

TensorFlow 平台下的视频目标跟踪 深度学习模型设计

刘 帆, 刘鹏远, 李 兵, 徐彬彬

中国人民解放军军械工程学院四系, 河北 石家庄 050003

摘要 训练模型复杂且训练集庞大导致深度学习的发展受到严重阻碍。使用 Google 最新开源的 TensorFlow 软件平台搭建了用于视频目标跟踪的深度学习模型。介绍了深度学习的原理和 TensorFlow 的平台特性, 提出了使用 TensorFlow 软件平台设计的深度学习模型框架结构, 并使用 VOT2015 标准数据集中的数据设计了相应的实验。经实验验证, 该模型具有较高的计算效率和识别精度, 并可便捷地调整网络结构, 快速找到最优化模型, 很好地完成视频目标识别跟踪任务。

关键词 机器视觉; TensorFlow; 深度学习; 计算机视觉; 目标跟踪

中图分类号 TP391 文献标识码 A

doi: 10.3788/LOP54.091501

Deep Learning Model Design of Video Target Tracking Based on TensorFlow Platform

Liu Fan, Liu Pengyuan, Li Bing, Xu Binbin

Forth Department, Ordnance Mechanical Engineering College of PLA, Shijiazhuang, Hebei 050003, China

Abstract Due to the complexity of training model and huge training set of deep learning, the development of deep learning is seriously hindered. We use an open-source platform called TensorFlow developed by Google to build deep learning model for video object recognition and tracking. Some basic theories are introduced including the principles of deep learning and TensorFlow's properties. The framework of deep learning model developed by TensorFlow is outlined. Experiments are designed based on the standard data in VOT2015. Experimental results show that the model has high computational efficiency and recognition accuracy, and it can adjust network structure easily, find optimal structural model fast and complete video object recognition and tracking task well.

Key words machine vision; TensorFlow; depth learning; computer vision; object tracking

OCIS codes 150.0155; 150.1135; 100.4999

1 引 言

视频目标识别与跟踪的本质是在连续视频帧中找到感兴趣的运动目标所处的位置。现有很多成熟的算法可以实现此功能, 如基于粒子滤波的目标跟踪算法和基于 meanshift 的目标跟踪算法等。有些学者在目标跟踪方法中融入机器学习的思想, 如支持向量机(SVM)和 AdaBoost 分类算法等, 这些算法都需要手动提取特征, 随着识别对象的改变, 通常需要重新设计特征, 而深度学习能够自动提取并学习更深层次的特征, 具有更强的适用性和智能性。

深度学习^[1-2]的目的是建立多层神经网络, 以期模仿人脑机制来分析和解释图像。深度学习通过组合浅层特征形成更加抽象的高层特征, 以发现数据的更深层次分布式特征表示。深度学习的产生引起了计算机

收稿日期: 2017-03-01; 收到修改稿日期: 2017-04-10

作者简介: 刘 帆(1995—), 男, 硕士研究生, 主要从事计算机视觉方面的研究。E-mail: 2434344286@qq.com

导师简介: 刘鹏远(1975—), 男, 博士, 副教授, 硕士生导师, 主要从事增强现实维修方面的研究。

E-mail: 18253163045@163.com

视觉领域的一场变革,目前在图像识别、场景识别和物体追踪等方面都已取得显著的效果,并表现出了极大的应用价值。由于深度学习需要处理大量的训练样本并提取高层次的特征,因此计算模型复杂,计算量极大,这在很大程度上限制了其推广应用。

自 2016 年 3 月以来,Google 公司开发的围棋人工智能程序“AlphaGo”风靡一时,其主要工作原理就是深度学习。而 TensorFlow 是 Google 于 2015 年 11 月开发的开源人工智能学习系统,在搭建深度学习模型时表现出独特的优势,很快得到诸多开发者的青睐。TensorFlow 提供 Python 和 C++ 编程接口,可从底层实现对线程和队列的运算,能够简单调用所有可用的硬件资源^[3-4];用户仅需提供目标函数,输入数据和节点结构等,TensorFlow 即可把节点分配给不同的设备,实现并行计算;其通过先描绘出一个操作交互图的方式,完全在 Python 语言外部进行一些费时的矩阵运算,避免了切换回 Python 环境时的操作开销。这些特点使得 TensorFlow 在进行深度学习时比以往的软件平台更加快速有效。

本文在 TensorFlow 平台下构建图像识别跟踪的深度学习过程,并设计视频目标识别跟踪的深度学习框架结构,在进行网络结构的性能调优后,与以往的软件平台进行性能对比,验证基于 TensorFlow 的深度学习模型快速识别的优势。

2 深度学习模型原理

深度学习的实质是构建具有多隐层的机器学习模型,对海量的训练数据进行学习得到更有用的特征,从而提升分类或预测的准确性。相比于传统的机器学习,其最大的特点是具有多层的网络结构。

通常的神经网络结构中^[5-7],隐层神经元与图像像素点进行全连接,当神经元数量过多时,权值参数庞大。在图像识别领域最适合的网络结构是卷积神经网络,其比普通的神经网络增加了卷积层和降采样层。其中,卷积层较好地提取相邻像素之间的共同特性,具有权值共享特性,能够大量减少神经网络训练过程中的参数数量;降采样层可以保证位移不变。

如图 1 所示,一般在卷积神经网络中卷积层与输入层相连^[7-10],从输入层获得输入图像,网络的中间层是由卷积层与降采样层交替连接而成,即每进行一次卷积运算,就进行一次下采样,之后继续进行卷积。经过数次卷积和下采样后,再使用全连接结构,最后得到整个网络的输出。输出层使用目标的标签信息,采用有监督的训练方式进行网络收敛计算。为降低运算复杂度,卷积神经网络将神经元平面上神经元的权重进行共享,通过不同的卷积核提取不同的特征,有效减少了卷积神经网络训练过程中的参数数量,降低了网络中参数选择的复杂度。卷积层中,前一层的特征图与一个可学习卷积核进行卷积运算,结果经过激活函数输出成为下一层特征图的神经元,从而构成下一层对应某一特征的特征图。使用不同的卷积核进行卷积可提取前一层特征图的不同特征,代表不同特征的特征图共同作为下一层子采样的输入数据。卷积层 l 中第 j 个神经元为

$$a_j^l = f(b_j^l + \sum_{i \in M_j} a_i^{l-1} * k_{ij}^l), \quad (1)$$

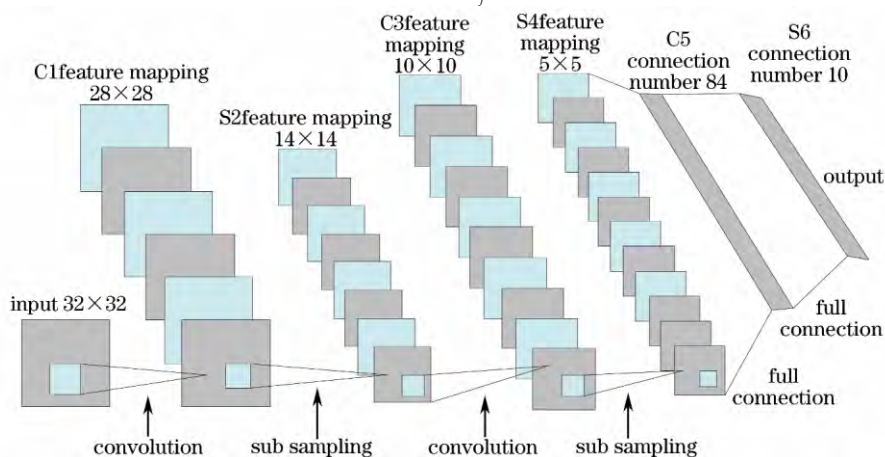


图 1 卷积神经网络结构

Fig. 1 Convolution neural network structure

式中 k 为卷积核, M 为输入层的感受野, b 为偏置, $f(\cdot)$ 表示卷积网络的激活函数。一个卷积层通常由几个特征图构成, 且各个特征图之间共享权值, 这样即可降低网络中自由参数的数量。

子采样层(又称池化层)通常跟随在卷积层之后, 可以将多个像素值压缩为一个, 通常选取多个像素的平均值或最大值, 其功能是提取特征以减少数据规模, 降低网络的空间分辨率, 从而实现畸变、位移稳健性。子采样得到的神经元的计算公式为

$$a_j^l = f[\beta_j^l \cdot \text{pooling}(a_j^{l-1}) + c_j^l], \quad (2)$$

式中 $\text{pooling}(\cdot)$ 表示池化函数, β 为权重系数。

卷积神经网络的训练过程可分为前向传播和后向传播两个阶段。在前向传播阶段, 信息从输入层开始向前逐层传播, 经过各个卷积层和全连接层直至输出层, 在此过程中, 网络执行的计算为

$$y_i = f_i \{ \cdots \{ f_2 [f_1 (X w_1) w_2] \cdots \} w_i \}, \quad (3)$$

式中 y_i 为卷积网络第 i 层的输出, $f_i(\cdot)$ 表示卷积网络第 i 层激活函数, w_i 为第 i 层卷积核的权值向量。

在后向传播阶段, 计算实际输出与标签信息的差值, 然后按照极小化误差的策略, 反向传播调整网络各层的权值, 通常采用随机梯度下降法。

基于卷积神经网络的深度学习模型具有权值共享、模型复杂度低以及权值数量少等优点^[11-13]。与传统识别算法相比, 其能够避免复杂的手动特征提取和数据重建过程, 可实现自动特征学习, 在处理大规模的图像识别时具有较大优势。

3 TensorFlow 平台特性

TensorFlow 支持 C、C++ 以及 Python 等编程语言, 其 Python 库中提供了大量辅助函数来简化构建图的工作, 简单易用。本文构建的实验框架主要基于 Python 库。TensorFlow 是一个使用图来表示计算任务的编程系统^[14]。图中的节点被称之为 op, 一个 op 获得零个或多个张量, 每个张量是一个类型化的多维数组。这种形式可以高效地构建深层神经网络, 并且可以方便地修改网络结构。TensorFlow 本身不包含各种运算函数, 所以在开始编程前, 需要先使用 import 命令连接需要使用的数据库, 如“import cv”语句表示调用 Open CV 图像处理函数库。

TensorFlow 程序可以分为构建图和执行图两个阶段, 如图 2 所示。

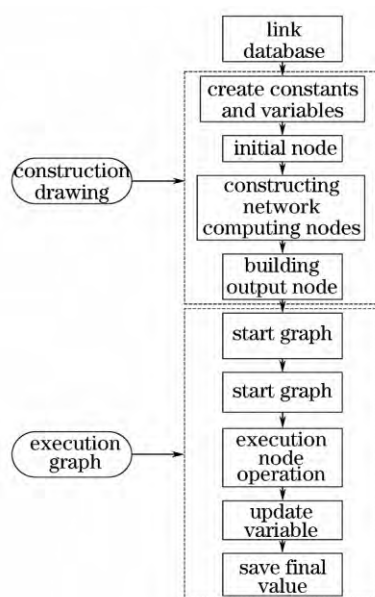


图 2 TensorFlow 开发流程

Fig. 2 Development process of TensorFlow

1) 构建图^[15-16]: TensorFlow 在执行计算操作之前需要完成计算图的构建。计算图由单个节点和节点间的相应操作组成, 另外, 需要定义所需的常量和变量。

2) 执行图: 图创建完成后, 才能运行。首先创建一个 Session 对象, 称为图的启动, 然后进行变量的初始化, 执行节点间相应操作并更新变量值, 最后保存最终数据。

4 基于 TensorFlow 搭建深度学习模型

4.1 深度学习模型框架结构图

基于 TensorFlow 设计的用于图像识别跟踪的深度学习模型框架结构如图 3 所示。

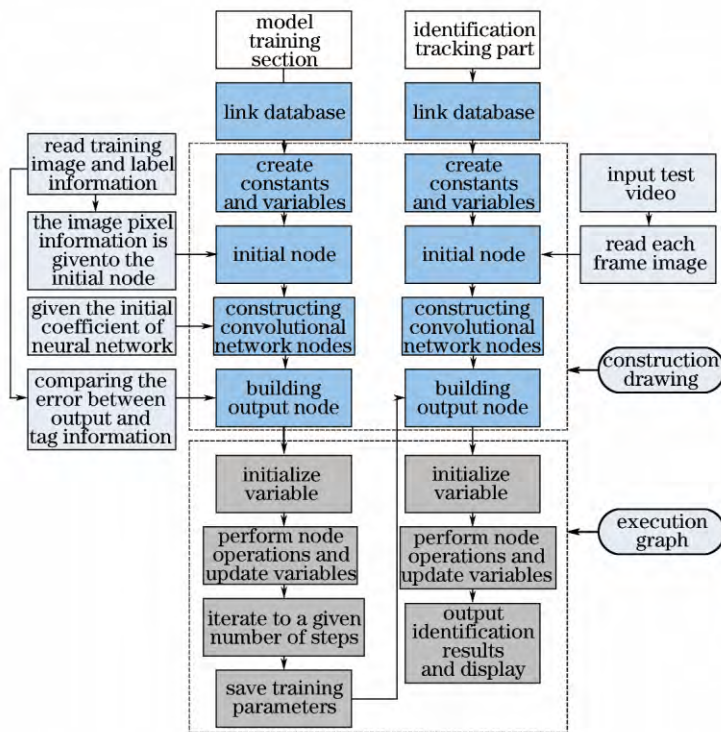


图 3 基于 TensorFlow 设计的深度学习模型框架结构

Fig. 3 Framework of deep learning model based on TensorFlow

实验分为模型训练和识别跟踪两部分, 这两部分的节点构造和节点间的操作相同, 主要区别在于权值变量的初始赋值。其中, 模型训练部分神经网络的初始权值人为给定, 通过训练样本调整卷积网络结构的各层权值, 使实际输出和理想输出误差较小; 而识别跟踪部分的权值直接使用模型训练部分得到的神经网络权值, 将测试视频信息通过各层网络映射, 得到的输出即为识别结果。

4.2 实验环境配置和初始准备

在 Window 系统中使用虚拟机运行 TensorFlow 进行实验, 主要配置参数为虚拟机 VMware Workstation, 运行内存 6 G, 虚拟机使用磁盘 60 G; 电脑配置为 Window 7 操作系统, 8 GB 运行内存, 显卡为 GTX960M。

实验使用的数据为 VOT2015 标准数据集集中的 Bird 图片集(200 张), 图片大小为 720 pixel×400 pixel, 使用 100 张图片作为训练样本, 另外 100 张图片组成视频流作为测试样本, 将识别出的目标用白色矩形框标出。对于训练样本集, 每张图片进行如下处理: 标出目标中心点像素位置 (x, y) , 以及矩形框长 h 和宽 w (矩形框设为同样大小); 然后对数据进行归一化, 即将输入特征的各个属性放至统一的区间内, 使得各个属性在分类中具有相同的贡献。

4.3 深度学习模型设计实现

实验开始阶段需要链接相关函数库, 程序中采用如下的语句:

```
import TensorFlow as tf
import cv
```

...

import TensorFlow as tf 是 TensorFlow 的启动语句,用来调用 TensorFlow 内的函数库;import cv 用来调用相应的 open cv 函数库。

构建图是深度学习模型设计的关键部分,其主要工作是完成节点的创建并定义节点间的运算方式,由于节点的建立和操作均依托张量进行,所以模型的节点连接以张量的连接形式表现,通常将一组图像集表示为一个 4 维浮点数数组,4 个维度分别为 [batch, height, width, channels]。

初始节点和输出节点的构建:

```
x__=tf.placeholder(" float", shape=[None, 720, 400, 3])
y__= tf.placeholder(" float", shape=[None, 4])
```

720 和 400 代表输入图片的分辨率,4 代表输出数据有 4 个数值,即目标中心点 x 、 y 轴坐标和矩形框的长、宽值。

网络结构:实验设定进行 5 次卷积和下采样,每次卷积后的当前层节点数目分别为 8、16、32、64、128、256。

卷积层节点的构建:

```
W_conv1 = weight_variable([3, 3, 3, 16])
b_conv1 = bias_variable([16])
W_conv2 = weight_variable([2, 2, 16, 32])
...
W_conv5 = weight_variable([3, 3, 128, 256])
b_conv5 = bias_variable([256])
```

全连接层和输出层节点的构建:

```
W_fc1 = weight_variable([4×4×256, 256])
b_fc1 = bias_variable([256])
W_fc2 = weight_variable([256, 256])
b_fc2 = bias_variable([256])
W_fc3 = weight_variable([256, 4])
b_fc3 = bias_variable([4])
```

在数组 [2, 2, 3, 16] 中,前两个参数代表使用 2×2 的卷积网格进行卷积,第三个参数代表 RGB 通道彩色图像,最后一个参数代表经过一次卷积后生成了 16 张特征图,数组 [2, 2, 16, 32] 中后两个参数代表由 16 张特征图与 32 张特征图连接。

定义神经网络节点间的操作运算:卷积和池化。

```
h_conv1 = tf.nn.relu(conv2d(x, W_conv1) + b_conv1)
h_pool1 = max_pool_2x2(h_conv1)
```

神经网络的构建按照先卷积一次,再池化一次的顺序交替进行。

极小化误差方法:

```
rmse=tf.sqrt(tf.reduce_mean(tf.square(y__ - y_conv)))
train_step=tf.train.AdamOptimizer(1e-3).minimize(rmse)
```

实验采用的学习方法是自适应矩估计(Adam)算法,Adam 算法根据损失函数对每个参数的梯度进行一阶矩和二阶矩估计,动态调整每个参数的学习速率。同时,Adam 算法是基于梯度下降的方法,每次迭代参数的学习步长都有一个确定的范围,不会因较大的梯度而导致较大的学习步长,因此参数值比较稳定。

5 实验结果分析

5.1 识别跟踪结果

训练完成后,将测试样本输入至识别跟踪模型。测试样本为 100 帧图片合成的视频,在识别跟踪模型中,测试样本需要通过同样的卷积结构,但是网络中的权值参数 w 由部分训练得到。选取一组具有代表性



图 4 视频目标识别跟踪结果

Fig. 4 Results of video object recognition and tracking

的视频帧显示识别跟踪结果,如图 4 所示。

从图 4 可以看出,当目标(具有黑色羽毛的鸟)不断移动时,绿色矩形框始终紧紧将目标包围在框内,即使干扰物(黄色羽毛的鸟)将目标部分遮挡,仍然能够较好地识别。这表明构建的视频目标跟踪深度学习模型效果理想,具有实时追踪,区分干扰物和抗部分遮挡的能力。

5.2 训练步数对识别误差的影响

识别误差是测试得到的目标中心点像素与实际目标中心点像素值的二次方根差,即中心位置误差,定义为

$$\epsilon = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}, \quad (4)$$

式中 (x_1, y_1) 为识别出的目标中心点位置, (x_0, y_0) 为标记的中心点位置,其为归一化值时,识别误差也为归一化误差。

该卷积模型每一步的训练时间为 10.378 s,每帧图像的识别时间为 0.4416 s,随着训练步数的增加,识别误差的变化如图 5 所示。

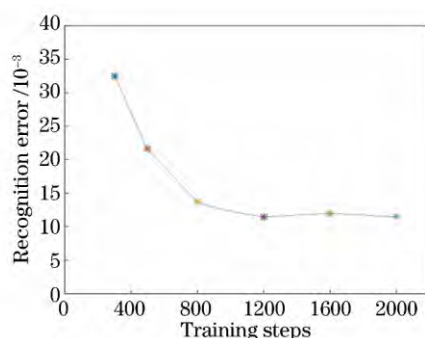


图 5 训练步数对识别误差的影响

Fig. 5 Effect of training steps on recognition error

从图 5 可以看出,随着训练步数的增加,识别精度也在不断提高,但当训练步数达到 1200 后,识别精度几乎没有明显变化,说明此时神经网络训练得到的权值已经达到稳定,因而后续实验都将采用 1200 步进行训练。

5.3 卷积网络层数和节点数对识别效果的影响

神经网络的层数和节点数对识别效果具有十分重要的作用,网络层数和节点数太少,识别精度会变差,而层数和节点数过多,训练时间和识别时间又会增加。经验表明,一般构建 5~7 层神经网络即可达到较好的效果;卷积节点数可以为几十到数百不等,但通常为偶数,如 16,32,64,...

使用 TensorFlow 平台改变网络结构进行实验。当最大节点数固定为 512 时,卷积网络层数对识别效果的影响如图 6 和图 7 所示。

当卷积网络层数固定为 6 时,卷积网络最大节点数对识别效果的影响如图 8 和图 9 所示。

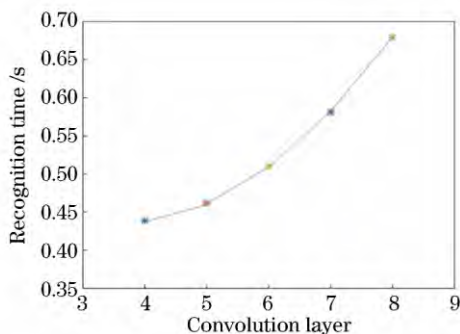


图 6 卷积网络层数对识别时间的影响

Fig. 6 Effect of convolution network layer on recognition time

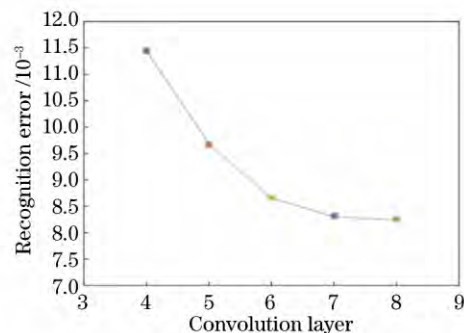


图 7 卷积网络层数对识别误差的影响

Fig. 7 Effect of convolution network layer on recognition error

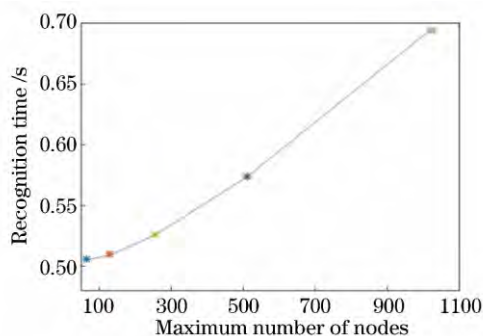


图 8 最大节点数对识别时间的影响

Fig. 8 Effect of maximum number of nodes on recognition time

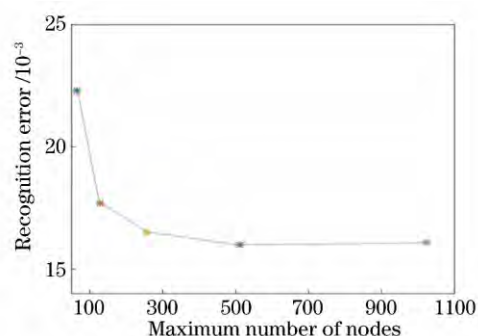


图 9 最大节点数对识别误差的影响

Fig. 9 Effect of maximum number of nodes on recognition error

从图 8 可以看出,通过增加卷积层层数和每层节点数,可以在一定程度上提升识别精度;且当网络层数或者节点数达到一定程度时,识别误差的变化趋于平缓,但是识别时间会不断增长,且增长越来越快。因此,实际应用中可以根据具体性能需求,兼顾识别时间和识别误差,找到最佳的平衡点。分析图 7 和图 9 可知,如果对精度的要求更高,可以选用 7 层以上的神经网络,最大节点数选择 512 或者更多;同样地,分析图 6 和图 8 可知,如果对实时性要求更高,可以选用不超过 5 层的神经网络,最大节点数选择 128;如果要兼顾识别时间和误差,可以选用 6 层神经网络,最大节点数选择 256。

基于 TensorFlow 软件平台设计的深度学习模型在调整网络结构时具有方便快捷的优势,同时其运行速度较快,可以在模型的设计搭建过程中快速找到合适的网络结构,提升深度学习的实现效率。

5.4 不同软件平台下深度学习模型的性能比较

Caffe 是当前深度学习方面应用较多的软件,文献[17-18]应用了 Caffe 深度学习框架,在 Matlab 环境中搭建深度学习模型,其每帧处理速度约为 1 frame/s,由图 6 和图 8 可知,对于基于 TensorFlow 软件平台和 Python 语言环境搭建的深度学习模型,当采用最优搭配 6 层数神经网络和 256 个最大节点数时,每帧处理速度达到了 0.53 frame/s,可见速度提升明显,并且可通过降低最大节点数和网络层数进一步优化。

采用准确率和成功率作为识别性能的衡量指标^[19]。准确率的定义为中心位置误差小于设定阈值的跟踪目标占有所有视频帧数的百分比,这里以 20 个像素点作为阈值。将成功率定义为

$$S = \frac{|R_t \cap R_a|}{|R_t \cup R_a|}, \quad (5)$$

式中 R_t 为跟踪目标的边界区域, R_a 为目标真实的边框区域, \cap 、 \cup 分别表示对 R_t 和 R_a 进行集合的交和并运算, $|\cdot|$ 表示集合中像素的个数。当 $S > 0.6$ 时,认为跟踪成功。表 1 和表 2 分别比较了本文搭建的深度学习模型与 Caffe 框架下几种算法的平均准确率和成功率。

表 1 不同算法的准确率对比

Table 1 Accuracy rate comparison of different algorithms

/%

	Proposed algorithm	CNN-BPF algorithm ^[18]	TLD algorithm ^[20]	Struck algorithm ^[21]	MIL algorithm ^[22]	IVT algorithm ^[23]
Accuracy rate	91.0	91.3	90.4	87.5	80.3	78.5

表 2 不同算法的成功率对比

Table 2 Success rate comparison of different algorithms

/%

	Proposed algorithm	CNN-BPF algorithm ^[18]	TLD algorithm ^[20]	Struck algorithm ^[21]	MIL algorithm ^[22]	IVT algorithm ^[23]
Success rate	83.0	84.6	83.2	80.5	76.3	75.5

从表 1 和表 2 可以看出,本文方法的识别准确率和成功率与 Caffe 框架下的较好方法基本持平。但是本文方法的处理速度几乎可以达到 Caffe 框架下方法的两倍,而对于深度学习,处理速度是学习性能的重要保证,一方面可以提升最终的识别速度,另一方面可以使网络参数的训练更有效,从而构建出识别精度更高的深度学习模型。

6 结 论

TensorFlow 具有编程简单、优化算法集成度高、使用灵活的特性,能够有效提升建模、编程和分析的效率。使用 TensorFlow 平台设计了用于图像目标跟踪识别的深度学习流程及框架,介绍了 TensorFlow 的平台特性和在深度学习应用中的优势,并据此进行了相应的实验。

使用 TensorFlow 搭建的高效深度学习模型,能够便捷调整网络层数和网络节点数,同时其具有并行式运算结构,可调用 GPU 进行运算,因此运算速度较快,可以快速找到运行效率最高的网络结构。相比于其他软件平台下的深度学习模型,该模型能够保证识别的准确率和成功率,在处理速度上具有较大优势,可见基于 TensorFlow 设计深度学习模型具有较大的开发应用潜力。

参 考 文 献

- [1] Lu Hongtao, Zhang Qinchuan. Overview of application of depth convolutional neural network in computer vision[J]. Data Acquisition and Processing, 2016, 31(1): 1-17.
卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述[J]. 数据采集与处理, 2016, 31(1): 1-17.
- [2] Krizhevsky A, Sutskever I, Hinton G E. Image net classification with deep convolutional neural networks [C]. Advances in Neural Information Processing Systems(NIPS), 2012: 1097-1105.
- [3] Zhang Wei. Design and implementation of intelligent home system based on machine learning [D]. Jilin: Jilin University, 2016: 25-37.
张 炜. 基于机器学习的智能家居系统设计与实现[D]. 吉林: 吉林大学, 2016: 25-37.
- [4] Abadi M, Agarwal A, Barham P, *et al.* TensorFlow: large-scale machine learning on heterogeneous systems[J/OL]. [2017-03-01]. <https://arxiv.org/pdf/1603.04467.pdf>.
- [5] Jia Y Q, Shelhamer E, Donahue J, *et al.* Caffe: convolutional architecture for fast feature embedding[C]. Proceedings of the 22nd ACM International Conference on Multimedia, 2014: 675-678.
- [6] Li Yandong, Hao Zongbo, Lei Hang. A survey of convolutional neural networks [J]. Journal of Computer Applications, 2016, 36(9): 2508-2515, 2565.
李彦冬, 郝宗波, 雷 航. 卷积神经网络研究综述[J]. 计算机应用, 2016, 36(9): 2508-2515, 2565.
- [7] Sun Zhiyuan, Lu Chengxiang, Shi Zhongzhi, *et al.* Research and development of deep learning[J]. Computer Science, 2016, 43(2): 1-8.
孙志远, 鲁成祥, 史忠植, 等. 深度学习研究与进展[J]. 计算机科学, 2016, 43(2): 1-8.
- [8] Hu Zhengping, Chen Junling, Wang Meng. New development of classification model based on convolutional neural network in pattern recognition[J]. Journal of Yanshan University, 2015, 39(4): 283-291.
胡正平, 陈俊岭, 王 蒙. 卷积神经网络分类模型在模式识别中的新进展[J]. 燕山大学学报, 2015, 39(4): 283-291.

- [9] Hu Peng, Yang Weiqing, Fan Yong. Multi object tracking algorithm based on region convolution neural network[J]. Journal of Southwest University of Science and Technology, 2016, 31(1): 67-72.
胡 鹏, 杨伟清, 范 勇. 基于区域卷积神经网络的多目标跟踪算法[J]. 西南科技大学学报, 2016, 31(1): 67-72.
- [10] Wang Zhen, Gao Maoting. Design and implementation of image recognition algorithm based on convolutional neural network[J]. Modern Computer, 2015(20): 61-66.
王 振, 高茂庭. 基于卷积神经网络的图像识别算法设计与实现[J]. 现代计算机, 2015(20): 61-66.
- [11] Rui Ting, Fei Jianchao, Zhou You. Pedestrian detection based on depth convolutional neural network[J]. Computer Engineering and Application, 2016, 52(13): 162-166.
芮 挺, 费建超, 周 游, 等. 基于深度卷积神经网络的行人检测[J]. 计算机工程与应用, 2016, 52(13): 162-166.
- [12] Wang Haidong, Ma Xiaoshan, Yang Zhen, *et al.* Artificial neural network for atmospheric point spread function[J]. Laser & Optoelectronics Progress, 2016, 53(10): 102801.
王海东, 马晓珊, 杨 震, 等. 人工神经网络计算大气点扩展函数[J]. 激光与光电子学进展, 2016, 53(10): 102801.
- [13] Wang Hongqiao, Cai Yanning, Fu Guangyuan, *et al.* Image recognition and tracking based on image sequences for slow moving targets in the ground[J]. Laser & Optoelectronics Progress, 2016, 53(5): 051501.
汪洪桥, 蔡艳宁, 付光远, 等. 基于图像序列的地面慢动多目标识别与跟踪[J]. 激光与光电子学进展, 2016, 53(5): 051501.
- [14] He Pengcheng. An improved convolution neural network model and its application[D]. Dalian: Dalian University of Technology, 2016.
何鹏程. 改进的卷积神经网络模型及其应用研究[D]. 大连: 大连理工大学, 2016.
- [15] Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.
- [16] Zhang Jun, Li Xin. Handwritten character recognition based on TensorFlow platform[J]. Computer Knowledge and Technology, 2016, 12(16): 199-201.
张 俊, 李 鑫. TensorFlow 平台下的手写字字符识别[J]. 电脑知识与技术, 2016, 12(16): 199-201.
- [17] Yang Xiangnan. Research on target tracking algorithm based on convolution neural network and nested network[D]. Xiamen: Huaqiao University, 2016.
杨向南. 基于卷积神经网络和嵌套网络的目标跟踪算法研究[D]. 厦门: 华侨大学, 2016.
- [18] Zhang Zifu. Research and implementation of object recognition and tracking algorithm based on convolutional neural network[D]. Jilin: Jilin University, 2015.
张子夫. 基于卷积神经网络的目标识别跟踪算法研究与实现[D]. 吉林: 吉林大学, 2015.
- [19] Mao Ning, Yang Dedong, Yang Fucui, *et al.* Adaptive target tracking based on hierarchical convolution[J]. Laser & Optoelectronics Progress, 2016, 53(12): 051502.
毛 宁, 杨德东, 杨福才, 等. 基于分层卷积特征的自适应目标跟踪[J]. 激光与光电子学进展, 2016, 53(12): 051502.
- [20] Yilmaz A, Javed O, Shah M. Object tracking: a survey[J]. ACM Computing Surveys, 2006, 38(4): 1-45.
- [21] Hare S, Golodetz S, Saffari A, *et al.* Struck: structured output tracking with kernels[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(10): 2096-2109.
- [22] Kalal Z, Matas J, Mikolajczyk K. P-N learning: bootstrapping binary classifiers by structural constraints[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010: 49-56.
- [23] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409-1422.