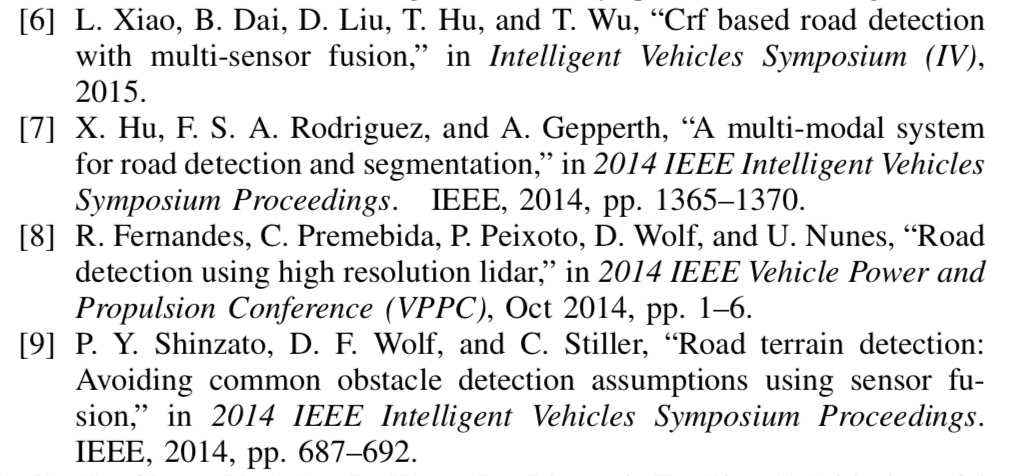
Fast LIDAR-based Road Detection Using Fully Convolutional Neural Networks

Abstract：

本文提出一种仅使用LiDAR数据的深度学习算法来进行road detection，从非结构point cloud开始，顶视图编码几种基本统计量，例如：平均海拔（mean elevation）、密度等。通过考虑顶视图表示，道路检测被简化为单一尺度问题，可以通过简单快速的完全卷积神经网络（FCN）来解决。FCN通过large感受野和高分辨率特征图谱来进行像素级语义分割任务。

Introduction：

1. 大多数使用深度神经网络的Road Detection方法都是基于单目摄像头的，但基于摄像头的方法受到环境照明的强烈影响。
2. LiDAR对环境照明不敏感，特别适合于实现高水平的自动驾驶。一些论文提出的算法仅仅是使用了LiDAR数据，或者是camera和lidar的fusion但是没有使用深度学习，且准确性表现的并不好：



1. 在本文中，road detection任务被构造成使用FCN在point cloud top-view图像中进行像素级语义分割任务，可以实现real-time且在KITTI数据集上获得了很好的准确性。

Point cloud top-view road detection：

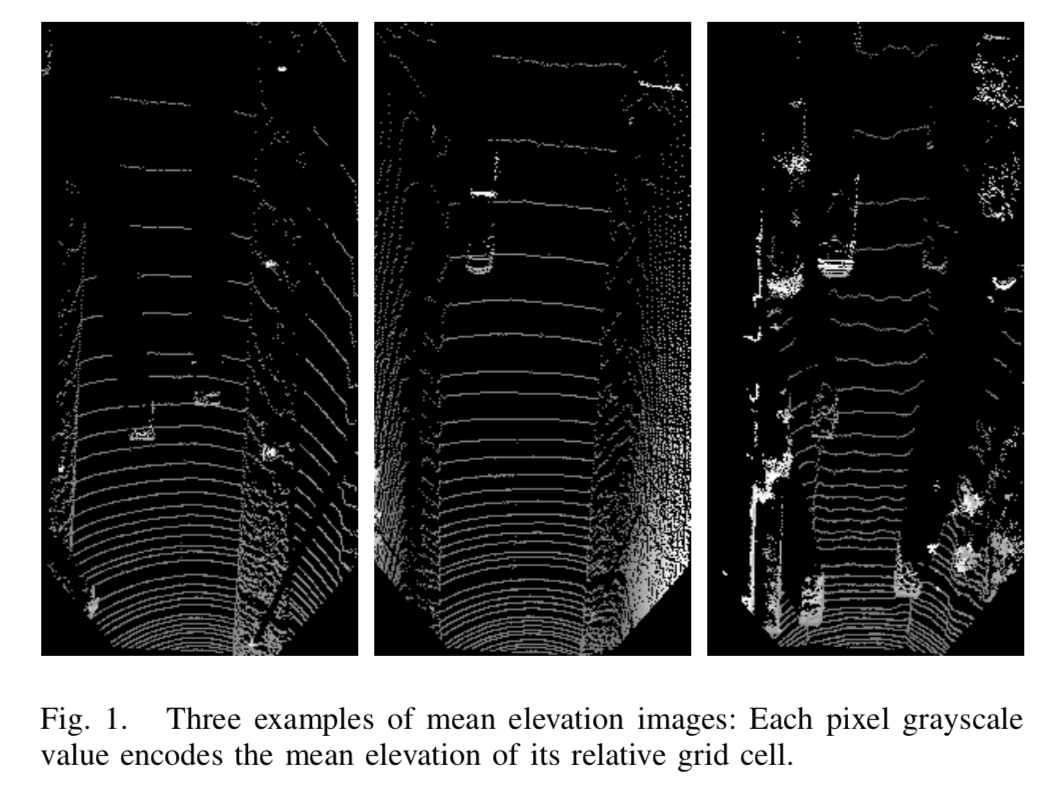
本文中的road detection被定义为评估所有可通行区域(而不是单一的lane),因此本文不考虑自我车道和迎面而来通向车道的区别。

从非结构化点云开始，生成车辆周围环境的俯视图像。 每个图像编码几个基本统计中的一个，例如平均高度和平均反射率。顶视图比相机透视图更合理（根据：J.Fritsch,T.Kuehnl,andA.Geiger,“Anewperformancemeasureand evaluation benchmark for road detection algorithms,” in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.）因为车辆控制和道路规划都是在这个2D world中执行的。更为重要的是，使用top-view images，分类任务可以被简化成单一尺度问题，只用考虑覆盖面积而不用考虑物体在image中的image。The neural network is fully convolutional and can therefore process images of any size. An advantage of this design choice is that road detection can be carried out in regions of interest (ROIs) that can be dynamically changed and, in the case of rotating LIDARs, can even span a 360◦ view around the vehicle.

1. From point cloud to top-view images:

An unstructured point cloud must be transformed into a suitable format before it can be used as input for an FCN.

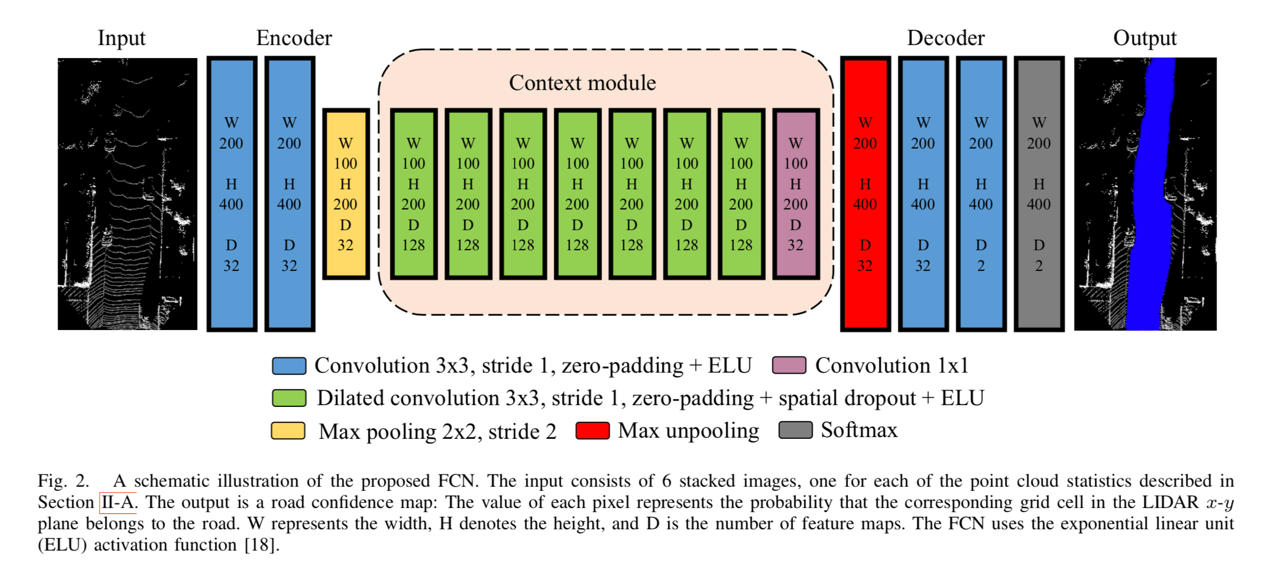
1. 在LIDAR的x-y平面中创建网格，并将点云的每个元素分配给其中一个单元格。网格是40\*20，其中y[-10,10], x[6,46],每一个cells的大小是0.10\*0.10
2. 为每一个gird cell计算基本统计量：points的数目、平均反射率、平均值、标准差、最大最小海拔。
3. 最后，通过将网格单元视为像素，生成六个图像，每个图像用于上述统计中的每一个。



即：最后生成的6张top-view分别表示6个统计量，这6张图中的像素点的灰度值就是每一个cells中计算出来的统计量，每一个cells就是一个像素，然后每个cells中包含的points计算相应的统计量，这个统计量就是这个cell表示的像素的像素值

1. FCN architecture

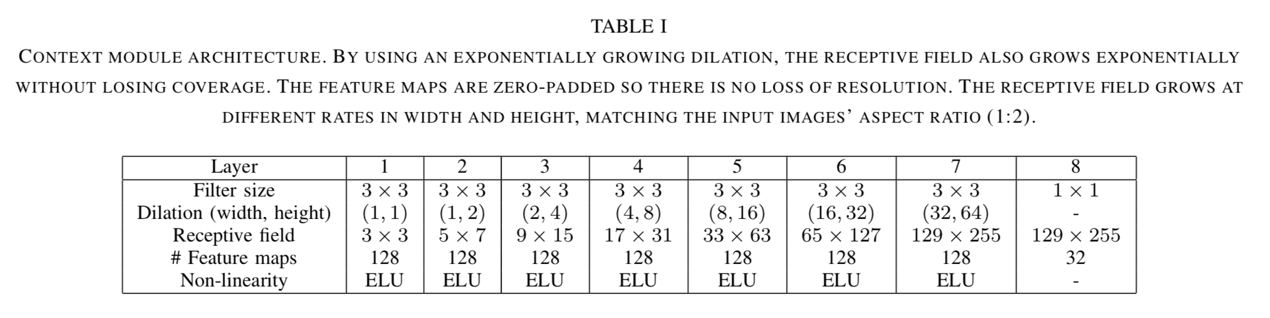
本文中，考虑到lidar和camera的数据有本质的不同，因此要从头开始训练一个FCN。训练数据仅使用KITTI的训练数据。



1. 1)  A six-channel input layer, one channel for each of the point cloud statistics described in Section II-A.
2. 2)  An encoder with the main purpose of subsampling the feature maps, thus reducing the FCN’s memory requirements. Subsampling is carried out by using a max pooling layer with a 2 × 2 window and stride 2.
3. 3)  A context module that aggregates multi-scale contextual information(多尺度上下文信息) by using dilated convolutions (扩展卷积)[16]. More details are provide in Section II-C.
4. 4)  A decoder that upsamples the feature maps back to the input size by using a max-unpooling layer [14] followed by two convolutional layers.
5. 5)  An output layer that returns a road confidence map, that is, an image where the value of each pixel represents the probability that its corresponding grid cell in the LIDAR x-y plane (see Section II-A) belongs to the road.
6. Context module

在保持参数和层数较小的同时扩展感受野的有效策略是采用扩张（膨胀）卷积算子/又称：空洞卷积算子，该算子支持感受野的指数扩展而不会丢失分辨率（即，特征图不会减小尺寸）或丢失覆盖范围。

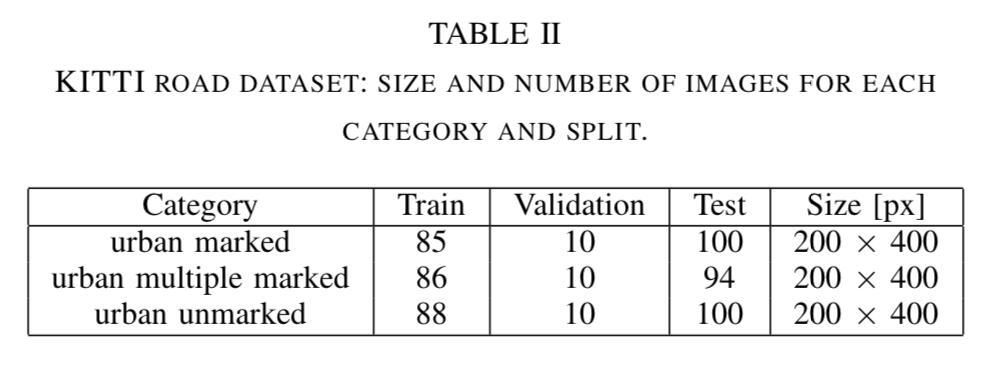
本文中采用的Context module结构如下：



Data Set AND Step:

1)KITTI Data set:

本文使用KITTI road benchmark data set，289张training images和290张test images。Ground truth注释在相机透视空间中表示，仅适用于训练集。Lidar point cloud数据作为数据集的扩展数据被提供。



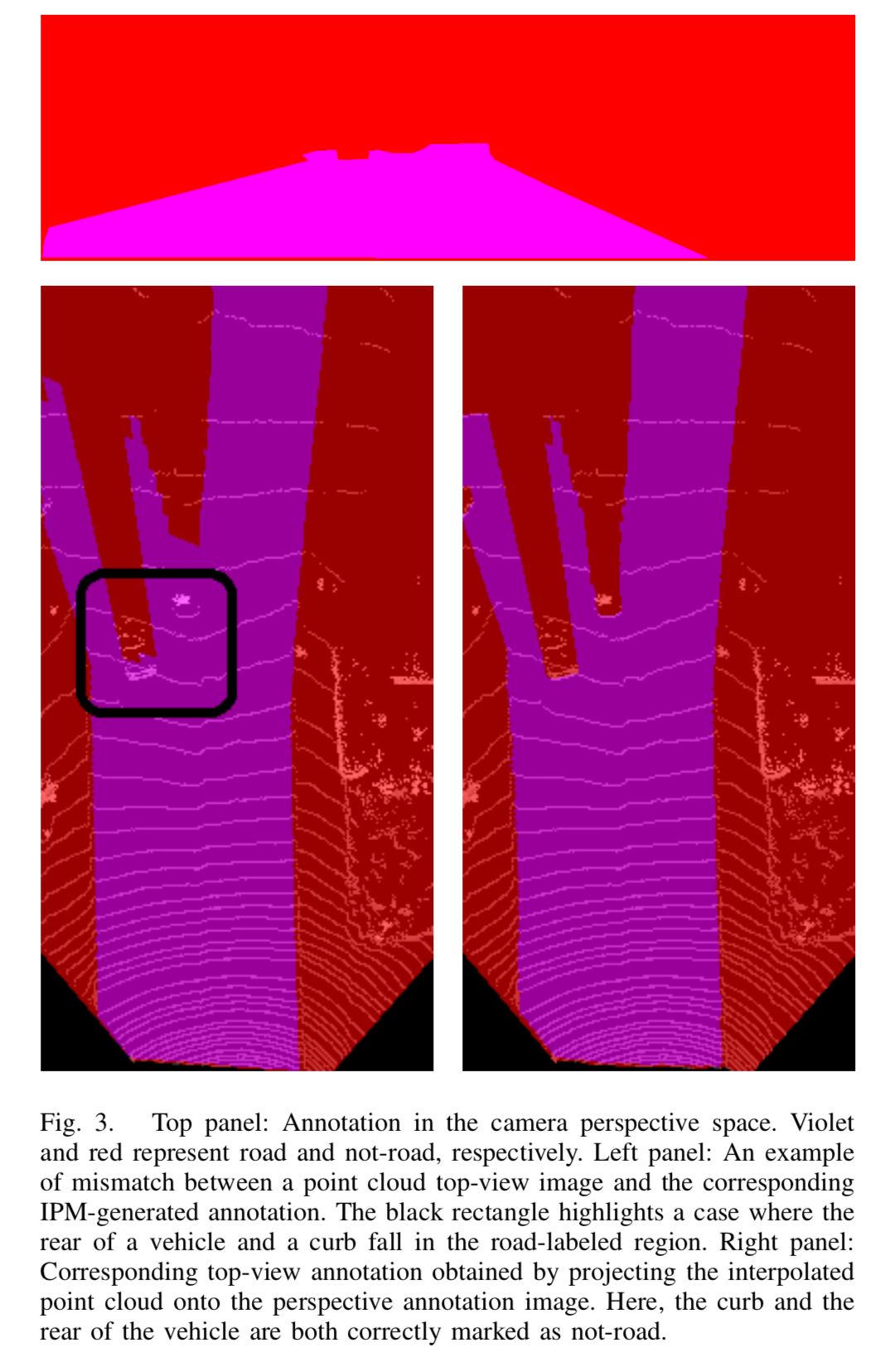
训练集用于计算目标函数和调整FCN的权重，而验证集用于决定何时停止训练和选择超参。

2）Data augmentation

防止过拟合，做一些数据扩充和增强

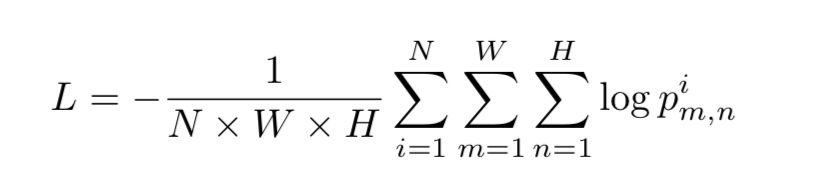
3）Inverse perspective mapping vs. point cloud projection

KITTI数据集的ground truth注释是相机透视空间的，而本文提出的算法是基于top-views images的，因此要将数据进行空间转化，IPM（逆透视映射）是一种方法，但是IPM假设平坦和无障碍道路，这在现实中不切实际，因此，有一种可行的办法。将点云投影到相应的摄像机视图注释中，以确定其哪些点属于道路，然后使用类似于第II-A节中描述的过程，但考虑class而不是elevation和reflectivity统计量。为了增加点的密度并获得密集的注释，在执行投影之前，将点云在窄圆扇区内线性插值。



1. Training

FCN采用Adam优化，学习率0.01，用cross-entropy作为目标函数，loss定义为：



W，H为softmax layer的output的width和height，N是batchsize，p是FCN正确类预测的概率。对于正则化，在每个扩张的卷积层之后添加dropout layer（pd = 0.25）[20]。 FCN使用Torch7框架实现，并在具有6GB内存的NVIDIA GTX980Ti GPU上进行了培训。