

基于深度神经网络的扶梯异常行为检测

吉训生, 滕彬*

江南大学物联网工程学院, 江苏 无锡 214122

摘要 针对 Tiny YOLOv3 算法在扶梯异常行为检测时存在高漏检率和低准确率的问题, 提出一种改进的 Tiny YOLOv3 网络结构用于扶梯异常行为检测。利用 K -means++ 算法对数据集中的目标边框进行聚类, 根据聚类结果优化网络的先验框参数, 使训练网络在异常行为检测方面具有一定的针对性。利用多层深度可分离卷积提取深层次的语义信息, 加深特征提取的网络结构; 增加一个尺度用于低层语义信息的融合, 改进原有算法预测层的结构; 使用 GPU 进行多尺度训练, 得到最优的权重模型, 对扶梯异常行为进行检测。实验结果表明, 优化后的模型与 Tiny YOLOv3 相比, 平均漏检率减小了 22.8%, 检测精度提高了 3.4%, 检测速度是 YOLOv3 的 1.7 倍, 更好地兼顾了检测的精度和实时性。

关键词 图像处理; 异常行为检测; 自动扶梯; 深度可分离卷积; Tiny YOLOv3

中图分类号 TP391

文献标志码 A

doi: 10.3788/LOP57.061010

Detection of Abnormal Escalator Behavior Based on Deep Neural Network

Ji Xunsheng, Teng Bin*

School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China

Abstract Because of the high missing rate and low accuracy of Tiny YOLOv3 algorithm in the detection of abnormal escalator behavior, an improved Tiny YOLOv3 network structure is proposed for the detection of abnormal escalator behavior. K -means++ algorithm is used to cluster the target boundaries in the data set. The *a priori* parameters of the network are optimized according to the clustering results to make the training network have a certain pertinence in abnormal behavior detection. The network structure of feature extraction is deepened by using multi-layer deep separable convolution to extract deep semantic information. A scale is added to fuse low-level semantic information to improve the structure of the prediction layer of the original algorithm. Finally, the GPU is used for multi-scale training. The optimal weight model is obtained to detect the abnormal behavior of escalators. The experimental results show that compared with Tiny YOLOv3, the optimized model improves the missed detection rate by 22.8%, the detection accuracy by 3.4%, and the detection speed by 1.7 times. It gives better consideration to the accuracy and real-time performance of the detection.

Key words image processing; abnormal behavior detection; escalator; deep separable convolution; Tiny YOLOv3

OCIS codes 100.4996; 100.3008; 040.7290

1 引言

对自动扶梯上异常行为的有效实时检测, 可以预防危险事故的发生。国内外在异常行为检测方面的研究已经取得一定进展。沈铮等^[1]针对公交车内异常人群识别率较弱的问题, 引入多尺度滑动窗口

算法确定识别区域, 改进卷积神经网络算法进行异常检测; 虽然检测精度有所提高, 但无法满足实时性要求。Chen 等^[2]利用光流法得到运动幅值和方向, 通过距离分割法和最近邻域搜索法检测运动物体的发散中心, 根据发散中心检测异常事发点。除此之外, 还有很多利用光流、梯度方向直方图、角点等特

收稿日期: 2019-07-26; 修回日期: 2019-08-20; 录用日期: 2019-08-28

基金项目: 国家自然科学基金(61771223)、江苏省重点研发计划(BE2018334)

* E-mail: 2660087950@qq.com

征组合来检测异常行为的方法,但这类方法都是利用人工提取特征,鲁棒性差,无法满足准确实时的异常行为检测要求。

相比于传统方法,深度神经网络^[3]在目标检测^[4]方面显现出巨大的优越性。常用的深度神经网络方法包括建议区域法和无建议区域法,目前广泛应用的 YOLO^[5]网络模型在检测速率、精度和模型的泛化能力方面都比较优秀,但在小目标识别率方面还有待提高。

为提高算法检测扶梯异常行为的准确率,降低漏检率,本文提出一种改进的 Tiny YOLOv3 网络算法。利用 K-means++ 聚类算法^[6]代替 Tiny YOLOv3 中的 K-means 算法^[7],对数据集中的边框进行聚类分析,根据聚类结果更新优化先验框参数。采用多层深度可分离卷积替代 7 层卷积+池化网络提取特征,增加网络深度,提高检测精度,减小计算量。通过在 Tiny YOLOv3 模型基础上增加上采样层,并与低层特征融合^[8],改进 Tiny YOLOv3 预测结构,可以实现快速、精准的扶梯异常行为检测。

2 检测算法

2.1 Tiny YOLOv3 算法原理

Tiny YOLOv3 将输入图像划分为 $S \times S$ 的网格,在每个网格内预测 B 个边界框,检测 C 类目标,输出每类目标的边界框和置信度。置信度由每个网格中包含检测目标的概率和输出边界框的准确度共同确定,其中输出边界框的准确度定义为预测边界框与真实边界框的交并比 (IOU, I)^[9],其计算公式为

$$Z = P_{r(\text{object})} \times I_{pre}^{\text{truth}}, \quad (1)$$

式中: Z 为边界框的置信度; $P_{r(\text{object})}$ 为网格中含有某类目标的概率。

每个网格预测的类别置信度为

$$Z_i = P_{r(\text{Class}_i|\text{object})} \times P_{r(\text{object})} \times I_{pre}^{\text{truth}}, \quad (2)$$

式中: i 代表检测类别数。

设定阈值排除低于类别置信度的边界框,剩下的边界框利用非极大值抑制^[10] (NMS) 方法进行筛选,最终得到包含 x 、 y 、 w 、 h 四个参数的最优边界框,其中 (x, y) 为位置坐标, (w, h) 为宽和高,最终网络输出大小为 $S \times S \times (5 \times B + C)$ 。

Tiny YOLOv3 的损失函数 (L) 主要从边界框坐标误差、边界框的置信度误差及分类预测误差三个方面定义,

$$\begin{aligned} L = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \\ & (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{(C \in \text{classes})} (p_{i(C)} - \hat{p}_{i(C)})^2, \end{aligned} \quad (3)$$

式中: λ_{coord} 为权重系数; λ_{noobj} 为惩罚权重系数; x_i 和 y_i 为预测目标坐标; \hat{x}_i 、 \hat{y}_i 为实际目标坐标; w_i 和 h_i 为预测目标的宽和高; \hat{w}_i 和 \hat{h}_i 为实际目标的宽和高; C_i 和 \hat{C}_i 分别为预测目标与实际目标的第 i 个类别; $p_{i(C)}$ 和 $\hat{p}_{i(C)}$ 分别为预测目标和实际目标的类别率; S 为网格总数; B 为边界框总数;obj 代表含有目标;noobj 代表不含目标。 1_{ij}^{obj} 表示第 i 个网格第 j 个 anchor box 是否负责这个 object,负责是 1,否则为 0。 1_{ij}^{noobj} 表示第 i 个网格第 j 个 anchor box 不负责这个 object。

2.2 Tiny YOLOv3 的网络结构

Tiny YOLOv3 是 YOLOv3^[11] 的简化模型,融合了最新的特征金字塔^[12] 网络 (FPN^[13]) 和全卷积网络 (FCN),模型结构更简单,检测速度更快。Tiny YOLOv3 的网络采用一个 7 层卷积+池化网络提取特征,采用 13 pixel \times 13 pixel、26 pixel \times 26 pixel 的分辨率探测网络进行预测,网络结构如图 1 所示。

3 Tiny YOLOv3 算法改进

3.1 深度可分离卷积原理

深度可分离卷积^[14] 将标准卷积分解成深度卷积和 1×1 的逐点卷积。标准卷积结合所有输入得到新的输出。深度可分离卷积对每个单独层进行滤波后再结合。这种分解可有效减少计算量以及模型的大小,提高卷积的实时性。

输入的特征映射 F 尺寸为 $D_F \times D_F \times M$,采用的标准卷积 K 如图 2 所示,尺寸为 $D_K \times D_K \times M \times N$,输出的特征映射尺寸为 $D_G \times D_G \times M$ 。标准卷积的计算公式为

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m}, \quad (4)$$

式中: M 为输入通道数; N 为输出通道数,对应的计算量为 $D_K \times D_K \times M \times N \times D_F \times D_F$ 。

将标准卷积 $K (D_K \times D_K \times M \times N)$ 拆分为深度卷积和逐点卷积。深度卷积负责滤波,尺寸为 $D_K \times$

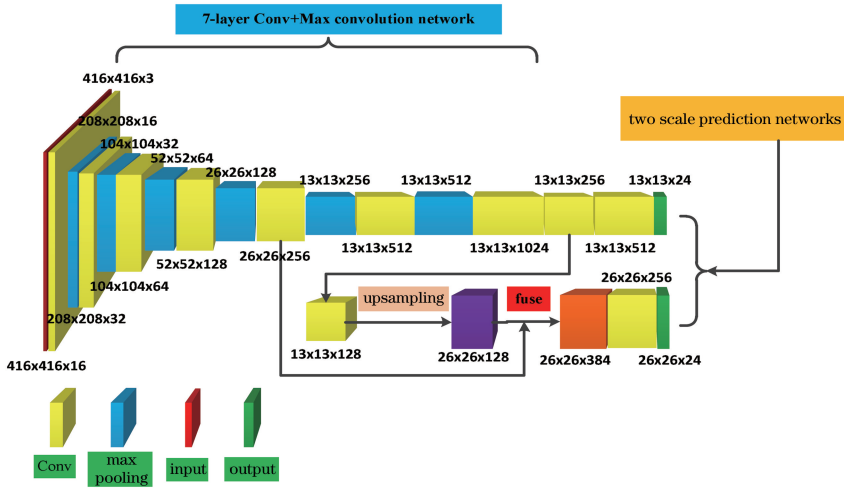


图 1 Tiny YOLOv3 网络结构

Fig. 1 Network structure of Tiny YOLOv3

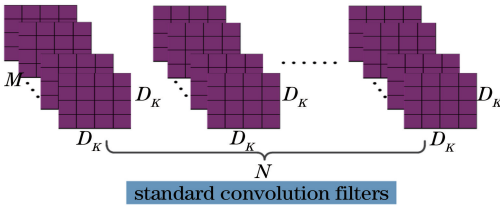


图 2 标准卷积结构

Fig. 2 Structure of standard convolution filters

$D_K \times 1 \times M$, 如图 3 所示, 输出特征映射尺寸为 $D_G \times D_G \times M$ 。深度卷积计算公式为

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m}, \quad (5)$$

式中: \hat{K} 为深度卷积; 第 m 个卷积核用在 F 的第 m

个通道, 输出在 \hat{G} 上的第 m 个通道。计算量为 $D_K \times D_K \times M \times D_F \times D_F$ 。

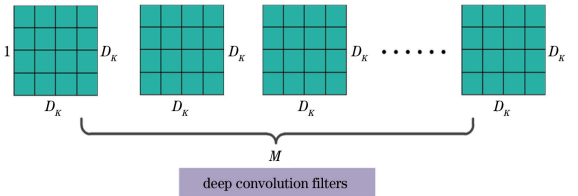


图 3 深度卷积结构

Fig. 3 Structure of deep convolution filters

逐点卷积负责转换通道, 如图 4 所示, 尺寸为 $1 \times 1 \times M \times N$, 输出特征映射尺寸为 $D_G \times D_G \times N$, 计算量为 $M \times N \times D_F \times D_F$ 。

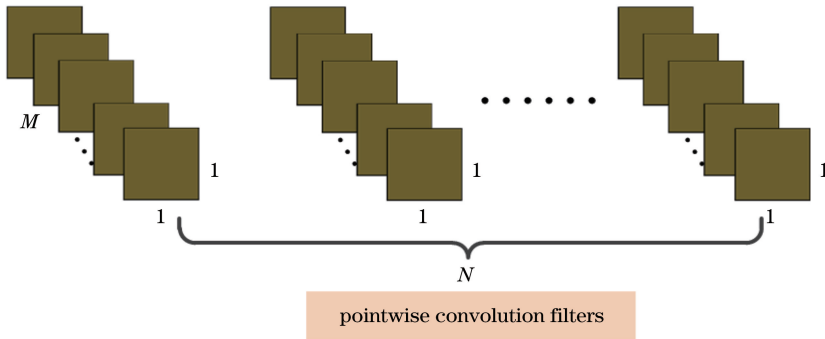


图 4 逐点卷积结构

Fig. 4 Structure of pointwise convolution filters

深度可分离卷积的计算量为 $D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$, 即深度卷积与逐点

卷积的和。深度可分离卷积与标准卷积计算量之比为

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = 1/N + 1/D_K^2. \quad (6)$$

卷积核 D_k 取 3 时,标准卷积计算量比深度可分离卷积大 8~9 倍。

3.2 Tiny YOLOv3 算法网络结构改进

针对 Tiny YOLOv3 模型检测精度低、小目标漏检率高的问题,提出利用 18 层深度可分离卷积网

络代替 7 层卷积+池化网络,同时增加一个尺度,以实现 3 尺度预测改进的 Tiny YOLOv3 算法。改进后的 Tiny YOLOv3 网络模型共 34 层,其中深度可分离卷积 18 层,标准卷积 8 层,上采样 2 层,特征 3 层,预测 3 层。具体结构如图 5 所示。

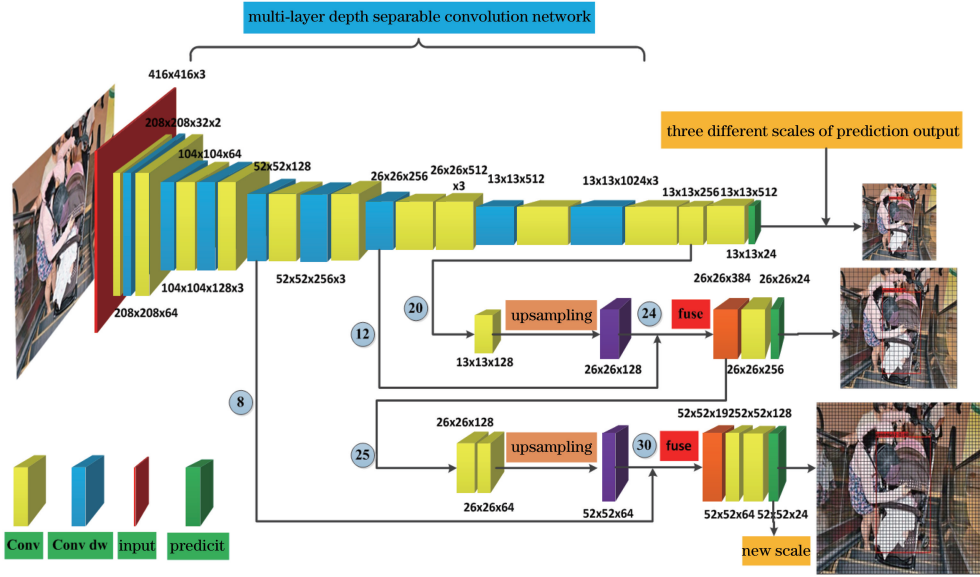


图 5 改进模型的网络结构

Fig. 5 Network structure of improved model

图 5 中网络输入图片尺寸为 $416 \times 416 \times 3$,使用步长为 2 的 3×3 标准卷积进行一次降维,输出向量为 $208 \times 208 \times 32$ 。交替使用 3×3 的深度卷积和 1×1 的逐点卷积进行异常行为的特征提取,减小特征图尺寸并增加通道数。为避免出现特征消失,只选用 18 层深度可分离卷积,并在卷积过程中引入残差网络进行特征融合。分别在 20 层、24 层和 30 层得到 $13 \times 13 \times 256$ 、 $26 \times 26 \times 384$ 和 $52 \times 52 \times 192$ 维的向量作为三个特征,卷积降维后送入预测层,并对三个尺度进行训练得到权重文件。对输入图片中的异常行为进行预测,输出标记的图片。

18 层深度可分离卷积结构如表 1 所示。表中 Conv dw 表示深度卷积,尺寸为 1×1 的 Conv 表示逐点卷积。

改进后的网络结构利用 9 个 3×3 的深度卷积提取异常行为的特征信息。其中 4 个步长为 2 的卷积核用以减小特征图的尺寸;9 个 1×1 的卷积用以变换通道数和提高特征表达能力。为充分利用特征信息,特征图尺寸每减小一半则相应通道数增加一倍。

改进后的算法流程如下:1) 将图像缩放为 3 通道长宽均为 416 的统一形式,作为整个网络的输入;

表 1 18 层深度可分离卷积结构

Table 1 18-layer deep separable convolution structure

Type/stride	Filter shape	Output
Conv dw/1	$3 \times 3 \times 32$	$208 \times 208 \times 32$
Conv/1	$1 \times 1 \times 32 \times 64$	$208 \times 208 \times 64$
Conv dw/2	$3 \times 3 \times 64$	$104 \times 104 \times 64$
Conv/1	$1 \times 1 \times 64 \times 128$	$104 \times 104 \times 128$
Conv dw/1	$3 \times 3 \times 128$	$104 \times 104 \times 128$
Conv/1	$1 \times 1 \times 128 \times 128$	$104 \times 104 \times 128$
Conv dw/2	$3 \times 3 \times 128$	$52 \times 52 \times 128$
Conv/1	$1 \times 1 \times 128 \times 256$	$52 \times 52 \times 256$
Conv dw/1	$3 \times 3 \times 256$	$52 \times 52 \times 256$
Conv/1	$1 \times 1 \times 256 \times 256$	$52 \times 52 \times 256$
Conv dw/2	$3 \times 3 \times 256$	$26 \times 26 \times 256$
Conv/1	$1 \times 1 \times 256 \times 512$	$26 \times 26 \times 512$
Conv dw/1	$3 \times 3 \times 512$	$26 \times 26 \times 512$
Conv/1	$1 \times 1 \times 512 \times 512$	$26 \times 26 \times 512$
Conv dw/2	$3 \times 3 \times 512$	$13 \times 13 \times 512$
Conv/1	$1 \times 1 \times 512 \times 1024$	$13 \times 13 \times 1024$
Conv dw/1	$3 \times 3 \times 1024$	$13 \times 13 \times 1024$
Conv/1	$1 \times 1 \times 1024 \times 1024$	$13 \times 13 \times 1024$

2) 利用 3×3 标准卷积进行一次卷积,随后堆积深度可分离卷积提取特征;3) 提取第 20 层输出向量作为第一个特征,并对该特征进行一次卷积和上采样;4) 将第 12 层输出量和第 24 层输出量进行特征

融合得到第二个特征,并对该特征进行卷积和上采样;5)将第8层输出量和第30层输出量进行特征融合得到第三个特征;6)将三个特征送入预测层分别生成 $13 \times 13 \times 24$ 、 $26 \times 26 \times 24$ 、 $52 \times 52 \times 24$ 的张量,对这些张量进行训练,得到最优的权重模型。

3.3 Tiny YOLOv3 先验框参数优化

Tiny YOLOv3 算法引入先验框参数。训练初始阶段,利用 K -means 聚类方法得到先验框,并通过网络不断学习修正先验框。先验框参数直接影响检测速度与精度, K -means 算法在初始聚类中心的选取效果不佳,可以利用 K -means++ 算法替代 K -means 算法分析样本标签聚类。

为避免预测框和先验框边框之间出现更多误差,选取两者的交并比替代原始算法中的欧氏距离作为目标函数,目标函数定义为

$$D_{(\text{box}, \text{centroid})} = 1 - I_{\text{centroid}}^{\text{box}}, \quad (7)$$

式中:box 为样本标签的预测框;centroid 为聚类中心。目标函数表示各个样本与聚类中心间的偏差。

4 实验与分析

4.1 数据集与标签

利用目标检测算法检测自动扶梯上的婴儿推车、人倒地和携带宠物三种异常行为的方法很少,且没有公用数据集。实验数据集来自超市和火车站扶梯出口的实际采样样本,包含迅达、三菱、西子奥的斯和沃克斯等多种不同厂家的扶梯,共 3600 张图片。其中每种异常行为各 1200 张图片,训练集和测试集按 4:1 进行分配。为避免训练模型出现过拟合的情况,对图片进行旋转、添加噪声和对对比度调暗等方法增加样本数量。利用 Labelimg 工具标注数据集,适应不同尺度图片的训练。

4.2 数据集标签聚类分析

由于 VOC2007^[15] 数据集中不包含与扶梯异常行为相关的数据,使用 Tiny YOLOv3 原始参数进行训练,会对训练时间和准确度造成一定影响,因此需要对异常行为标签重新进行聚类分析,以得到更具针对性的先验框参数。使用 K -means 算法和 K -means++ 算法对标签进行维度聚类分析,选择不同数目的先验框,两种算法的平均交并比(IOU)如图 6 所示。

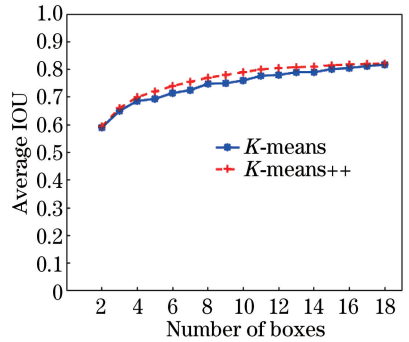


图 6 不同先验框数量的交并比

Fig. 6 IOU of different number of priori boxes

由图 6 可以看到,随着先验框数目不断增多,两种算法的平均交并比都在不断增大,但 K -means++ 算法的准确度更高,曲线更为平滑,趋势更为稳定,可在一定程度上减小聚类偏差。由表 2 可以看到,先验框的数目 k 值超过 9 时会出现大小较为相近的聚类结果,产生数据冗余。增加先验框的数量会导致模型的检测速度变慢,但综合考虑检测的准确性,实验最终选择 K -means++ 算法聚类生成的 9 个先验框,具体的参数为 (60, 59), (75, 105), (100, 147), (136, 81), (146, 205), (192, 293), (210, 133), (278, 385) 和 (280, 218)。

表 2 不同 k 值对应的先验框宽高

Table 2 Width and height of priori box corresponding to different k values

$k=7$	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$
(62, 61)	(59, 62)	(60, 59)	(55, 60)	(56, 60)	(54, 59)
(81, 117)	(83, 123)	(75, 105)	(75, 106)	(75, 106)	(73, 109)
(123, 172)	(125, 78)	(100, 147)	(100, 148)	(100, 148)	(102, 70)
(140, 83)	(131, 183)	(136, 81)	(100, 152)	(122, 72)	(102, 65)
(186, 255)	(185, 280)	(146, 205)	(146, 206)	(140, 209)	(135, 222)
(231, 146)	(206, 128)	(192, 293)	(178, 112)	(177, 120)	(151, 96)
(287, 323)	(271, 210)	(210, 133)	(192, 291)	(189, 312)	(183, 183)
	(278, 372)	(278, 385)	(242, 162)	(215, 216)	(188, 323)
		(280, 218)	(266, 398)	(267, 415)	(230, 131)
			(302, 247)	(264, 414)	(235, 246)
				(315, 258)	(287, 388)
					(323, 210)

4.3 实验环境与训练结果

实验使用的电脑配置为 Core (TM) i7-7700CPU, 3.6 GHz, 显卡为 Nvidia GeForce GTX 1080, 运行内存为 8G。软件平台为 Python3.6.3、Inter(R) win10 系统、CUDA8.0、CUDNN6.0、Tensorflow-gpu1.4.0 和 Opencv3.0。

迭代次数与平均损失函数的关系如图 7 所示。可以看到, 经过 8000 次迭代, 模型逐渐趋于拟合并最终收敛。调用训练得到的权重文件, 对携带宠物、婴儿推车和人倒地三种异常行为的图片分别进行测试。对比 Faster RCNN^[16]、SSD^[17]、YOLOv3 和 TinyYOLOv3 四种常用算法, 检测结果如

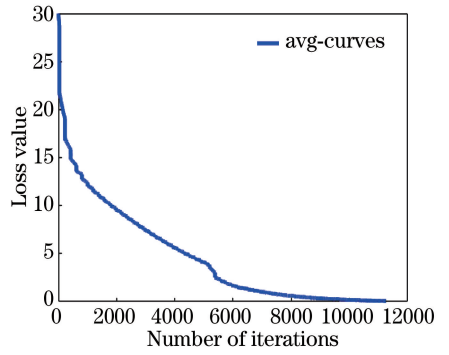


图 7 损失函数变化趋势

Fig. 7 Tendency of loss function

图 8~12 所示。



图 8 Faster RCNN 检测结果

Fig. 8 Detection results of Faster RCNN



图 9 SSD 检测结果

Fig. 9 Detection results of SSD



图 10 YOLOv3 检测结果

Fig. 10 Detection results of YOLOv3

4.4 评价指标

对于扶梯上的异常行为检测, 因为涉及到安全问题, 所以对漏检率和实时性的要求较高。为了分析改进的实验效果, 采用平均漏检率(A)、每秒检测帧数(FPS)、准确率(P)、召回率(R)及 F_1 辅助值作

为评价指标来评估模型, 分别定义为

$$A = \frac{F_n}{F_n + T_p}, \quad (8)$$

$$P = \frac{T_p}{T_p + F_p}, \quad (9)$$



图 11 Tiny YOLOv3 检测结果

Fig. 11 Detection results of Tiny YOLOv3



图 12 本文算法检测结果

Fig. 12 Detection results of our algorithm

$$R = \frac{T_p}{T_p + F_n}, \quad (10)$$

$$F_1 = \frac{2PR}{P + R}, \quad (11)$$

式中： T_p 是真实的正样本数量； F_p 是虚假的正样本数量； F_n 是虚假的负样本数量。

4.5 测试性能对比

训练过程中,为加快训练速度,防止过拟合,根据先验知识选用随机梯度下降(SGD)^[18]作为基础迭代器,设置初始学习率为 0.001,选用 Adam 优化器^[19],设置权重衰减系数为 0.005,迭代次数为 12000,选用经过 K-means++ 聚类得到的宽高作为先验框参数。

对比 Faster RCNN、SSD、YOLOv3、Tiny YOLOv3 四种常用算法在同一数据集上对扶梯异常行为的表现,五种算法的 P-R 曲线如图 13

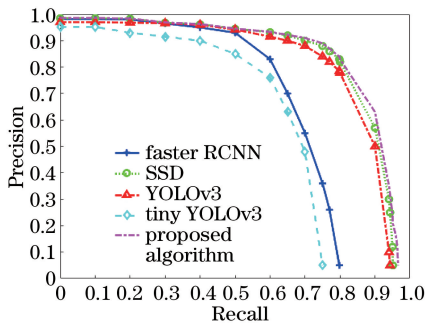


图 13 五种算法 P-R 曲线

Fig. 13 P-R curves of five algorithms

由图 13 可以看到,改进后的 Tiny YOLOv3 算法,在精确率和召回率方面与 Tiny YOLOv3 相比分别提高 3.4% 和 22.8%,与其他算法相比存在一定优势。不同模型的平均漏检率、FPS、 F_1 值如表 3 所示。

表 3 五种不同算法对异常目标的检测性能分析

Table 3 Analysis of abnormal target detection performance for five different algorithms

Detection algorithm	A / %	FPS / (frame · s ⁻¹)	F_1 / %
Faster RCNN	20.20	4.52	88.05
SSD	5.00	33.33	96.72
YOLOv3	5.50	25.64	95.63
Tiny YOLOv3	26.20	50.00	83.15
Proposed algorithm	3.40	43.48	97.60

由表 3 可以看到,改进后的 Tiny YOLOv3 算法在漏检率方面与 Faster RCNN、SSD、YOLOv3 和 Tiny YOLOv3 相比分别减小了 16.8%、1.6%、2.1% 和 22.8%;在 FPS 方面与 Faster RCNN、SSD、YOLOv3 相比分别增加了 38.96、10.15、17.84 frame/s,检测速度是 YOLOv3 的 1.7 倍; F_1 值比 Tiny YOLOv3 提高了 14.45%,系统总体性能相较于改进之前显著提升。

针对广泛使用的 YOLO 算法存在小目标识别率不高的问题,对改进后的 Tiny YOLOv3 算法、Tiny YOLOv3 算法、YOLOv3 算法重点做了不同大小目标的对比实验,结果如图 14 和图 15 所示。

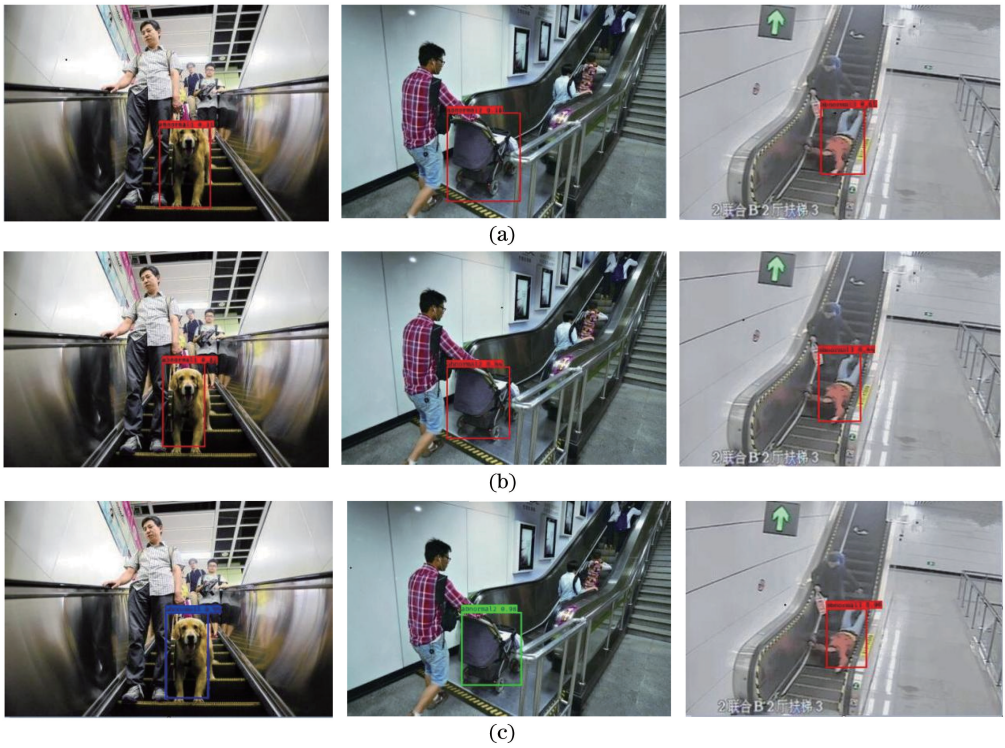


图 14 三种算法大目标检测结果。(a) YOLOv3;(b) Tiny YOLOv3;(c)本文算法

Fig. 14 Detection results of three algorithms for large targets. (a) YOLOv3; (b) Tiny YOLOv3; (c) proposed algorithm

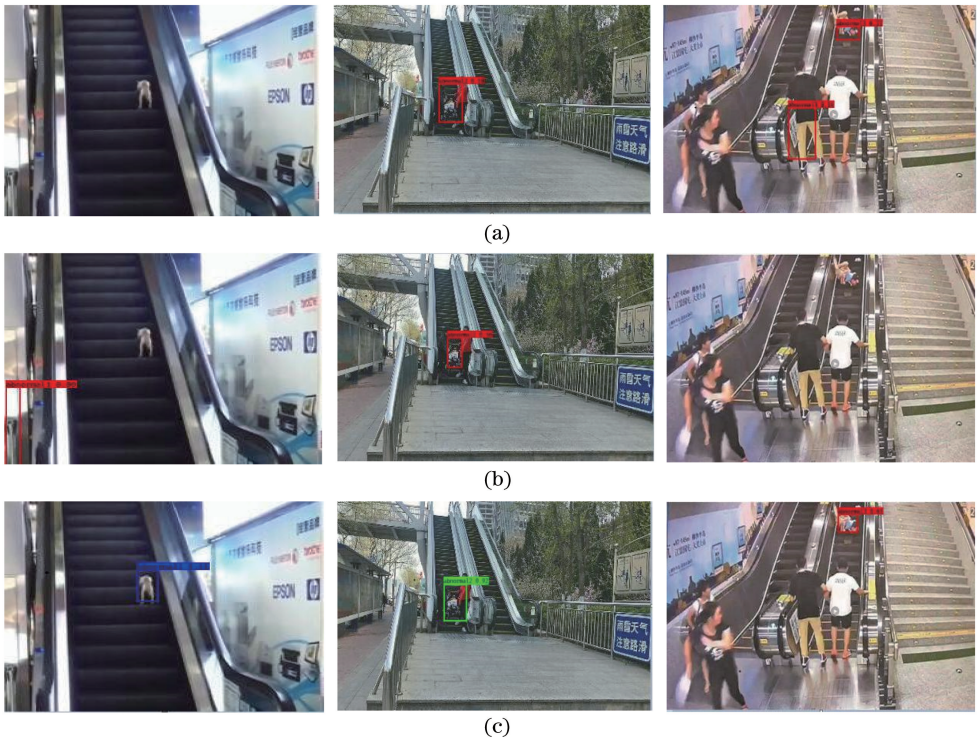


图 15 三种算法小目标检测结果。(a) YOLOv3;(b) Tiny YOLOv3;(c)本文算法

Fig. 15 Detection results of three algorithms for small targets. (a) YOLOv3; (b) Tiny YOLOv3; (c) proposed algorithm

由图 14 可以看到,三种算法都能较好地检测识别大目标的异常行为。由图 15 可以看到,对小目标

的异常行为检测,YOLOv3 算法和 Tiny YOLOv3 算法都存在部分误检和漏检的问题,而改进的算法

有效地减少了误检和漏检,使改进后的 Tiny YOLOv3 算法的小目标识别率有所提升。

综上所述,改进的 Tiny YOLOv3 算法在漏检率、准确率和小目标识别率方面均优于原算法,能达到实时检测的要求,并具有一定的鲁棒性。

5 结 论

提出了一种改进的 Tiny YOLOv3 算法用于扶梯异常行为检测。使用 K -means++ 算法对先验框参数进行优化,利用多层深度可分离卷积网络进行特征提取,利用残差网络^[20]增加一个尺度,融合低层特征以实现多尺度检测扶梯异常行为。根据实际数据的分析结果可以得到以下结论:1) 利用目标检测的深度网络算法可以实现对扶梯上的携带宠物、婴儿推车和老人倒地三种异常行为端到端的检测;2) 使用聚类误差更小的 K -means++ 算法替代 K -means 算法,可以对先验框参数进行优化,利用多层深度可分离卷积网络替代 Tiny YOLOv3 的 7 层卷积+池化网络提取特征,增加网络深度,利用残差网络增加一个尺度进行预测。

实验结果表明,改进的 Tiny YOLOv3 与原算法相比平均漏检率减小了 22.8%,检测精度增加了 3.4%,检测速率达到 43.48 frame/s,完全可以满足自动扶梯异常行为实时检测的需求。

参 考 文 献

- [1] Shen Z, Wu W. Abnormal crowd behavior detection on bus based on image processing algorithm [J]. Computer Engineering and Design, 2018, 39(1): 165-171.
沈铮, 吴薇. 基于图像处理的公交车内人群异常情况检测 [J]. 计算机工程与设计, 2018, 39(1): 165-171.
- [2] Chen C Y, Shao Y. Crowd escape behavior detection and localization based on divergent centers [J]. IEEE Sensors Journal, 2015, 15(4): 2431-2439.
- [3] Liu F, Shen T S, Ma X X, et al. Ship target recognition based on multi band depth neural network [J]. Optical Precision Engineering, 2017, 25(11): 2939-2946.
刘峰, 沈同圣, 马新星, 等. 基于多波段深度神经网络的舰船目标识别 [J]. 光学精密工程, 2017, 25(11): 2939-2946.
- [4] Wu S, Xu Y, Zhao D N. Survey of object detection based on deep convolutional network [J]. Pattern Recognition and Artificial Intelligence, 2018, 31(4):

335-346.

吴帅, 徐勇, 赵东宁. 基于深度卷积网络的目标检测综述 [J]. 模式识别与人工智能, 2018, 31(4): 335-346.

- [5] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 779-788.
- [6] W T, W W B, Y L, et al. Insulator Defect Detection Method for Lightweight YOLOv3 [J/OL]. [2019-7-18]. <http://kns.cnki.net/kcms/detail/31.1289.TP.20190307.1353.006.html>
吴涛, 王伟斌, 于力等. 轻量级 YOLOv3 的绝缘子缺陷检测方法 [J/OL]. [2019-7-18]. <http://kns.cnki.net/kcms/detail/31.1289.TP.20190307.1353.006.html>
- [7] Wei Y M, Quan J C, Hou Y Q Y. Aerial image location of unmanned aerial vehicle based on YOLOv2 [J]. Laser & Optoelectronics Progress, 2017, 54(11): 111002.
魏湧明, 全吉成, 侯宇青阳. 基于 YOLOv2 的无人机航拍图像定位研究 [J]. 激光与光电子学进展, 2017, 54(11): 111002.
- [8] Liu F, Shen T S, Ma X X. Convolutional neural network based multi-band ship target recognition with feature fusion [J]. Acta Optica Sinica, 2017, 37(10): 1015002.
刘峰, 沈同圣, 马新星. 特征融合的卷积神经网络多波段舰船目标识别 [J]. 光学学报, 2017, 37(10): 1015002.
- [9] Redmon J, Farhadi A. YOLO9000: better, faster, stronger [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 6517-6525.
- [10] Neubeck A, van Gool L. Efficient non-maximum suppression [C] // 18th International Conference on Pattern Recognition (ICPR06), August 20-24, 2006, Hong Kong, China. New York: IEEE, 2006: 9210072.
- [11] Redmon J, Farhadi A. Yolov3: an incremental improvement [J/OL]. (2018-04-08) [2019-07-25]. <https://arxiv.org/abs/1804.02767>.
- [12] Ren Z J, Lin S Z, Li D W, et al. Mask R-CNN object detection method based on improved feature pyramid [J]. Laser & Optoelectronics Progress,

- 2019, 56(4): 041502.
- 任之俊, 蔺素珍, 李大威, 等. 基于改进特征金字塔的 Mask R-CNN 目标检测方法[J]. 激光与光电子学进展, 2019, 56(4): 041502.
- [13] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 936 - 944.
- [14] Liu Q F, Zhang H L, Wang Y L. Real-time pixel-wise classification of agricultural images based on depth-wise separable convolution [J]. Scientia Agricultura Sinica, 2018, 51(19): 3673-3682.
- 刘庆飞, 张宏立, 王艳玲. 基于深度可分离卷积的实时农业图像逐像素分类研究[J]. 中国农业科学, 2018, 51(19): 3673-3682.
- [15] Everingham M, Winn J. The PASCAL visual object classes challenge 2007 (VOC2007) development kit [J]. International Journal of Computer Vision, 2006, 111(1): 98-136.
- [16] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C] // Advances in Neural Information Processing Systems, December 7-12, 2015, Montreal, Quebec, Canada. Canada: NIPS, 2015: 91-99.
- [17] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[M] // Leibe B, Matas J, Sebe N, et al. Computer vision-ECCV 2016. Lecture notes in computer science. Cham: Springer, 2016, 9905: 21-37.
- [18] Goyal P, Dollár P, Girshick R, et al. Accurate, large minibatch SGD: training ImageNet in 1 hour[J/OL]. (2018-04-30) [2019-07-25]. <https://arxiv.org/abs/1706.02677>.
- [19] Kingma D P, Ba J. Adam: a method for stochastic optimization [J/OL]. (2017-01-30) [2019-07-25]. <https://arxiv.org/abs/1412.6980>.
- [20] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 26-July 1, 2016, Las Vegas, Nevada. New York: IEEE, 2016: 770-778.