计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(1):10−19 [doi: 10.15888/j.cnki.csa.006154]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于深度编解码网络的运动目标检测算法①

侯　畅, 董兰芳

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)
通讯作者: 董兰芳, E-mail: lfdong@ustc.edu.cn

摘　要: 运动目标检测算法在视频监控等领域应用广泛, 但是现实场景中由于噪音、光照变化等因素导致背景复杂多变, 传统的运动目标检测算法往往效果不佳. 为了提升算法效果, 提出了一种新的基于深度编解码网络的运动目标检测算法, 将问题转化为像素级的语义分割问题. 事先使用大量数据离线训练出一个编解码网络, 来学习背景与视频帧之间的差异性, 实际应用中首先使用高斯混合模型进行背景建模, 之后将所得背景与视频帧作为网络输入即可直接获取检测结果. 该方法利用了深度卷积网络在抗噪及特征学习等方面的优点, 无需进行复杂的参数调优即可实现高性能的运动目标检测. 我们在 CDnet2014 数据集上进行了实验评估, 实验结果显示我们所提出的算法较原 GMM 算法有很大提升, 甚至在一些场景中的表现优于现有的一些顶尖算法. 另外得益于非常简单的背景建模方法以及网络结构, 我们的算法在使用 GPU 的情况下能够近乎实时地进行运动目标检测, 实用性很强.

关键词: 运动目标检测; 深度学习; 卷积神经网络; 高斯混合模型

引用格式: 侯畅,董兰芳.基于深度编解码网络的运动目标检测算法.计算机系统应用,2018,27(1):10–19. http://www.c-s-a.org.cn/1003-3254/6154.html

Moving Object Detection Algorithm Based on Deep Encoder-Decoder Neural Network

HOU Chang, DONG Lan-Fang

(College of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Moving object detection algorithms are widely used in video surveillance and other fields. But due to noise, illumination changes and other interference, traditional algorithms are often ineffective. To get a better performance, we transform the problem into a pixel-wise segmentation problem, and propose a novel algorithm based on deep encoder-decoder neural networks. We train an encoder-decoder network offline to learn the differences between the background and the video frame. We firstly use the Gaussian Mixture Model (GMM) to generate a background, and then feed video frames and the background into the encoder-decoder network to get detection results. This method utilizes the advantages of deep convolution network in anti-noise and feature learning, and performs well without complicated parameter tuning. We experiment on the CDnet2014 dataset, and results show that the algorithm we propose performs much better than the original GMM algorithm, and even outperforms some top algorithms in some scenes. Due to the simple network architecture, our algorithm is capable of almost real-time processing using a GPU, which shows its great practicality.

Key words: moving object detection; deep learning; convolutional neural network; Gaussian Mixture Model

1　引言

运动目标检测算法即根据历史视频帧将视频中的每个像素点分类为背景或前景, 以获取运动目标, 被广泛应用于视频监控等领域[1].

① 收稿时间: 2017-04-07; 修改时间: 2017-04-26; 采用时间: 2017-05-08; csa 在线出版时间: 2017-11-14

一种最简单的运动目标检测算法就是基于图像灰度值, 使用视频图像减去事先给定的背景图像, 与阈值进行比较来获得检测结果. 然而由于自然场景的动态性 (如图像噪声、光照变化、动态背景以及前景目标间歇性运动等), 这种方法所获取的结果通常很不理想, 如何实现一个适用于不同场景的运动目标检测算法一直是该领域所面临的主要挑战.

很多学者就该问题进行了大量研究, 运动目标检测算法可简单分为基于采样的、基于概率统计的、基于编码本的以及基于深度学习的. 早期比较偏向于使用基于统计或采样的方法来解决该问题, 如 Stauffer 与 Grimson 提出使用高斯混合模型 (Gaussian Mixture Model, GMM) 来进行运动目标检测[2], 该模型假定每个背景像素点颜色值都是服从概率分布的, 而其概率分布函数 (Probability Distribution Function, PDF) 可看作一个高斯混合模型, 且邻近像素点间是相互独立的, 这样输入视频帧中的颜色值与高斯分布均值的差值在一定范围内的像素点即为背景, 反之为前景像素点, 同时使用一种期望最大化 (Expectation Maximization, EM) 算法来学习高斯混合模型中的参数[3]; 类似地, Elgammal 等提出一个非参数化概率方法来进行背景建模, 同样假定背景像素颜色值服从某种概率分布函数, 但是对于每个像素点的评估使用核密度估计 (Kernel Density Estimation, KDE) 算法[4]; Barnich 等与 Kim 等分别提出基于采样的以及基于编码本的背景建模法[5,6]. 后来 Varadarajan 等[7]提出了一种基于区域的高斯混合模型, 从方形子图像块中提取特征来进行建模; St-Charles 等[8]引入局部二值相似度特征 (Local Binary Similarity Patterns, LBSP) 来作为额外特征来改善背景模型, 并针对阈值的确定提出了一些启发式的改进, 虽然这些方法在一定程度上改善了检测结果, 但是时间复杂度增加, 很难做到实时.

近几年鉴于卷积神经网络 (Convolutional Neural Network, CNN) 在特征学习上的成功, 很多人尝试使用基于深度学习的方法来解决运动目标检测问题. Babaee 等结合 St-Charles 等的成果, 训练一个通用的 CNN 模型, 用来对比背景图像与视频帧, 效果很好, 但是其背景建模方法是结合了几种现有算法, 时间复杂度很高, 在比较好的计算平台上 (英特尔 E5-1620 v3 处理器、英伟达 GeForce Titan X 显卡) 也只能做到 10 帧每秒 (Frame Per Second, FPS)[9].

总结起来, 传统基于概率统计、采样等技术的运动目标检测算法没能很好利用图像特征来改进背景去除结果, 另一方面近年来基于深度学习的方法并没有充分挖掘 CNN 的特征学习能力, 且大部分算法时间复杂度高, 不适用于实时任务.

在图像处理领域很多解决问题的方法或范式都可以进行一定程度的推广, 比如 2012 年在图像分类领域基于深度卷积网络的 Alexnet 后来被广泛应用到其他图像处理任务中[10]; 源自人脸识别领域图像比对的思想近年来也被应用到目标跟踪等任务中[11]. 启发自图像比对及图像语义分割的思想, 本文提出了一种新的基于深度编解码网络的运动目标检测算法, 我们一方面使用计算复杂度较低的高斯混合模型作为背景建模方法, 另一方面充分利用 CNN 的特征学习能力, 采用事先训练好的一个基于反卷积的编解码网络来识别视频帧与背景图像间的差异. 实际应用中首先用高斯混合模型进行背景建模, 之后将所得背景与视频帧作为网络输入即可直接获取检测结果. 该方法利用了深度卷积网络在抗噪及特征学习等方面的优点, 无需进行复杂的参数调优即可实现高性能的运动目标检测. 我们在 CDnet2014 数据集上进行了实验评估, 其结果显示我们所提出的算法在很多指标上优于现有的大部分算法. 另外得益于较为简单的网络结构, 我们的算法在使用 GPU 的情况下能够近乎实时地进行运动目标检测, 实用性很强.

2 基于编解码网络的运动目标检测算法

一个典型的运动目标检测系统如图 1 所示.

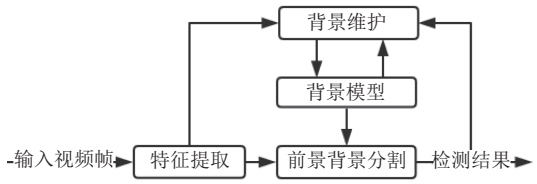

图 1 运动目标检测系统模块图

图 1 中背景模型就是当前场景中背景的一种描述, 在运动目标检测算法中通常作为输入视频帧的参照物. 一般使用最开始的部分视频帧来初始化背景模型, 初始化完成后对每个输入的视频帧首先提取其特征, 之后跟背景模型进行比较来获取检测结果. 输入的视频帧与检测结果有时也用来维护更新背景模型.

本文使用高斯混合模型来进行背景建模, 将所得背景图像与视频帧作为深度卷积神经网络的输入即可直接获取运动目标检测结果. 下面来分别介绍我们所用的背景建模方法以及编解码网络模型.

2.1 背景建模

目前常用的背景建模方法[12]主要有中值法、均值法、卡尔曼滤波器模型[13]、码本法[6]、单高斯模型以及混合高斯模型[14,15]等. 其中中值法与均值法难以适应现实场景中光照等动态变化, 后几种方法中混合高斯模型鲁棒性相对较好, 且实现简单、计算复杂度不高, 因此本文采用 GMM 来进行背景建模, 下面我们来详细介绍 GMM.

在时刻 t 的 RGB 或其他空间中的一个像素对应的值用 来表示, 基于像素的背景减除法涉及到对一个像素是前景 (FG) 还是背景 (BG) 进行决策, 贝叶斯决策 R 的公式如下:

$$R = \frac{p(BG|\vec{x}^t))}{p(FG|\vec{x}^t))} = \frac{p(\vec{x}^t|BG))p(BG))}{p(\vec{x}^t|FG))p(FG))} \tag{1}$$

通常情况下, 不知道前景对象的信息, 如什么时候出现, 出现的频率等, 因此我们假设$p(FG) = p(BG)$和前景对象的出现符合均匀概率分布即$p(\vec{x}^t|FG) = C_{fg}$. 可以使用式 (2) 来对某个像素是否属于 BG 进行判别, 如果满足公式, 则对应的是 BG:

$$p(\vec{x}^t|BG) > C_{thr} \tag{2}$$

其中, C_{thr} 是一个阈值, 称$p(\vec{x}^t|BG)$为背景模型. 从训练集 χ 中来估计对应的背景模型, 得到的模型用$\hat{p}(\chi, BG)$表示. 由于在实际的应用中, 场景中亮度可能是逐渐的改变 (如户外场景的天气的变化) 或者突变 (户内场景的灯光的切换) 以及场景中新对象的出现或者对象的消失都会对场景背景建模有一定的影响. 为了适应这种变化, 通过增加新的样本和排除旧的样本来更新训练的样本集, 选择一个合理的时间间隔 T, 在时刻 t 有$\chi_T = x^t \cdots x^{(t-T)}$, 当有新样本到来的时候, 都需要更新训练集 χ_T 和重新估计$\hat{p}(\vec{x}|\chi_T, BG + FG)$. 然而来自老的样本中可能会存在一些值是属于前景对象的, 因此我们应该用$p(\vec{x}^t, BG + FG)$来估计, 使用 M 个组件的 GMM, 对应的公式如下:

$$p(\vec{x}^t|\chi_T, BG + FG) = \sum_{m=1}^{M} \hat{\pi}_m N(\vec{x}, \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \tag{3}$$

其中, $\hat{\vec{\mu}}_1, \cdots, \hat{\vec{\mu}}_M$表示高斯组件估计的均值, $\hat{\sigma}_1, \cdots, \hat{\sigma}_M$ 表示高斯组件估计的方差, 协方差矩阵被假设为对角单位阵 I, 权重用$\hat{\pi}_m$表示, 而且是非负的并且权重之和为 1. 在时刻 t 给定一个新的样本, 使用下面的公式来更新高斯模型:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \tag{4}$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m \tag{5}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \hat{\sigma}_m^2) \tag{6}$$

其中, $\vec{\delta}_m = \vec{x}^t - \hat{\vec{\mu}}_m$代替前面提到的使用时间间隔 T, 这里的常量 α 描述一个包含了以指数下降速度来限制旧的数据对高斯模型的影响, 保持使用同样的标记并且近似得到 α=1/T. 将一个新样本产生的与最大$\hat{\pi}_m$的“最近”的高斯组件的$o_m^{(t)}$设为 1, 其他的为 0. 定义一个样本和一个组件是“近”的, 是通过样本与高斯组件的 Mahalanobis 距离来度量的, 例如小于 3 倍的标准差, 对来自第 m 个高斯组件平方距离的计算为: $D_m^2(\vec{x}^t) = \vec{\delta}_m^T\vec{\delta}_m/\hat{\sigma}_m^2$. 如果没有和新样本“近”的高斯组件, 则一个新的高斯组件产生, 而且$\hat{\pi}_{M+1} = \alpha, \hat{\vec{\mu}}_{M+1} = \vec{x}^t$和$\hat{\sigma}_{M+1} = \sigma_0$, 其中 σ_0 是某个合适的初始方差; 如果达到了像素对应的最多高斯组件的数量, 那么就删除具有最小元$\hat{\pi}_m$的高斯组件.

这个算法呈现了一个在线的聚类算法, 而且通常入侵的对象由一些具有小权重$\hat{\pi}_m$的聚类来表示, 因此使用前 B 个最大的聚类来近似背景模型:

$$p(\vec{x}^t|\chi_T, BG) \sum_{m=1}^{B} \hat{\pi}_m N(\vec{x}, \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I) \tag{7}$$

如果每个像素对应的高斯组件按照权值$\hat{\pi}_m$降序排列, 则可以得到:

$$B = argmax_b(\sum_{m=1}^{M} \hat{\pi}_m > (1 - c_f)) \tag{8}$$

其中, c_f 表示一个属于前景对象但是不会干扰背景模型的最大比例值. 例如, 如果有一个新对象进入场景并在场景中保持静止一段时间, 那么该对象就很有可能产生一个额外稳定的聚类, 由于背景被遮挡, 产生的额外的聚类的权重π_{B+1}的值会持续增长, 如果对象保持足够长的静止时间, 那么对应的权重慢慢会超过 c_f, 则其就会被当成是背景. 从式 (4) 能够知道对象只需要大约静止为$\log(1 - c_f)/\log(1 - \alpha)$帧, 就会被认为是背景的一部分, 例如 c_f=0.1 和 α=0.001, 那么就可以知道其需要 105 帧. 为了更好的适应环境的变化, 用式 (8) 来替换式 (4), 可以得到权重的更新公式为:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T \quad (9)$$

其中, $c_T=c/T$, c 对应的是支持一个高斯组件的样本数目, 例如可以选择 $\alpha=1/T$, 那么至少需要 $c=0.01*T$ 样本数来支持一个高斯组件, 那么就可以得到 $c_T=0.01$, GMM 对应的具体流程如图 2 所示.

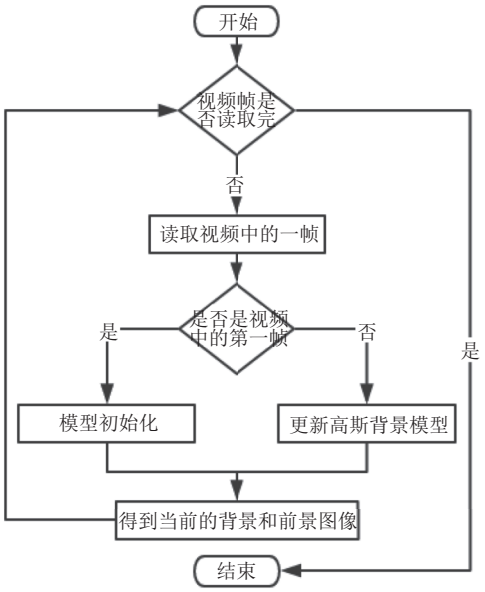

图 2 高斯混合模型流程图

通过以上介绍不难看出, GMM 作为一种像素级的背景建模方法, 并没有利用到区域性特征如边缘信息, 如果前景目标部分与背景颜色相似, 则所得的运动目标检测结果很容易产生“空洞”, 如图 3 所示.

图 3 高斯混合模型检测结果中的“空洞”

而另一方面深度卷积神经网络 (Deep Convolutional Neural Network, DCNN) 有着很强的特征学习能力, 能够学习到除颜色特征之外区域级的特征, 可以有效解决该问题, 所以本文中先使用 GMM 来生成背景图像, 前景背景的分割则采用一个基于反卷积的编解码网络来实现.

2.2 深度编解码网络

运动目标检测是对于每个像素点进行背景或前景的二分类, 从另一个角度看来就是一种像素级的语义分割, 如图 4 所示.

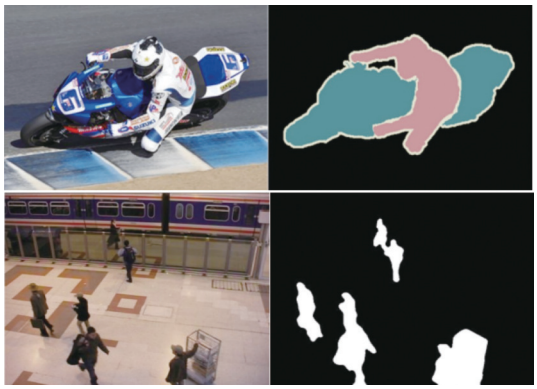

图 4 图像语义分割 (上) 与运动目标检测 (下)

在图像语义分割领域, 目前比较成功的模型都是基于深度神经网络的, 其中 FCN[16]、SegNet[17]、DeconvNet[18]、DeepLab[19]是其中比较成功的模型, 这些网络都是首先使用卷积等操作来获取描述原图像的特征图, 之后从较低分辨率的特征图 (Feature Maps) 解码获取原图大小的像素级分类结果, 整个网络类似一个编解码器, 我们不妨称之为编解码网络.

其中 SegNet 的作者在 CamVid 数据集[20]上使用相同的学习速率分别训练了这几种模型进行测试, 其结果如表 1 所示.

表 1 不同模型在迭代过程中的表现

算法		SegNet	DeepLab-largeFOV	FCN	Deconv-Net
40 k	G	**88.81**	85.95	81.97	85.26
	C	59.93	**60.41**	54.38	46.40
	mIoU	50.02	**50.18**	46.59	39.69
	BF	**35.78**	26.25	22.86	27.36
80 k	G	**89.68**	87.76	82.71	85.19
	C	**69.82**	62.57	56.22	54.08
	mIoU	**57.18**	53.34	47.95	43.74
	BF	**42.08**	32.04	24.76	29.33
>80 k	G	**90.40**	88.20	83.27	89.58
	C	**71.20**	62.53	59.56	70.24
	mIoU	**60.10**	53.88	49.83	59.77
	BF	46.84	32.77	27.99	**52.23**
最大迭代次数		140 k	140 k	200 k	260 k

表 1 中的 G 代表整体准确率 (global accuracy), 指在所有 10 种类别 (动物、行人、卡车等 10 种) 上的分类准确率 (正确分类的像素数除以总像素数); C 代表

类别平均准确率 (class average accuracy), 指在所有类别上的平均准确率; mIoU 代表平均交叠率 (mean intersection over union), 指分割结果与真实数据之间的交叠率:

$$\mathrm{IoU}=\frac{Result\cap GroundTruth}{Result\cup GroundTruth} \tag{10}$$

BF 代表边缘 指标, 指针对边缘像素点的 F_1 指标 (综合考虑准确率与召回率):

$$F_1=\frac{2\times Precison\times Recall}{Precision+Recall} \tag{11}$$

通过结果不难看出 SegNet 能快速收敛到比较好的结果, 且其对于边缘的描绘较其他几种模型好很多, 因此我们选择以 SegNet 为基础设计网络结构.

2.2.1 网络结构

参考 SegNet 我们设计了两种编解码网络, 第一种网络包含 4 个编码层 (不妨称之为 SubNet-4), 其结构如图 5 所示.

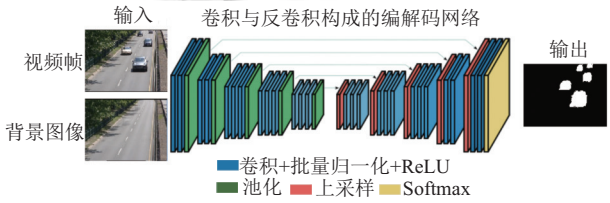

图 5 SubNet-4 网络结构示意图

整个网络包含一个编码网络与对应的解码网络, 最后接一个像素级的分类层来获取分类结果, 网络的输入为视频帧与背景图像, 输出为检测结果.

编码网络中的每个“编码器”首先进行卷积操作, 卷积核大小为 7×7, 边缘填充 3 个像素, 保证卷积后特征图大小与原图相等, 然后批量归一化[21](Batch Normalization), 接着进行像素级的线性整流 (Rectified Linear Unit, ReLU), 再进行窗口大小为 2×2、步长为 2 的最大值池化操作来得到特征图, 这样每经过一层编码特征图大小会缩放到上一层的四分之一.

为了能得到原输入图像大小的特征图, 解码网络中的“解码器”首先使用对应层的“编码器”中最大值池化的池化掩模 (记录了进行池化操作时选择了哪个位置的激活值作为池化结果) 进行一次“上采样”, 如图 6 所示.

这样的上采样操作很明显丢失了特征图中的一些信息, 所以每个上采样层紧接着一个可训练的卷积层来还原原来的特征图. 类似编码器中的设置, 将卷积核大小设定为 7×7, 边缘填充 3 个像素. 这样通过结合上采样与卷积操作实现类似反卷积的效果, 每次解码将特征图缩放到上一层的 2 倍大小, 最终得到原输入大小的特征图用于像素点的分类.

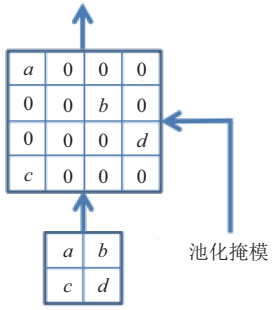

图 6 SegNet 中的上采样

为了对比不同深度的编解码网络在运动目标检测任务上的表现, 我们进一步加深了网络层数, 得到第二种编解码网络, 其编码网络包含 13 个卷积层, 结构类似 VGG16 网络[22](一个经典的用于目标分类的深度卷积网络) 的前 13 个卷积层, 对应的解码网络也有 13 层 (不妨称之为 SubNet-13), 各个编解码器结构与 SubNet-4 相同, 这里不再赘述.

2.2.2 计算复杂度分析

我们所提出的算法主要包含两个模块: GMM 以及编解码网络, 下面我们简单分析下这两个模块的计算复杂度.

GMM 背景建模算法中为了优化对于模型参数的求解实际采用的是 K-means 聚类算法, K-means 算法的计算复杂度一般为 $O(n\times k\times t)$, 其中 n 为待聚类的点的个数, 在 GMM 中即为历史帧的个数, k 为聚类中心个数, 即为 GMM 中高斯组件的个数, t 为直到收敛时的迭代次数. 如果数据本身就有一定的聚类结构, 那么收敛所需的迭代数目通常是很少的, 并且进行少数迭代之后, 再进行迭代的话, 对于结果的改善效果很小. 鉴于上述原因, 该模块对于单个像素点的建模在实践中可以认为几乎是线性复杂度的, 其整体计算复杂度 $O(M\times N\times n)$ 取决于图像的大小.

一般的 DCNN 由卷积层以及全连接层构成, 而由于卷积层采用了局部连接及权值共享等手段, 其计算复杂度较全连接层要低. SubNet 中并没有采用全连接层, 以 SubNet-4 为例, 其整体可以看作一个 8 层的全卷积网络, 其计算复杂度可看作 $O(8\times M\times N\times m\times n)$, 其

中 M、N、m、n 分别代表每层图像以及卷积核的大小.

综上可以得知我们算法中较为耗时的模块是编解码网络, 然而由于 SubNet 无全连接层, 故与使用了普通 DCNN 进行运动目标检测的算法相比, 有一定的速度优势.

2.2.3 准备数据

我们使用 CDnet2014 数据集来进行训练及测试, 该数据集中包含了 10 个类别的场景总共约 140 000 帧的视频数据, 其中有标注的数据大约 50 000 帧, 图像中的每个像素点分别以不同灰度值被标注为五类, 如图 7 所示.

1) 灰度值 0: 静止的像素点.

2) 灰度值 50: 属于阴影的像素点.

3) 灰度值 85: 不在感兴趣区域内的像素点.

4) 灰度值 170: 运动状态未知的像素点 (通常在运动目标边缘, 源自运动模糊等因素).

5) 灰度值 255: 运动的像素点.

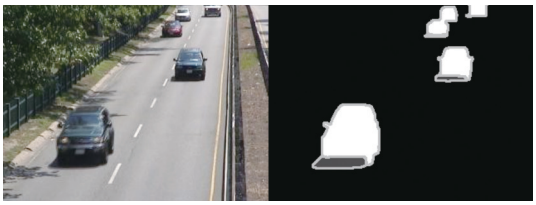

图 7 示例数据

我们的模型有两个输入, 分别是视频帧与背景图像, 视频帧、背景图像与真实数据共同组成一条训练数据. 其中背景图像是我们使用高斯混合模型 (高斯混合模型的模型个数为 5, 历史帧数为 100 帧, 平方 Mahalanobis 距离阈值固定为 16) 从视频中生成的, 具体每个实验所用到的训练以及测试数据我们在实验部分有对应的说明.

此外, 为了适应于网络的输入, 我们使用最近邻插值 (Nearest Neighbor Interpolation) 视频帧、背景图像与标注数据统一缩放到 360×480 的大小.

3 实验

为了验证模型及算法的有效性, 我们先使用基准场景中的部分数据训练我们的 SubNet 模型并测试, 对比了不同深度的网络模型的表现, 并且与原 GMM 算法以及当前比较先进的算法进行对比, 之后我们在一些新的场景中对模型进行了进一步的微调及测试.

3.1 评估指标与评估方法简介

衡量一个运动目标检测算法质量的指标主要包括:

1) 真阳性 (True Positive, TP): 结果中的前景像素点确为前景像素点.

2) 伪阳性 (False Positive, FP): 结果中的前景像素点并非前景像素点.

3) 真阴性 (True Negative, TN): 结果中的背景像素点确为背景像素点.

4) 伪阴性 (False Negative, FN): 结果中的背景像素点并非背景像素点.

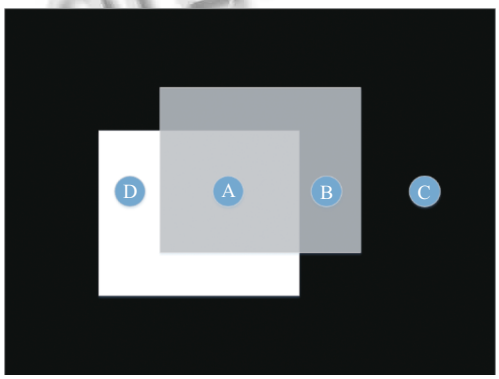

图 8 结果示意图

图 8 是实验结果的示意图, 其中白色矩形框是真实数据, 灰色矩形框是算法结果, 图中 ABCD 四个圆形区域内的点即分别为 TP、FN、TN、FN 的像素点.

继而可进一步得出以下统计指标:

1) 召回率 (Recall, Re):

$$Recall = TP/(TP + FN)$$

2) 特异度 (Specificity, SPC):

$$SPC = TN/(FP + TN) = 1 - FPR$$

3) 准确度 (Accuracy, ACC):

$$ACC = (TP + TN)/(P + N)$$

4) F 评分 (F Measure, FM):

$$FM = 2 \times ACC \times Re/(ACC + Re)$$

5) 伪阳性率 (False Positive Rate, FPR), 又称错误命中率, 假警报率 (False Alarm Rate, FAR):

$$FPR = FP/N = FP/(FP + TN)$$

6) 伪阴性率 (False Negative Rate, FNR):

$$FNR = FN/(TP + FN)$$

这里我们特别关注下 F 评分, 从 F 评分的计算公式不难看出其结果是综合考虑了多个评估指标, 有较高的参考价值, 较为鲁棒的算法通常有更高的 F 评分.

此外, 考虑到实用性, 我们也会考察算法的速度, 采用每秒帧数作为参考指标.

CDnet2014 数据集提供了所有现有算法的结果以及各个算法与真实数据对比所得的统计指标, 同时给出两种评估方法: 一是在线评估, 将算法在所有场景上的结果上传到服务器进行评估; 二是离线使用他们给出的工具包进行评估, 评估结果可能跟在线方式有细微区别, 但整体不会差别太大. 考虑到 GMM 模型的应用场景, 我们仅在部分场景上进行了训练以及测试, 所以我们使用离线的方式来评估我们的算法, 并使用同样的方式评估对比算法.

3.2 算法在 CDnet2014 基准数据集上的实验结果

我们在 CDnet2014 基准 (baseline) 数据集中 highway、office、pedestrians 场景中随机选取了 10% 的真实数据 (约 800 条) 作为训练数据来分别训练 SubNet-4 与 SubNet-13.

参考 SegNet 的训练过程, 两个模型都采用交叉熵函数[18]作为损失函数, 用随机梯度下降算法 (Stochastic Gradient Descent, SGD) 在 Caffe 框架[23]上进行训练. 在训练 SubNet-4 时将学习速率固定为 0.01, 根据实际硬件条件将批量大小 (batch size) 设置为 10; 训练 SubNet-13 时将学习速率固定为 0.001, 批量大小设置为 4. 我们观察到在大约训练 15 个周期 (epoch, 指在所有训练数据上都进行一次训练) 后两个模型都已经基本收敛, 为了对比不同深度的模型的表现, 我们进一步将两个模型各自训练到约 30 个周期, 之后在所得模型上进行测试. 两个模型训练过程中损失函数值变化过程如图 9 所示.

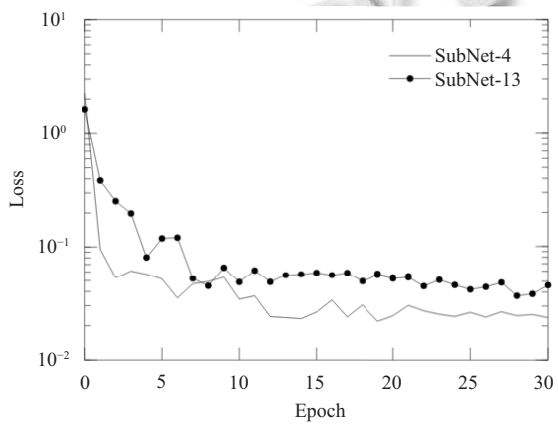

图 9 训练过程中损失函数值变化曲线

训练完成后, 我们使用 SubNet-4 与 SubNet-13 分别在这些场景中进行了测试, 将测试结果与原 GMM 算法以及当前效果比较好的 SuBSENSE[8]及 DeepBS[10]算法进行对比, 结果如表 2 所示.

表 2 不同算法在三个场景上的整体表现对比

算法	DeepBS	SuBSENSE	GMM	SubNet-4	SubNet-13
Re	0.9702	0.9481	0.6710	0.9552	0.9596
SPC	0.9985	0.9980	0.9930	0.9922	0.9867
FPR	0.0014	0.0020	0.0069	0.0078	0.0132
FNR	0.0297	0.0519	0.3289	0.0447	0.0404
FM	**0.9718**	**0.9551**	**0.7464**	**0.9108**	**0.8712**
ACC	0.9735	0.9621	0.8409	0.8704	0.7978

进一步分析 SubNet-4 与 SubNet-13 在基准数据集中各个场景下的表现, 如表 3、表 4 所示 (其中“平均”是统计所有场景下 TP、FN、TN、FN 的结果, 而不是简单计算统计指标的均值).

表 3 SubNet-4 在基准数据集中各个场景下的表现

场景	Highway	Office	Pedestrian	平均
Re	0.9852	0.9337	0.9876	0.9552
SPC	0.9926	0.9888	0.9977	0.9922
FPR	0.0074	0.0111	0.0023	0.0078
FNR	0.0148	0.0662	0.0124	0.0448
FM	**0.9370**	**0.8958**	**0.8115**	**0.9109**
ACC	0.8932	0.8609	0.8910	0.8704

表 4 SubNet-13 在基准数据集中各个场景下的表现

场景	Highway	Office	Pedestrian	平均
Re	0.9836	0.9430	0.9775	0.9596
SPC	0.9889	0.9794	0.9964	0.9867
FPR	0.0110	0.0205	0.0036	0.0133
FNR	0.0163	0.0569	0.0224	0.0404
FM	**0.9114**	**0.8494**	**0.8362**	**0.8713**
ACC	0.8491	0.7728	0.7305	0.7978

我们发现在 highway 场景中算法表现良好, 但是在 office、pedestrian 场景中表现较差. 为了研究算法表现不佳的原因, 我们选取了 SubNet-4 在 office 场景下的部分结果进行观察, 如图 10 所示.

通过结果可以发现, 因为我们的模型是通过背景图像与视频帧之间的差异性来找出前景目标, 然而由于 GMM 模型本身的缺点, office 场景中的前景目标在场景内长时间停留后, 导致 GMM 将其误看作背景. 鉴于 office 为一个背景变化不大的场景, 我们手动选择了一张背景图片作为全局背景 (图 10 中第 600 帧时生成的背景图像), 使用 SubNet-4 模型进行测试, 测试结果如表 5 所示.

测试结果验证了我们的猜想, 模型的表现有了很明显的提升进步.

另外算法各个模块的计算耗时以及不同算法之间性能对比结果如表 6、表 7 所示.

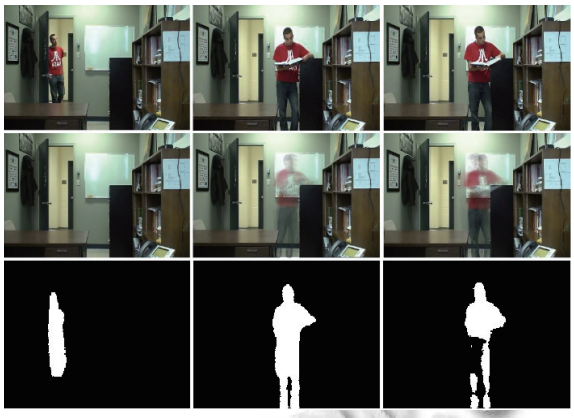

图 10 Office 场景中第 600、1200、1800 帧的测试结果: 视频帧 (上)、背景图像 (中) 与检测结果 (下)

表 5 不同背景下 SubNet-4 在 office 场景中的表现

指标	动态背景	静态背景
Re	0.9337	0.9967
SPC	0.9888	0.9960
FPR	0.0111	0.0040
FNR	0.0662	0.0033
FM	0.8958	0.9718
ACC	0.8609	0.9482

表 6 各个模块的计算耗时 (单位: ms)

模块	SubNet-4	SubNet-13
背景建模	5	5
编解码网络	59	94

实验中我们使用 GTX 1080 GPU 对算法的各个模块进行了加速, 与同样使用 GPU 加速的的 DeepBS 算法对比, 我们在较弱的硬件条件下 SubNet-13 取得了与 DeepBS 同等性能, 而 SubNet-4 的性能更好, 达到了 15FPS, 已经近乎实时.

表 7 不同算法的性能对比

算法	DeepBS	SuBSENSE	GMM	SubNet-4	SubNet-13
CPU	Intel E5-3.5 GHz	Intel i5-3.3 GHz	Intel i7-3.4 GHz	Intel i7-3.4 GHz	Intel i7-3.4 GHz
GPU	GTX Titan X	无	无	GTX 1080	GTX 1080
计算能力	很强	一般	强	强	强
FPS	10	30	21	15	10

总结该阶段的实验结果可以得知:

1) 我们的算法较原 GMM 算法有不小的提升.

2) 我们的算法的表现已经比较接近于当前的顶尖算法, 这一结果还是在没有进一步优化背景建模算法及检测结果的情况下实现的 (比如 DeepBS 在获取检测结果后又使用时间中值滤波对结果做了进一步的处理), 表明我们的算法有很具竞争力, 也很有优化潜力.

3) 对比 SubNet-4 于 SubNet-13, 4 层编解码网络的表现已经足够好, 进一步增加网络深度反而导致模型过拟合, 降低了泛化能力; 另一方面, 考虑到算法性能及实用性, 我们建议实际应用中不需要采用过深的网络.

3.3 在 CDnet 数据集上的进一步训练及测试

为了进一步验证模型的泛化能力, 我们在 CDnet 数据集的其他场景上进行了实验及测试. 参考之前不同深度编解码网络的对比结果, 我们这里仅使用 SubNet-4 进行了相关的实验.

3.3.1 使用基准模型在其他场景上进行测试

首先我们使用在基准数据集上训练得到的 SubNet-4 在 CDnet2014 数据集中 badWeather 类别下的部分场景上进行了实验, 其结果如表 8 所示).

表 8 SubNet-4 在不同场景上的表现

场景	Blizzard	Skating	Snowfall
Re	0.7051	0.9384	0.8313
SPC	0.9994	0.9953	0.9990
FPR	0.0006	0.0047	0.0009
FNR	0.2949	0.0616	0.1686
FM	**0.8015**	**0.9252**	**0.8539**
ACC	0.9287	0.9124	0.8776

结果发现算法在 skating 场景中表现良好, 但是另一些场景中表现不佳, 原因在于恶劣天气下场景中有飘舞的雪花等干扰, 基于背景-视频帧对比的话不一定能得到很好的结果, 我们需要进一步对模型进行微调优化.

3.3.2 使用不同场景数据微调模型并测试

深度神经网络模型有着很强的抗噪以及特征学习能力, 因此我们尝试使用新的场景中的数据对 SubNet-4 模型进行微调.

具体来说, 我们随机选取了上面几个场景的部分数据 (分别取各个场景的 10% 的数据, 总共约 2000 条

数据) 对 SubNet-4 进行了进一步的训练微调, 同样训练了约 30 个周期待模型收敛后在这些场景中进行了测试. 为了研究微调对模型的影响, 我们同时测试了微调后的模型在 CDnet2014 baseline 类别中 highway、office、pedestrians 场景下的平均表现, 测试结果如表 9、表 10 所示.

表 9 微调后的 SubNet-4 在不同场景上的表现

场景	Blizzard	Skating	SnowFall	Baseline
Re	0.9795	0.9859	0.9941	0.9143
SPC	0.9990	0.9982	0.9985	0.9970
FPR	0.0010	0.0018	0.0015	0.0030
FNR	0.0205	0.0141	0.0059	0.0857
FM	**0.9484**	**0.9761**	**0.9133**	**0.9284**
ACC	0.9191	0.9666	0.8446	0.9429

表 10 不同算法在 badWeather 部分场景中的平均表现

算法	DeepBS	SuBSENSE	GMM	SubNet-4
Re	0.7776	0.8233	0.7557	0.9855
SPC	0.9998	0.9991	0.9989	0.9987
FPR	0.0001	0.0008	0.0010	0.0013
FNR	0.2224	0.1767	0.2443	0.0145
FM	**0.8692**	**0.8766**	**0.8267**	**0.9516**
ACC	0.9853	0.9372	0.9126	0.9200

通过结果不难看出微调后的模型较原模型在新场景中的表现有了很大的提升, 在 badWeather 场景上的平均表现甚至超过了现有最好的算法. 其中一个很有趣的发现是微调后的模型在基准数据集上的表现有所提升 (F 评分从 0.9109 提升到 0.9284), 表明增加数据量有助于提高模型的鲁棒性.

3.3.3 与原 GMM 算法的直观性对比

我们选取了部分测试结果与原 GMM 算法进行了对比, 结果如图 11 所示.

不难看出我们的算法较原 GMM 算法有很大的提升, 并且在有效地解决“空洞”问题的同时大幅度提高了抗噪能力.

3.4 讨论

通过多个实验的结果可以得出我们的算法在原 GMM 算法上有了很大的提升, 特别是针对特定场景微调模型后, 算法的表现甚至超越了现有的一些顶尖算法, 证明了我们所提出的算法的有效性.

实验结果也同样说明了这种算法虽然有很强的学习与泛化能力, 在使用特定场景的数据进行微调后能提升效果, 但还是很依赖于背景建模方法, 容易受 GMM 模型弊端的影响, 然而也从另一个角度说明了如果配合更好的背景建模方法, 其效果能进一步地提升.

图 11 测试结果对比, 从上到下分别是视频帧、真实数据、GMM 算法结果与我们的算法结果

4 总结

受现实场景动态性的影响, 传统的运动目标检测算法往往效果不佳. 为了提升算法效果, 本文提出了一种新的基于编解码网络的运动目标检测算法, 将该问题看作像素级的语义分割问题, 结合 GMM 与深度神经网络, 无需进行复杂的参数调优即可实现高效的运动目标检测. 并且算法模型非常简单, 在使用 GPU 的情况下能够近乎实时地进行检测, 实用性很强. 另外由于前景背景分割模块是使用深度编解码网络实现的, 独立于背景建模方法, 如果配合更好的背景建模方法能够进一步的提升算法效果, 还有很大的优化空间.

总结得出论文的主要贡献在于:

1) 将运动目标检测问题转化为图像语义分割问题, 使用 GMM 结合基于反卷积的编解码网络有效地解决了 GMM 算法中的“空洞”等问题.

2) 证明了只需使用深度卷积网络进行前景背景分割, 无需较为复杂的背景建模方法以及参数调优就能很好地进行运动目标检测.

3) 本文的算法仍然依赖于 GMM, 在未对模型输出做任何形式的优化的情况下仍取得了很好的结果, 表明该方法很具潜力, 仍有很大的改进空间.

4) 我们的模型十分简单, 在使用 GPU 加速的情况下能够近乎实时地实现运动目标检测, 很具实用性.

下一步的研究工作一方面可以尝试使用递归神经网络 (Recurrent Neural Network, RNN) 等适合处理时序数据的网络模型来改进背景建模方法, 同时可以集

成为一个可端到端学习的深度网络模型, 来提升算法效果与效率; 另一方面可以探索使用更高效的语义分割模型来提升算法速度.

参考文献

1 Sajid H, Cheung SCS. Background subtraction for static & moving camera. 2015 IEEE International Conference on Image Processing (ICIP). Quebec City, QC, Canada. 2015. 4530–4534.

2 Stauffer C, Grimson WEL. Adaptive background mixture models for real-time tracking. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Fort Collins, CO, USA. 1999. 252.

3 Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1977, 39(1): 1–38.

4 Elgammal A, Harwood D, Davis L. Non-parametric model for background subtraction. European Conference on Computer Vision. Dublin, Ireland. 2000. 751–767.

5 Barnich O, Van Droogenbroeck M. ViBe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing, 2011, 20(6): 1709–1724. [doi: 10.1109/TIP.2010.2101613]

6 Kim K, Chalidabhongse TH, Harwood D, *et al*. Real-time foreground-background segmentation using codebook model. Real-Time Imaging, 2005, 11(3): 172–185. [doi: 10.1016/j.rti.2004.12.004]

7 Varadarajan S, Miller P, Zhou HY. Region-based mixture of Gaussians modelling for foreground detection in dynamic scenes. Pattern Recognition, 2015, 48(11): 3488–3503. [doi: 10.1016/j.patcog.2015.04.016]

8 St-Charles PL, Bilodeau GA, Bergevin R. SuBSENSE: A universal change detection method with local adaptive sensitivity. IEEE Transactions on Image Processing, 2015, 24(1): 359–373. [doi: 10.1109/TIP.2014.2378053]

9 Babaee M, Dinh DT, Rigoll G. A deep convolutional neural network for background subtraction. arXiv: 1702.01731, 2017.

10 Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2012. 1097–1105.

11 Bertinetto L, Valmadre J, Henriques JF, *et al*. Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision. Amsterdam, The Netherlands. 2016. 850–865.

12 余家奎. 基于视频的火花和烟雾检测算法研究[硕士学位论文]. 合肥: 中国科学技术大学, 2015.

13 夏梁, 何波. 基于卡尔曼滤波的背景更新算法. 电脑知识与技术, 2014, 10(6): 1242–1243.

14 Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction. Proceedings of the 17th International Conference on Pattern Recognition. Cambridge, UK. 2004. 28–31.

15 Reynolds D. Gaussian mixture models. Encyclopedia of Biometrics. US. 2015. 659–663.

16 Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 640–651. [doi: 10.1109/TPAMI.2016.2572683]

17 Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv: 1511.00561, 2015.

18 Noh H, Hong S, Han B. Learning deconvolution network for semantic segmentation. Proceedings of the 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1520–1528.

19 Chen LC, Papandreou G, Kokkinos I, *et al*. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv: 1606.00915, 2016.

20 Brostow GJ, Fauqueur J, Cipolla R. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 2009, 30(2): 88–97. [doi: 10.1016/j.patrec.2008.04.005]

21 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv: 1502.03167, 2015.

22 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.

23 Jia YQ, Shelhamer E, Donahue J, *et al*. Caffe: Convolutional architecture for fast feature embedding. Proceedings of the 22nd ACM International Conference on Multimedia. Orlando, Florida, USA. 2014. 675–678.