文章编号：2095-2163(2021)03-0177-04 中图分类号：TP183；TP391.41 文献标志码：J

基于孪生卷积神经网络的目标跟踪算法研究

邹 超，杨国平

（上海工程技术大学 机械与汽车工程学院，上海 201620）

摘 要：传统的目标跟踪算法采用人工特征描述物体特征，这类人工设计的特征不能全面地表达一个物体的特点，在跟踪过程中这些特征点容易受到外界因素的影响，导致跟踪效果不稳定。基于卷积神经网络的目标跟踪算法由于采用卷积神经网络提取物体的深层次特征，这类特征能够模仿人脑描述学习一个物体的深层特征，使得在跟踪中具有较高的稳定性，目标不容易丢失且跟踪的准确性更高，能够适应复杂多变的环境鲁棒性更好。本文提出的算法采用 Tensorflow 搭建网络框架，离线训练模型，然后利用 OpenCV 调用训练好的模型进行目标跟踪实验。算法在确保较高的跟踪准确性基础上，又得到了较快的跟踪速度、且显示出很强的实时性，具有一定的实际应用价值。

关键词：目标跟踪；卷积神经网络；Tensorflow；OpenCV

Research on a target tracking algorithm based on Siamese convolutional neural network

ZOU Chao，YANG Guoping

（School of Mechanical and Automotive Engineering，Shanghai University of Engineering Science，Shanghai 201620，China）

【Abstract】 The traditional target tracking algorithm uses artificial features to describe the characteristics of the object，this kind of artificial features can not fully express the characteristics of an object，so in the practical application of environmental changes these characteristic points will lead to inaccurate tracking，and the consequences become serious when the target is lost. The target tracking algorithm based on convolutional neural network uses convolutional neural network to extract the deep-level features of an object，which can describe the features of an object in a more comprehensive way，thus resulting in higher stability in the tracking process，less target loss and higher tracking accuracy，and being able to adapt to complex and changeable environments. The algorithm proposed in this paper uses Tensorflow to build network framework and offline train model，and then uses OpenCV to call the trained model for target tracking experiment. The algorithm not only keeps the tracking accuracy，but also has fast tracking speed and real time，which has certain practical application value.

【Key words】 target tracking；convolutional neural network；Tensorflow；OpenCV

0 引 言

随着深度学习[1]的不断发展成熟，其应用范围也在不断扩大。最近很多学者将其引用到目标跟踪中。目标跟踪是计算机视觉领域的一个经典应用，通过跟踪单个或者多个移动目标，可以获取目标的位置、运动速度、方向等信息为相关的任务提供信息支持和消除误差。其应用诸如手势操作、行人行为预测、车辆跟踪、机器人抓手等。经典的目标跟踪算法主要有 CamShift[2]为代表的运用目标的色彩特征的跟踪算法、以 KCF[3]代表的核相关滤波算法、卡尔曼滤波算法[4]、粒子滤波算法[5]以及利用物体的特征点实现跟踪诸如基于 SIFT 特征点、SURF 特征点的目标跟踪算法等。这些算法均通过人工设计的特征来表达物体，然后通过分析人工特征实现对目标的跟踪。

Mean-Shift、CamShift 通过物体的色彩信息计算出色彩分布直方图，然后利用直方图的分布实现对目标的跟踪。基于 SIFT 和 SURF 特征点的算法主要运用人工设计的角点、即一些灰度突变的位置点实现对目标的跟踪。这些方法受限于人工设计的特征的局限性和容易受到诸如光照变化、物体形变、旋转、运动模糊等因素的影响，导致跟踪的准确性不够高、目标漂移甚至丢失目标。核相关滤波算法（Kernel Correlation Filter，KCF），是在 2014 年由 Henriques 等人提出来的。以 KCF 为代表的核相关滤波跟踪算法在速度上具有较大的优势，能够满足实时性的要求，但是其难以应对目标的尺度变化、遮

作者简介：邹 超（1993-），男，硕士研究生，主要研究方向：机器视觉、图像处理、目标跟踪；杨国平（1962-），男，博士，教授，主要研究方向：汽车产品虚拟样机技术、汽车系统建模与计算机仿真、车辆流体传动与控制技术。

通讯作者：邹 超 Email：zouchao1231@163.com

收稿日期：2020-10-26

挡、重叠等问题。

针对这些问题,很多学者提出了一些改进算法结合多尺度、多通道等方法提高核相关算法的准确性。所以找到一种合适的特征能够全面、准确地描述一个物体成为目标跟踪领域里的研究热点,从色彩特征到角点特征,从单通道特征到多通道特征都不能完全满足跟踪的需求,此外融合的特征信息越多必然会造成算法结构的冗余、庞大的数据处理对算法的实时性造成影响。卷积神经网络 CNN 处理图片类似于人的大脑,处理得到的是更高层次、更高维度的信息,所以将其应用在目标识别领域是必然的趋势,其处理图像信息的方法同样可以运用在目标跟踪领域中用于提取目标的特征。

1 网络结构与模型

近年来,孪生卷积神经网络被广泛运用在目标跟踪领域,一些使用孪生卷积神经网络的目标跟踪算法在各项比赛中都取得了不错的成绩。有些算法为了提高跟踪的准确性和算法的鲁棒性将网络结构设计得比较复杂、深度较深,在准确性提高的同时也导致了算法在跟踪速度上的表现不够优秀。并且,结构越深、越复杂的网络参数也会越多,算法的运行对硬件条件要求更高且难以满足实际工程应用的要求,不具有实际应用价值。所以本文设计一种类似于孪生卷积神经网络[6]的结构神经网络,网络结构如图 1 所示。

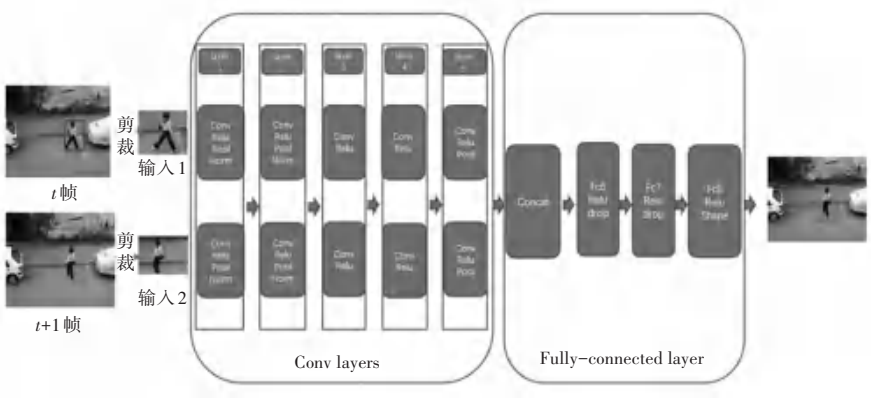

图 1 网络结构示意图

Fig. 1 Schematic diagram of network structure

网络结构采用双路五层卷积网络提取物体特征,然后合并连接上三层全连接层通过回归的方式得出目标所在位置。这样的卷积网络深度比较合适、结构简单,五层卷积参数较少,对硬件的需求不高,可以达到实时的跟踪效果。算法设计流程拟详述如下。

Step 1 输入 1:输入图像大小为 127 * 127 * 3,以目标图像位置为中心,假设目标框的大小为 $w * h$,则以目标框的位置为中心剪裁大小为 $2w * 2h$ 的图像;

输入 2:输入图像大小为 227 * 227 * 3,以目标图像位置为中心,假设目标框的大小为 $w * h$,则以目标框的位置为中心剪裁大小为 $2w * 2h$ 的图像。

Step 2 将输入 1 和输入 2 输入到 CNN 中,进行卷积提取目标特征。

Layer1:使用 11 * 11 * 96 的卷积核进行卷积,卷积步长为 4,使用 *ReLU* 函数作为激活函数,最大池化操作,使用 3 * 3 的核步长为 2,再进行正则化后输入到 Layer2 中。

Layer2:使用 5 * 5 * 256 的卷积核进行卷积,边缘补充 2 行 2 列,卷积步长为 2, *ReLU* 激活函数,最大池化,采用 3 * 3 的核,步长为 2,再进行正则化输入到 Layer3 中。

Layer3:卷积核为 3 * 3 * 384, 激活函数为 *ReLU*。

Layer4:卷积核为 3 * 3 * 384, 激活函数为 *ReLU*。

Layer5:卷积核为 3 * 3 * 256,补充边缘一行一列进行卷积操作,激活函数为 *ReLU*, 最大池化,采用 3 * 3 的核,步长为 2。

Step 3 将 2 个输入经过卷积得到数据互相关运算到一起。

Step 4 将互相关运算之后的数据输入到全连接层。

Fc-layer6　输出 4 096 维数据，激活函数为 *ReLU*，*drop_out_ratio* 为 0.5。

Fc-layer7　输出 4 096 维数据，激活函数为 *ReLU*，*drop_out_ratio* 为 0.5。

Fc-layer8　输出 4 维数据（目标框的左上角和右下角坐标位置）。

卷积部分是并行的 2 个卷积网络，同时对目标和待搜索区域进行卷积。考虑物体的移动特性，一般情况下目标在下一帧的位置不会距离上一帧的目标位置太远，所以对下一帧的卷积并不是对整个区域进行卷积，以上一帧中目标的位置为中心，通过裁切得到 2 倍于目标框大小的搜索图像送入卷积网络提取特征，最后通过全连接层学习出 t 帧和 $t+1$ 帧的目标位置关系，得出 bounding box。返回的 bounding box 由 4 个数据组成，分别为跟踪得到的目标框的左上角坐标位置和右下角坐标位置。

2　重要参数设定

激活函数选用 *ReLU* 激活函数。常用的激活函数有 *sigmoid* 激活函数、tanh 激活函数以及 *ReLU* 激活函数。对于线性函数而言，*ReLU* 的表达能力更强，尤其体现在深度网络中。而对于非线性函数而言，ReLU 由于非负区间的梯度为常数，因此不存在梯度消失问题使得模型的收敛速度维持在一个稳定状态。*ReLU* 函数图像如图 2 所示。

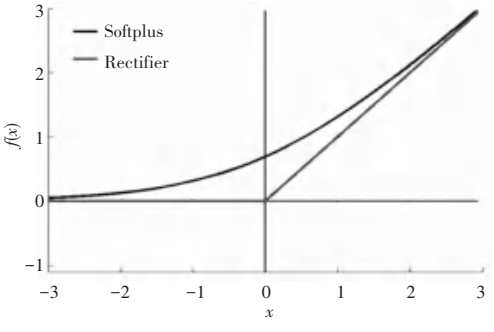

图 2　*ReLU* 函数图像

Fig. 2　*ReLU* function image

由图 2 可知，*ReLU* 函数的数学定义形式如下：

$$ReLU(x)=\begin{cases}x, & \text{if } x>0,\\ 0, & \text{if } x\leqslant 0.\end{cases} \tag{1}$$

损失函数采用 L_1-Loss。损失函数可以避免误差在反向传播时出现梯度消失或者梯度爆炸的情况，表现在训练时网络过快或者过慢的收敛，导致网络无法学习到特征。这里需用到如下数学公式：

$$L=\sum_{i}^{n}|y_i-d_i|. \tag{2}$$

其中，y 表示实际的输出，d 表示真实值。

在训练网络时，由于参与模型的计算参数太多，而训练样本有限，训练出来的模型很容易产生过拟合的现象。过拟合顾名思义就是训练出来的模型过度拟合训练集的数据，表现出来就是在训练数据上损失函数较小，预测准确率较高；但是在测试数据上损失函数比较大，这样训练出来的模型泛化能力不强。为了提高模型的泛化能力，对网络中的神经元进行剔除很有必要，采用的 *dropout-ratio* 为 0.5，即随机地让全连接层一半的神经元不参与运算。

3　模型训练与实验

3.1　模型训练

一个神经网络模型的优劣不仅与网络的结构、算法等因素有关，在很大程度上也取决于训练网络数据的质量。一个标注准确、复杂程度高、场景全面的数据集可以使训练得到的模型准确率高、鲁棒性好和泛化能力强。训练的样本使用阿姆斯特丹普通视频跟踪库的 307 个视频和 OTB100[7] 中的 98 个类别 100 个视频序列。OTB100 中的视频序列见图 3。阿姆斯特丹普通视频跟踪库涵盖了多种视频情况，例如：照明、透明度、镜面反射、与类似物体混淆、杂乱、遮挡、缩放、严重形状变化、运动模式、低对比度等。OTB100[7] 中标记了 98 个物体，每一帧图片中的物体位置均有相应的标记，100 个视频序列对应 100 个 groundtruth_rect.txt，存放着每个序列中目标的位置。模型使用 Tensor flow 搭建并训练，测试集则是来源于 VOT 2014 Tracking Challenge 中的 25 部视频，视频的每一帧都注释有遮挡、光照变化、运动、大小变化、机位移动等变化形式。

使用 Tensorflow 训练好模型并保存，使用 OpenCV 调用模型进行跟踪实验。目前，OpenCV dnn 模块支持 Caffe、TensorFlow、Torch、PyTorch 等深度学习框架。

3.2　实验结果

实验结果如图 4 所示。实验发现所设计的网络结构可以很好地提取到物体的特征，实现跟踪。跟踪的准确性较高、速度较快，具有一定的实际应用价值。例如，图 4(a) 中的人脸有遮挡、变形，图 4(b) 中的汽车有距离、尺度变化，图 4(c) 中的踢足球的人有大小、形变等变化，图 4(d) 中出现多个与目标相似的行人和自行车。算法能够应对这些变化并实现跟踪。

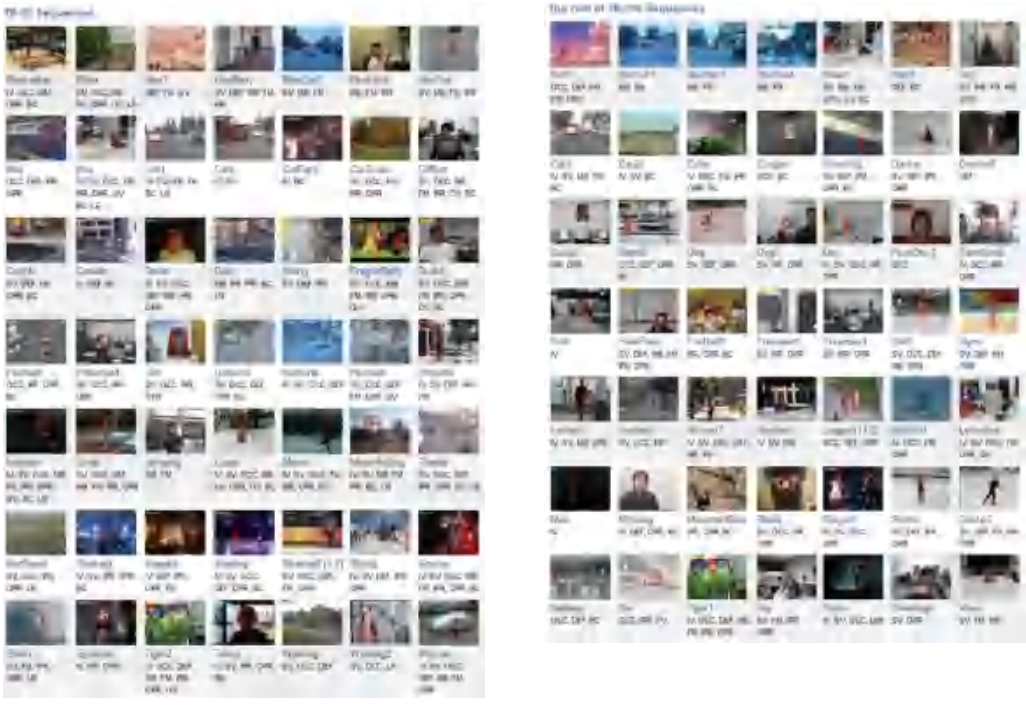

图 3 OTB100 中的视频序列

Fig. 3 Video sequences in OTB100

(a) 人脸

(a) Human face

(b) 汽车

(b) Cars

(c) 踢球的人

(c) Football player

(d) 与目标相似的多个结果

(d) Multiple results similar to the goal

图 4 实验结果

Fig. 4 Experimental results

4 结束语

模型采用了五层卷积再加上三层全连接,结构类似于 CaffeNet。实验发现算法能够应对遮挡、形变、运动模糊、尺度变化等挑战实现良好的跟踪效果。就整体而言,算法在跟踪的准确性和速度上表现都还好。算法在 GTX1050TI+INTER I5 平台上能够达到 100+FPS 的效果,即使在只有 CPU 的条件下也能够稳定在 20+FPS,有着不错的实时性表现。算法的泛化能力依然有较大的改进空间,需要加大训练的数据量,微调参数来提高模型的泛化能力。

参考文献

[1] 罗海波, 许凌云, 惠斌,等. 基于深度学习的目标跟踪方法研究现状与展望[J]. 红外与激光工程, 2017, 46(5):6-12.

[2] 邬大鹏, 程卫平, 于盛林. 基于帧间差分和运动估计的 Camshift 目标跟踪算法[J]. 光电工程, 2010, 37(1):55-60.

[3] HENRIQUES J F, RUI C, MARTINS P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis Machine Intelligence, 2015, 37(3): 583-596.

[4] 汪颖进, 张桂林. 新的基于 Kalman 滤波的跟踪方法[J]. 红外与激光工程, 2004, 33(5):505-508.

[5] 王法胜, 鲁明羽, 赵清杰,等. 粒子滤波算法[J]. 计算机学报, 2014, 37(8):1679-1694.

[6] HELD D, THRUN S, SAVARESE S. Learning to track at 100 FPS with Deep Regression Networks[M]//LEIBE B, MATAS J, SEBE N, et al. Learning to track at 100 Fps with deep regression networks. Computer Vision - Eccv 2016. Lecture Notes In Computer Scienc. Cham:Springer, 2016,9905:749-765.

[7] WU Yi, JONGWOO L, YANG M H. Object tracking benchmark [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2018,37(9):1834-1848.