文章编号：2095-2163(2021)06-0005-09 中图分类号：TP391 文献标志码：A

基于深度学习的连续帧车道线检测网络

孔 健，李 烨，尹 婷
（上海理工大学 光电信息与计算机工程学院，上海 200093）

摘 要：为了改善单帧图像检测复杂背景中车道线性能较差问题，例如车道线受到阴影影响、污渍污损或人车遮挡等情况时性能较差的问题。本文提出了一种基于连续帧的车道线检测网络，实现了卷积神经网络（Convolutional Neural Network，CNN）和长短期记忆网络 LSTM（Long Short-Term Memory，LSTM）的融合。首先，编码器 CNN 对连续帧进行特征提取，生成多尺度特征映射；其次，输入对应的双层 ConvLSTM 网络，捕获连续帧的时空信息；最后，捕获的时空信息在解码器 CNN 中进行多尺度特征融合，产生车道线预测的分割图。实验结果表明，所提网络的准确率、召回率和 F1 值较高，分别达到了 85.8%、96.1%和 90.0%，总体上 F1 相对于原始 CNN 网络提高了约 4%，在某些复杂路况下 F1 的提升在 10%以上。与其它网络相比，本文提出的网络具有较高的准确率、召回率和 F1 值，同时运行时间并没有大幅增加，实时性得到保障。

关键词：车道线检测；卷积神经网络；LTSM；多尺度特征融合

Continuous frame lane line detection network based on deep learning

KONG Jian, LI Ye, YIN Ting
(School of Optoelectronic Information and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, China)

【Abstract】 In order to improve the performance of single-frame image detection of lane lines in dealing with complex road conditions, such as lane lines affected by shadows, stains, or occlusion by people and vehicles, the performance of the lane line is poor. A lane line detection network based on continuous frames is proposed, which realizes the fusion of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). First, the encoder CNN performs feature extraction on continuous frames to generate a multi-scale feature map, and then enters the corresponding double-layer ConvLSTM network to capture the spatiotemporal information of the continuous frames. Finally, the captured spatiotemporal information is feature fused in the decoder CNN to produce lane line prediction. Segmentation diagram. Experimental results show that the accuracy, recall, and F1 value of the proposed network are high, reaching 85.8%, 96.1%, and 90.0% respectively. In general, F1 is increased by about 4% compared to the original CNN network. F1′s improvement in road conditions is more than 10%. Compared with other networks, the proposed network has a higher accuracy rate, recall rate and F1 value. At the same time, the running time has not increased significantly, and the real-time performance is guaranteed.

【Key words】 lane detection; convolutional neural network; LTSM; multi-scale feature fusion

0 引 言

传统的车道线检测方法依赖于手工特征提取来检测车道线。手工特征一般是基于颜色、边缘等，这些特征可与霍夫变换或卡尔曼滤波器结合在一起预测车道线。这些方法很简单，实时性好，对平台的要求也比较低。但是，其性能取决于测试环境，例如照明条件和是否遮挡，面对复杂路况时检测鲁棒性很差，甚至检测不出车道线，对于安全行驶决策是致命的。

近期车道线检测采用深度学习网络来提取特征，在复杂的场景中具有出色的性能，其中卷积神经网络（Convolutional Neural Network，CNN）方法在计算机视觉领域的表现尤为突出。车道线检测通常基于语义检测任务，对图像每一个像素分配一个二进制标签，以指示其是否属于车道线。尽管这些方法取得了出色的性能，但由于其采用多分类方法来区分每个车道，因此只能应用于固定车道数量的场景。SCNN 网络按照一定方向（上、下、左、右），按照顺序进行卷积，适用于车道线这种持续延伸的目标[1]；LaneNet 将问题归结为实例分割，使用用于特征提取的共享编码器和 2 个解码器，其中一个解码器执行二进制车道线分割，另一个解码器进行实例分割，使得该网络可以检测任意数量的车道线[2]；PointLaneNet 结合关键点检测与点云实例分割进行车道线检测，也可适用于检测任意场景和任意数量的车道线[3]。

由于驾驶场景是连续的，在相邻帧之间存在大量重叠画面，因此相邻帧中车道线的位置具有高度

作者简介：孔 健（1996-），男，硕士研究生，主要研究方向：深度学习、图像检测。

收稿日期：2021-03-28

的相关性。更准确地说,即使车道线可能会受到阴影、污渍和遮挡带来的损坏或退化,当前帧中的车道线仍可以通过前面多个帧进行预测。使用多帧进行车道线检测,涉及到时域信息的提取,RNN 具有连续信号处理、序列特征提取和综合等优点,但是单纯使用 RNN 进行图像处理会产生大量参数,造成沉重的计算负担。同时车道线大小长宽不固定,采用单一尺度特征图进行检测,效果不是很好。

为了改善这些问题,提出了一种基于连续帧的车道线检测网络。

(1)提出了一种新的融合策略,将 CNN 与长短期记忆网络 LSTM(Long Short-Term Memory,LSTM)融合;

(2)LSTM 被实现为双向 ConvLSTM,捕获来自正反方向的时空信息;

(3)编码器 CNN 生成多尺度特征映射,输入相应的双层 LSTM 网络捕获连续帧的时空信息,最后在解码器 CNN 中进行特征融合。

1 网络结构设计

本文提出网络的总体架构如图 1 所示,由 3 个主要部分组成:编码器、双向 ConvLSTM 和解码器。编码器和解码器是 2 个对称的卷积网络,训练时选取 5 幅连续图片和最后一帧的地面真实情况作为输入训练所提出的网络,并在最后一帧识别车道;编码器生成多尺度特征映射;输入双向 ConvLSTM 捕获连续帧的时空信息;多尺度特征映射的时空信息被输入解码器进行特征融合。

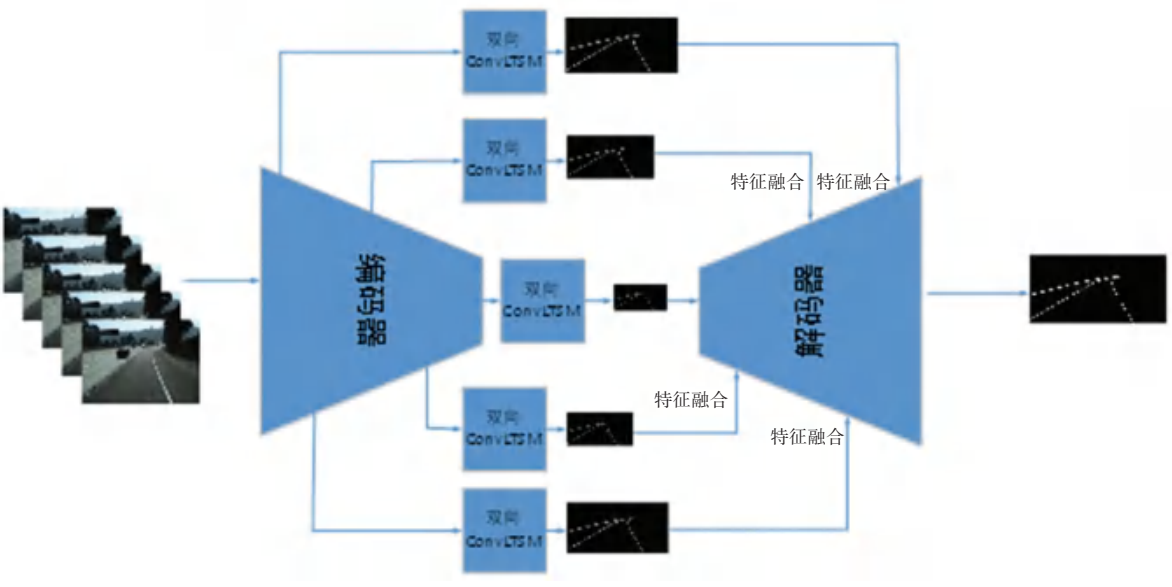


图 1 总体网络结构图

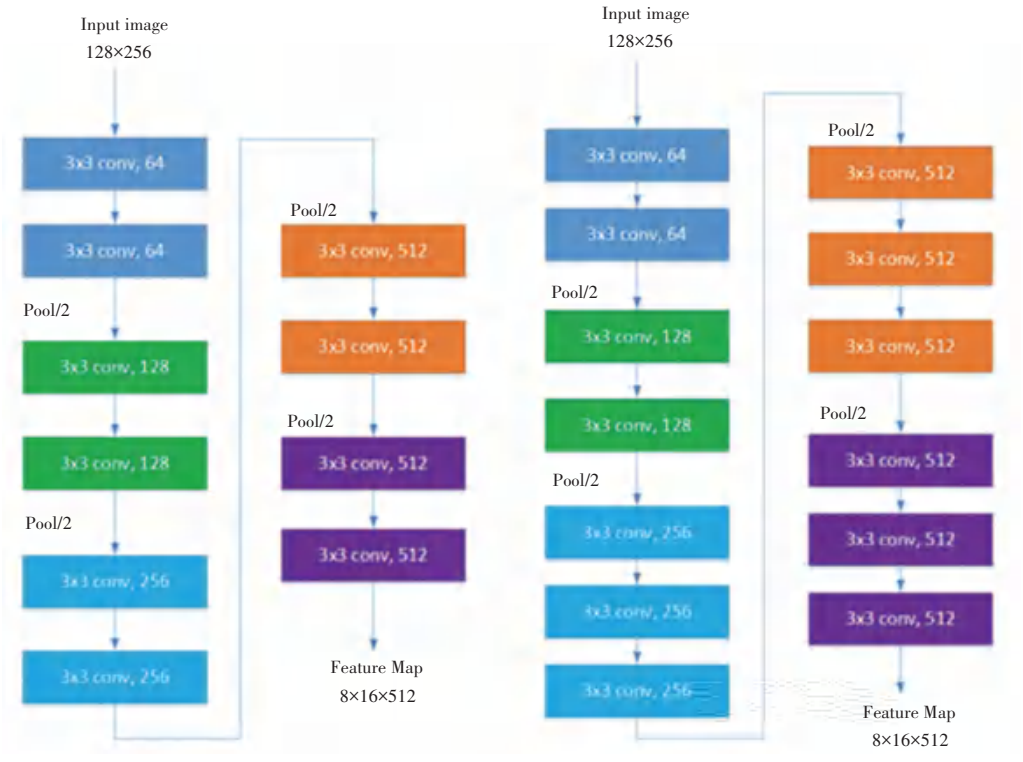
Fig. 1 Overall network structure diagram

1.1 多尺度编码器网络

在编码器部分进行特征提取,并生成多尺度特征映射。受 SegNet 和 U-Net 在语义分割方面的成功启发,参考 SegNet 和 U-Net 设计了编码器,并通过改变卷积核数和 Conv 层对其进行了优化如图 2 所示,可以在准确率和效率之间取得平衡。在 U-Net 中,编码网络的一个块包括 2 个卷积层,卷积核数是最后一个块的 2 倍,池化层用于特征映射的下采样。经过此操作后,特征映射的大小将缩小到一半,而通道数量将加倍,表示高级语义特征。在优化的 U-Net 编码器中,最后一个块没有使卷积层的核数加倍,如图 2(a)所示。因为车道通常可以用颜色和边缘等来表示,即使使用较少的通道,原始图像中的信息也能得到很好的表达。SegNet 采用 VGGNet 的 16 层卷积结构作为编码器,如图 2(b)所示。

编码器生成多尺度特征映射,然后将不同大小的特征块用作 ConvLSTM 模块的输入。假设(S_t, S_{t-1}, S_{t-2}, S_{t-3}, S_{t-4})是在时间(t, $t-1$, $t-2$, $t-3$, $t-4$)的输入,这里使用(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k, f_{t-4}^k)(其中 $k=1,2,3,4,5$)来表示(S_t, S_{t-1}, S_{t-2}, S_{t-3}, S_{t-4})的第 k 层的特征映射。换句话说,当使用 S_t 作为输入时,f_t^1 将是编码器网络的第一个块的特征映射,如式(1)。

$$(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k, f_{t-4}^k) = Encoder^k(S_t, S_{t-1}, S_{t-2}, S_{t-3}, S_{t-4}),$$
$$\text{where } k = 1, \cdots, 5. \tag{1}$$



（a）U-Net 编码器　　（b）SegNet 编码器

（a）U-Net encoder　　（b）SegNet encoder

图 2　编码器架构

Fig. 2　Encoder architecture

1.2　双向 LSTM

为了提取连续帧的时空信息，LSTM 块将编码器提取的特征映射作为输入。采用 LSTM 是因为其遗忘不重要信息和保留重要特征的能力优于传统的 RNN 网络，而全连接 LSTM 耗时且计算量大，本文网络中使用卷积 LSTM（ConvLSTM）。ConvLSTM 用卷积运算代替 LSTM 中每个门的矩阵乘法运算，广泛应用于时间序列数据的端到端训练和特征提取[4]。

下面将详细描述这些组件，一般的 ConvLSTM 在 t 时刻的激活可以表述为式(2)~式(6)：

$$C_t = f_t \circ C_{t-1} + i_t \circ \tan h(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \quad (2)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * W_{t-1} + W_{cf} * C_{t-1} + b_J f), \quad (3)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * W_{t-1} + W_{co} * C_{t-1} + b_o), \quad (4)$$

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * W_{t-1} + W_{ci} * C_{t-1} + b_i), \quad (5)$$

$$H_t = o_t \circ \tan h\ (C_t). \quad (6)$$

其中，X_t 表示编码器在时间 t 提取的输入特征映射；C_t、H_t 和 C_{t-1}、H_{t-1} 分别表示在时间 t 和 $t-1$ 的存储器和输出激活；C_t、i_t、f_t 和 o_t 分别表示单元、输入门、遗忘门和输出门；$\boldsymbol{W}_{xi}$ 是输入 X_t 对输入门的权重矩阵；b_i 是输入门的偏差；其它 W 和 b 的含义可以从上述规则中推断出来；$\sigma(.)$ 表示 sigmoid 运算；$\tan h(.)$ 表示双曲正切非线性；“$*$”和“$\circ$”分别表示卷积运算和阿达玛积。

网络中第 k 个（$k = 1,2,3,4,5$）ConvLSTM 模块将公式(1)中提到的特征映射（$f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k, f_{t-4}^k$）作为其输入，这个 ConvLSTM 模块产生一个输出特征映射（表示为 g^k），捕获这些个帧的时空信息。操作总结如式(7)：

$$g^k = ConvLSTM^k(f_t^k, f_{t-1}^k, f_{t-2}^k, f_{t-3}^k, f_{t-4}^k), \quad \text{where } k = 1,\cdots,5. \quad (7)$$

得益于语音识别方面的进步，在这里进一步将 ConvLSTM 模块扩展到双向 ConvLSTM，以使用前向和后向 2 种方向对时空信息进行建模。

本文提出的双向 ConvLSTM 模块如图 3 所示。输入特征映射 $f_{t-4}^k, \cdots, f_t^k$ 被馈送到 2 个 ConvLSTM 模块：$ConvLSTM^{forward}$ 和 $ConvLSTM^{backward}$。$ConvLSTM^{forward}$ 计算从时间步 $t-4$ 到 t 的前向隐藏序列 $\overrightarrow{H}_{t+1}$，而 $ConvLSTM^{backward}$ 通过在从时间步 t 到 $t-4$ 的后向方向上迭代输入来计算 $\overleftarrow{H}_{t+1}$。最后将 $ConvLSTM^{forward}$ 和 $ConvLSTM^{backward}$ 的输出连接起来，得到特征映射 g_s^k，并将其转发给解码器进行后续处理。双向 ConvLSTM 中

的操作,如式(8)~(10)所示:

$$\overrightarrow{H}_s, \overrightarrow{C}_s = ConvLSTM^{forward}(f^k_{s-1}, \overrightarrow{H}_{s-1}, \overrightarrow{C}_{s-1}), \quad (8)$$

$$\overleftarrow{H}_s, \overleftarrow{C}_s = ConvLSTM^{backward}(f^k_{s+1}, \overleftarrow{H}_{s+1}, \overleftarrow{C}_{s+1}),$$

$$\text{where } s = t-4, t-2, t-1, t, \quad (9)$$

$$g^k_s = concat(\overrightarrow{H}_t, \overleftarrow{H}_{t-4}). \quad (10)$$

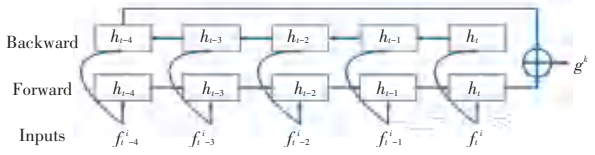


图 3 双向 ConvLSTM 模块体系结构

Fig. 3 Bidirectional ConvLSTM module architecture

1.3 解码器网络

在解码器中,接收的特征映射的大小和数量与编码器特征映射相同,但方向相反,以便更好地恢复。在各个网络模型中,ResNet 等采用的 element-wise add(简称 add)来融合特征[5],而 DenseNet 等则采用 concat 来融合特征[6]。add 是特征映射的相加,concat 是通道的合并,add 的计算量要比 concat 的计算量小得多,因此选用 add 进行特征融合。

解码器每个子块中的上采样和卷积匹配编码器的子块中的相应操作。解码器获取 5 个双向 ConvLSTM 模块的输出 $(g^1, g^2, g^3, g^4, g^5)$,并为时间 $t+1$ 生成未来的语义分段掩码 S_{t+1}(假设提前一步预测)。解码器结构如图 4 所示。

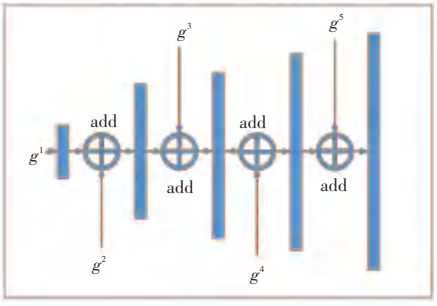


图 4 解码器结构

Fig. 4 Decoder structure

采用 1×1 卷积,在 g^1 上进行上采样,以匹配 g^2 的尺寸;然后通过 add 与 g^2 结合起来。g^2, g^3, g^4 和 g^5 操作类似,最后 1×1 卷积和上采样来获得 S_{t+1},式(11)~式(13)。

$$z^1 = g^1, \quad (11)$$

$$z^k = UP(C_{1\times1}(z^{k-1})) + g^k, where k = 2,3,4,5, \quad (12)$$

$$S_{t+1} = UP(C_{1\times1}(z^5)). \quad (13)$$

$C_{1\times1}(.)$ 和 $UP(.)$ 分别表示 1×1 卷积和上采样操作。

2 实验设置

2.1 数据集设置

Robust Lane Detection 中自行构建了一个连续帧车道线数据集[7],使用 TuSimple 数据集和 Robust Lane Detection 的一个数据集组成的综合数据集。TuSimple 车道数据集包括3 626个图像序列。这些图片是高速公路上的前额驾驶场景,每个序列包含一秒钟内收集的 20 个连续帧,最后一帧即第 20 张图像车道线被标记。这里为了增加数据集,在每个序列中额外标记了第 13 幅图像的车道线。Robust Lane Detection 车道线数据集包括1 148个乡村道路图像序列,使用这 2 个数据集结合,大大扩展了车道线数据集的多样性,详细信息见表 1 和图 5。

表 1 数据集的结构和内容

Tab. 1 The structure and content of the data set

数据集	包含数据集名称	有标记的帧	图片总数
训练集	TuSimple	第 13,20 帧	7 252
	Robust Lane Detection	第 13,20 帧	2 296
测试集	测试集 1	第 13,20 帧	540
	测试集 2	所有帧	728

(a) 训练集中的图像

(a) Images in the train set

(b) 测试集 1 中的图像

(b) Images in the test set 1

(c) 测试集 2 中的图像

(c) Images in the test set 2

图 5 数据集图像

Fig. 5 Pictures of date set

训练时,选取 5 幅连续图片和最后一帧的车道线标注作为输入,训练所提出的网络,并在最后一帧检测车道线。为使所提出的网络能够在不同的行驶速度下进行车道检测,对输入图像进行了 3 种不同的采样,即 1 帧、2 帧和 3 帧的步长,见表 2。

在数据扩充中,采用了旋转、翻转和裁剪等操作,生成了38 192个用于训练的标记图像组。输入将随机改变为不同的光照情况,这有助于训练的模型更加健壮。

为了测试,抽取 5 幅连续图像来识别最后一帧

的车道，将其与最后一帧的地面真实值进行比较。构建了 2 个测试集，测试集 1 是在正常的 TuSimple 测试集上构建的，测试集 2 由在不同情况下路面图片组成，针对模型稳健性进行评估。

表 2　连续输入图像的采样方法

Tab. 2　Sampling method of continuous input image

步长	输入帧	有标记的帧
1	第 9、10、11、12、13 帧	第 13 帧
2	第 5、7、9、11、13 帧	第 13 帧
3	第 1、4、7、10、13 帧	第 13 帧
1	第 16、17、18、19、20 帧	第 20 帧
2	第 12、14、16、18、20 帧	第 20 帧
3	第 8、11、14、17、20 帧	第 20 帧

Robust Lane Detection 数据集中用细线来标注车道，然而在语义分割任务中，网络必须学习像素级标签。所以对图像进行低分辨率采样，因为当图像变小时，车道线宽度接近一个像素，如图 6 所示。

（a）输入图像　　（b）车道线真值

（a）Input image　　（b）Truth value of lane line

图 6　低分辨率图像

Fig. 6　Low resolution image

2.2　超参数设置和损失函数

（1）ImageNet 是一个用于分类的大型基准数据集[8]，提出的网络在 ImageNet 上进行预先训练。利用预先训练好的权值进行初始化，不仅可以节省训练时间，而且可以将适当权值传递给所提出网络[9]；

（2）以 N 幅连续的驾驶场景图像为输入，进行车道线识别。因此在反向传播中，ConvLSTM 的每个权重更新系数应该除以 N。在实验中，设置 $N=5$，还研究了 N 对影响车道线检测性能的影响；

（3）基于加权交叉熵构造了一个损失函数来求解区分性分割任务，式（14）：

$$\varepsilon_{loss}=\sum_{x\in\Omega}w(x)\log\left(p_{\iota(x)}(x)\right). \tag{14}$$

其中，$\iota:\Omega\rightarrow\{1,\cdots,K\}$ 是每个像素的真实标签，$w:\Omega\rightarrow\mathbb{R}$ 是每个类的权重，目的是平衡车道线类。其被设置为整个训练集中 2 个类的像素数的比率。$p_{\iota(x)}$ 定义为式（15）：

$$p_{\iota(x)}=\exp(a_k(x))/\sum_{k}^{K}\exp\left(a_k^{'}(x)\right). \tag{15}$$

其中，$a_k(x)$ 表示特征通道 k 在像素位置 $x\in\Omega,\Omega\in\mathbb{Z}^2$ 处的激活，k 是类的数目；

（4）为了有效地训练所提出的网路，在不同的训练阶段使用不同的优化器。一开始使用自适应矩估计（Adam）优化器，其具有更高的梯度下降率，但很容易陷入局部极小。为了避免这种情况，当网络被训练到一个相对较高的精度时，转而使用随机梯度下降优化器（SGD），其在寻找全局最优解方面具有更高的性能。

在更换优化器时，需要进行学习速率匹配，否则学习过程会受到干扰，导致收敛的混乱或停滞。学习率匹配公式（16）~（18）：

$$w_{k_{sgd}}=w_{k_{Adam}}, \tag{16}$$

$$w_{k-1_{sgd}}=w_{k-1_{Adam}}, \tag{17}$$

$$w_{k_{sgd}}=w_{k-1_{sgd}}-\alpha_{k-1_{sgd}}\hat{\nabla}f(w_{k-1_{sgd}}). \tag{18}$$

其中，w_k 表示第 k 次迭代中的权重；α_k 是学习率；$\hat{\nabla}f(.)$ 是由损失函数 $f(.)$ 计算的随机梯度。实验中，初始学习率设为 0.01，当训练精度达到 90% 时，改变优化器。

3　实验

在实验中，车道检测图像的采样分辨率为 256×128。实验配置为 E5-2630@ 2.3GHz，64GB 内存和 2 个 GeForce GTX TITAN-X GPU。批量处理大小为 16，epoches 为 100。

3.1　与原始网络以及修改版本比较

将本文提出的网络命名为 UNet_2ConvLSTM 和 SegNet_2ConvLSTM，与其原始基线以及一些修改版本进行了比较。包括以下方法：

（1）SegNet：一种经典的用于语义分割的编解码结构神经网络，编码器与 VGGNet 相同；

（2）SegNet_Cat：在 SegNet 基础上，在编码器和解码器之间添加多尺度编码特征融合；

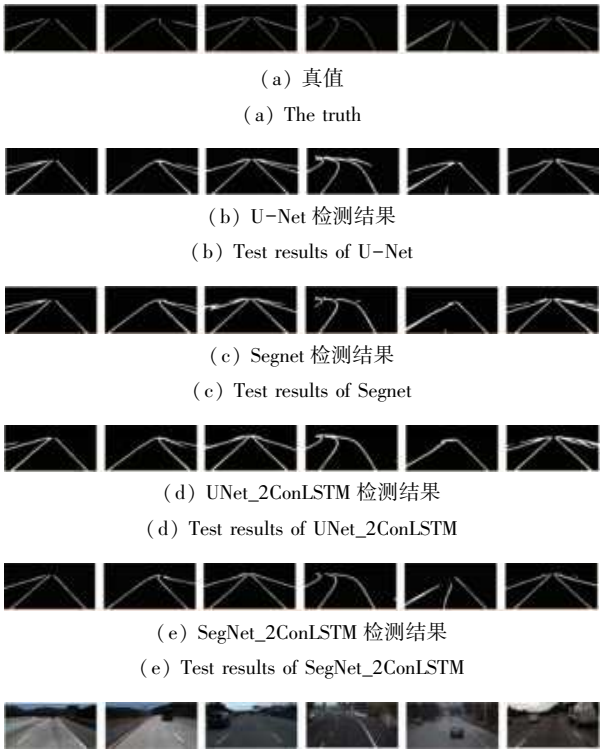
（3）SegNet_ConvLSTM：在 SegNet_Cat 中添加单向 LSTM；

（4）SegNet 3D：通过将连续图像叠放，利用三维卷积核得到混合的空间和序列特征；

（5）UNet 相关网络：将 SegNet 的编码器和解码器替换为修改后的 UNet，生成相应网络。

在对上述网络进行训练后，对测试集的结果进行了比较。首先从视觉上检查不同方法得到的结果；然后对其进行定量比较；并证明所提出框架的先进性。

如图 7 所示,所提网络识别出了输入图像中的每一条车道线,当车道线被遮挡或形状不规则时,也能够完整地识别出,避免将一条连续的车道线检测成多条断裂的车道,而其它方法很容易将其它边界识别为车道线或者识别不连续。



(a) 真值

(a) The truth

(b) U-Net 检测结果

(b) Test results of U-Net

(c) Segnet 检测结果

(c) Test results of Segnet

(d) UNet_2ConLSTM 检测结果

(d) Test results of UNet_2ConLSTM

(e) SegNet_2ConLSTM 检测结果

(e) Test results of SegNet_2ConLSTM

(f) 输入图像

(f) Input image

图 7 车道线检测结果

Fig. 7 Lane line detection results

SegNet_Cat 表明了在编码器和解码器之间添加多尺度特征融合的有效性,见表 3。SegNet_ConvLSTM 表明了在 SegNet_Cat 中的编码器网络之后使用传统的单向 ConvLSTM 的有效性。而本文所提网络即使用多尺度编特征融合,又将传统的单向 ConvLSTM 拓展为双向 ConvLSTM,性能进一步提升,同时也胜过 SegNet_3D,相对于 SegNet 的准确率分别提高了约 1.5%,FN 和 FP 也有一定降低。UNet_2ConvLSTM 表现同理。但是由于代表车道线的像素远远小于代表背景的像素。因此,准确度只能看作是一个参考指标。

在车道线检测任务中,将车道线设置为正类,背景设置为负类。根据公式(19)(20),其中 *TruePositive* 表示正确预测为车道线的像素数,*FalsePositive* 表示错误预测为车道线的像素数,*FalseNegative* 表示错误预测为背景的像素数,UNet_2ConvLSTM 的 *Precision* 比 UNet 提高了 8%,*Recall* 仅下降了 1.5%。对于 SegNet,加入双向 ConvLSTM 后,*Precision* 提高了 6%,*Recall* 也略有提高,见表 4。

表 3 各网络在测试集 1 上准确度对比

Tab. 3 Comparison of each network on test set 1

Model	*Acc*/%	*FP*	*FN*
SegNet	96.45	0.035 7	0.040 1
UNet	96.39	0.034 4	0.032 4
SegNet_Cat	96.64	0.036 8	0.038 7
UNet_Cat	96.74	0.038 9	0.033 8
SegNet_ConvLSTM	96.76	0.032 1	0.033 5
UNet_ConvLSTM	96.89	0.035 8	0.031 8
SegNet_3D	96.45	0.034 1	0.039 4
UNet_3D	96.37	0.036 9	0.030 1
SegNet_2ConvLSTM	97.88	0.021 0	0.023 6
UNet_2ConvLSTM	98.03	0.020 3	0.021 7

表 4 各网络在测试集 1 上的测试结果

Tab. 4 Test results of each network on test set 1

Model	*Precision*	*Recall*	*F*1
SegNet	0.769	0.959	0.852
UNet	0.771	0.976	0.861
SegNet_Cat	0.784	0.960	0.863
UNet_Cat	0.797	0.968	0.874
SegNet_ConvLSTM	0.810	0.961	0.879
UNet_ConvLSTM	0.815	0.967	0.884
SegNet_3D	0.795	0.961	0.870
UNet_3D	0.785	0.984	0.873
SegNet_2ConvLSTM	0.828	0.963	0.890
UNet_2ConvLSTM	0.858	0.961	0.900

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}, \quad (19)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}, \quad (20)$$

考虑到 *Precision* 或 *Recall* 只反映车道检测性能的一个方面,又引入 *F*1 测度作为一个整体进行评价。*F*1 定义为式(21)。所提方法的 *F*1 测量值比原始版本提高了约 4%,见表 4。这些显著的改进表明,多帧比单帧预测车道线更加准确和双向 ConvLSTM 在语义分割框架中对序列数据的有效性。

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (21)$$

从表 4 可以更明显的看出,SegNet 基础上添加多尺度特征融合的有效性。由于 ConvLSTM 能够接受高维张量作为输入,因此其可以在原始基线的基

础上提高精度，而双向 ConvLSTM 性能提升更加明显，相对于 SegNet 的 $F1$ 提高了约 3.8%。UNet_2ConvLSTM 表现同理，相比于 UNet 网络的 $F1$ 提高了约 4.1%。

三维卷积核在立体视觉问题中非常普遍，其也可以通过将图像堆积到三维体积来处理连续图像。也测试了 SegNet_3D 和 UNet_3D 的性能，见表 4。然而由于三维卷积核对时间序列特征的描述能力不强，导致没有很好的性能表现。

本文提出网络以一系列图像为输入，可能会增加运行时间。如果处理全部 5 帧，则所提出的网络比仅处理一个图像的网络要耗费更多时间。因为之前帧的特征已经被抽象出来，编码器可以重用前 4 帧的特征，只需要处理当前帧，而且性能与全部处理 5 帧相差无几，由于 ConvLSTM 块可以在 gpu 并行执行，对于以单个图像作为输入的模型，其运行时间与原网络几乎相同，具体表现见表 5。

表 5　在测试集 1 上测试结果和运行时间

Tab. 5　Test results and running time on test set 1

Model(frames)	*Pre*	*Rec*	*F*1	*Time* /ms
SegNet(1)	0.789	0.959	0.866	5.2
UNet(1)	0.791	0.976	0.874	4.8
SegNet_2ConvLSTM(5)	0.829	0.965	0.891	25.3
UNet_2ConvLSTM(5)	0.857	0.963	0.902	24.3
SegNet_2ConvLSTM(1)	0.828	0.963	0.890	6.8
UNet_2ConvLSTM(1)	0.858	0.961	0.900	5.9

SegNet_2ConvLSTM 如果将所有 5 个帧作为新的输入进行处理，则运行时间大约为 25 ms。如果存储并重用前 4 帧的特征，则运行时间为 6.8 ms，比原 SegNet 的 5.2 ms 稍长。同样，U-Net_2ConvLSTM 运行的平均时间约为 5.9 ms，比 U-Net 的 4.8 ms 稍长。

3.2　与 TuSimple 车道线检测竞赛中方法比较

为了进一步验证所提出方法的性能，将所提出的方法与 TuSimple 车道线检测竞赛中的方法进行了比较。这里训练集是基于原始图像大小的 TuSimple 数据集。PointLaneNet 可以在单个网络中同时执行位置预测和车道线分类；ENet-SAD 与现有方法相比精度属于前列，且参数量更少[10]；SCNN 由于网络结构较深，并且采用行列卷积的形式，也取得了不错的精度；LaneNet 由于设计了一个带分支结构的多任务网络，取得了较好的结果；ERFNet 核心元素是一个新层，利用跳跃连接和 1D 卷积核，也在检测精度上得到了提升[11]。从表 6 可看出，所提网络的 FN 和 FP 都接近最佳结果，在所有方法中具有最高的精确度。

表 6　与 TuSimple 车道线检测竞赛先进算法比较

Tab. 6　Comparison of advanced algorithms with TuSimple lane line detection competition

Model	*Acc* /%	*FP*	*FN*
PointLaneNet	96.34	0.0467	0.0518
ENet-SAD	96.36	0.0345	0.0321
SCNN	96.50	0.0289	0.0287
LaneNet(+H-net)	96.24	0.0363	0.0326
ERFNet	96.27	0.0378	0.0309
SegNet_2ConvLSTM	97.08	0.0220	0.0246
UNet_2ConvLSTM	97.17	0.0213	0.0224

3.3　鲁棒性测试

尽管在之前的测试集上取得了很高的性能，但是仍需要测试所提网络的鲁棒性。因为即使是细微的错误识别也会增加交通事故的风险[12]。一个好的车道检测模型应该能够处理各种不同的驾驶场景，如城市道路和高速公路等日常驾驶场景，以及具有挑战性的驾驶场景，如乡村道路、照明不良、人车遮挡等。

在鲁棒性测试中，使用具有不同的驾驶场景的测试集 2 进行测试。测试集 2 包含 728 个图像，包括农村、城市和高速公路场景中的车道线图片，是一个综合性和挑战性的测试集，其中一些车道线人眼也很难检测到。

UNet_2ConvLSTM 在所有场景的 Precision 上都优于其它方法，并且有很大的提高，见表 7。表 8 所示在大多数场景中也都达到了最高的 $F1$ 值，这说明了所提网络具有很好的鲁棒性。大多数实验 UNet_2ConvLSTM 的性能优于 SegNet_2ConvLSTM。

3.4　超参数分析

网络中主要有 2 个参数可能会影响本文所提出网络的性能，一个是输入的连续帧总帧数，另一个是采样步长。这 2 个参数共同决定第一帧和最后一帧之间的范围。

当更多的帧作为网络输入时，网络可以生成包含更多附加信息的特征映射，这可能有助于最终的预测结果。但是如果使用太多的前面的帧，结果也可能不好，因为距离当前帧较远的前面帧中的车道线情况可能与当前帧显著不同。在这里首先分析了输入帧数对检测结果的影响，将输入帧数设置为 1~5，同时也分析了在输入帧数相同情况下，采样步长对检测结果的影响。

表 7 在 12 种挑战性场景中的表现(Precision)

Tab. 7 Performance in 12 challenging scenarios (Precision)

Method	curve	shadow	bright	occlude	dirty	blur	tunnel
SegNet	0.733 2	0.728 2	0.600 3	0.594 3	0.214 4	0.504 7	0.429 5
UNet	0.714 6	0.767 8	0.669 5	0.609 8	0.339 9	0.426 0	0.613 6
SegNet_Cat	0.743 0	0.708 9	0.592 4	0.603 5	0.216 7	0.489 4	0.438 9
UNet_Cat	0.719 6	0.777 2	0.681 0	0.640 7	0.396 5	0.638 9	0.624 1
SegNet_ConvLSTM	0.749 0	0.733 8	0.594 2	0.640 1	0.276 9	0.572 0	0.552 1
UNet_ConvLSTM	0.727 8	0.816 9	0.738 1	0.676 3	0.456 1	0.693 3	0.699 4
SegNet_3D	0.630 4	0.494 5	0.384 2	0.369 7	0.135 7	0.440 9	0.367 4
UNet_3D	0.592 8	0.531 7	0.471 9	0.413 6	0.298 1	0.530 1	0.400 9
SegNet_2ConvLSTM	0.716 8	0.730 9	0.583 2	0.674 1	0.268 3	0.567 8	0.713 4
UNet_2ConvLSTM	0.759 2	0.859 6	0.774 3	0.745 8	0.526 7	0.772 4	0.796 9

表 8 在 12 种挑战性场景中的表现(*F*1 度量)

Tab. 8 Performance in 12 challenging scenarios (*F*1 measurement)

Method	curve	shadow	bright	occlude	dirty	blur	tunnel
SegNet	0.713 8	0.775 0	0.620 8	0.602 8	0.209 1	0.560 2	0.404 8
UNet	0.704 8	0.757 5	0.693 5	0.503 4	0.311 0	0.419 8	0.582 9
SegNet_Cat	0.722 1	0.736 7	0.689 2	0.632 3	0.142 7	0.536 1	0.642 3
UNet_Cat	0.711 1	0.746 8	0.723 4	0.547 2	0.297 5	0.653 4	0.499 2
SegNet_ConvLSTM	0.755 4	0.739 5	0.688 6	0.658 8	0.269 5	0.625 5	0.674 0
UNet_ConvLSTM	0.756 5	0.738 8	0.770 5	0.582 0	0.340 7	0.682 6	0.568 7
SegNet_3D	0.669 2	0.563 8	0.501 4	0.400 4	0.162 3	0.537 0	0.437 5
UNet_3D	0.674 3	0.556 6	0.602 1	0.385 1	0.298 0	0.601 8	0.316 7
SegNet_2ConvLSTM	0.831 1	0.793 1	0.705 0	0.734 9	0.306 1	0.621 7	0.767 1
UNet_2ConvLSTM	0.807 2	0.742 8	0.834 1	0.580 7	0.314 3	0.726 4	0.607 6

这里以 UNet_2ConvLSTM 为例,在测试集 1 上进行了测试。从表 9 可以看出,在相同的采样步长下,当使用更多的连续图像作为输入时,精确度和 F1 测量值都会增加,这说明了使用所提出的网络结构使用连续帧作为输入的有用性,采用多帧的方法比仅使用一幅图像作为输入的方法有显著的改进。随着步幅的增加,性能的增长趋于稳定。例如,从 4 帧到 5 帧的性能改善要小于从 2 帧到 3 帧的性能改善,因为从较远的前一帧得到的信息对车道预测和检测的帮助要小。分析了另一个参数:2 幅连续输入图像之间的采样步长的影响。从表 9 可以看出,当帧数固定时,所提出的模型在不同的采样步长下获得了非常接近的性能,这表明抽样步幅的影响较小。

表 9 UNet_2ConvLSTM 在不同参数设置下在测试集 1 的表现

Tab. 9 UNet_2ConvLSTM performance in test set 1 under different parameter settings

步长,总帧数	3,5	3,4	3,3	3,2	2,5	2,4	2,3	2,2	1,5	1,4	1,3	1,2	0,1
跨度数	12	9	6	3	8	6	4	2	4	3	2	1	0
Acc/ %	98.0	97.9	97.6	97.5	97.9	97.8	97.6	97.5	98.0	97.8	97.7	97.5	97.1
Precision/ %	85.6	85.0	83.4	81.7	85.6	84.9	83.4	81.6	85.6	84.9	83.4	81.6	80.1
Recall/ %	95.8	95.8	96.0	96.1	95.8	95.8	95.9	96.2	95.8	95.8	96.0	96.2	98.5
F1/ %	90.5	89.8	89.4	88.3	90.5	89.8	89.4	88.4	90.5	89.8	89.5	88.3	88.3

上述结果有助于理解双向 ConvLSTM 的有效性。单帧输入时,编码器提取的特征地图不能包含车道的全部信息,在某种程度上,解码器必须想象来预测结果;当使用多帧作为输入时,双向 ConvLSTM

将从连续图像中提取的特征映射集成在一起，得到更全面、更丰富的车道线信息，有助于解码器做出更准确的预测。

4　结束语

本文提出了一种连续帧车道线检测网络。首先，在该网络中用编码器提取输入帧的多尺度特征信息；其次，用双向 ConvLSTM 对多尺度的序列特征进行处理；最后，将双向 ConvLSTM 的输出输入解码器进行特征融合和车道线预测。与其它网络相比，提出的网络具有较高的精确率、召回率和 $F1$ 值。此外，该网络在一个具有挑战性驾驶场景的数据集上进行了测试，检验其鲁棒性。结果表明，本文所提出的网络能够在各种情况下稳定地检测出车道，并能很好地避免错误。在参数分析中，采用较长的输入序列来提高检测性能，进一步证明了多帧比单一图像更有利于车道检测的策略。

参考文献

[1] PAN X, SHI J, LUO P, et al. Spatial as deep: Spatial CNN for traffic scene understanding[C]//IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017:1-8.

[2] NEVEN D, BRABANDERE B D, GEORGOULIS S, et al. Towards End-to-End Lane Detection: an Instance Segmentation Approach[J]. IEEE, 2018: 286-291.

[3] CHEN Z, LIU Q, LIAN C. PointLaneNet: Efficient end-to-end CNNs for Accurate Real-Time Lane Detection[C]// IEEE Intelligent Vehicles Symposium (IV), 2019: 2563-2568.

[4] SHI X, CHEN Z, WANG H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting [J]. Advances in neural information processing systems, 2015: 802-810.

[5] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[6] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

[7] ZOU, QIN, et al. Robust lane detection from continuous driving scenes using deep neural networks[J]. IEEE transactions on vehicular technology, 2019, 69.1 (2019): 41-54.

[8] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[J]. IEEE Conference on Computer Vision and Pattern Recognition, 2009:248-255.

[9] HE K, GIRSHICK R B, DOLLAR P. Rethinking imagenet pre-training[J]. CoRR, 2018.

[10] HOU Y, MA Z, LIU C, et al. Learning Lightweight Lane Detection CNNs by Self Attention Distillation[C]//In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019:1013-1021.

[11] ROMERA E, ALVAREZ J M, BERGASA L M, et al. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation[J]. IEEE Transactions on Intelligent Transportation Systems, 2017 (1): 1-10.

[12] LI L, OTA K, DONG M. Humanlike driving: Empirical decisionmaking system for autonomous vehicles[J]. IEEE Transactions on Vehicular Technology, 2018, 67(8):6814-6823.

(上接第 4 页)

4　结束语

本文提出了一种相似案例推荐算法，该算法利用法律要素作为标签训练神经网络，并利用网络结构中学习到法律要素信息的输出层的前一层的输出向量作为一个案例的向量表示，本文利用该向量计算任意一对案件的相似度，找出相似度最高的案例集合作为给定案例推荐的相似案例。本文的算法取得了良好的效果，但法律要素预测模型的 $F1$ 分数不高，今后的研究方向是设计网络来提高法律要素预测的准确性，进一步提高相似案例推荐的精度。

参考文献

[1] OPIJNEN M V. Citation Analysis and Beyond: in Search of Indicators Measuring Case Law Importance[C]// International Conference on Legal Knowledge and Information Systems, 2012: 95-104.

[2] WAGH R S, ANAND D. 2017. Application of citation network analysis for improved similarity index estimation of legal case documents: a study[C]// IEEE international conference on current trends in advanced computing (ICCTAC), 2017: 1-5.

[3] MINOCHA A, SINGH N, SRIVASTAVA A, et al. Finding Relevant Indian Judgments using Dispersion of Citation Network [C]// The Web Conference, 2015: 1085-1088.

[4] MANDAL A, MANDAL A, CHAKI R, et al. Measuring similarity among legal court case documents[C]//Proceedings of the 10th annual ACM India compute conference, 2017: 1-9.

[5] MANDAL A, GHOSH K, BHATTACHARYA A, et al. Overview of the FIRE 2017 IRLeD track: information retrieval from legal documents. FIRE (Working Notes), 2017: 63-68.

[6] ASHLEY K D, WALKER V R. From Information Retrieval (IR) to Argument Retrieval (AR) for Legal Cases: Report on a Baseline Study[C]//JURIX, 2013: 29-38.

[7] CARNEIRO D, NOVAIS P, ANDRADE F, et al. Retrieving information in online dispute resolution platforms: a hybrid method[C]//Proceedings of the 13th International Conference on Artificial Intelligence and Law. ACM, 2011: 224-228.

[8] XIA C, HE T, LI W, et al. Similarity Analysis of Law Documents Based on Word2vec[C]//IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2019: 354-357.

[9] VO N P, PRIVAULT C, GUILLOT F, et al. Experimenting word embeddings in assisting legal review[C]// International Conference on Artificial Intelligence and Law, 2017: 189-198.