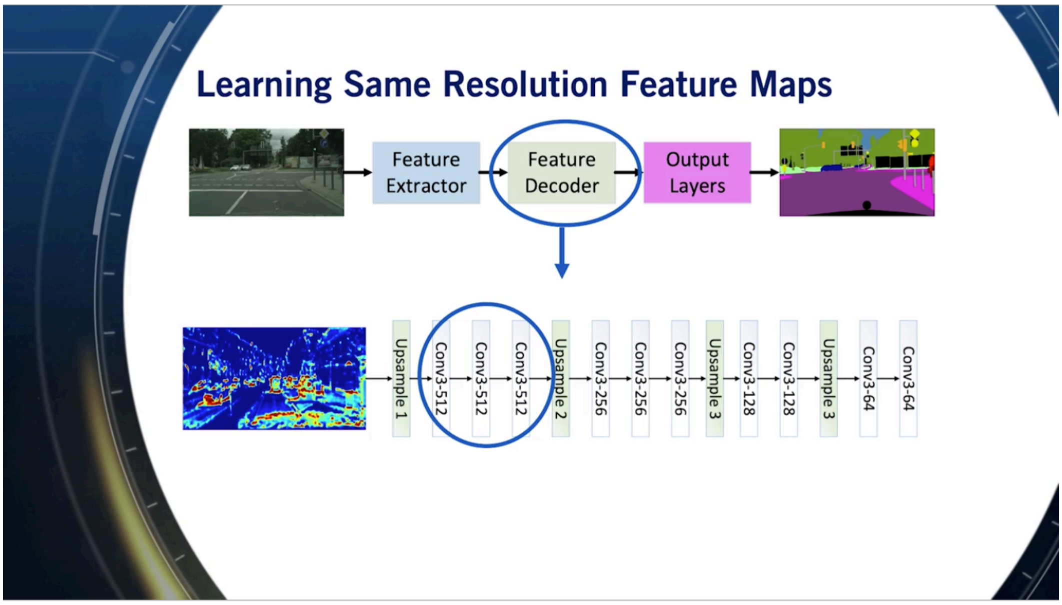


Evaluation Metrics
- True Positive (TP): The number of correctly classified pixels belonging to class X.
- False Positive (FP): The number of pixels that do not belong to class X in ground truth, but are classified as that class by the algorithm.
- False Negative (FN): The number of pixels that do belong to class X in ground truth, but are not classified as that class by the algorithm.
- $IOU_{class} = \dfrac{TP}{TP + FP + FN}$.
    - 注意：Class毎に計算するんだ！
    - 上記例のClass Sidewalkの場合：
        - $TP = 4$.
        - $FP = 2$.
        - $FN = 0$.
        - $IOU_{sidewalk} = \dfrac{TP}{TP + FP + FN} = \dfrac{4}{6}$.
- Class IOU over all the data is calculated by computing the sum of TP, FP, FN for all images first.
    - Computing the IOU per image and then averaging will actually give you an incorrect class IOU score.
- Averaging the class IOU is usually not a very good idea!
    - つまりclass毎にIOUを計算するまで。
- CitySpaces Segmentation Dataset.
    - The CitySpaces benchmark is one of the most used benchmarks for evaluating semantic segmentation algorithms.

Lesson 2: ConvNets for Semantic Segmentation

内容
- How to use convolutional neural networks to perform the semantic segmentation task.
- Different layers required for the good performance of semantic segmentation models.

- Using ConvNets for semantic segmentation is actually a little easier than using them for object detection.
- Unlike ConvNets for object detection, the training and inference stages are practically the same for semantic segmentation.



The Feature Extractor（大事）（Feature Decoderが必要である理由）
- Pooling Layerを使っちゃだめかも、pooling layerを使うと、feature mapのサイズが縮むので、Semantic Segmentationのpixel毎のClassificationと矛盾するでしょう。
  - Naive Upsamplingをやっても変わらない。
  - Naive Upsamplingじゃなく、Feature Decoder！

Learning Same Resolution Feature Maps
- The feature decoder can be thought of as a mirror image of the feature extractor.
- The Upsampling usually using nearest neighbor methods achieves the opposite effect to pooling, but results in an inaccurate feature map.
- The following convolutional layers are then used to correct the features in the upsampled feature map with learnable filter banks.
- This correction usually provides the required smooth boundaries as we go forward through the feature decoder.
- Output Layers: Softmax
  - This layer provides a k-dimensional vector per pixel with the kth element being how confident the neural network is that the pixel belongs to the kth class.

Classification Loss

$$L_{cls} = \frac{1}{N_{total}} \sum_i CrossEntropy(s_i^*, s_i).$$

- N_{total} is the number of pixels in all images of our minibatch.
- s_i is the output of the neural network.
- s_i^* is the ground truth classification.

論文：
- [2017] Pyramid Scene Parsing Network: https://arxiv.org/abs/1612.01105。またSenseTimeの論文！
- [2016] SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation: https://arxiv.org/abs/1511.00561

Lesson 3: Semantic Segmentation for Road Scene Understanding

内容
- How to use the output of semantic segmentation models to perform drivable space estimation.
- How to use the output of semantic segmentation models to perform semantic lane estimation.

大事な質問：drivable spaceの予測のメインの手法は本当にsemantic segmentationですか?

3D Drivable Surface Estimation
- In the context of semantic segmentation, the drivable surface includes all pixels from the road, crosswalks, lane markings, parking spots, and even sometimes rail tracks.
- Estimating a drivable surface is very important as it is one of the main steps for constructing occupancy grids from 3D depth sensors.

1. Generate semantic segmentation output.
2. Associate 3D point coordinates with 2D image pixels.
   1. either from stereo data or by projecting a LiDAR point cloud to the image plane.
3. Choose 3D points belonging to the Drivable Surface category.
4. Estimate 3D drivable surface model.
   1. The complexity of this model can range from a simple plane to more complex spline surface models.

2. splineという言葉はMatLabのDrivingScenario toolboxにも見たことがある。点列から道路を作る時！

Fit a planar drivable surface model given segmented image data and lidar points
- Plane Model: $ax + by + z = d$.
- Least squares formulation:
  - $p = [a, b, d], \underset{A}{argmin}(Ap - B)$.
  - $A = \begin{bmatrix} x_1 & y_1 & -1 \\ x_2 & y_2 & -1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & -1 \end{bmatrix}, B = \begin{bmatrix} -z_1 \\ -z_2 \\ \vdots \\ -z_N \end{bmatrix}$.
  - Solution: $p = (A^T A)^{-1} A^T B$.

Outliers対応（Batch Least Squares簡単ですが、OutliersのせいでPlaneの品質が悪くなったりする）
- Minimum number of points to estimate model: 3 non-collinear points.
- RANSAC algorithm:（Matched FeaturesからEssential Matrixを計算する関数のディフォルト手法）
  1. From your data, randomly select 3 points.
  2. Compute model parameters a, b, and d using least squares estimation.
  3. Compute number of inliers, N.
  4. If $N >$ threshold, terminate and return the computed plane parameters. Else, go back to step 1.
  5. Recompute the model parameters using all the inliers in the inlier set.
     1. EssentialMatrixを計算する関数にこれをやったかは不明。多分やったと思う。なぜなら、これをやらないと、僕maskを使ってinlier setを作って、また同じ関数を使ってEssential matrixを計算することになっちゃうので、不合理です。多分inliersだけでessential matrixを計算しているし、inliersのmaskも返す。
     2. また、最終的にinliersでessential matrixを計算しないと、RANSACアルゴにならないでしょう。。。

Semantic Lane Estimation
- Estimate the lane, the area where the car can drive on the drivable surface.
- Estimate what is at the boundaries of the lane:
  - Curb
  - Road（白線の意味だと思う）
  - Car
- The self-driving car has to base its maneuvers on what objects occur at the boundary of the lane, especially during emergency pull overs.
- Semantic lane estimationのタスクはestimating the lane and what occurs at its boundaries.

1. Extract segmentation mask from pixels belonging to lane separators such as lane markings or curbs.
2. Extract edges from this segmentation mask using an edge detector.
   1. Here, use canny edge detector.
   2. The output are pixels classified as edges that will be used to estimate the lane boundaries.
      1. つまり白線のedge以外の部分は使わないよ。
3. Linear Lane Model: Use the Hough transform to detect lines in the output edge map.
   1. Given an edge map, the Hough transform can generate a set of lines that link pixels belonging to edges in the edge map.

    2. The minimum length of the required lines can be set as a hyperparameter to force the algorithm to only detect lines that are long enough to be part of lane markings.
4. Filter lines based on slope to remove horizontal lines.
5. Remove any line that does not belong to the drivable space.
6. Determine which classes occur at the boundary of the lane.
    1. 例えば車のedgeがboundaryに出たら。

- Semantic lane detection and drivable surface estimation are the most prominent uses for semantic segmentation models in the context of self-driving cars.
    - 多分mobileyeのいわゆる白線はこのsemantic segmentationの応用でしょう。
- Semantic Segmentation Results can be used to aid in 2D object detection and localization.
- Semantic segmentation is a powerful tool for self-driving cars and a core component of the high level perception stack.