# 第十三期 车道线检测 LaneNet

论文：<https://arxiv.org/abs/1802.05591?context=cs>

代码：https://github.com/MaybeShewill-CV/lanenet-lane-detection

作者：张松

**核心思想：**

采用实例分割的方式得到每条车道线的像素点集，通过学习出路面的透视投影矩阵，将前视图转换到鸟瞰图视角，拟合每条车道线的曲线方程，最后逆变换回原视图。

目录：

一、label数据

二、框架结构

三、实验参数

四、评价标准

# 一、label数据

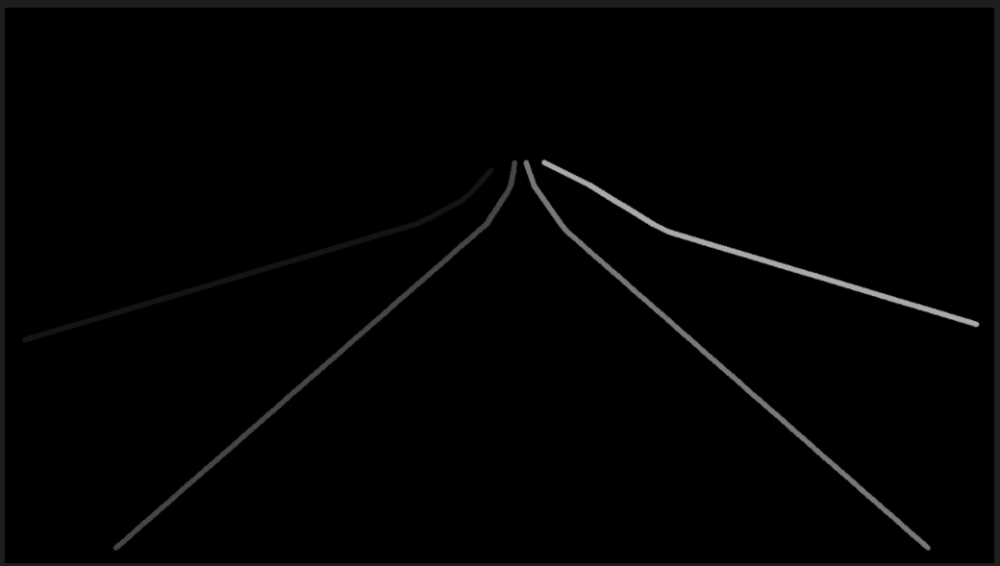
数据集：[Tusimple](https://github.com/TuSimple/tusimple-benchmark/issues/3)

包含三个部分：

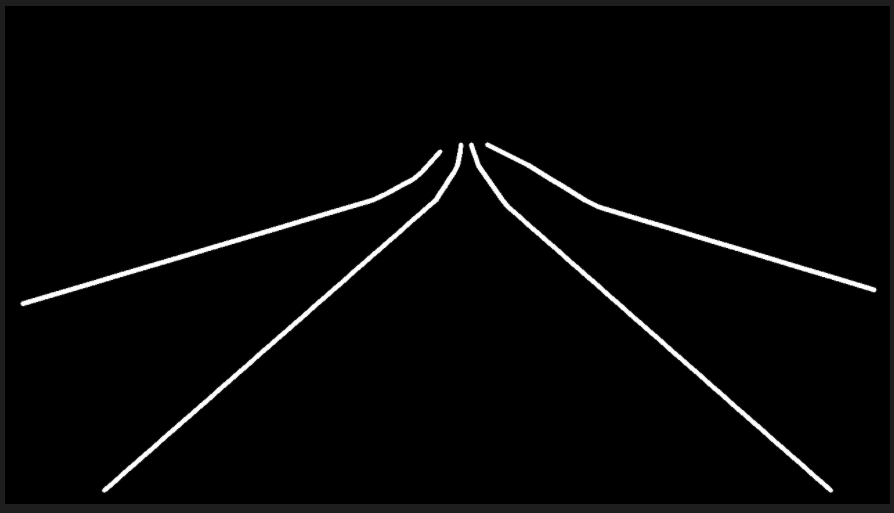
原图：



gt\_instance\_image实例分割图：

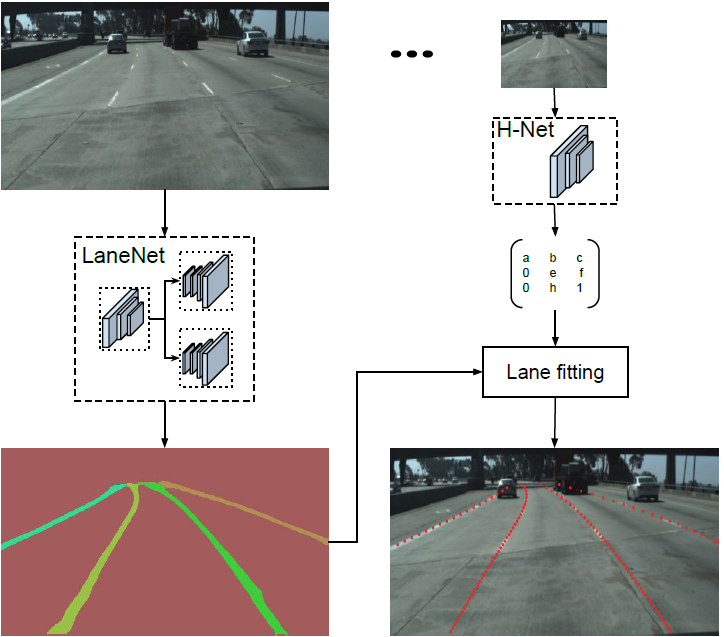


gt\_binary\_image二进制图片

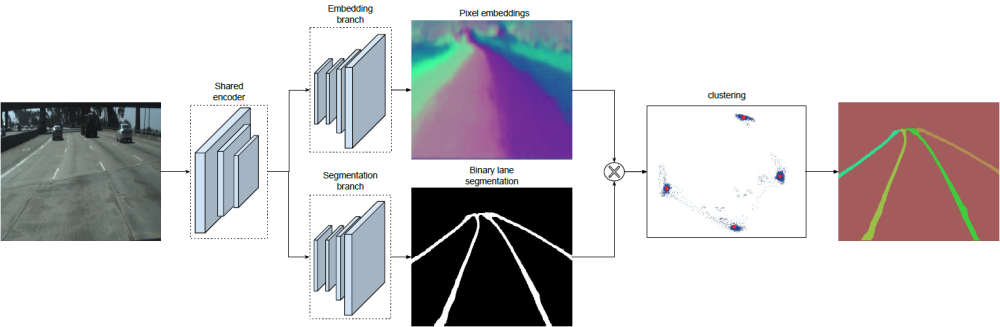


# 二、框架结构

本文提出一种端到端的车道线检测算法，包含 **LanNet** + **H-Net** 两个网络模型，两个网络模型分开独立训练。其中 LanNet 是一种将语义分割和对像素进行向量表示结合起来的**多任务模型**，最后利用**聚类**完成对车道线的实例分割。H-Net 是有个小的网络结构，负责预测**变换矩阵** **H**，使用转换矩阵 H 对同属一条车道线的所有像素点进行重新**建模**（使用 y 坐标来表示 x 坐标）。



## LanNet



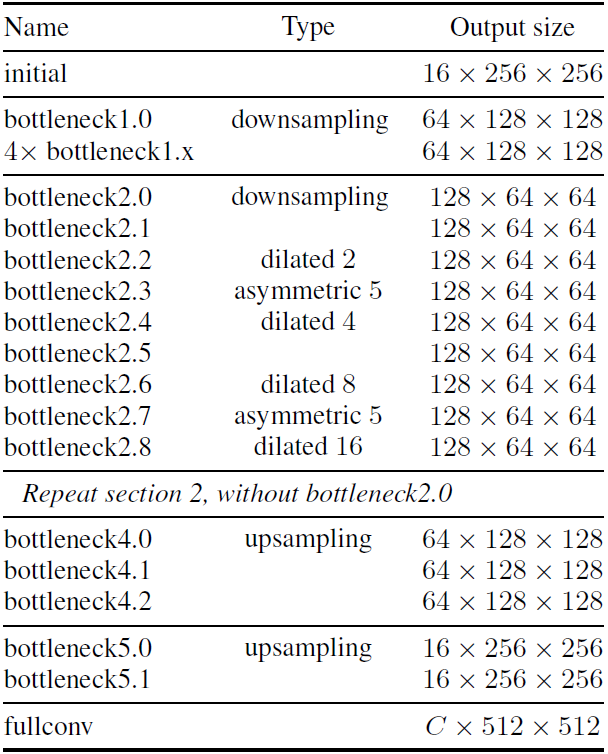
论文中将实例分割任务拆解成语义分割（LanNet 一个分支）和聚类（LanNet一个分支提取 embedding express, Mean-Shift 聚类)两部分。如上图所示，LanNet 有两个分支任务，分别为 a lane segmentation branch and a lane embedding branch。Segmentation branch 负责对输入图像进行语义分割（对像素进行二分类，判断像素属于车道线还是背景）；Embedding branch 对像素进行嵌入式表示，训练得到的 embedding 向量用于聚类。最后将两个分支的结果进行结合利用 Mean-Shift 算法进行聚类，得到实例分割的结果。

## 网络结构

LaneNet是基于[ENet](https://arxiv.org/abs/1606.02147) 的encoder-decoder模型，如图5所示，ENet由5个stage组成，其中stage2和stage3基本相同，stage1,2,3属于encoder，stage4,5属于decoder。

如上图所示，在LaneNet 中，语义分割和实例分割两个任务共享 stage1 和 stage2，并将 stage3 和后面的 decoder 层作为各自的分支(branch)进行训练；其中，语义分割分支(branch)的输出 shape 为W\*H\*2，实例分割分支(branch)的输出 shape 为W\*H\*N，W,H分别为原图宽和高，N 为 embedding vector 的维度；两个分支的loss权重相同。

[ENet](https://arxiv.org/abs/1606.02147)网络结构如下：

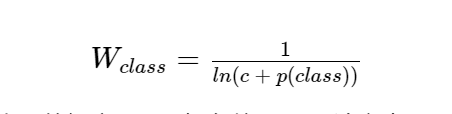


## 语义分割

在设计语义分割模型时，论文主要考虑了以下两个方面：

1.在构建 label 时，为了处理遮挡问题，论文对被车辆遮挡的车道线和虚线进行了还原（估计）；

2. Loss 使用 **softmax\_cross\_entropy**，为了解决样本分布不均衡的问题（属于车道线的像素远少于属于背景的像素），参考论文[ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation](https://arxiv.org/abs/1606.02147) ，使用了 **bounded inverse class weight** 对 loss 进行加权:

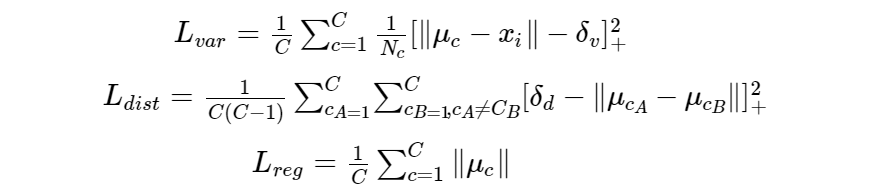


其中，p 为对应类别在总体样本中出现的概率，c 是超参数（ENet论文中是1.02）。

## 像素 embedding

为了区分车道线上的像素属于哪条车道，embedding\_branch 为每个像素初始化一个 embedding 向量，并且在设计 loss 时，使属同一条车道线的表示向量距离尽可能小，属不同车道线的表示向量距离尽可能大。

这部分的 loss 函数是由三部分组成：方差 loss(Lvar) 和距离 loss(Ldist)：



其中，C 是车道线数量，Nc 是属同一条车道线的像素点数量，μc是车道线的均值向量，xi是像素向量（pixel embedding)，[x]+=max(0,x)。注意这里先执行 + 操作，再执行 2操作。

该 loss 函数源自于论文 [《Semantic Instance Segmentation with a Discriminative loss function》](https://arxiv.org/abs/1708.02551v1.pdf)

**同一车道线的像素向量，距离车道线均值向量 μc 超过 δv时， pull force(Lvar) 才有意义，使得 xi靠近 δd；**

**不同车道线的均值向量 μcA和 μcB之间距离小于 δd时，push force(Ldist) 才有意义，使得 μcA 和 μcB 彼此远离。**

## 聚类

注意，聚类可以看做是个后处理，上一步里 embedding\_branch 已经为聚类提供好的特征向量了，利用这些特征向量我们可以利用任意聚类算法来完成实例分割的目标。

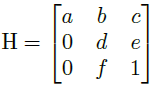
为了方便聚类，论文中设定δd>6δv。

在进行聚类时，首先使用 [mean shift](https://www.cnblogs.com/xfzhang/p/7261172.html)聚类，使得簇中心沿着密度上升的方向移动，防止将离群点选入相同的簇中；之后对像素向量进行划分：以簇中心为圆心，以2δv 为半径，选取圆中所有的像素归为同一车道线。重复该步骤，直到将所有的车道线像素分配给对应的车道。

## H-Net

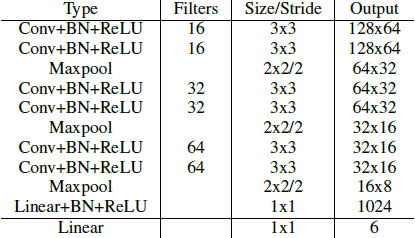
LaneNet的输出是每条车道线的像素集合，还需要根据这些像素点回归出一条车道线。传统的做法是将图片投影到 bird’s-eye view 中，然后使用 2 阶或者 3 阶多项式进行拟合。在这种方法中，变换矩阵 H 只被计算一次，所有的图片使用的是相同的变换矩阵，这会导致地平面（山地，丘陵）变化下的误差。

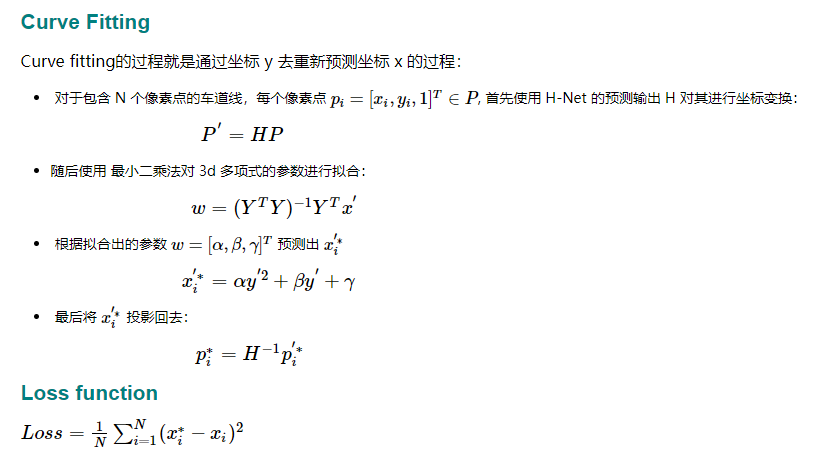
为了解决这个问题，论文训练了一个可以预测变换矩阵 H 的神经网络 H-Net，网络的输入是图片，输出是变换矩阵 H：



**通过置 0 对转置矩阵进行约束，即水平线在变换下保持水平。（即坐标 y 的变换不受坐标 x 的影响）**

由上式可以看出，转置矩阵 H 只有6个参数，因此H-Net的输出是一个 6 维的向量。H-Net 由 6 层普通卷积网络和一层全连接网络构成，其网络结构如图所示：





# 三、实验参数

## LanNet

Dataset : Tusimple

Embedding dimension = 4

δ\_v=0.5

δ\_d=3

Image size = 512\*256

Adam optimizer

Learning rate = 5e-4

Batch size = 8

## H-Net

Dataset : Tusimple

3rd-orderpolynomial

Image size =128\*64

Adam optimizer

Learning rate = 5e-5

Batch size = 10

# 四、评价标准

设定 G1 代表 GT二值图里像素值为 1 部分的集合，P1 表示检测结果为 1 的集合。

官方:





