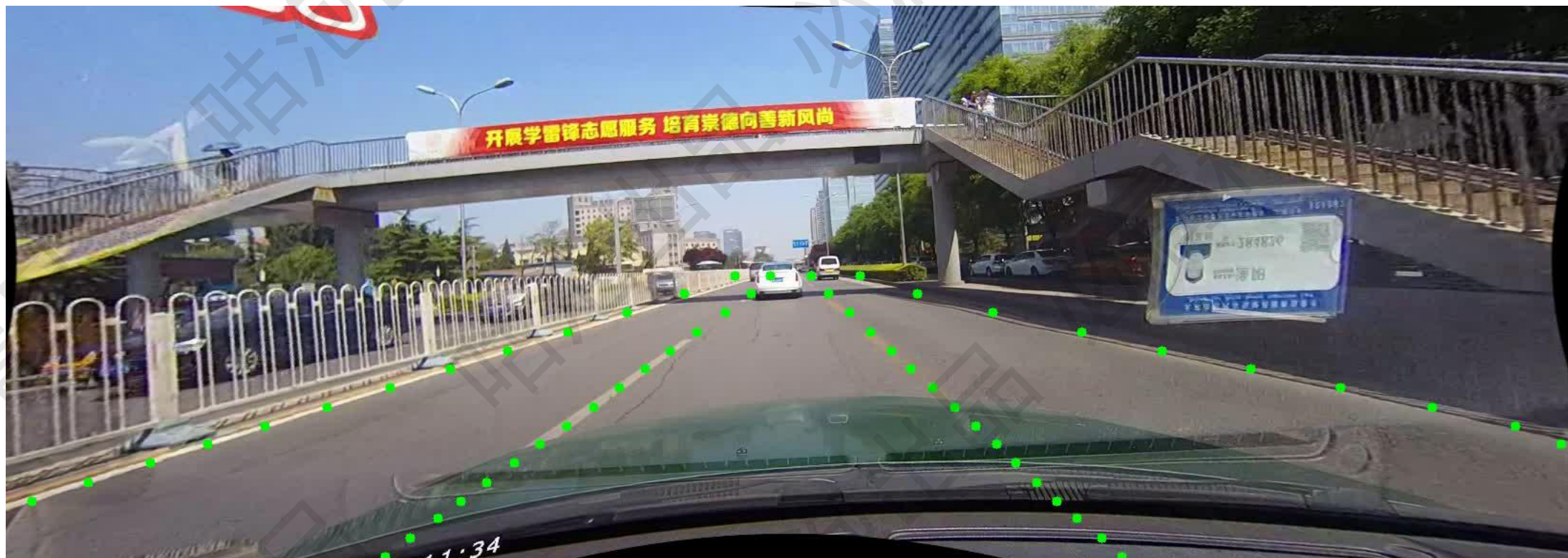


Deep Lane Detection

✓ 检测效果

📎 项目车道线检测输出结果展示

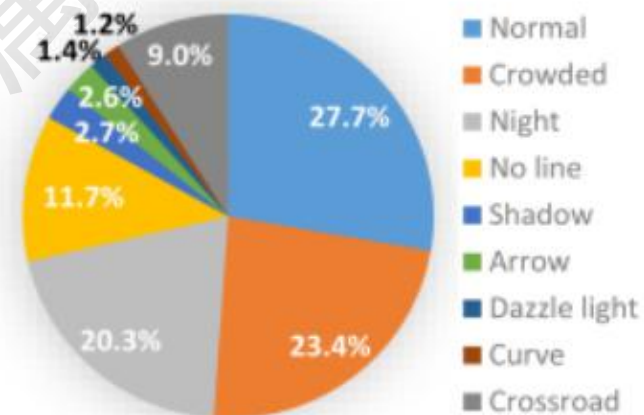
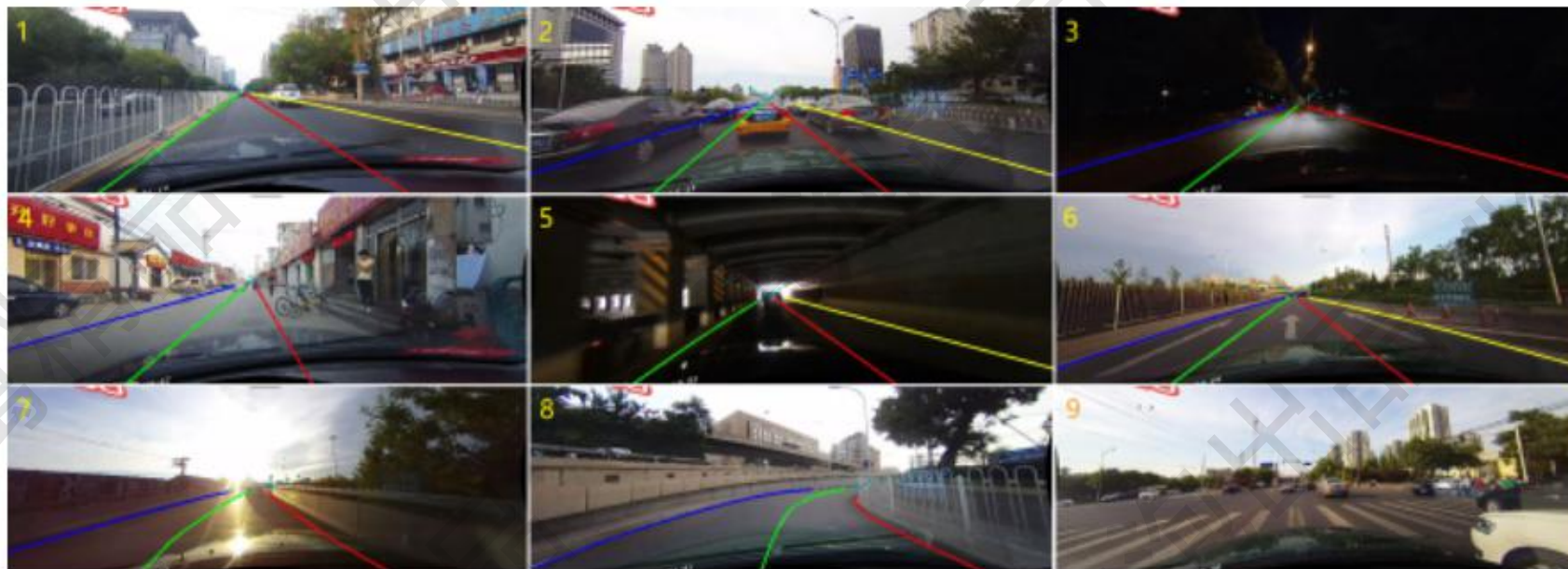


Deep Lane Detection

✓ 数据集介绍:

✎ 常用数据集: CULane Dataset, Tusimple(特点就是非常大, 视频文件)
<https://xingangpan.github.io/projects/CULane.html>

✎ made in beijing, 国内的车道线检测benchmark



Deep Lane Detection

✓ 数据增强

✎ 主要就是平移操作，小幅度旋转等



(a) Original anaotation



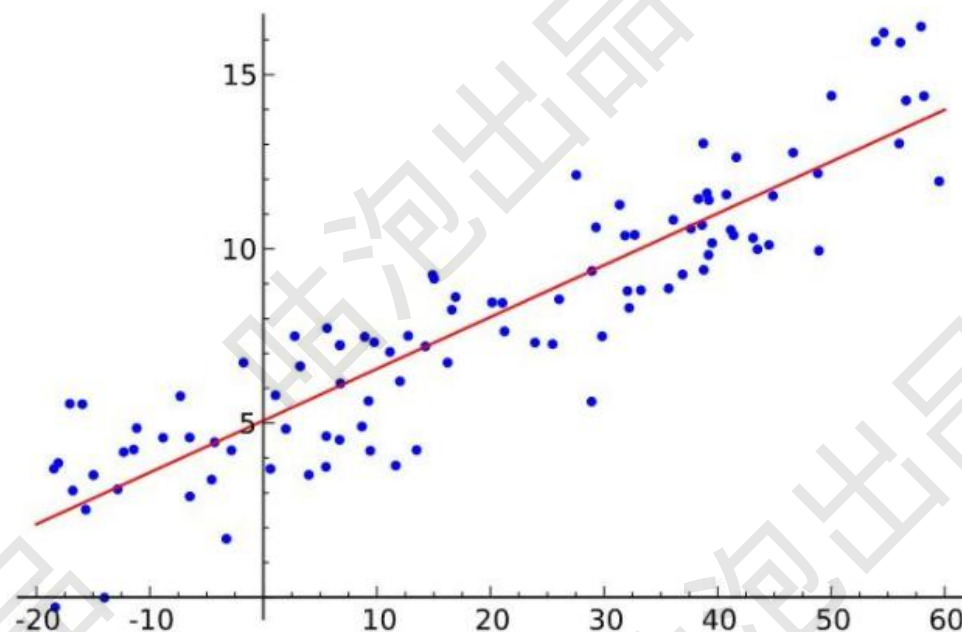
(b) Augmentated result

Deep Lane Detection

✓ 标签延伸

✎ 标注数据中很多车道线没到尽头就没了，可以再补全标签

✎ 通过一个线性拟合来得到没到尽头的那些线该咋延伸

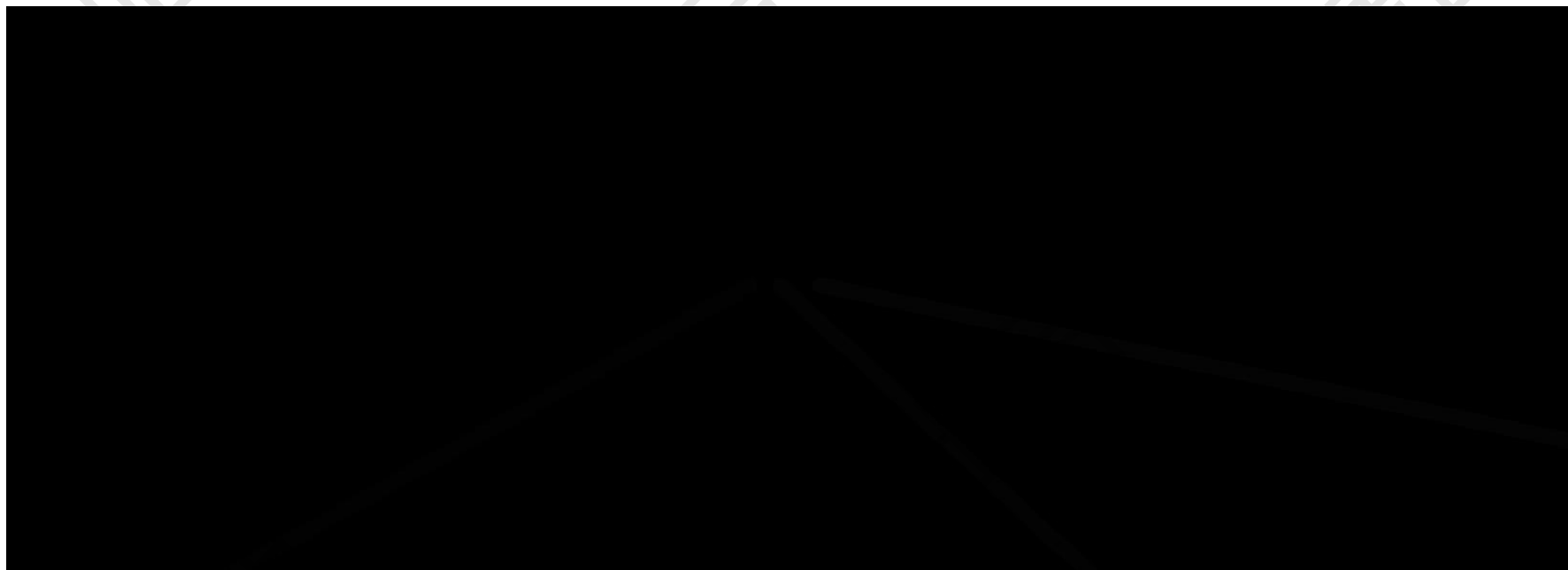


Deep Lane Detection

✓ 标签与输出结果

✎ 这标签咋看起来一片黑呢，其实里面有信息，源码里才能读出来

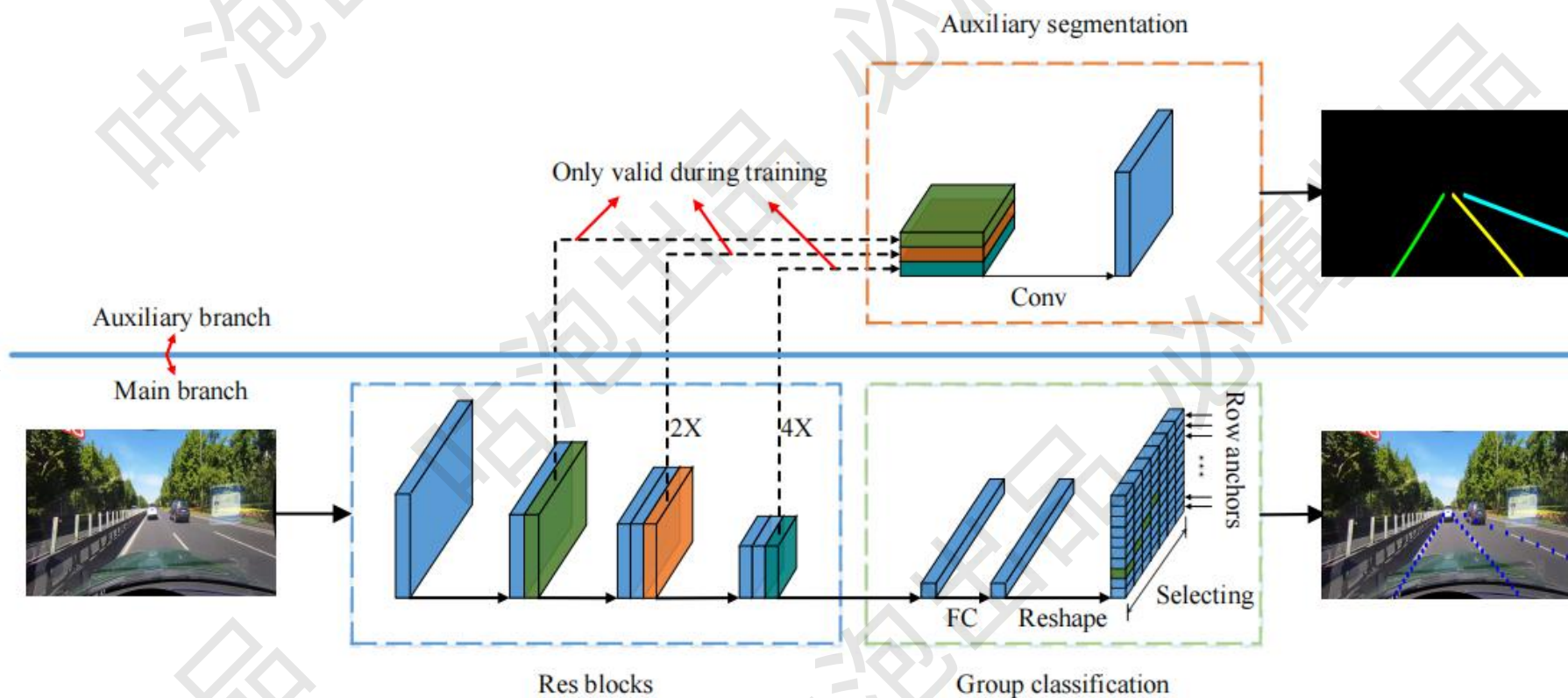
✎ 根据模型需要，输出标签为[4,18]矩阵，表示4条车道线在18个位置上的具体点



Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

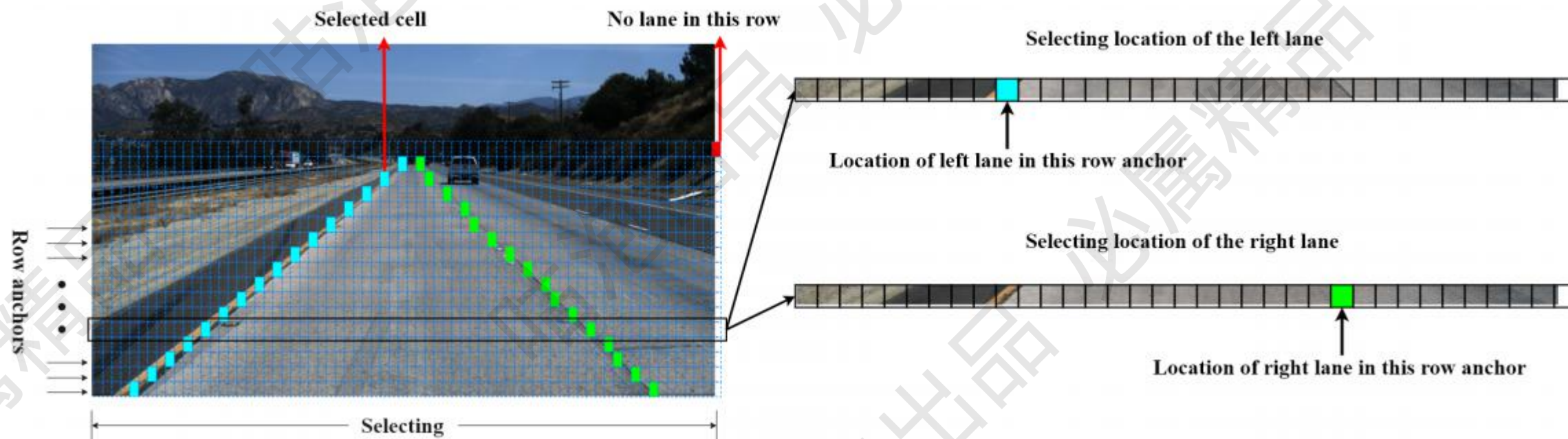
✎ 输入图像: $[\text{batch}, 3, 288, 800]$ 返回backbone中三层的结果(分割要用的)



Deep Lane Detection

✓ 任务分析

✎ 其实就是把H和W两个维度展开，做成网格来进行预测任务



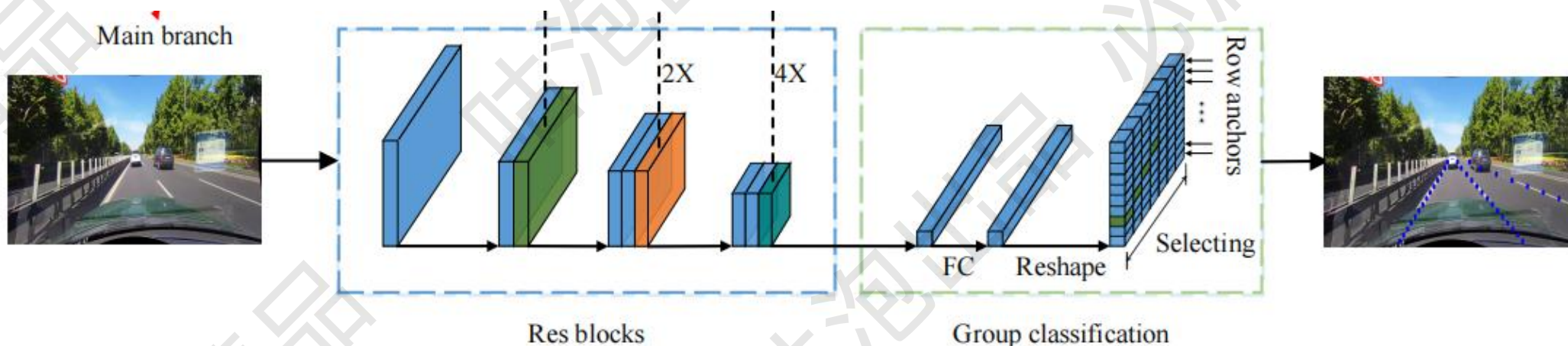
Deep Lane Detection

✓ 注意咱们最后完成的是分类任务

✎ 最后全连接输出14472个特征，为啥是这个数呢？

✎ $14472 = [201, 18, 4]$ ；其中 $[18, 4]$ 是要与标签对应，201则是分类概率

✎ 其中0-199表示位置类别，201表示是否不存在车道



Deep Lane Detection

✓ 损失函数

✎ 1. 分类损失, Focal Loss : $L_{cls} = \sum_{i=1}^C \sum_{j=1}^h L_{CE}(P_{i,j,:}, T_{i,j,:})$, (源码中考虑了样本权重)

✎ 2. 相似损失: $L_{sim} = \sum_{i=1}^C \sum_{j=1}^{h-1} \|P_{i,j,:} - P_{i,j+1,:}\|_1$, (相邻的位置分类结果应接近)

✎ 3. 期望替代绝对: 预测得到的实际位置点由200个大众评委期望得到

✎ 常规计算: $Loc_{i,j} = \underset{k}{\operatorname{argmax}} P_{i,j,k}$, s.t. $k \in [1, w]$ $Prob_{i,j,:} = \operatorname{softmax}(P_{i,j,1:w})$,

预测位置: $Loc_{i,j} = \sum_{k=1}^w k \cdot Prob_{i,j,k}$, 损失计算: $L_{shp} = \sum_{i=1}^C \sum_{j=1}^{h-2} \| (Loc_{i,j} - Loc_{i,j+1}) - (Loc_{i,j+1} - Loc_{i,j+2}) \|_1$,

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息

Deep Lane Detection

✓ 特征匹配完之后能做什么呢?

✎ 两张图像匹配后，我们可以知道它俩的位姿(位移，角度等)变化

✎ 这就是咱们在机器人导航和三维重构中非常重要的一个模块

✎ 图像相似度计算(基于匹配到的点)，图像检索与匹配等

✎ 相当于可以根据关键点的匹配特征得到很多位置相关的信息