南京大學

研 究 生 毕 业 论 文

（申请硕士学位）

论　文　题　目　单类别实时目标检测算法与系统研究

作　者　姓　名　董学文

学科、专业名称　计算机科学与技术

研　究　方　向　人工智能

指　导　教　师　申富饶教授

2021 年 **5** 月 **29** 日

学　　号：**MF1833017**

论文答辩日期：**2021** 年 **05** 月 **24** 日

指 导 教 师：　　（签字）

Research on Real-time One-class Object Detection

by

Xuewen Dong

Supervised by

Furao Shen

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER
in
Computer Technology

Department of Computer Science and Technology
Nanjing University

May 29, 2021

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：单类别实时目标检测算法与系统研究

计算机科学与技术 专业 2018 级硕士生姓名：董学文

指导教师（姓名、职称）：申富饶教授

摘　要

目标检测作为目前发展较为成熟的计算机视觉技术，在监控定位，交通国防等实际应用领域发挥着越来越重要的作用。针对多类别目标检测问题的神经网络模型从最初的 R-CNN 系列，再到后来的 SSD、YOLO 等受到学术界的广泛关注，随着模型的发展其检测精度逐步增加。然而通过分析实际需求可以发现，很多场景下目标检测网络并不需要识别过多的类别，甚至只需要识别某一个类别的场景也屡见不鲜，相对而言实时性的需求反而更加重要。若直接采用上述多类别目标检测模型，会由于网络自身运算参数过多，导致上述网络不能在普通算力计算机上进行实时检测，为实际应用带来困难。

本文就实时性单类别目标检测问题出发，针对当前目标检测网络模型运行速度不达标的问题，提出两个优化模型参数的方法，并在上述优化算法的基础上搭建了一个室内定位系统。本文的主要工作包括：

1. 为获得能够快速检测的单类别目标检测模型，本文对特征提取网络基模型 mobilenet v3 提出了一种优化策略，修改了 SE(Squeeze-and-Excitation) 层的实现细节和 depthwise,SE,pointwise 层的连接顺序，结合 warmup 训练方式，最终得到了比原 mobilenet v3 网络参数量更少，精度更高的新网络模型 mobilenet v3 $small_2$。实验将其作为 SSD 网络的骨干模型，与其它网络模型相比有明显速度上的优势，为后续实现可实时性单类别目标检测任务打下基础。
2. 为进一步提高上述单类别目标检测模型的速度与精度，本文提出了一种针对目标类别的网络剪枝算法，该算法可以根据目标类别的变化适应性地选择不同的卷积核进行删减。同时，本文从可解释性的角度证明该剪枝算法的合理性。实验证明相比于剪枝前网络，该算法剪枝后的网络在目标类别检测的速度、精度上都具有一定优势。综合上述两个工作，本

文提出的单类别目标检测模型在普通算力的计算机上运行即可达到实时检测的要求。

3. 在上述两个工作的基础上，本文训练了一个针对单类别 (人物) 的目标检测模型，并使用此模型搭建了一个室内定位系统。该系统在普通算力的计算机上就能达到实时性检测的效果，同时具有较高的检测精度。

相关实验表明，本文提出的优化算法及剪枝策略可以有效的提升目标检测网络在单一类别检测上的运行速度和精度。同时，在此基础上实现的室内定位系统证明了上述算法的实用性及可行性。

关键词： 神经网络　单类别目标检测　网络剪枝　实时性

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Real-time One-class Object Detection

SPECIALIZATION: Computer Technology

POSTGRADUATE: Xuewen Dong

MENTOR: Furao Shen

Abstract

The object detection as a relatively mature computer vision technology, is playing an increasingly important role in practical applications such as surveillance and localization, traffic and defense. The neural network models for multi-category object detection problem, which develop from the initial RCNN series to the later SSD and YOLO, have received a lot of academic attention and their detection accuracy is gradually increasing. However, the neural networks are not required to detect too many categories in many practical scenarios, and scenarios where only one category needs to be identified are common. Thus, the need for real-time detection is more important. If the above-mentioned multi-category object detection models are adopted directly, the detection cannot be completed in real time on ordinary arithmetic computers for the networks have too many operational parameters, which brings difficulties for practical applications.

For the problem of real-time one-class object detection, we propose two methods to reduce current model parameters to speed up their computation, and build an indoor localization system based on the above optimization algorithms. The main works of this paper are as follows:

1. For achieving real-time one-class object detection, we propose an optimization strategy for the feature extraction base model mobilenet v3, modifying the implementation details of the SE (SqueezeandExcitation) layers and the connection sequence of depthwise, SE, and pointwise layers. We finally obtain a new network model mobilenet v3 small_2 with less number of parameters and higher accuracy than the original mobilenet v3 combining with the warmup training method. It

is used as the backbone model of the SSD network in following experiments for the following real-time one-class object detection, which has obvious speed advantage compared with other network models.

2. To further improve the precision and speed of the above model, we propose a network pruning algorithm for one-class object detection, which adaptively selects different convolutional kernels for culling according to the object categories. Also, we analyze the pruning strategy from the viewpoint of interpretability. It is demonstrated in experiments that the pruned network of this algorithm has certain advantages in terms of speed and accuracy of object category detection compared with the pre-pruned network.
3. Based on the above two methods, an one-class (person) object detection model is trained and an indoor localization system is built on it. The system can achieve real-time detection and high detection accuracy on a computer with ordinary computing power.

Relevant experiments show that the optimization algorithm and pruning strategy proposed in this paper can effectively improve the computation speed and accuracy of object detection network for one-class detection. Meanwhile, the indoor localization system proves the practicality and feasibility of the above algorithms.

keywords: neural networks one-class object detection network pruning real-time

目　　录

插图清单

附表清单

第一章　绪论

1.1　研究背景及意义

1.1.1　研究背景

目标检测 (Object Detection) 指的是输入一幅图片或一段视频，网络模型可以自动检索图片或视频中它感兴趣的类别，例如人、花、杯子、桌子等。需要注意的是目标检测的目的是检测模型感兴趣的类别中所有物体，而非类别下某个特定的物体。目标检测现已应用在广泛的生活场景中，包括国防监控，交通指挥，商场观测等。

现如今主流目标检测模型（包括 R-CNN 系列、MobileNet 系列、YOLO 系列等等）都具备识别多种类别的能力，可识别类别个数一般为 20 – 200 不等。但在实际生活应用中，往往所需要检测的类别个数较少，有些场合甚至只需要检测某一种类别。研究者将只需要检测某一种类别的目标检测问题定义为单类别目标检测问题 (One-class Object Detection)。针对单类别目标检测任务，一个较为直接的解决方案是利用上述多目标检测模型来解决单类别检测任务。但是，多类别目标检测模型存在检测速度慢，实时性差等问题。

多目标检测模型随着网络结构的不同，模型的检测速度也各不相同，一般而言在普通 cpu 上每秒能够做检测 5 – 15 张图片。由于人眼每秒可分辨帧数约为 24张/秒[1]，直接使用上述检测模型进行实时性检测会出现卡顿的情况，因此许多场景下的实时性检测一般由检测模型得到的结果结合追踪算法完成，即在检测的间隔中使用追踪算法替代检测的功能 [2]。该方法通常能够满足实时检测的要求，但追踪算法对同一场景下多个目标的追踪表现较差。更好的方法是提出一个针对单类别目标检测任务的优化方案。本文的主要思想是通过结构优化及剪枝策略减少现有目标检测模型的参数量使其既能够满足实时性检测的速度需求，同时与优化前的多目标检测网络相比在目标类别上的检测精度相近甚至更优。

因此本文着眼于单类别目标检测中网络模型参数数量问题。目前目标检测任务大多需要骨干网络和头部网络联合解决，其中在骨干网络方面 mobilenet

v3 网络具有参数量少，提取效果佳的优点，适合作为最终网络模型的骨干网络基模型。为了进一步提升最终模型的计算效率，本文对现有特征提取网络 mobilenet v3 进行部分结构的调整，减少网络参数。同时提出了一种针对单类别目标检测任务通用的网络剪枝算法，使网络在检测速度方面得到了进一步的提升。而且后续的实验和分析表明，通过该剪枝算法修改过的网络模型检测精度与原网络相当，甚至略高。最后该剪枝算法能从神经网络可解释性方面加以诠释，使其无论从理论证明亦或视觉观感的角度都可以证明文中所述算法的优越性。

1.1.2 应用方向

单类别目标检测在实际的生产生活中有很重要的应用，涵盖了商场监控、国防安全、科教文卫等多个方面。下面分别从实际生活和学术研究两大方面举例说明单类别目标检测的具体应用。

首先是在生活检测方面。在一些特定的场景中，往往我们仅仅关注某一个类别的位置及活动，这种情况包括在教室中查看学生的行为、或是商场中检测买家与柜员的互动、再者是可以在工厂或是国防安全方面随时随地检查是否有人进入了危险区域等等。图 1-1是一个在商业街监测人物活动的例子说明。在这些场景下，我们并不关注其他事物本身的位置，而且检测多种类别反而会对目标类别的检测造成影响。同时在这些场景下往往对目标检测的速度有较高要求，因此在这些场景下速度更快的单一类别目标检测模型有较大优势。在这些情况下，单类别目标检测不但可以完美的匹配需要检测的类别事物，更重要的是，相比于多类别目标检测模型而言，仅仅针对单类别的目标检测模型在速度上有明显的提升。

其次单类别目标检测的作用在于替代多类别目标检测模型。事实上，当需要检测的类别较少时，用多路单类别目标检测器检测图片或视频场景的效率要高于单个多类别目标检测模型。试想这样一个场景，现有一组单类别目标检测模型 $S\{A\}$，每个模型 A_i 单独检测一种动物类别 (狗、猫、等等)，检测一幅图片需要约 35ms。还有一个通用目标检测模型 B，它自身能够检测 $S\{A\}$ 中的所有类别，但是每次检测所需的时长约 40ms，如图 1-2所示。在所有目标检测器精度几乎相当的情况下，同时运行 $S\{A\}$ 的检测模型之后将结果整合的速度显然要快于单独使用检测模型 B 的速度。至于如何利用这一组检测器 $S\{A\}$ 输出一个最终结果，无论是利用 one-hot 独热码或是 softmax 得到每个类别的归一化概

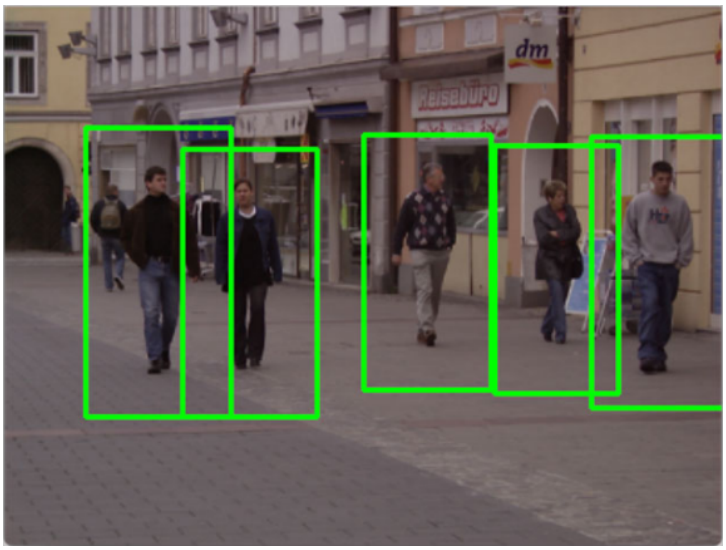


图 1-1: 单类别目标检测模型在商业街的表现

率，都是一个可靠的答案。

图 1-2: 多个单类别目标检测模型与一个多类别目标检测模型的运行时间对比

1.2 研究现状

目前主流界公认的目标检测问题在 2001 年提出，其由 Viola Jones 算法代表的人脸检测问题引申而来 [3]。这是最早的目标检测模型，其实也是一个特殊的单类别目标检测模型。随着后续计算机视觉方面的发展，目标检测问题逐渐不再局限于对单个类别的检测，而是扩展成为了现今的多元目标检测问题。

对于解决单类别目标检测问题，一种直观的思路是直接使用多类别目标检测模型进行单类别目标的检测。但是这样会导致场景中出现一些不关注的类别被模型捕捉到，造成对目标类别检测效果的干扰。如果对原模型进行微调，将

其改成仅识别单类别的目标检测模型，又会因为模型原本可识别类别较多，现在仅仅检测单一的类别而造成参数冗余，从而导致检测速度过慢。目前来看，现阶段的目标检测模型无法达到在普通算力计算机上进行实时性检测的要求，因此本文希望通过对现有网络模型进行针对单个类别的参数优化，以满足在普通算力计算机上能够实时性运行的要求。现阶段目标检测问题的处理方法可主要分为两种：基于传统机器学习的目标检测方法和基于神经网络的目标检测方法。

较为早期的目标检测方法常通过提取人脸或目标共同属性、计算相关积分图像与矩形特征，同时引入 AdaBoost[4] 辅助删除冗余的矩形特征进行检测 (Viola Jones 算法)。也有使用方向梯度直方图 (Histogram of Oriented Gradient,HOG) 作为目标检测的特征描述算子，辅以色彩归一化、区间归一化等方法进行行人检测的方式 [5]。后续工作在 HOG 的基础上引申出 DPM 模型，在当时取得了很好的效果 [6]。这些算法在单幅图片上的检测速度较快，但都有检测效果一般、算法本身不稳定、疏漏小目标的缺点，在现今的主流目标检测界用处较少。

基于神经网络的目标检测算法最早是 Ross Girshick 在 2014 年提出的 R-CNN 算法 [7]。这种新型的目标检测方式在当时引起了广泛关注。而后越来越多的研究者开始专注于用神经网络方法解决目标检测问题，目标检测领域的发展也愈加迅速。此类方法能够达到传统机器学习算法难以实现的精度，在速度方面也逐渐逼近传统机器学习算法，俨然成为现今目标检测的主流手段。其中以模型小、速度快见长的 SSD 模型在移动端目标检测上取得了成功应用 [8]。本文将在 mobilenet-ssd 网络模型的基础上进行改进，使之在单类检测上有速度和精度的提高。同时调整网络结构使之变得更加合理，从而解决上述困难。

1.3 研究内容

针对单类别目标检测问题，研究需要使目标检测网络突破现有模型的速度瓶颈。这些策略将在现有的特征提取模型基础上进行结构优化，减少模型的参数量。同时提出了一个针对单类别目标检测问题的剪枝算法，并在这些工作的基础上实现了一个基于神经网络的室内定位系统，该系统可以在普通机器上执行目标检测任务并近乎达到实时性的要求。其中的主要研究包括以下几点：

1. 面对单类别检测问题的 mobilenet v3 优化。本文提出了一种基于 mobilenet

v3 网络模型的优化策略，修改了 depthwise、SE 模块、和 pointwise 的实现和它们之间的连接顺序，并加入 BatchNorm 模块使网络表现更加稳定。与 mobilenet v3 small[9] 相比，优化后的 mobilenet v3 small 模型参数量减少约 7%，且在图像分类任务上的表现更好。同时在实验中使用 warmup 训练方法，使模型的特征提取能力进一步提升，最终获得 mobilenet v3 $small_2$ 网络模型，成为下一步工作的基础。

2. 面向单类别目标检测问题的网络剪枝算法。本文提出了一种针对单类别目标检测的网络剪枝方式，该方法减少了目标检测模型中骨干网络最后一个卷积层的卷积核数量，适用于目前所有骨干——头部类型的目标检测网络。利用这种方式可以使得剪枝后的网络具有更快的检测速度，同时与剪枝前网络相比，提升了目标类别上的检测精度。本文还从可解释性和可视化的角度出发讨论了该剪枝策略的合理性。实验表明在该剪枝算法的作用下，新的网络模型无论是作为目标类别图像分类还是作为目标检测骨干网络都有速度和精度上的提升。
3. 基于单类别目标检测模型的室内定位系统。本文在上述两个工作的基础上，搭建了一个只识别人物类别的室内定位系统，该系统可以检测室内人的位置，且能满足室内监测的精度需求，更重要的是，该系统可以在普通算力的计算机上实时性运行。

1.4 论文结构

本文主要探究面向单类别目标检测问题的优化方式，并对现有的神经网络目标检测模型进行优化，提升了原网络的运行速度和检测精度。同时基于该单类别目标检测器搭建了一个室内定位系统。

全文一共分为六章：第一章为绪论，主要介绍单类别目标检测的问题和实用价值；第二章介绍了当前主流目标检测界的相关研究工作；第三章介绍了作者针对 mobilenet v3 网络模型的优化，使其在减少参数量的同时达到更优的效果；第四章主要介绍作者提出的基于单类别目标检测的剪枝方法，并在可解释性和可视化的角度直观验证该剪枝方式的合理性；第五章介绍了基于上面工作所搭建的室内定位系统；最后第六章总结全文，并对未来工作进行展望。

第二章　相关工作

目标检测问题可以追溯到 2001 年，由于其应用场景多、影响力大，至今 20 年间众多研究者们参与了相关研究，并取得了很大的进展。尤其是近十年来深度学习算法的崛起，使研究者们在计算机视觉 [10]、语音识别 [11]、自然语言处理 [12] 等领域都取得了重大突破。目前主流的目标检测方法也大多为深度神经网络算法，不同的网络模型之间检测方式各不相同，速度与精度之间的权衡各有偏重。

虽然最初提出的人脸检测模型针对的是单类别目标检测问题，但是由于研究者们更关注多类别目标检测问题，目前关于单类别目标检测的研究很少，许多场景下研究者们采用多类别目标检测模型直接代替单类别目标检测模型去解决单类目标检测问题。但实际上直接使用多分类目标检测模型会不可避免地造成网络参数冗余，增大单个类别目标检测所需的时间消耗，无法满足在普通算力计算机上达到实时性单类目标检测的要求。

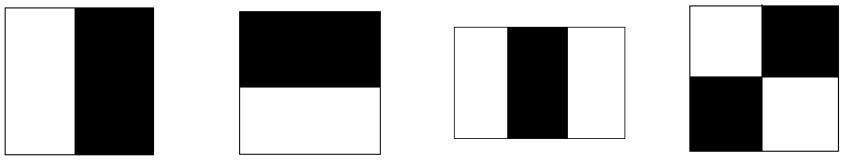
因此本文根据目标检测中神经网络模型的结构特性和卷积核的运算特性，将现有的目标检测神经网络进行针对单类别目标检测任务的改进，使结构更改后的网络模型可以达到在不损失原网络检测精度的前提下提高检测速度的效果。为介绍这些年来主流目标检测模型的基本原理作为后续工作的基础，本章主要介绍出现较早的基于传统机器学习解决目标检测问题的方法以及近几年来流行的基于人工神经网络解决目标检测问题的方法。最后会对不同的方法进行对比，分析几类主流检测方法的优势和缺点，解释了人工神经网络用于目标检测的优势，最后在其基础上提出本文的创新点。

2.1　基于传统机器学习的目标检测方法

本节主要介绍在神经网络算法出现之前目标检测问题的一般解决办法。在深度学习兴起之前，人们已经尝试通过用不同大小的滑动窗口结合图像特征匹配来寻找目标类别。但这类方法通常因为计算开销过大，导致算法运行速度较慢，因此早期算法大都尝试如何快速提取特征以及迅速筛选掉不满足条件的窗口。

2.1.1 Viola-Jones 算法

早在 2001 年 Viola 和 Jones 两人就提出了基于 Haar 特征提取和 AdaBoost 级联的特征检测算法 [3]。虽然该算法当时为人脸检测算法，但可以将其视为一个特殊的单类别目标检测算法，同时该算法也是第一个被正式提出的目标检测算法。该算法采用了滑动窗口结合特征匹配来进行人脸检测，使用 Haar 特征来进行人脸匹配。Haar 特征主要反应了图像灰度的变化情况，其一般采用预先设定好的模版来匹配人脸，其中一部分模版如图 2-1所示。



(a) 模版 A (b) 模版 B (c) 模版 C (d) 模版 D

图 2-1: Haar 特征模版示例

特征数值的计算方式为 $\sum_{\text{白色区域}} - \sum_{\text{黑色区域}}$。计算时要求不同颜色的像素个数一致，所以对于图 2-1(c)而言是 $\sum_{\text{白色区域}} -2 * \left(\sum_{\text{黑色区域}} \right)$。作者通过模版的匹配找到图像中的相应组件，进而根据人脸特征（比如眼窝部分颜色比鼻梁更深，嘴附近比脸颊更深等等）拼凑出图像中的人脸，如图 2-2所示 [3]。

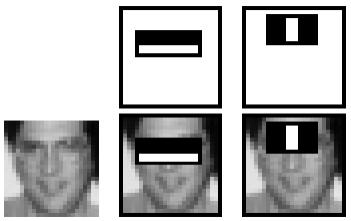


图 2-2: 图像中人脸匹配示例

这种人脸检测算法在当时并非首次提出，但类似算法运行速度很慢，原因是因为模版的大小和候选框的大小都可以随意调整，位置也不能确定。甚至一个很小的图片 (24×24 像素图) 就有十几万种可能情况 [13]。因此 Viola-Jones 采用动态规划的方式 (积分图) 来统计一个区域的像素之和。首先构建二维数组 $ii(x, y)$ 有：

$$ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y') \tag{2-1}$$

其中 $i(x', y')$ 表示该点的像素值。根据动态规划相关知识可知，该二维数组 $ii(x, y)$ 可以在 $O(m + n)$ 求得，其中 m 和 n 分别为图像的宽高。显然在有了 $ii(x, y)$ 矩阵的基础上，任何一个矩形的像素值加和都可以在极短时间内算取，极大提升了匹配的效率。最后判断策略引入 AdaBoost 算法，通过将弱分类器组合形成一个强分类器，将强分类器级联，若第一层强分类器判断该区域不为人脸则直接舍弃，当所有强分类器均判定该区域为人脸时输出。

2.1.2 HOG 结合 SVM 进行目标检测

方向梯度直方图 (Histogram of Oriented Gradient, HOG) 结合 SVM 做目标检测的算法最早在 2005 年提出 [5]，当时主要用于行人的检测。同样的，行人检测也可视为单类目标检测的一种特化情况。该算法使用了图像的梯度边缘特征，因此为了尽量排除光线、阴影的影响，需要先进行图像校正，一般采用 Gamma 校正方式 [14]。校正过的图像对比度会发生变化，后续的标准化操作只在检测颜色变化较小的物体时采用。之后需要对图片进行去噪平滑，一般采用高斯平滑。

将图像预处理后接下来需要求得每个像素点处的梯度，梯度包括大小和方向，因此需要先计算像素点在水平和垂直方向的梯度，如式2-2和式2-3所示。其中，$G_x(x, y)$ 为图片在 (x, y) 处的水平梯度，$G_y(x, y)$ 为图片在 (x, y) 处的垂直梯度，$I(x, y)$ 为图片在 (x, y) 处的像素值。

$$G_x(x, y) = I(x + 1, y) - I(x - 1, y) \tag{2-2}$$

$$G_y(x, y) = I(x, y + 1) - I(x, y - 1) \tag{2-3}$$

根据上述信息可以求出该像素点的梯度幅值 $G(x, y)$ 和梯度方向 α，如式2-4和式2-5所示。

$$G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \tag{2-4}$$

$$\alpha = \arctan\frac{G_y(x, y)}{G_x(x, y)} \tag{2-5}$$

在求出每个像素点的梯度值和方向后，将相邻的像素点划分为细胞单元 (cell)，然后统计该细胞单元内的像素点直方图。在统计所有细胞单元直方图后需要进行归一化，具体方法是将个细胞单元整合成一个大的块 (block)，求出块

内的对比度来调整细胞单元，使得不同块之间的直方图相差不大。块与块之间可以重叠，虽然有一部分的计算冗余，但是实验发现这样做可以显著提高检测精度 [5]。最终将得到所有块内细胞单元生成的直方图向量。

提取过特征后将得到的特征输入分类器。由于梯度方向的区间取值不定，因此往往把梯度方向分为几个区间来表示。在检测过程中会将一个块内多细胞单元的直方图串联形成高维特征，同时块会滑动形成新的特征，一般而言步长为一个细胞单元大小。最终将串联得到的高维特征输入 SVM 用于训练和最终测试 [15]，达到检测的目的。

2.1.3 基于 DPM 的目标检测算法

HOG 在目标检测领域取得了重大的突破，可它也有自身的显著问题：对于遮挡/形变物体识别率很低。这是由于边缘梯度检测无法检测遮挡物后面的物体。在 HOG 的基础上，Felzenszwalb 在 2008 年提出了 DPM(Deformable Part Model) 目标检测算法 [6]。这种目标检测方式在当时引起了广泛关注，更重要的是，里面很多细节的处理影响了后续基于神经网络的目标检测方法。

DPM 目标检测算法一定程度上参照了上文的 HOG 算法，但优化程度更高。在特征提取方面，DPM 算法去掉了 HOG 中的块结构，只用细胞单元作为基本的检测单元，同时只选取该细胞单元对角线邻域区域进行归一化，这样会得到与 HOG 十分相近的结果并且减小计算量。另外，DPM 引入了有符号和无符号梯度相结合的策略，这样保证了多类目标都能被有效地检测到。

除此之外，DPM 最重要的突破在于多个根滤波器和部件滤波器。考虑到相同类型的物体根据视角和姿势的不同也会出现不一样的形态，比如一个站立的成人宽高比可能是 1:2.5，躺下就可能变成 8:1。因此 DPM 引入多个根滤波器进行训练，具体做法是将数据集中的目标单元长宽比例进行分类，根据不同类别训练不同的根滤波器，而后将其整合。同时由于物体会发生形变，还引入了多个部件滤波器，用于检测目标物体的多个部件。不同部件之间组合可以用于判断识别到了相关目标，检测的同时也会判断部件之间的距离不会过大。同时设定部件滤波器训练图片的分辨率为根滤波器训练图片分辨率的两倍，以便分辨更清晰的细节，更精准的提取特征。模型依然使用了支持向量机 (SVM) 作为基模型。

DPM 进行检测的过程比较简单，首先对输入图像进行特征提取，来得到原图的特征图和 2 倍分辨率下的特征图；而后分别在两个特征图下进行根滤波器

和部件滤波器的得分计算；最后去掉部件之间搭配过远的组合，同时合并可能的组合得分，求出总得分并输出合并框和分数。具体计算公式如式2-6：

$$\text{score}(x_0, y_0, l_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b \tag{2-6}$$

由于先前已经训练好了两类模型的参数，后面只需在图像特征图上求取点积计算得分即可。式2-6中的得分包括三个部分：根滤波器的得分 $R_{0,l_0}(x_0, y_0)$，部件滤波器的得分 $\sum_{i=1}^{n} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i)$ 以及偏移量 b。

2.1.4 小结

在目标检测问题提出伊始，由于算法复杂度的限制，研究者们提出的模型通常只针对单类别或少量类别进行检测。他们倾向于用滑动窗口结合特征提取的方式去解决该问题。而以往特征提取计算量过大，因此 Viola、Jones 等人使用积分图的形式极大限度地缓解了该问题。在检测方面也引入了级联的方式来达成速度的提升。这种方法在当时比其他检测方式要快上百倍，开创了目标检测可行性的先河。该算法固然有它的自身缺陷 (性能不稳定，性能较差，小物体难以识别)，但仍是目标检测领域最初的标志性算法。

而后研究者们在其基础上引入了边缘梯度信息，更快也更精准的提取图像边缘特征结合支持向量机进行检测识别。在 HOG 算法的基础上，Felzenszwalb 等人提出了 DPM 算法，该算法特征提取的准确性直到 2013 年才被后续算法超越 [16]。DPM 算法至今也有着广泛的应用，它的一些检测思想已经根植于目标检测主流深度学习算法之中。虽然现在来看它的精度已经无法与深度学习算法相比拟，但它在传统机器学习领域的地位无可动摇。

2.2 基于人工神经网络的目标检测方法

基于人工神经网络的目标检测方法大都为多类别目标检测算法。虽然目前运算速度最快的目标检测算法已经超过了传统机器学习算法的检测速度，但是其所需要计算的参数量仍然很大，无法直接在普通算力计算机上实时运行。

最早提出的基于神经网络的目标检测方法是 R-CNN[7]。随后，各大基于深度学习的目标检测方法以井喷的形式发展。早期深度学习的目标检测方法主要分为两类：一步检测 (one-stage) 和两步检测 (two-stage)。其中，两步检测简

而言之就是先得到候选边界框，而后校正候选框同时将框内物体分类；一步检测是直接在候选框所覆盖范围内校正检测。

上述深度学习方法无论是一步或两步检测都是基于锚点 (Anchor)，而锚点和候选框本身的选取对整个网络检测速度和精度的影响很大，某种程度上制约了相关算法的性能。因此近年来基于 Anchor-free 的目标检测方法开始逐渐流行。Anchor-free 算法去除了锚点，克服了这部分缺陷，不过也带来了一些后续问题隐患。接下来介绍现今主流的深度学习目标检测算法，而后将其加以对比总结。

2.2.1 基于 two-stage 的目标检测算法

基于 two-stage 的目标检测算法简而言之是先在图片上生成候选区域 (Region Proposal)，而后在候选区域上使用分类器进行分类的算法，由于它具有明显的二阶段性，因此该类算法被统称为 two-stage。接下来介绍一些经典的 two-stage 算法。

最早提出应用卷积神经网络解决目标检测的模型是 2014 年的 R-CNN[7]，该算法主要采用启发式的选择性搜索 (Selective Search) 算法来对图片进行候选框的生成，每幅图片大致生成 2000 个候选框。而后分别在候选框所选取的范围内运行卷积神经网络 (CNN) 进行特征提取，最后将特征提取的结果输入 SVM 进行分类，同时还会输入到一个线性回归器中，便于校正候选框以便更好的检测目标物体。

在上述工作的基础上，学者们设计了 Fast R-CNN 网络模型。Fast R-CNN 先对图像进行特征提取而后用启发式算法生成候选框。这样做的好处是原本需要在所有候选框中运行的卷积神经网络模型现在只需要在图片输入初始运行一次，大大提升了检测效率 [17]。同时放弃 SVM 的分类方式引入 softmax，这种输出预测概率的方式后续也成为了目标检测算法的主流。

后续提出的 Faster R-CNN 利用 RPN 网络代替之前的启发式算法来获取候选框，这种方式较之前的选择搜索算法快很多 [18]。同时引入锚点策略来帮助选取目标检测的候选框 [19]。具体而言，RPN 网络先在特征图上利用 3×3 大小的窗口进行滑动，同时在滑动过程中，RPN 以中间的像素点为中心，将预设好的多个比例套用在特征图上得到锚点框 (anchor box)，最后在这些锚点框上进行预测，得到边界框包含物体的概率，将包含目标物体概率较高的锚点框坐标输入到下一分类网络，RPN 具体行为如图2–3所示。

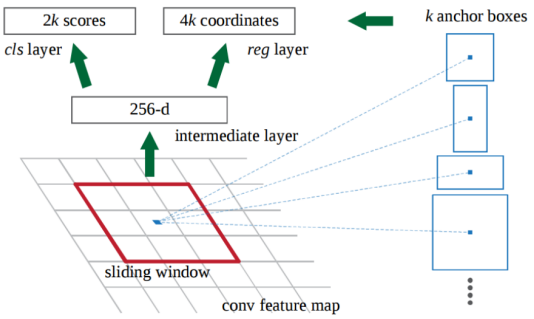


图 2-3: RPN 网络工作原理

接下来的工作与 Fast R-CNN 类似，图像特征通过 pooling 层、全连接层计算后进入 softmax 层进行类别区分以及边框回归器 (bounding box regressor) 中最后校正。

上述的 two-stage 目标检测模型都基于卷积神经网络模型。他们在精度上较之前基于传统机器学习的检测方式有很大提升，但他们也有一些自身的缺点。比如卷积模式固定，无论输入什么样的图片，CNN 都会按相同的模式进行卷积、特征提取、pooling 和分类，这样就很容易漏判一些特殊情况。假设训练集中所有的物体都是正向摆放的，那么如果验证集中有一个侧放甚至倒放的物体，这些网络可能会漏检。解决这种问题可以采用数据增强 [20] 或者用滑动窗口 [21] 等办法，但是数据增强只能得到同类型的多张图像，滑动窗口会增大时间消耗。因此可变形卷积策略 (Deformable Convolutional Networks, DCN) 就此应运而生 [22]。DCN 较之前的算法基础上采用了额外的空间采样点以及偏移量，并可以从目标任务中学习优化偏移量，无需额外的监督资源。新的卷积模型可以替代之前的卷积神经网络模型，依然支持反向传播训练，从而实现可变形的卷积神经网络。现阶段 DCN-v2 在 two-stage 目标检测算法中精度最高基本上被主流界公认 [23]，在整个目标检测领域内也占据着重要的地位。

2.2.2 基于 one-stage 的目标检测算法

one-stage 算法舍弃了 two-stage 算法中的感兴趣区域生成步骤 (即 R-CNN 中的启发式算法、Faster R-CNN 中的 RPN 等等)，对于一个输入图像，one-stage 算法可以在单个步骤中完成上述两个任务，即在处理图像的同时进行边框预测和分类。不同 one-stage 算法的运行方式也不尽相同，下面介绍几个经典的 one-stage 目标检测算法。

最早由 2016 年提出的 SSD 是第一个 one-stage 目标检测算法 [24]。其思想

是先行预设好锚点和备选框的比例，而后遍历整个特征图以每个像素点为中心来获得备选框。这样做可以起到代替 RPN 网络的作用，但缺陷是备选框过多，绝大多数备选框都是负样本，即框内不存在目标物体或同一物体周围被多次框选。因此 SSD 利用非极大抑制技术 [25] 来将高度重叠的框整合成一个最逼近的框，同时使用最佳负样本筛选技术来使正负样本保持平衡 (1:3)。具体而言是 SSD 只选取那些使训练损失达较大的负样本，使其个数约为正样本数的 3 倍。鉴于 SSD 小物体识别率较高、速度较快的优秀特性，我们后续的实验也使用了基于 SSD 的目标检测模型。

另一种 one-stage 目标检测算法 YOLO[26] 则采用了其它策略：它将目标检测问题看成一个回归问题解决。对于输入图像而言，YOLO 将其划分为 $S * S$ 个格子，同时检测每个格子是否为某个目标的中心。每个格子包含 5 个参数，分别为 $x, y, w, h,$ confidence。其中 (x, y) 是当前格子预测所得的 bounding box 中心坐标，w, h 表示以 (x, y) 为中心所得框的宽度和高度，并进行归一化。而 confidence 由如下公式 2-7确定：

$$\text{confidence} = P(\text{object}) * \text{IOU} \tag{2-7}$$

在式 2-7中，若 YOLO 判断该 bounding box 包含物体，则 $P(\text{object}) = 1$，否则 $P(\text{object}) = 0$。IOU 表示预测框与真实框之间的交集。在最后输出的时候，每个格子只选择预测 IOU 最高的 bounding box 作为输出，即每个格子最多只检测一个物体。这也间接导致一个问题，当物体本身很小的时候 YOLO 有较大概率漏检。YOLO 是当前目标检测领域较快的模型，在几轮的迭代优化后已经逐渐趋于成熟 [27][28][29]。

2.2.3 Anchor-free 相关目标检测算法

由于之前的检测算法绝大多数引入锚点来先行画框之后再对其进行取舍，而锚点的选取对最终检测结果的影响很大，因此近年来许多研究者们把重心转移到如何去掉预先构建框集合的问题上，相继提出一系列不需要预设框集合的 Anchor-free 的检测算法。

ConerNet[30] 的研究重点为舍弃 Anchor 后如何描述框。该算法模型提出用 (x_1, y_1, x_2, y_2) 形式来描述，其中 (x_1, y_1)，(x_2, y_2) 分别为框的左上角和右下角。在此基础上利用热力图 (heatmap)[31] 判断每个点是 corner 的概率，另一方面利

用 embedding 输出每个位置数据某个类别的概率。最后加入偏移量 offsets 对点的位置进行修正，类似于之前方法的预测框最后回归。

在 ConerNet 基础上，研究者们提出 CenterNet 网络模型 [32]。由于 ConerNet 只能提供目标的边缘信息，对于检测而言比较困难。因此 CenterNet 在此基础上增加了对中心点的检测，来帮助筛选初步得到的候选结果。与 CornerNet 不同的是，CenterNet 新增加了不进行 embedding 的网络分支，用来预测特征图上每个点是中心点的概率。当有候选框被算出时，网络会自动将候选框划分为 3×3 或者 5×5 的格子，然后判断中间格子中是否有中心点，如果没有则舍弃该候选框。这种判断形式较 CornerNet 而言避免了很多误检。

2019 年提出了新的检测网络 ExtremeNet[33]，其借鉴了目标标注方法和 CornerNet 的部分思想，它认为在图片中寻找角落是比较复杂的问题，毕竟很多时候角落并非在这个物体之中，往往在背景 (background) 上。这无论是对训练还是最终预测都会有较大干扰。因此 ExtremeNet 提出用 (top-most, bottom-most, left-most, right-most, center) 这样一种描述方式去检测所需的目标类别，它的输出框也不再拘泥于矩形结构。其它网络结构与 CornerNet 类似，但 ExtremeNet 舍弃了 CornerNet 的 embedding 分支，采用穷举联合中心点判断的方式判断最终的类别归属。

2.2.4 小结

从 2014 年第一个基于深度学习的目标检测算法 R-CNN 开始，短短几年时间便涌现了几十种目标检测方法。基于深度学习的目标检测算法在类别不是非常多 (少于一万种) 的情况下相比于传统的目标检测算法有着很大的优势。从 YOLO 模型被提出开始，基于深度学习的目标检测模型就在速度和精度上全面超越了传统目标检测算法。可以预见的是基于深度学习的目标检测算法将会越来越快、越来越精准，与传统的目标检测算法拉开更大距离。

2.3 本章小结

由于现今基于深度学习的目标检测算法在精度和速度方面已经全方位超越了传统的目标检测算法 [34]，因此后续的比较不再以传统机器学习检测算法为参考，仅仅针对深度学习算法内部而言进行比较和讨论。

在使用锚点 (Anchor) 的检测方法中，可大致将其分为 two-stage 和 one-stage

两种结构。two-stage 因为对预设框进行了筛选处理，同时将筛选后的框进行分类和回归，因此整体的精度会略高于 one-stage 的目标检测算法，但是在速度方面处于劣势。one-stage 的主要缺点在于预先设定好的锚点框有极多的负样本，因此需要对负样本进行清洗，筛选出对模型训练有最大帮助的负样本进行训练。因此许多研究者专注于如何在这么多负样本中找出优质的负样本集合，因为负样本数过多和无法取出优质负样本是 one-stage 目标检测算法的瓶颈。自 2017 年 Focal loss[35] 被提出后，one-stage 算法的精度突飞猛进，目前已经与 two-stage 相差无几，因此最近两年基于锚点的目标检测算法由 one-stage 占据主流，它们具有速度上的优势同时在精度上也毫不逊色。

由于预设的锚点需要被指定，同时不同类别锚点策略也不统一，因此部分研究者认为这违背了深度学习理念的初衷 (模型自适应输出结果)，因此最近几年放弃锚点转而研究 Anchor-free 算法的研究者越来越多。整体而言，Anchor-free 算法相较于之前的 Anchor-based 算法有以下几点优势：首先不需要预设锚点和框的比例，达到真正的深度学习自适应判别输出；其次是放弃了锚点会导致冗余的候选框大大减少，使网络结构更加自然容易理解。但是 Anchor-free 也存在缺陷：基于 Anchor-free 的检测方法往往需要判断目标的边界点或者中心点，这会导致若两个目标的中心点距离很近，那么有极大的概率会误检甚至错检，因为大多数 Anchor-free 算法都选取重合部分最优的一个结果进行输出。虽然现阶段有研究者采用 FPN 网络和上文提到的 Focal loss 去缓解上述问题，但效果一般。

目前针对目标检测问题的网络模型主要的发展方向在于提升精度。在检测速度方面，虽然 SSD 和 YOLO 以参数量少，检测速度快而著称，但搭配普通的特征提取网络仍然做不到在普通算力的计算机上实时性检测。如上文所述，若直接用多类别目标检测网络执行单类别目标检测任务，会造成参数的冗余，从而影响模型运行速度。因此本文通过优化骨干网络模型的方式来减少目标检测模型整体的运算量，完成在普通算力计算机上进行实时单类别目标检测的任务。

本文后续实验在 SSD 的基础上进行。因为 SSD 相较于其它目标检测算法而言，可以根据不同卷积层下的 Anchor 策略和框选比例来判断各个大小的目标，可以极大限度地缓解小目标物体检测难题，这对于我们后续进行的单类别目标检测十分重要，也是不选择 YOLO 的主要原因。同时与其它 Anchor-free 算法相比，SSD 的检测效率比较稳定，在后续的实验中比较容易能够看出算

法对于整体模型所带来的提升。而且在现阶段的优化下基于 SSD 的目标检测在高算力计算机 (gpu) 上的检测速度也能达到实时性的要求 (40+FPS)，因此我们选择 SSD300 目标检测模型作为后续的实验模型，骨干网络选择参数较少的 mobilenet 网络。

本文在标准 mobilenet-SSD 基础上根据单类别目标检测的特殊性，面对不同的目标类别对骨干网络进行有针对性地更改。本文的目标是更改后的网络模型可以拥有更高的速度以及与原模型不相上下的精度，从而达到在普通算力 (cpu) 的计算机上实时检测的效果。

第三章　单类别目标检测基于 mobilenet v3 的骨干网络架构优化

当前的计算机视觉任务中，骨干网络 (backbone) 联合针对特定任务的网络头部共同组成解决目标检测任务的网络系统已经成为主流。新的骨干网络模型在近几年层出不穷。由于本文最终任务是实现一个可以在普通算力计算机上实时运行的目标检测模型，因此在骨干网络的选择方面，参数量的多少是作者选择的重要标准。在本章实验中将选用 mobilenet v3[9] 网络作为骨干网络，参与特征提取。主要原因是该特征提取网络具有参数量少，运行速度快、精度较高的多重优点，是目前计算机视觉领域优秀的特征提取网络。

mobilenet v3 是神经架构搜索 (Neural Architecture Search, NAS)[36] 的产物，它继承了 mobilenet v1 的深度可分离卷积以及 mobilenet v2 的倒残差结构 [37] 等优点，并改进了激活函数 swish(x)[38]。与现今主流特征提取网络相比，有速度快、效果良好的优势，由于本文更关注实时性的目标检测，因此本节实验基于 mobilenet v3 进行。

为构建实时性的单类别目标检测模型，本章节通过改变 mobilenet v3 的部分结构和训练方式使其在单类检测中精度更高，参数量更少，能够更快更精准地提取到图像特征。最终本文将会用本章的方法结合第4章算法的网络模型设计一个针对人物类别的室内定位系统。

3.1　目标检测模型中的骨干网络

实际上，当前主流目标检测框架由骨干网络 (backbone) 负责提取特征，头部网络负责目标检测，进而实现端对端的目标检测任务。前文 2.2节中提到基于深度学习的目标检测方法都是头部网络的算法和结构，实际上2.2节中绝大多数网络的输入都是特征图片，而特征图片由骨干网络提取输出。一般而言，一个完整的目标检测模型的步骤如图3-1所示。

如何能够更好地，更精准地提取出图片中的特征信息，对于目标检测网络而言是至关重要的。近年来研究者们提出了大量优秀的骨干网络，从最初的

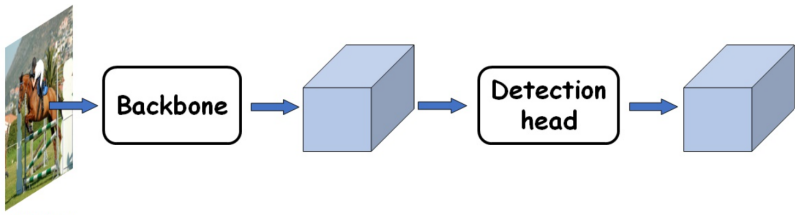


图 3-1: 目标检测模型的网络结构

resnet 到 vgg 再到后来的 darknet、mobilenetnet、shufflenet [39] 等等。不同网络模型的侧重点不同，就实时性的单类别目标检测而言，我们需要得到一个在普通机器上可以达到近乎实时性要求的目标检测网络模型，因此需要一个体积小、参数少的骨干网络模型。因此 mobilenet 作为本文的主要骨干网络结构。

由于本文的目的是建立一个能够在普通算力计算机上做到实时性检测的单类别目标检测网络模型，因此相比于精度我们更关心参数数量和运算量的大小。在本章使用参数量较少的 mobilenet v3 作为特征提取网络的基模型，尽管它依旧无法满足实时性要求。由于 mobilenet v3 的参数量远多于 ssd，因此本文对 mobilenet v3 网络结构进行改进，使其参数量变少，运算速度加快，更加充分地发挥目标检测骨干网络的特征提取作用，近乎达到实时性检测的要求。

3.2 卷积计算方式与参数数量

本节介绍基本 CNN 网络模型中的卷积计算方式和所需的参数量，通过对比可以直观的感受到深度可分离卷积 (Depthwise Separable Convolution) 相比于普通卷积在参数量方面的优越性。进一步地，这一节将会解释为什么相同效果下深度可分离卷积中的 pointwise 卷积核参数量要小于 depthwise 卷积核，为后续优化 mobilenet v3 网络模型打下基础。

3.2.1 普通卷积

卷积运算在计算机视觉中有不可替代的作用。卷积的计算方法如式 3-1所示：

$$(f * g)(n) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(n-\tau) \tag{3-1}$$

其中,$f(\tau)$ 为待卷积区域，一般为 $n \times n$ 的正方形，大小取决于输入图片的大小和上一层计算后的特征图尺寸。$g(n-\tau)$ 为卷积核，是一个 $m \times m$ 的正方形。一般 $3 \times 3, 5 \times 5$ 为佳 [40]。假定一层有 k' 个卷积核，每个卷积核会输出一张特征图，因此通过这层卷积，将会输出 k' 个特征图，这里的 k' 称为通道。

实际上在卷积层中每个卷积核有 k 维，k 是该卷积层接收的原图或上层输出的特征图片的通道数量。卷积核从左上角开始从左向右、从上到下在待卷积区域上滚动。其具体计算方式如图3-2所示。

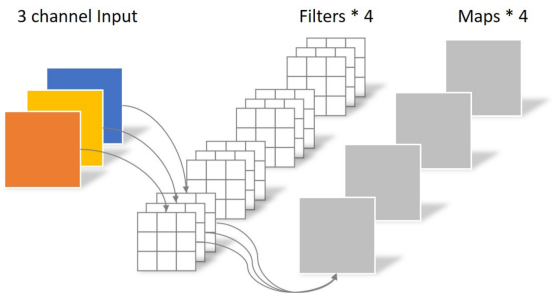


图 3-2: 普通卷积运算形式

图3-2中左侧为待卷积区域 $f(\tau)$，中间部分是卷积核，这里使用了 3×3 的卷积核作为算子，而右侧灰色图像为输出的特征图。可以看出图3-2中 $m = 3, k = 3, k' = 4$。理论上，卷积神经网络可以拟合任意线性变换 [41]，只不过需要构建特殊的卷积核、连接层和追加深度。可以看出，神经网络计算时间的瓶颈在于参数数量。在图3-2中可以看到，该层卷积运算使用了 4 个卷积核，每个卷积核大小为 $3 \times 3 \times 3$，因此该层的参数量为 $4 \times 3 \times 3 \times 3 = 108$。

3.2.2 深度可分离卷积

深度可分离卷积 (depthwise separable convolution)[42] 的目的在于采用更少的参数得到相同的效果。因此它在运算的过程中将卷积拆分成了 Depthwise Convolution 和 Pointwise Convolution 来替代原有的卷积模式。

首先参与运算的是 depthwise 部分，它与上文中的普通卷积模式有两点不同：其一是卷积核的个数并非该层所输出的通道数，而是与上一层的原图/特征图通道数一致；其二是该层中每个卷积核的维度都是 1 维，不与普通卷积方式中上一层原图/特征图一致。同样假设卷积核的大小是 $m \times m$，上一层输出的原图或者特征图维度为 k，那么该层所需的参数数量应该为 $m \times m \times k$。具体 depthwise 卷积部分如图3-3所示。 从图3-3中可以看出，$m = 3, k = 3$。因此执

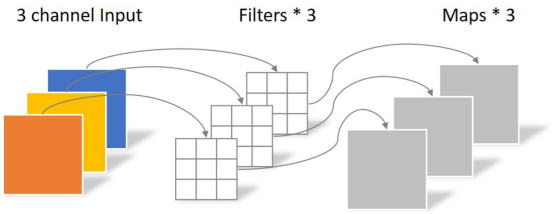


图 3-3: 深度可分离卷积中的 depthwise 卷积

行此过程的所需的参数数量为 $3\times3\times3=27$。

可以看出，depthwise 操作只能得到 k 个 $m\times m\times1$ 的特征图片。这与卷积层最终输出 k' 个 $m\times m$ 特征图的要求不符。因此在 depthwise 之后还会引入 pointwise 操作以改变其通道数。具体而言，pointwise 与普通卷积类似，只不过卷积部分由 k' 个 $1\times1\times k$ 的卷积核组成，所需的网络参数量为 $1\times1\times k\times k'$。其运算过程如图 3-4所示。

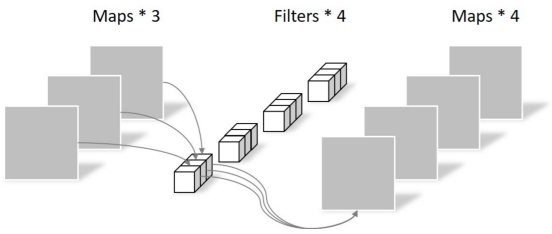


图 3-4: 深度可分离卷积中的 pointwise 卷积

从图 3-4中可以看出，这种卷积模式可以将 depthwise 所得到的不同通道卷积结果进行组合，同时还可以通过控制 pointwise 过程的卷积核数量来影响该层神经网络的输出通道数。显然，在图 3-4中该过程所需的参数量为 $1\times1\times3\times4=12$。深度可分离卷积一共所需的参数数量为 $27+12=39$。

由此可见，与普通卷积形式相比，深度可分离卷积模式可以在远少于原参数量的基础上实现卷积的操作，这也是 mobilenet 系列参数量比较少的根本原因。

3.3 单类别目标检测下 mobilenet v3 的性能优化

mobilenet v3 的提出无疑为图像分割、目标检测等任务提供了一个关键有效的骨干网络模型。本节将在现有的 mobilenet v3 small 的基础上进行改动，使其更加适用于实时性的单类别目标检测任务，进而将其作为后续演示系统的基模型。

3.3.1 SE 模块的调整和优化

mobilenet v3 性能提升的关键就在于引入了 SE(Squeeze-and-Excitation[43]) 模块。该模块的主要作用在于判定同一个卷积层中不同卷积核之间大致重要程度，具体流程如图 3-5所示。

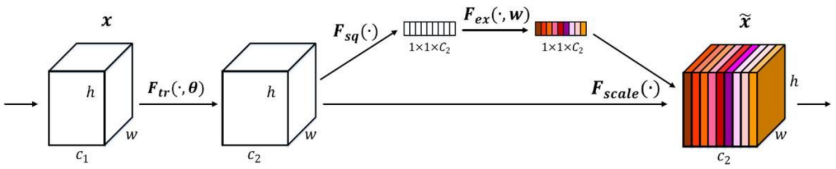


图 3-5: mobilenet v3 中的 SE 模块

SE 模块的思想是先用一个全局 pooling 得到通道长度为 C 的向量（即图 3-5中 c_2），其数学形式如式 3-2所示。

$$P_k = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} f_k(i,j) \tag{3-2}$$

其中，$f_k(i,j)$ 代表第 k 个通道下的卷积核在 (i,j) 处的特征值，卷积核宽度为 W，高度为 H。最终第 k 个通道输出的全局 pooling 结果记为 P_k。因此，对该层所有特征图进行全局 pooling 操作后将会得到一个 $1 \times 1 \times C$ 的向量，整个过程称作压缩 (Squeeze)。后续的激励 (Excitation) 操作将在这个向量上进行，对于 mobilenet v3 而言，激励过程如图 3-6所示。

图 3-6: mobilenet v3 SE 模块中的激励部分

从图3-6中可以看出，整个结构由一个全局 pooling 和两个全连接层、两个激活层组成。与普通卷积神经网络全连接层不同的是，SE 模块中的全连接层为了防止运算量过大，引入了 SE Ratio 参数来削减通道数，在原版 mobilenet v3 中取值为 $\frac{1}{4}$。也就是说经过第一个全连接层后，输出通道变为输入通道的 $\frac{1}{4}$，经过第二个全连接层后，最终输出通道与最初输入通道一致。两个全连接层部分的数学表示分别如式3-3和式3-4所示：

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ W_{i1} & \vdots & \ddots & \vdots \\ W_{m1} & W_{m2} & \cdots & W_{mn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{3-3}$$

$$\begin{bmatrix} a'_1 \\ a'_2 \\ \vdots \\ a'_n \end{bmatrix} = \begin{bmatrix} W'_{11} & W'_{12} & \cdots & W'_{1m} \\ W'_{21} & W'_{22} & \cdots & W'_{2m} \\ W'_{i1} & \vdots & \ddots & \vdots \\ W'_{n1} & W'_{n2} & \cdots & W'_{nm} \end{bmatrix} \times \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_m \end{bmatrix} + \begin{bmatrix} b'_1 \\ b'_2 \\ \vdots \\ b'_n \end{bmatrix} \tag{3-4}$$

其中式3-3为第一个全连接层的计算过程，式3-4为第二个全连接层的计算过程。全连接层可以看成是多个向量组合的形式。对于式3-3，全连接层的权重矩阵表示为 $[W_1, W_2, \cdots, W_m]$。其中每个单元 W_i 自身又是一个长度为 n 的向量。$[x_1, x_2, \cdots, x_n]$ 是输入向量，它的长度等于上文中的通道数 C，$[b_1, b_2, \cdots, b_n]$ 表示偏置向量，输出向量为 $[a_1, a_2, \cdots, a_m]$。式3-4同理。由于 $\mathrm{SE\,Ratio} = \frac{1}{4}$，因此式3-3和式3-4中有 $n = 4m$。

在通过第一个全连接层后，mobilenet v3 选择 ReLU 函数进行激活，通过第二个全连接层后的向量使用 H-swish 函数激活。最终通过激活函数得到一个长度为 C 的向量，其中每个值表示其对应特征图的权重，在输入到下一层时将向量对应单元的值乘以特征图中的相应通道图像作为实际输入。因此激活模块所需的计算量如式3-5所示。

$$Params_E = 2 \times C^2 \times \mathrm{SE\,Ratio} = \frac{1}{2} \times C^2 \tag{3-5}$$

由于 mobilenet v3 中引入了 SE Ratio 参数，在几乎一样的效果下全连接层部分减少了近 $\frac{3}{4}$ 计算量 [9]，而在原版 mobilenet v3 中 SE 块置于 depthwise 与 pointwise 之间，因此在 mobilenet v3 中 SE 模块全局 pooling 部分所占的运算比

重较大。同时由于 depthwise 与 pointwise 是一个卷积过程的整体，中间插入 SE 模块显得复杂冗余，因此在本文工作中将 SE 模块插入到 pointwise 后面。这样的优化不但是网络的整体逻辑易于理解，更重要的是，这种优化使得 mobilenet v3 网络模型的参数量大大减少，使实现实时性目标检测成为了可能。

但单单改变 SE 模块的位置会由于 pointwise 通道数较多，获取的权重向量的方差会变得更加不稳定，因此本文实验中还在 SE 块中加入 BatchNorm 模块 [44]，使其计算而得的卷积特征图之间的结果更加稳定。至于扩张部分仍然选取原文的 $\frac{1}{4}$。后续的实验表明通过这种优化方式确实减少了参数量，而且由于 BatchNorm 的加入，新的网络模型在精度方面也有了一定的提升，使得优化后的网络模型在后续任务上的表现优于原 mobilenet v3 网络模型。

3.3.2 训练方式调整

在上述优化的基础上，新的 mobilenet v3 模型得到了比原来更好的结果。但事实上 mobilenet v3 还可以在其基础上进行进一步优化以提升精度。

普通的网络模型训练是先设定一个学习率 α，而后根据训练轮数 (epoch) 的增加缓慢降低 α，网络参数趋于稳定后得到收敛的模型。神经网络的初始化往往是随机的，而模型最初训练的学习率较大会导致最开始训练的几轮数据重要性占比增加，这时候如果这几轮训练数据不能够满足整体验证集的分布，会导致模型“偏离”整体数据轨迹，想要靠后续较小的学习率回调是十分困难的。

本文采用 linner warmup 来缓解该问题 [45]，即训练过程中采用学习率先增加再减小的方式来缓解普通训练所出现的学习偏移情况。初始学习率设定为 10^{-4}，缓慢增大到第 15 个 epoch 时为最大，此时 $\alpha = 10^{-3}$，而后再缓慢降低，直到第 100 个 epoch 时回到初始学习率值，即 10^{-4}。学习率变化如图 3-7所示。

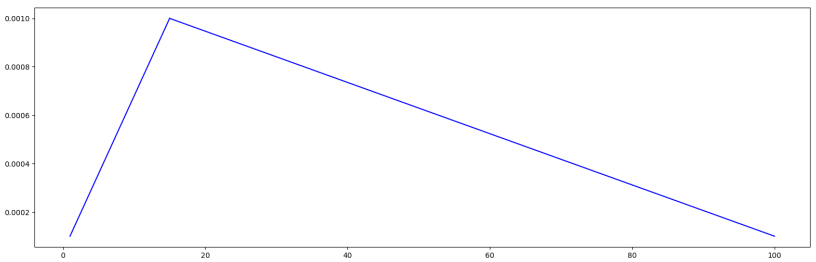


图 3-7: 改进的 mobilenet v3 学习率变化情况

从后续的实验可以看出，使用这两种策略进行优化后，基于 mobilenet v3 small 的改进网络较原模型在速度和精度上都有提升。本文将两种策略优化后的

mobilenet v3 small 网络记作 mobilenet v3 $small_2$。

3.4 实验及分析

为了验证以上两部分优化在 mobilenet v3 small 上的有效性，本节将对优化后的模型与原模型及其它主流模型进行对比。由于图像分类的结果可以一定程度上反映网络特征提取程度的好坏，因此本文的对比任务主要分为目标检测和图像分类。除此之外，在模型的参数数量也进行了对比，从而体现本文优化策略的效果。本章节实验包括：

1. 比较多个特征提取模型的参数数量和计算量，突出 mobilenet v3 $small_2$ 相比于其它模型在运行速度上的优势。
2. 将新训练的 mobilenet v3 $small_2$ 接入 SSD，验证优化后的网络在单类别目标检测领域的性能，进而对后续系统的搭建提供数据论证基础。
3. 比较现阶段主流图像分类模型的精度和速度，进一步验证 mobilenet v3 $small_2$ 的实验效果。

3.4.1 实验设计

3.4.1.1 数据集

由于本章节实验同时涉及到目标检测和图像分类，因此将对检测和分类两方面的主流数据集进行介绍，后续的实验也在这些数据集上进行。目标检测相关数据集包括：

- VOC2012[①]：全名为 Pascal VOC2012 数据集，由 Pascal VOC 挑战赛提供[46]。该挑战赛是世界级计算机视觉挑战赛，主要挑战方向为目标的分类、检测、分割、动作识别等等。主要分为 4 大类别：人、常见动物、室内家具、车辆。细化下来又分为 20 个类别，是当前网络模型在目标检测任务中测试最频繁的数据集。本文由于执行单类别目标检测任务，因此在实验中只采取单一类别作为参考。
- COCO[②]：COCO 数据集全名为 Microsoft Common Objects in Context，是一个大型的物体检测、分割数据集 [47]。起源于 2014 年，由微软出资标

[①]相关数据集下载地址：http://host.robots.ox.ac.uk/pascal/VOC/
[②]相关数据集下载地址：http://cocodataset.org

注。该数据集主要从复杂的生活场景中提取目标类别，在图像中目标通过精确的分割进行位置的确定。整个数据集包含 80 个类别目标，在目标检测领域占有很重要的地位。

图像分类方面的数据集介绍：

- CIFAR-10/CIFAR-100[①]：这两个数据集都来自于一个规模更大的图像数据集 80 million tiny images dataset [②][48]。其中 CIFAR-10 包含 10 个类别，每个类别有 6000 张图片，50000 张图片用于训练，10000 张图片用于测试。CIFAR-100 包含 100 个不同的类别，每个类别有 600 张图片，500 张用于训练而 100 张用于测试。不同类别之间的图片没有交叉，很多网络和算法都会在上述两个数据集上进行对比。
- Caltech 101 [③]：这个数据集是李飞飞于 2004 年参与研究的数据集，全名为加利福尼亚理工学院 101 类图像数据集。它包含 101 个不同的图像类别，一共包含 9000 多张图片。有趣的是该数据集的各个类别分布不均，每个类别有 40-800 张图片不等。因此有时也将该数据集作为类别不均/小样本实验的数据集 [49]。由于本文基于目标单类别的图像识别实验，因此在本实验中我们选择其中数据量较多的飞机 (airplane) 作为我们的目标类别。
- ILSVRC2012[④]：这是 ImageNet2012 竞赛所用的数据集的一个子集。竞赛包含了不同的计算机视觉方向，该类别数据集分为 1000 个类别，每个类别有大约 1000 张图片用于训练。同时每个类别 50 张图片作为验证集，除此之外还有大约 15 万张测试图像。该数据集是目前图像分类领域最常用的数据集之一。

3.4.1.2 评价指标

现如今在目标检测领域，AP (Average precision) 是主流的评价指标。在多类别目标检测里引入 mAP (mean Average Precision)，即不同类别的 AP 取均值。其中 AP 的计算需要涉及重复区域比例 (IoU), 精度 (precision) 和召回率 (recall) 的概念。

IoU (Intersection over union) 衡量两个区域的重叠程度，具体计算方法是利

[①]相关数据集下载地址：http://www.cs.toronto.edu/ kriz/cifar.html
[②]相关数据集下载地址：http://groups.csail.mit.edu/vision/TinyImages/
[③]相关数据集下载地址：http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[④]相关数据集下载地址：http://www.image-net.org/challenges/LSVRC/2012/downloads

用两个区域重叠的面积除以两个区域的总面积。公式表示如下：

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}} \tag{3-6}$$

如图3-8所示，蓝色框是能够覆盖我们目标的最佳框选，绿色框是算法计算预测的实际框选，那么由公式3-6可知，当前情况下 $IoU = \frac{\text{黑色区域}}{\text{黑色区域+白色区域}} = \frac{①}{①+②+③}$。

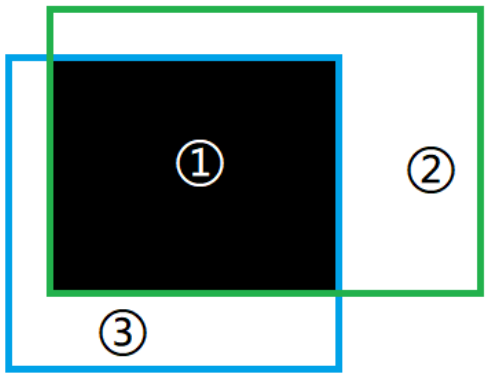


图 3-8: IoU 计算实例

在目标检测任务中，一般而言模型/算法预测框与实际的最优框之间的 IoU 比值大于某个阈值 (通常为 0.5) 时即认为模型输出了正确的框。

目标检测问题可以理解成检测框是否合格的二分类问题。对于二分类问题而言，可以根据真实情况与预测情况将预测结果分为真正例 (true positive)、假正例 (false positive)、真反例 (true negative)、假反例 (false negative) 四种情况。其混淆矩阵可以如表3-1所示 [50]。

表 3-1: 分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP(真正例)	FN(假反例)
反例	FP(假正例)	TN(真反例)

其中，TP 表示为与某个标准框匹配 IoU ≥ 0.5 的预测框数量 (每个标准框最多只能有 1 个最接近的预测框作为 TP)；FP 表示为与任意标准框都有 IoU < 0.5 的预测框数量与匹配同一个标准框的多余框数量的加和；FN 表示为没有检测

到的标准框数量。查准率 P 的定义为所有预测框中有意义的预测框比例，查全率 R 的定义为所有有意义的标准框被找到的比例。因此根据查准率 P 与查全率 R 的定义 [51] 可知：

$$P = \frac{TP}{TP + FP} \tag{3-7}$$

$$R = \frac{TP}{TP + FN} \tag{3-8}$$

P 值和 R 值当然是越高越好，但实际上它们是相互矛盾的。如果想提高 P 值，自然希望每个预测的结果都是真正例，那么会倾向于策略比较保守的估计；如果想提高 R 值，则希望尽可能多的把所有真实情况中的正例覆盖，估计策略会比较宽泛，而宽泛的估计自然会降低 P 值。实际生产和生活中往往根据 P/R 的不同需求来定制相应的策略。但是如果想要综合评定哪种算法策略更优，主流公认的方法一般通过绘制 P-R 曲线来判断。

在目标检测领域，一般先将得到的预测结果框进行置信度排序，然后逐个判断是否为 TP/FP/FN，并计算当前框数下的 P/R 值。然后根据每个置信度下的预测框所计算出的 P/R 值找到坐标系中的对应点，最后根据置信度的顺序将点连接就得到了该检测模型下的 P-R 曲线。而目标检测模型在该数据集中的平均精度 (AP) 就是这个曲线纵坐标 (P) 的平均值。不过一般而言我们通过积分的方式求出 AP：

$$AP = \int_0^1 P(r)dr \tag{3-9}$$

其中 $p(r)$ 表示横坐标为 r 时曲线的值。但由于一旦遇到了假正例，P-R 曲线就会垂直下降，给最终结果造成较大误差。因此很多时候研究者用平滑后的 P-R 曲线下面积来表示该模型在此种类别上的 AP 值。如式3-10所示。其中 P_{smooth} 为当前点右侧最大的 P 值 (含该点)，公式表示如式3-11所示。

$$AP = \int_0^1 P_{smooth}(r)dr \tag{3-10}$$

$$P_{smooth}(r) = \underset{r' \geq r}{max} P(r') \tag{3-11}$$

在本文的后续实验中，AP 的计算均按照 $IoU \geq 0.5$ 作为阈值，因此后续基于 AP 的评价标准将会标注成 AP@0.5。

除了目标检测之外，本文还采用了图像分类的相关指标，即使用 top-1 的精确度 (accuracy) 来度量算法的好坏。值得注意的是，因为模型本身也要防止

过拟合/假正例数量过多，因此除了 top-1 之外还会引入 F1 数值来进行另一个尺度的度量。也就是说真正模型的好坏不光是要在我们关注的类别上增加 top-1 的值，还要尽可能减小 FP 的数量，不希望一个模型把任何一张图片都归类为我们感兴趣的类别。其中 F1 计算公式如式 3-12所示，P、R 分别代表着查准率和查全率，具体计算方法如式 3-7和 3-8所示。

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{3-12}$$

目标检测方面采用上文叙述过的 AP@0.5 来进行性能的度量。同时因为我们是基于单类别目标检测任务的剪枝算法，在这个特殊情况下有 $mAP = AP$。

除此之外，本节的对比实验还涉及模型参数数量与计算量，下面分别介绍这两部分的评价标准。

模型参数数量指的是模型训练好后参与运算的参数个数，对于普通卷积而言，模型参数量 $Params$ 如式 3-13所示。其中 C_i 和 C_o 分别表示该卷积层的输入/输出通道数，k_w 和 k_h 分别表示卷积核自身的宽和高。后面的常数表示偏离率 b。对于卷积核是正方形的网络而言，上式可简化为 3-14，其中 m 为卷积核自身的边长。

$$Params_1 = C_o \times (k_w \times k_h \times C_i + 1) \tag{3-13}$$

$$Params_2 = C_o \times (m^2 \times C_i + 1) \tag{3-14}$$

对于深度可分离卷积而言，由 3.2.2可知，可由式 3-15表示，m 依然表示为正方形卷积核的边长。

$$Params_3 = m^2 \times C_i + 1 \times 1 \times C_i \times C_o \tag{3-15}$$

对于全连接层而言，若输入的特征向量权重为 N_i，输出特征向量的权重为 N_o，那么全连接层的参数数量如式 3-16所示。

$$Params_4 = (N_i + 1) \times N_o \tag{3-16}$$

在神经网络模型计算量方面，FLOPs 表示为浮点运算次数，可以大致表示计算量。而浮点运算次数包含乘法和加法，因此我们在此用更加规范的乘加次数 MAdds 作为计算量的衡量标准 [52]。网络模型自身的参数量和计算量都可以用现有的函数库 torchstat 直接计算而得，后续的结果也基于上述函数库的直接

输出。

由于本文的目的是最终建立一个单类别实时性目标检测系统，因此目标检测模型的检测速度在本文是一个重要的评价指标。本文采用计算机每秒可以处理的图片数 (Frames Per Second, FPS) 作为参考标准。由于人眼能够分辨的图片数量为 24 张/秒，因此实验中将 24FPS 作为一个实时性检测的参考阈值。

3.4.1.3 实验环境

由于我们最终的目的是得到一个轻量级的单类别目标检测网络模型，使其能够在普通算力的计算机上达到实时性检测的效果。因此本文所有实验在 cpu 上进行，具体参数为 i7 6700, 显卡为 GTX960。除此之外，为了证明本文提出的 warmup 训练方式的有效性，将优化后的网络模型进行了不同方式的训练。其中 mobilenet v3 small′ 为正常按步缩减的学习率训练模型，初始学习率为 10^{-3}，mobilenet v3 $small_2$ 为优化学习率训练模型，初始学习率为 10^{-4}，15 轮后上升为 10^{-3}，最后回到初始值。

3.4.2 主流特征提取网络参数量的比较

首先我们进行比较的是主流特征提取网络模型在参数量方面的情况。本次参与比较的网络分别为 VGG-16 网络 [53]、inception v3 网络 [54]、mobilenet v1[55]、mobilenet v2[37]、mobilenet v3 small[9]、mobilenet v3 $small_2$。

表 3-2: 不同网络之间的参数量对比

网络模型	Params(Million)	MAdds(Million)
VGG-16	138	15300
inception v3	23.2	5000
mobilenet v1	4.2	569
mobilenet v2	3.4	326
mobilenet v3 small	2.9	66
mobilenet v3 $small_2$	**2.7**	**63**

从表3-2中可以看出，参数量的多少一定程度上反应了模型计算量的大小。但由于后续 mobilenet 自身模型在改进版残差单元、纺锤卷积形态等方面的优化 [37]，可以使得 mobilenet v2 以及后面的网络得以做到计算量更小。但

整体而言，我们的模型 mobilenet v3 $small_2$ 无论是从大小还是计算量的角度都在现有的主流特征提取网络中都拥有绝对的优势。与优化前的 mobilenet v3 small 相比也有一定程度的提高。优化后的 mobilenet v3 $small_2$ 网络模型相比于 mobilenet v3 参数量减少了约 6.9%，运算量减少了约 4.5%。

3.4.3 优化后模型在单个目标类别上的检测表现

针对单类目标检测问题，本文将原 mobilenet v3 small 模型、mobilenet v3 small′ 模型与最终优化后的 mobilenet v3 $small_2$ 模型分别作为 SSD 的骨架模型，分别选取 COCO 数据集的人物、飞机、马及鸟四个类别作为目标类别，评价指标为 AP@0.5 和 FPS。训练过程中轮流选取其中一个类别作为正类，其它作为负类，最终将结果取均值后记录。由于本文的最终目的是得到一个可以在普通算力计算机上运行的目标检测网络模型，因此 FPS 是我们评价的主要标准，在此基础上 AP@0.5 是第二评价标准。作为参考，实验中选择 YOLO v2 作为主流多类别目标检测模型的代表。

表 3-3: 不同网络模型在单类别目标检测任务上的平均表现

网络模型	COCO AP@0.5	FPS
YOLO v2	**43.81**	12.5464
mobilenet v1 ssd	40.24	16.0845
mobilenet v2 ssd	40.61	15.4497
mobilenet v3 small ssd	34.68	21.2832
mobilenet v3 small′ ssd	35.94	**22.9544**
mobilenet v3 $small_2$ ssd	36.05	**22.9544**

从表 3-3中可以看出，mobilenet v3 $small_2$ ssd 网络模型无论是从检测速度还是精度方面都优于 mobilenet v3 small ssd 网络。相同的优化条件下，本文使用的 warmup 训练结果也要好于常规训练结果。同时在速度方面，优化后的网络能够在普通计算机的算力基础上达到接近 23 张/秒的速度，这与人类肉眼可分辨的 24 张十分接近。因此基于单类目标检测在普通机器上的实时性得以保证，本文将该网络模型作为后续室内定位系统的目标检测基模型，尝试在其基础上增加检测精度及加快检测速度。另外，从实验结果中可以看出 YOLO v2 虽然在精度上高于其它检测模型，但是在检测速度上相差很大，无法用于普通算力计算机上的实时检测模型。

3.4.4 优化后的网络模型在图像分类任务上的表现

为展示该优化策略提升了网络特征提取的质量，本节对比 mobilenet v3 $small_2$ 和主流特征提取网络在 imagenet 数据集 [56] 上的表现，主要从速度和精度两个方面进行比较。速度上采用平均单张耗时，精度上采用 top-1。

表 3-4: 不同网络在 imagenet 数据集上的表现

网络模型	top-1	单张耗时 (ms)
VGG-16	71.5	469.3
inception v3	**76.9**	252.8
mobilenet v1	70.1	124.4
mobilenet v2	71.7	162.0
mobilenet v3 small	67.4	68.4
mobilenet v3 small′	68.3	**62.6**
mobilenet v3 $small_2$	68.6	**62.6**

从表 3-4中可以看出，与优化前的网络 mobilenet v3 small 相比，SE 模块和训练方法优化后的 mobilenet v3 $small_2$ 模型无论是在图片预测速度还是精度方面相比于优化前都有了一定的提升。同时由于我们的目标是在后续的实验中能够训练出一个基于单类别目标检测的模型，该模型在普通的机器上也能够达到近乎于实时性的效果，而这些主流特征提取网络的运行速度将会成为实时性的瓶颈，因此本文优化后的 mobilenet v3 $small_2$ 网络无疑是解决单类目标识别问题的最优选择。

但是在精度方面，mobilenet 系列整体表现一般。一部分原因在于深度可分离卷积自身造成的特征提取损失，另一部分在于模型在精简设计的过程中必然损失一些必要的特征提取步骤。因此如果在 gpu 甚至更高算力的计算机上执行单一类别实时性目标检测任务，mobilenet 系列将不会是最佳的网络骨架模型。

3.5 本章小结

为实现一个单类别快速目标检测的网络，本章在现有网络模型 mobilenet v3 small 的基础上进行结构的优化和调整，同时更改了训练方式使最终获得的模型能够拥有更小的参数量和更优的效果。该模型的最大特点就是运行速度快，因

此考虑将其作为单类别目标检测任务的基模型。如前文所述，本文使用了 SSD 作为前端模型以最大程度上消除小物体难以检测的影响。

实验结果表明，本节的优化算法提升了原模型的检测速度和精度，使得网络检测的实时性基本能够得以保证，但在精度方面略逊于其它网络模型，因此后续研究目标为进一步提升该网络进行单类别目标检测的速度和精度，从而得到应用需要的实时性单类别目标检测的网络模型。同时由于 COCO 数据集自身的复杂性，在实际生活应用的过程中该模型的表现基本要优于本章实验数据。

第四章　单类别目标检测基于投票策略的剪枝优化

由于本文的最终目的是得到一个可以在普通算力计算机上进行实时性单类别目标检测的模型，在第3章的实验中作者发现使用改进后的mobilenet v3 $small_2$ 模型已经接近了实时性检测的要求。在此基础上本章提出一个针对单类别目标检测网络中骨干网络的剪枝算法以进一步提升检测速度。该算法的优点是能够根据类别的不同动态删减不同的卷积模块，达到精简网络模型的效果。同时由于对骨干网络进行单一类别的针对性删减，剪枝后的网络模型在目标类别的精度上与原模型相比几乎没有损失。

为突出实验效果，本章就目前主流的目标检测模型 mobilenet v1-SSD300 进行剪枝优化，可以看到本文的算法将会在特定类别上得到更快的速度和更高的精度，并可以推广到其它特征提取网络中，包括之前使用的 mobilenet v3 网络。同时本章展示了该剪枝算法的可视效果，验证了该算法在神经网络可解释性层面上剪枝的合理性。

更重要的是，在剪枝网络的过程中往往需要进行参数微调 (finetune) 甚至重新训练 (retrain)。很多剪枝后的网络参数微调所需训练的时间不亚于重新训练一个模型。基于本文剪枝策略方法所剪枝后的网络不需要进行参数微调即可得到与原网络相当的效果，大大缩减了网络训练所需要的时间。

4.1　剪枝策略

由于本文需要优化网络模型使其在目标类别上检测速度更快，可想而知减少网络模型的参数量可以达到网络模型更小、运行速度更快的结果，因此我们考虑用网络剪枝的方法降低网络的参数量，这里以 mobilenet v1 为例。

mobilenet v1 的网络参数形式 [55]，如图 4–1所示。一般而言，剪枝策略选择剪掉网络中一层或多层的部分卷积核 (kernel)。事实上，剪掉前面的卷积核对后面相关层无论是通道个数还是特征提取都影响较大。更重要的是，神经网络的参数分布一般集中在后面的卷积层和全连接层 [57]。因此我们对最后一个卷

积层中的卷积核进行剪枝，由图 4-1可知 mobilenet v1 最后一个卷积层中存在 1024 个卷积核，因此尝试删去对检测当前类不重要的卷积核，来达到不影响甚至提升目标类别的精度，同时提高检测速度的目的。

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

图 4-1: mobilenet v1 网络结构

传统的网络剪枝策略包括 L_1/L_2 剪枝方式。一般而言，一个 3×3 大小的卷积核如图4-2所示。L_1/L_2 算法的基本思想是根据式4-1或者式4-2求出该层下所有卷积核的 L_1 或者 L_2 范数，

$$L_1 = \sum_{i=1,j=1}^{i=3,j=3} |a_{ij}| \tag{4-1}$$

$$L_2 = \sqrt{\sum_{i=1,j=1}^{i=3,j=3} |a_{ij}|^2} \tag{4-2}$$

而后根据范数排序以及最终需要剪枝的卷积核数量决定删除相应范数从低到高

a_{11}	a_{12}	a_{13}
a_{21}	a_{22}	a_{23}
a_{31}	a_{32}	a_{33}

图 4-2: 一个通用的 3×3 卷积核形式

的对应卷积核。这种方法的优点是剪枝过程运算的很快，同时十分直观。缺点是没有根据不同的类别侧重来删除最合适的卷积核，在速度加快的同时检测精度会稳定下降 [58]。

本文针对单类目标检测问题，提出一种新的网络模型剪枝方案。该方案可以针对不同类别进行不同卷积核的修剪，以达到精度损失最小甚至检测精度提升的目的，同时由于网络参数的减少，模型的检测速度将会快于原网络模型。更重要的是，该种剪枝方法抛弃了以往的原模型——剪枝——重新训练的路径，剪枝之后的网络不需要参数微调依然可以在该类别上表现很好，省去了大量的重新训练时间。

4.2 基于单类别目标检测的剪枝算法

由于剪枝网络无法提取原网络那么完整的特征信息，往往剪枝后的网络精度会不可避免的有所损失。因此想要做到损失最小甚至与原模型相比略有提高是十分困难的。本节提出了一种新的剪枝算法，该算法可以自行判断网络层中对目标类别不敏感的卷积模块，进行有选择地修剪。最终的实验证明了其对网络精度的影响很小，甚至在某些类别上精度略有提高。

4.2.1 卷积核的重要程度

在神经网络中，同一层的不同卷积核所侧重的特征情况不尽相同：有的卷积核对中间区域的颜色差异比较敏感，有的卷积核对整体图片的色调比较敏感，而有的卷积核则擅长捕捉图片中细小的边角。各司其职的卷积核合理地搭

配在了一起，才能得到一幅图片有效的特征，进而帮助神经网络妥善完成后续的分类和检测任务。

这样的判断在神经网络的类激活热力图中 [59] 也有所体现。如图 4-3[59] 为一个神经网络识别茶壶的热力图 (heatmap)。

图 4-3: 神经网络识别茶壶的热力图

图中显示神经网络主要通过“认出”茶壶的壶盖和壶嘴来判断图片中的物体类别是茶壶 (teapot)。结合上面的讨论可知，如果仅仅需要一个识别茶壶类别的神经网络，那么显然负责识别茶壶盖、壶嘴这两部分的卷积核是最重要的，其次判断壶盖和壶身相连的卷积核也比较重要，其余卷积核则不是那么关键。除此之外，整幅图片和纹理相关的卷积核和其它位置的卷积核也并非可以完全舍弃，毕竟他们能够帮助排除一些其它情况的干扰，但是相比于上面的那些卷积核而言，他们中一小部分可以选择性舍弃，来达到我们精简网络模型，获得更快的单类别目标检测网络的目的。

因此后续的任务可以转化成如何判断该卷积核对识别目标类别的作用，并选择性的将与目标类别相关性较低的卷积核放弃。

4.2.2 卷积核重要性的评价指标

最后一层卷积神经网络输出 1024 个 7×7 的矩阵，记为 $M[0, 1, \ldots, 1023]$。每个对应的 $M[i]$ 由上层的输出结果与该层第 i 卷个积核计算得到。因此 $M[i]$ 矩阵的表现与所对应的卷积核 $K[i]$ 有直接关系。虽然 $M[i]$ 的表现与前面几层的特征提取和卷积运算也微弱相关，但影响最大的还是最后一次参与计算的卷积核 $K[i]$。因此我们采用如下策略来作为卷积核筛选的标准：

首先可以把神经网络的剪枝转化成一个优化问题，如式4-3所示：

$$\min_{W'} \left| C(D|W') - C(D|W) \right| \quad \text{s.t} \quad \|W'\|_0 \leq B \tag{4-3}$$

其中，C 表示模型表现能力，由于剪枝过程中往往会造成精度损失，因此用最小绝对值之差来进行限制约束。D 代表训练样本集合，W 代表原始网络权值，W' 表示剪枝后的网络权值，B 代表网络压缩阈值。在网络参数愈加减少的情况下，需要保留更多的有效信息。与其它网络剪枝策略不同的是，单类目标检测问题需要保留识别某个特定的类别的信息，而不关心网络在其它类别上的表现。设 Z 为卷积神经网络最终的输出结果，N 为当前类别所有图片数量，$A_i(x_n)$ 表示第 i 个卷积核对图片 x_n 的激活矩阵，那么可以求得最终输出基于该卷积核对整幅图片输出的偏导数。将这个偏导数称作该卷积核对这个图片的“激活程度”γ_i，如式4-4所示。

$$\gamma_i = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial Z}{\partial A_i(x_n)} \tag{4-4}$$

依照这种方式可得到最后一个卷积层中每一个卷积核对该类别的激活程度。需要注意的是，激活程度并不一定是一个正值。接下来将分析不同策略对剪枝结果带来的影响以及采取何种策略剪枝方式更为合理。

4.2.3 基于投票法则的策略

在4.2.1节中提到，卷积层的不同卷积核在图像识别、特征提取方面作用各不相同。而根据上节的偏导计算，可以求出某个卷积核对整个图片最终分类的激活情况。假定使用策略 $\gamma' = Y(\gamma)$，其中 Y 代表了某种通过激活程度数值进行剪枝的策略，那么算法伪代码如算法4.1所示：

可以看出，当前剪枝问题最重要的是如何制定合理的策略 Y，能够有效地利用现有的卷积核在目标类别上的激活情况 γ_i，从而做到剪枝过程中尽量删除不重要的卷积核，保留该类别下的优质卷积核。

算法 4.1 单类别目标检测的网络剪枝算法

输入: 初始网络模型参数 W, 目标类别数据集, 数据集大小为 N, 每张图片表示为 x_j, 确定要修剪的卷积核数量 p, 待剪层卷积核数为 m
输出: 剪枝后的网络模型参数 W'

初始化 γ 数组，其中 $\gamma_i = 0, i = 0, 1, \ldots, m$
while 该类别下的图片 x_j **do**
  求出卷积层的最终输出 Z
  求出最后一层每个卷积核对图片的激活情况 $A_i(x_j)$
  $\gamma_i := \gamma_i + \frac{\partial Z}{\partial A_i(x_j)}$
end while
for γ_i in γ **do**
  $\gamma_i := \gamma_i / N$
end for
用策略 Y 将 γ 排序，得到重要度序列 γ_i'
将 γ_i' 序列中前 p 个卷积核删除

基于以上分析，可选策略一共有以下三种：

A. 直接将 γ_i 中数值最低的 p 个卷积核删除，原因是分类的过程中他们的激活程度最低，大都为抑制，不利于网络在该类别上的特征提取和分类，因此需要删掉这些卷积核。
B. 在计算 γ_i 的时候直接计算他们的绝对值之和，即 $\gamma_i' = \epsilon_i = \gamma_i + \left|\frac{\partial Z}{\partial A_i(x_j)}\right|$，而后根据从小到大的排序删除 p 个卷积核。这样做的原因是认为卷积核对目标类别的极端抑制也是有效的激活情况，因此贡献较少的是那些反应程度较低，几乎看不出来是支持还是抑制的卷积核。
C. 获得 γ_i 后求出所有卷积核的平均值 γ_{mean}，而后求出每个卷积核对应的 $\beta_i = |\gamma_i - \gamma_{mean}|$，最后将 β_i 中最小的 p 个卷积核删除。这种策略认为距离平均值较近的卷积核对这幅图片的贡献较小，其余的无论是激活还是抑制都对分类结果影响更大，因此应该删除距离中心值较近的一些卷积核。

实际上，最终输出 Z 与不同卷积核输出 $A_i(x_j)$ 的偏导结果 (激活程度值) 一定程度上反应了该卷积核赞不赞成将这个图片中的特征类别分为目标类。正数可以理解成赞成，负数可以理解成不赞成。正数与负数的绝对值大小恰恰可以反应出赞成与否定的程度，因此整个网络对该图片的评判是基于所有卷积核综合而定的结果。可以做一个恰当的比喻：把在最后一层的卷积核视作专家，让他们为该图片是否属于目标类别来打分，其中正数表示赞同该图片属于目标类别，负数表示不赞同，其数值的绝对值大小表示其赞同/不赞同的程度。本文将

该过程称为投票法则。

基于投票法则，方法 A 是不合理的。因为方案 A 的操作将会删去不赞成程度较高的卷积核。这样的结果是对于非目标类别的图片，只要与相关类别有一点边角或纹理的相似，那么这些多数的赞同票就会倾向于把该类别归为目标类别，这会导致存在着大量的假正例，影响最终的判别效果。

方案 B 和方案 C 的优劣可以通过如下方式判断：假设神经网络不认为该图片属于目标类别，那么最后一个卷积层的卷积核输出很有可能多数为负值。反而言之，若神经网络倾向于将该图片分类为目标类别，很可能多数卷积核激活程度值为正。后续的实验也会证明这一点。基于这个结果我们可以考虑一个比较极端的情况：现在卷积核激活的平均值 (γ_{mean}) 为 5×10^{-5}，是一个正数，而激活值为负数的卷积核大多数输出都在 -10^{-7} 左右，距离零点较近。因此这个时候如果采用方案 B 来将这些比较抑制的卷积核删除，那么将会造成与方案 A 一样的后果。同时方案 B 不考虑卷积核的激活平均值也会导致网络的效果不稳定。最后的实验结果也将证明，大多数情况下方案 B 与方案 C 的结果相差不多，少数情况下方案 C 优于方案 B，方案 C 更加稳定，整体而言优于方案 B 的效果。

由上文的分析可知，方案 B 和方案 C 的效果比方案 A 更好，同时方案 C 比方案 B 更稳定。因此本文中基于单类别目标检测的剪枝策略采用算法 4.2。接下来本文将从可解释性的角度来论述该方案的合理性，同时在后续实验上证明相比于方案 A 和方案 B，方案 C 的表现更加出色。而且与同类型的其它剪枝方案相比，方案 C 对于目标类别的表现更好。同时，不仅在目标检测任务方面，方案 C 剪枝策略在图像分类任务的表现上对目标类别的分类精度也全面超过传统的基于 L_1/L_2 的剪枝方案和 A 剪枝方案。

4.3 投票剪枝策略的可解释性

本节从神经网络的可解释性方面探究上节剪枝策略的合理性，并结合不同的剪枝方式作为比较，从视觉上展示该剪枝方法的通用性。

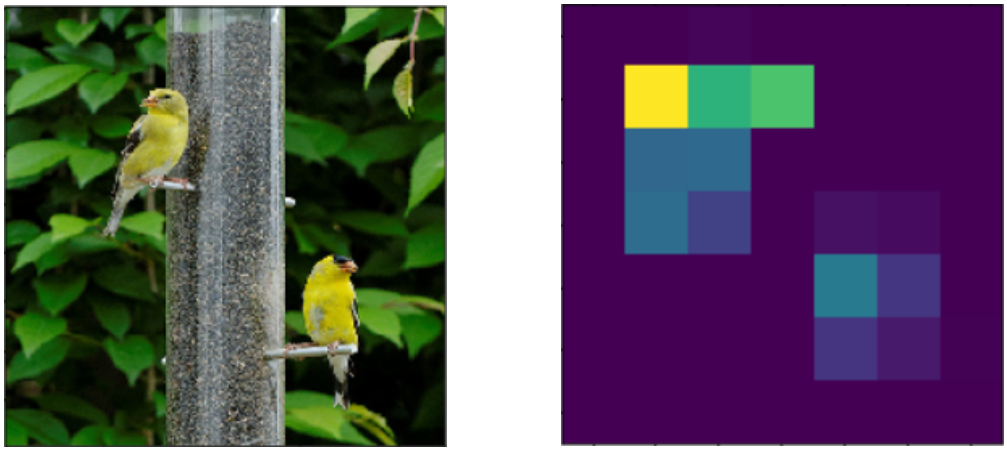
上文已经论述过卷积神经网络中不同的卷积核在特征提取及预测上发挥的作用不同，本节将利用直观的数据和视图展示其中的具体差异。如图 4-4(a)所示是 ILSVRC2012 分类数据集中的一张金丝雀图片，在 mobilenet v1 中识别过程如图 4-4(b)所示：

算法 4.2 使用策略 C 的单类别目标检测网络剪枝算法

输入: 初始网络模型参数 W, 目标类别数据集, 数据集大小为 N, 每张图片表示为 x_j, 确定要修剪的卷积核数量 p, 该层总共的卷积核数量 m

输出: 剪枝后的网络模型参数 W'

初始化 γ, β 数组, 其中 $\gamma_i = \beta_i = 0, i = 0, 1, \ldots, m$
function($kernal\_pruning(W, A_i, p)$)
while 该类别下的图片 x_j **do**
    求出卷积层的最终输出 Z
    求出最后一层每个卷积核对图片的激活情况 $A_i(x_j)$
    $\gamma_i := \gamma_i + \frac{\partial Z}{\partial A_i(x_j)}$
end while
$\gamma_i := \gamma_i / N$
$\gamma_{mean} = 0$
for γ_i in γ **do**
    $\gamma_{mean} := \gamma_{mean} + \gamma_i$
end for
$\gamma_{mean} := \gamma_{mean} / m$
for β_i in β **do**
    $\beta_i := |\gamma_{mean} - \gamma_i|$
end for
将 β 按照数值从小到大排列，删除前 p 个卷积核
end **function**



(a) 神经网络输入原图　　(b) 网络模型最终识别区域

图 4-4: 神经网络对目标类别的激活演示

从图中可以直观的看出，神经网络分类的依据主要在于左上的金丝雀。因为网络对于该金丝雀激活程度更高，对于右下的金丝雀虽然也有一定程度的激活，但相对来说程度更低。再细致观察可以发现，神经网络判断该图存在金丝雀的主要依据为图像中是否存在金丝雀的喙部。上节提到即使是同一层的不同

卷积核，它们对整幅图片的关注重点也不尽相同。就该幅图片而言，mobilenet v1 的最后一层卷积核的特征激活程度如图 4-5 所示。

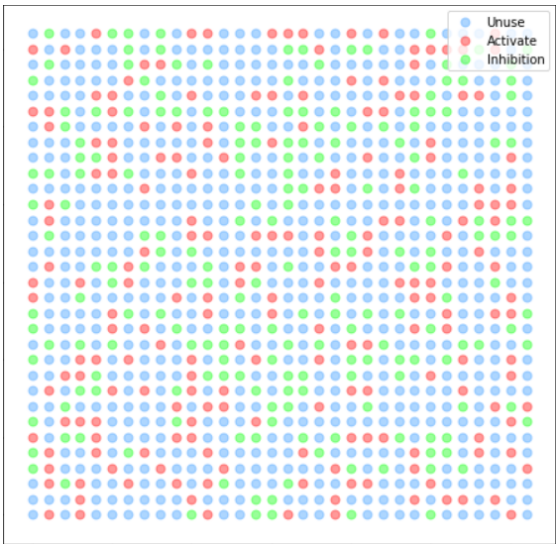


图 4-5: mobilenet v1 最后一层的卷积核对 4-4(a)的激活程度

图 4-5 中每一个圆点代表 mobilenet v1 最后一层的一个卷积核，其颜色代表卷积核对该幅图片的激活程度。本文将激活值位于前 20% 的卷积核定义为激励卷积核，在图中用红色标识。将激活值位于后 20% 的卷积核定义为抑制卷积核，在图中用绿色标识。其它卷积核为非敏感卷积核，在 4-5中用蓝色标识。在这张图中，绿色圆点与蓝色圆点的分界值为 $-2.0374743 \times 10^{-7}$，蓝色圆点与红色圆点的分界值为 2.6893653×10^{-7}。这个数值对我们后续判断卷积核是否发生了激活程度的改变有所帮助。如果采用某种剪枝方式之后再对网络进行参数微调，在得到的新模型上再次检测剩余卷积核的激活程度值，发现激活程度改变过大的卷积核数量较少，则说明更改后的网络模型与未剪枝前的网络模型差异较小，该剪枝策略比较成功。若微调后网络模型有大量的的卷积核发生了激活程度的改变，那就说明剪枝后参数微调的过程中这些卷积核发挥的作用发生了改变，其不稳定性增加，可靠性降低。更重要的是这种情况下关键卷积核被删除的可能性较大，因为剩余卷积核改变了其激活程度来弥补被删掉卷积核的作用。

综上所述，本文提出了卷积核激活程度值求得的法则和一系列剪枝的方案，接下来就要用这些剪枝方案结合卷积核的前后激活程度变化图来说明这种剪枝方式针对单类别特征提取的合理性，和上文讨论的结果一致，如果两种颜色的阈值与之前原模型的结果相近，同时尽可能保证每个卷积核在剪枝前和剪

枝后都保持相同的激活颜色，则可认为在识别目标类的任务中，该卷积核的职能变化不大。但是由于必须剪掉一些卷积核，所以所有卷积核保持相同颜色难以实现，因此可以通过阈值偏移量的大小和卷积核颜色变化的多少来量化表示剪枝方案的可靠性。

首先进行比较的是基于 L_1/L_2 的剪枝方法。一般而言，L_2 的剪枝策略比 L_1 要稳定且效果稍好 [60]。本文先将 L_1/L_2 剪枝方式进行对比，可以看出如图 4-6(a)和图 4-6(b)所示。

(a) 通过 L_1 剪枝后不同卷积核的激活变化情况　　(b) 通过 L_2 剪枝后不同卷积核的激活变化情况

图 4-6: L_1/L_2 剪枝法对最后一层卷积核激活程度的影响

上图所得的结果在模型剪枝之后，参数微调之前。同时作者将剪枝掉的卷积核激活值置为 0，方便相应位置对比和查看。上述讨论中我们知道，在原来颜色重合度两者相差不大的前提下，与图 4-5中阈值更接近的特征提取效果将会更好。在除去剪枝掉的卷积核后，我们可以求出 L_2 与原图颜色重合的卷积核个数为 527，L_1 剪枝法剪枝后与原图颜色重合的卷积核个数为 529。但在阈值方面 L_2 阈值范围为 $[-1.3130453 \times 10^{-5}, 1.266676 \times 10^{-5}]$，而 L1 阈值范围为 $[-4.2683823 \times 10^{-5}, 3.188434 \times 10^{-5}]$，因此整体而言 L_2 剪枝方法所得到的模型效果要略好于 L_1 剪枝法所得到的模型。

本文将会在后续的实验中给出以上所有剪枝方案的阈值结果和匹配情况，同时展示实际目标检测任务中这些剪枝方法对最终结果的影响，来进一步验证以上观点。

4.4 实验及分析

为了验证上文剪枝算法的有效性，作者将从多个模型及数据集对不同剪枝算法进行比较。由于剪枝任务本身不希望把过多的时间浪费在剪枝之后的参数微调上，因此本文将会分别比较这些算法在参数微调之前的性能。与3.4节一致，为了体现投票算法在包含目标类别的图片上保持了良好的特征提取效果，本节将通过目标类别的图像检测和目标分类两个方面来验证投票算法的有效性。具体而言包含以下几个方面：

1. 将 mobilenet v1 网络按不同策略剪枝后未进行参数微调之前，多种策略剪枝后的 mobilenet v1 网络对相同目标类别不同图片的激活情况变化，包括阈值变化和激活神经元个数的变化进行对比；
2. 将 mobilenet v1 网络用不同算法剪枝后未进行参数微调之前，同时作为 SSD 的骨架网络进行目标类别的目标检测，对比 AP 结果，为了验证投票策略的有效性，同时加入 mobilenet v3 $small_2$ 网络上使用方案 C 的剪枝结果作为对比，突出算法的性能优势。
3. 将 mobilenet v1 网络用不同算法剪枝后未进行参数微调之前，对比目标类别图像分类的结果，其中包括 L_1 剪枝策略、L_2 剪枝策略、本文的方案 A、方案 B、方案 C 以及原网络 mobilenet v1 的分类结果，结合实验 1，可以进一步证明方案 C 的投票策略保留了对目标类别重要程度更高的卷积核。

4.4.1 实验设计

本实验的环境配置与3.4.1小节所述基本一致，其中相关实验所使用的数据集、评价方式以及数据处理流程、对比方法等等都与3.4.1相同。为了保证最后一层特征提取的效果同时加快网络的运算速度，在 mobilenet v1 上的相关剪枝实验中选择利用不同的剪枝策略削减削减最后一层 1024 个卷积核中的 320 个。此时削减后的网络参数约为原来的 84.68%，与不同剪枝算法及原网络的特征提取效果对比明显。同时本文将会对第3章得到的 mobilenet v3 $small_2$ 网络进行基于投票策略的剪枝，所削减的卷积核为最后一个卷积层中的 160 个，以此来证明该剪枝算法在不同特征提取网络上都可以有效运行。

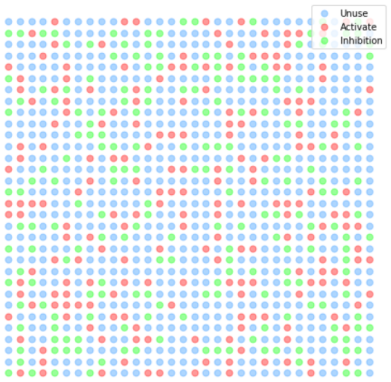
4.4.2 不同剪枝方案对该卷积层的影响

本节展示在不同的剪枝方案下，最后一个卷积层中的卷积核在剪枝前后的变化。为了保证实验的公平性，在本节中所有的剪枝策略都在 mobilenet v1 上进行。其中卷积核的激活程度值由4-4计算得到。本节以 ILSVRC2012 数据集中的金丝雀 (goldfinch) 图片为例，描述不同剪枝方案下卷积核前后激活情况的变化。其中，inhi_bound 和 act_bound 分别表示抑制到不敏感的阈值以及不敏感到激活的阈值，stable 表示卷积核在剪枝前后激活情况不变的个数。参与比较的剪枝算法包括 L_1 剪枝法，L_2 剪枝法，4.2.3中的方案 A(del_mini，A 策略)，方案 B(del_near_zero，B 策略)，方案 C(del_near_mean，C 策略，即上文所述的投票策略)。

表 4-1: 不同剪枝策略对该层卷积核的影响——金丝雀

剪枝策略	剪枝之前		剪枝之后		stable
	inhi_bound	act_bound	inhi_bound	act_bound	
L1 剪枝策略	-2.04×10^{-7}	2.69×10^{-7}	-4.27×10^{-5}	3.19×10^{-5}	529
L2 剪枝策略			-1.31×10^{-5}	1.27×10^{-5}	527
A 剪枝策略			-1.05×10^{-12}	8.31×10^{-12}	400
B 剪枝策略			-4.28×10^{-7}	5.59×10^{-7}	637
C 剪枝策略			$\mathbf{-2.52\times10^{-7}}$	$\mathbf{3.40\times10^{-7}}$	**664**

通过上文分析可知，剪枝操作会带来特征损失导致精度下降，因此剪枝后的网络模型在单类别目标检测任务上的表现与原网络模型相差越小越好。相差幅度小意味着表4-1中 inhi_bound 和 act_bound 与未剪枝前的 mobilenet v1 网络相关值接近。进而从表格4-1中可以看出，无论是从剪枝之后阈值变化幅度方面，还是从激活程度不变的卷积核个数方面，方案 B 和方案 C 都要远远好于其它三种剪枝策略。在大量的实验数据测试后，我们发现在该评价标准下方案 B 和方案 C 稳定好于其它三种剪枝策略，多数情况下方案 C 强于方案 B，这与本文之前的分析相符。实验证明了一般而言投票算法是一个在网络剪枝之后参数微调之前令网络在该目标类别上被剪枝卷积层中其它卷积核激活情况变化最小的剪枝策略。其中方案 A、B、C 所得到的最后一层激活情况图如图4-7所示。通过对比也可发现4-7(c)与原图4-5拟合程度最高。



(a) 方案 A 剪枝后卷积核的激活程度变化

(b) 方案 B 剪枝后卷积核的激活程度变化

(c) 方案 C 剪枝后卷积核的激活程度变化

图 4-7: 不同方案对最后一层卷积核激活程度的影响

4.4.3 不同剪枝方案在目标类别检测任务上的对比

如上文所述，本实验采用 SSD300 作为所有骨干网络模型的检测部分，因为本文最终需要得到一个可以在普通算力计算机上进行实时性单类别检测的网络模型，因此作为特征提取的骨架网络则抛弃了原论文中的 VGG16[24]，选用上文中参数更少的 mobilenet v1 和后续的剪枝网络模型。除此之外，为证明该算法的泛用性，将会用第 3章所述的 mobilenet v3 $small_2$ 以及结合剪枝策略 C 的结果作为综合对比，突出算法的优势和剪枝后的网络速度情况，进一步说明剪枝操作的泛用性。

本次实验选取的评价指标为上文介绍的 AP@0.5，由于任务是检测模型能在可实时性速度的基础上尽可能有精度的提升，因此实验中最重要的性能指标是 FPS。在满足实时性检测的 FPS 需求下检测精度值越高越好。由于在实验中不同剪枝方案均使得剪枝后的网络模型约是原模型参数量的 85%，因此实验结果中 mobilenet v1 模型采用不同剪枝策略所得到的 FPS 结果相等。数据集选用

VOC2012 和 COCO 数据集，目标类别采用以下两种方案：第一次使用两种数据集中的人物、飞机、马及鸟四个类别作为目标类别进行四次实验，每次实验选取其中一种作为正类，其它类别作为负类，正负类别验证集比例为 1 : 4，将实验结果的平均值进行记录；第二次将单独人物类别为目标类别的结果进行记录，为本文后续设计的室内定位系统打下基础。作为精度上限的参考，实验依旧选用 YOLO v2 作为对比模型[①]。与之前实验一致，本文希望目标检测模型能够在满足实时性检测的基础上精度尽可能高。

表 4-2: 不同剪枝策略对单类别目标检测的影响——四种类别的平均值

网络模型	VOC2012	COCO	FPS(每秒帧数)
YOLO v2	**72.7**	**43.81**	12.5464
原 mobilenet v1 ssd 模型	71.20	40.24	16.0845
原 mobilenet v3 $small_2$ ssd 模型	67.32	36.05	22.9544
mobilenet v1 L_1 剪枝	35.57	15.58	16.6477
mobilenet v1 L_2 剪枝	41.48	20.07	
mobilenet v1 ssd-A 方案	40.62	8.36	
mobilenet v1 ssd-B 方案	69.36	39.14	
mobilenet v1 ssd-C 方案	71.92	38.46	
mobilenet v3 $small_2$ ssd-C 方案	69.06	36.92	**24.1361**

从表4-2和表4-3中可以看出，基于投票法则所设计的方案 C 策略在模型关注的单个类别上对网络有速度上的明显提升。对于剪枝前的网络而言，其精度也有部分提升。而且该剪枝策略不需要进行微调和重新训练，省去了网络模型重新训练的时间，是可行有效的针对单类目标检测问题的剪枝策略。同时在 mobilenet v3 $small_2$ 上采用该剪枝策略可以使模型所需要的计算量减小，从而达到前文所述的实时性 (24FPS) 要求。在检测精度方面，可以看出虽然 YOLO v2 在上述模型中达到了最高精度，但是由于其检测速度很慢，无法完成单类别实时性目标检测的要求。本文后续系统核心采用的单类别目标检测模型即选择该实验中的 mobilenet v3 $small_2$ ssd-C 方案网络模型。

[①]注：实验环境依旧为 cpu i7 6700, 显卡为 GTX960。

表 4-3: 不同剪枝策略对单类别目标检测的影响——人物类别

网络模型	VOC2012	COCO	FPS(每秒帧数)
YOLO v2	**72.06**	**42.06**	12.9365
原 mobilenet v1 ssd 模型	71.14	40.88	16.1893
原 mobilenet v3 $small_2$ ssd 模型	68.09	36.12	22.0683
mobilenet v1 L_1 剪枝	33.86	16.49	16.7092
mobilenet v1 L_2 剪枝	41.23	20.31	
mobilenet v1 ssd-A 方案	44.29	9.72	
mobilenet v1 ssd-B 方案	68.94	38.61	
mobilenet v1 ssd-C 方案	71.49	39.71	
mobilenet v3 $small_2$ ssd-C 方案	70.18	37.41	**24.4359**

4.4.4 不同剪枝方案在目标类别分类任务上的对比

4.4.2节通过实验证明了在直接剪枝不经过网络参数微调的条件下，方案 C 中剪枝后网络层中的卷积核普遍变化最小，且整体的激活程度也与原网络近乎一致。这一节将用数据说明与剪枝前网络 inhi_bound 和 act_bound 相近的剪枝后网络表现与剪枝前网络接近，从而进一步证明方案 C 剪枝策略的有效性。本实验选用 3 个图像数据集，分别是 ILSVRC2012 数据集中的金丝雀类别、cifar-10 中的马类别以及 Caltech 101 中的飞机类别。网络模型依旧选择 mobilenet v1 ssd 网络模型。具体精确值和 F1 如表4-4、4-5所示。由于不同数据集中其它类别数据量与目标类别数据量比例不一致，而且只需考虑目标类别的检测情况，因此实验中测试集的目标类别: 非目标类别比例设定为 1 : 4。

从表4-4和表4-5中可以看出，方案 B 和方案 C 在网络模型剪枝后在目标类别上进行图像分类的结果影响较少，甚至出现 F1-score 略微超过原网络模型的情况。而方案 A 虽然真正例数较多，但是由于假正例 (FP) 严重多于方案 B 和 C，因此方案 A 整体而言误判较高，导致 F1 很低，算法性能较差。而 L_1/L_2 剪枝法比较而言 L_2 要略优于 L_1 剪枝策略，但仍与方案 C 有差距，与方案 B 整体比较而言方案 C 略优。

值得注意的是，当网络模型在数据集上的整体表现已经较好的情况下 (如表4-5)，该剪枝策略的表现会略逊原网络模型。整体表现出 FP 数量增多，有过拟合的趋势，不过结果仍然在可接受范围之内。

表 4-4: 不同剪枝策略对网络分类的影响——金丝雀、飞机

剪枝策略	金丝雀		飞机	
	Top-1 acc	F1-score	Top-1 acc	F1-score
L_1 剪枝策略	0.48	0.657895	0.557	0.284426
L_2 剪枝策略	0.50	0.675325	0.625	0.630911
A 剪枝方案	**0.98**	0.902655	**0.932**	0.703564
B 剪枝方案	0.92	0.959184	0.889	0.891765
C 剪枝方案	0.94	**0.969697**	0.894	**0.896547**
原模型 (mobilenet v1)	0.80	0.891304	0.876	0.890224

表 4-5: 不同剪枝策略对网络分类的影响——马

剪枝策略	Top-1 acc	F1-score
L_1 剪枝策略	0.593	0.73346
L_2 剪枝策略	0.655	0.77930
A 剪枝方案	**0.989**	0.769650
B 剪枝方案	0.944	0.927764
C 剪枝方案	0.958	0.926947
原模型 (mobilenet v1)	0.916	**0.936127**

该实验结果也进一步验证了上一节所讨论的阈值变化越少，激活情况不变的卷积核越多，最终模型在目标类别上精度变化越小的猜想。

4.5 本章小结

本章从单类目标检测问题出发，分析当前目标检测网络的内部结构，判断其网络特征提取部分最高的计算消耗在最后一层卷积的计算中。而后提出三种策略，将最后一个卷积层中的卷积核进行针对目标类别的合理删减，来达到比原模型速度更快，精度更高的目的。本章实验中削减了最后一个卷积层中的 320 个卷积核，使整个特征提取网络参数量减少 15.32%，模型的运行速度得以提升。

其次，通过实验和分析发现最后一个卷积层中不同卷积核在不同类别的图片上产生了不一致的激活，但是对于同一个类别而言，同一个卷积核激活情况

变化不大。最终得出了每个卷积核负责判定的图片区域、纹理都不同的结论，进而针对目标类别的具体特征进行剪枝。文中先是通过偏导来量化每个卷积核对该类别的激活情况，激活值为正则理解成激活/支持该图片分类为目标类别，值为负则理解成抑制/反对该图片分类为目标类别。在获得量化信息后提出基于投票策略的 A/B/C 方案，经过分析和实验得出了方案 C 最优的结果，具体过程如算法 4.2所示。同时将上述方案与经典 L_1/L_2 剪枝算法进行对比，确定了方案 C 剪枝后的卷积核激活阈值范围更接近原网络，激活发生变化的卷积核更少，验证了我们先前的猜想。

最后通过目标检测和图像分类实验，证明了使用方案 C 剪枝后的网络无论是对比其它剪枝网络还是原网络都有其速度、精度上的优势。另外，相比于 L_1/L_2 剪枝策略，基于投票策略的剪枝算法在剪枝后不需要进行重新训练，可以直接用于目标类别的特征提取和检测，大大缩减了剪枝后有效模型的生成时间。

本章节的算法思想与本文第3章所讨论的内容有一部分相似，都是探讨经过不同卷积核卷积后得到的特征图像之间的关系。不同之处在于 Squeeze-and-Excitation 网络模块是在得到的特征图基础之上训练部件，该部件可以显式地描述输出特征图不同通道的重要程度，而后将每个特征图像的重要程度乘以特征图像中的数值作为下一层的输入。而第4章所提出的算法是通过得到的特征图反推卷积核，进而确定卷积核对于目标类别的关键程度，从而进行单类别目标检测剪枝操作。4.4.3节的相关实验表明，结合上述两种策略的 mobilenet v3 $small_2$ ssd-C 方案网络模型可以在普通算力的计算机上达到实时检测的效果，同时在精度方面优于原 mobilenet v3 small ssd 网络模型。

第五章　基于单类别目标检测模型的室内定位系统

为了检验上述优化方案的有效性，本章将介绍一个实时性单类别目标检测问题的具体实例，即室内定位系统。该系统使用上述两个方案优化后的检测网络对室内人员位置进行实时定位，且能够在普通算力计算机上运行，展示了该优化方案的实用性。

5.1　室内定位系统的介绍

5.1.1　相关背景

近年来，随着社会的发展，市面上对活动范围内的人物监测需求也逐渐增多。无论是商场超市的人物流动范围监测，还是工厂工地中检查工人们的活动区域是否合理，甚至于国防的监控布防，都存在人物定位和监控的需求。在这样的大环境下，室内定位系统应运而生。

室内定位系统指的是可以准确判断室内人物位置坐标的系统，具体人物与人物之间不需要加以区分。其概念提出较早，如今已经有很多较为成熟的技术。但不同技术之间各有优劣，暂未出现一个优于所有解决方案的系统，因此现如今主流的室内定位系统仍然有多种表现形式。市面上成熟的室内定位系统原理不外乎以下几种：

首先是 wifi 定位技术。wifi 定位技术是目前十分成熟且应用广泛的技术[61]，目前许多公司 (微软、苹果等等) 在这个领域都有自己的研究机构。其基本原理是根据现在 wifi 已经普及的现状，在室内布控 wifi 信号发射器。现今大多数基于 wifi 的室内定位法用“近邻法”判断，即距离哪个热点最近就可以大致判断为在哪个位置。想要精确设备或是人的位置往往采用在室内配置多个 wifi 信号发射器，用交叉定位 (三角定位)[62] 等基础几何知识来判断。基于 wifi 的室内定位有很多优点，包括系统布置简单，现如今大多数移动设备都具备连接 wifi 的功能，同时对复杂的大范围定位友好，方便组网。但 wifi 定位也有明

显的问题：一是 wifi 定位具有一定的欺骗性，可以利用 wifi 定位移动设备而非人类来“欺骗”wifi 网络定位到了人，实际上可能移动设备与人相距甚远；二是 wifi 定位具有明显的同频干扰现象。不同系统之间会互相影响，其它的同频信号发射器也会影响 wifi 定位系统，使其失效。

其次是蓝牙定位技术。蓝牙 (bluetooth) 定位技术 [63] 主要基于信号场强指示 (Received Signal Strength Indication，RSSI) 值来判断目标距离基点的远近。蓝牙定位一般分为终端侧定位和网络侧定位，和 wifi 定位一样，蓝牙定位如果想要做到精确定位仍需要在室内布控多个信号发射点。一般而言，目标与发射点的距离和信号强度值如式 5-1所示。其中，|RSSI| 是该处 (x,y) 所接收到的信号强度，A 为距离信号一米处接收到的信号强度，n 是环境衰减因子。由此可以求出目标与该发射器 (x_1,y_1) 之间的距离 d_1。

$$d_1 = 10^{\frac{|\mathrm{RSSI}|-A}{10n}} \tag{5-1}$$

同理可求出与发射器 (x_2,y_2) 和 (x_3,y_3) 的距离 d_2,d_3，即可求出目标的坐标 (x,y)。根据线性代数相关知识可知有式 5-2，化简后可求出坐标 (x,y)，如式 5-3所示。

$$\begin{bmatrix}(x-x_1)^2+(y-y_1)^2\\(x-x_2)^2+(y-y_2)^2\\(x-x_3)^2+(y-y_3)^2\end{bmatrix}=\begin{bmatrix}{d_1}^2\\{d_2}^2\\{d_3}^2\end{bmatrix} \tag{5-2}$$

$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}2(x_1-x_3)\times 2(y_1-y_3)\\2(x_2-x_3)\times 2(y_2-y_3)\end{bmatrix}^{-1}\begin{bmatrix}{x_1}^2-{x_3}^2+{y_1}^2-{y_3}^2+{d_3}^2-{d_1}^2\\{x_2}^2-{x_3}^2+{y_2}^2-{y_3}^2+{d_3}^2-{d_2}^2\end{bmatrix} \tag{5-3}$$

相比于 wifi 定位方式，基于蓝牙的室内定位系统精度更高，在很多大型工厂中有所使用。但蓝牙定位系统也有其自身的缺陷。比如蓝牙信号受噪声干扰较为明显，被障碍物遮挡后还会出现阴影效应，其接受信号强度下降会影响对距离的判断进而造成偏差。而且蓝牙定位也需要携带接收或发射信号设备，存在被欺骗的风险。

现如今射频识别定位技术逐渐进入了大众的视野。射频识别定位 (Radio Frequency Identification，RFID) 技术 [64] 指的是利用无线射频方式进行双向的射频通信，以此来达到确定物体相对于基站位置的定位方法，是物联网的支撑技术。RFID 定位方法一般分为有源定位和无源定位，在生产生活中有源定位

居多。一般将 RFID 标签贴在物品或者人物外套上，通过 RFID 阅读器来对人或者物体进行定位。RFID 定位方法有很多优点，比如 RFID 标签可以重复利用，间接地减小定位成本；在遮挡条件下不影响 RFID 的准确定位 (无视距干扰)；更重要的是，RFID 定位方法的精度有所保障，一般而言误差在 5 米以内。但是 RFID 定位法有其自身的缺陷，主要是射频距离过短，一般而言在几十米以内。因此较大的区域下无法使用该定位方式成为了其通用性的瓶颈。

5.1.2 基于目标检测模型的室内定位系统优缺点

在本系统出现以前，也曾有研究者将基于神经网络的目标检测模型用于室内中人物的检测。整体而言，基于目标检测模型的室内定位系统主要具备以下优点：

- 配置代价低：一般而言，训练一个网络模型就可以重复用于室内定位，所要修改的无非是坐标的规定以及仿射变换 [65] 等等，因此整体而言，基于目标检测模型的室内定位系统所需要的经济花费较低。
- 定位相对准确：在预先的计算完成后，只需要运行目标检测模型即可自动的判断室内中感兴趣的类别，从而计算其相对于室内的坐标。而且精度误差往往在两米以内，这比前面介绍的几种室内定位方式要准确很多。
- 抗干扰能力强：前面说过，无论是基于 wifi 还是蓝牙的室内定位系统都存在同频干扰和信号接收差的风险，而基于神经网络目标检测模型的室内定位系统由于整体而言是个离线的模型计算，不涉及任何信号的发射和输出，因此抗干扰能力很强。
- 锁定性强：无论是基于 wifi 还是蓝牙，甚至 RFID 定位方法都有接收信号 (移动设备、蓝牙接收设备、RFID 标签等等) 遗漏甚至脱离目标的风险。在生活中可能有的人没有连接 wifi 或是蓝牙，甚至身上没有携带手机，RFID 标签有无意间贴错位置的风险。但是在基于目标检测的室内定位系统中将不会存在这些问题，基于目标检测的定位系统仅仅关心目标类别，而不关心目标类别身上是否携带其它物品。

但是基于目标检测模型的室内定位系统也有其自身的缺陷，主要包括模型能力不稳定，有些模型可能训练结果较差出现误检；运行速度较慢，毕竟平时生产应用的计算机运算能力较低，很难达到实时性目标检测的要求；无法摆脱

视距限制，在遮挡情况下增加误判的可能等等。

本文基于可实时性检测问题的思考，结合前文所述的优化方法训练了一个可以在普通计算机上完成室内实时检测人物位置的系统，在一定程度上缓解了上述基于目标检测模型室内定位系统的缺陷。

5.2 基于单类目标检测模型的室内定位系统

上文详细介绍了基于目标检测模型的室内定位系统相比于其它方式进行室内定位的优缺点，本节将用单类别目标检测模型作为我们室内定位系统的内核，结合其它计算机视觉方面的技术来搭建一个室内定位系统，这个室内定位系统只识别人物类别，在普通算力的计算机上可以达到实时性的效果。

5.2.1 系统整体架构

系统的整体架构如图 5-1所示：

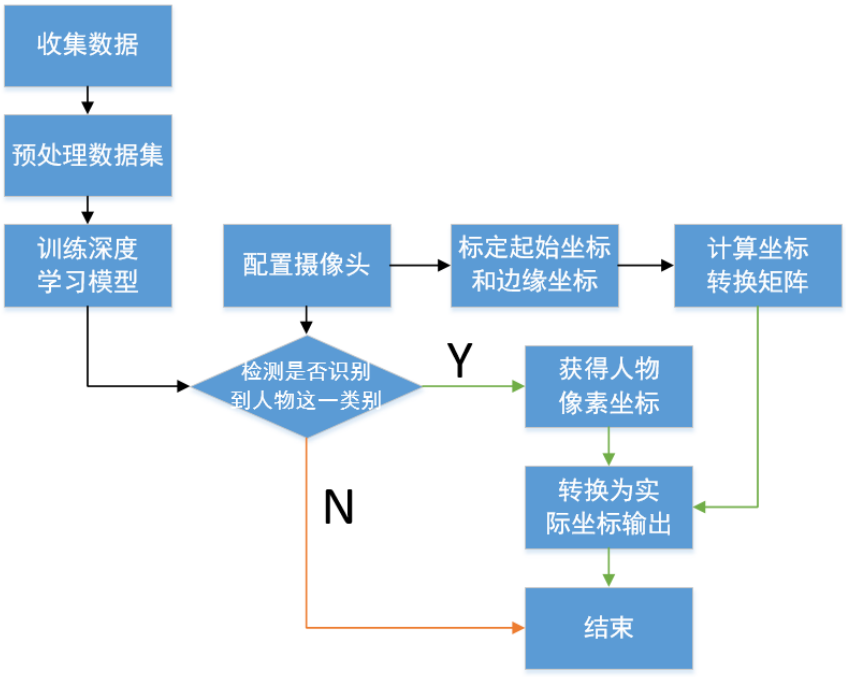


图 5-1: 使用单类别目标检测模型搭建的室内定位系统流程图

整个系统可以分为单类别目标检测的模型训练和室内实际坐标的标定两个部分。其中模型训练的部分包括数据集选取，数据集图片采样，网络模型的优

化等等，在坐标标定方面包括仿射矩阵的计算和实际坐标线性映射等等。具体的实现细节和逻辑由下文详述。

5.2.2 系统具体实现

本节只针对相关技术 (模型训练、坐标标定) 来进行介绍，至于摄像头位置、型号等等细节可以根据实际应用进行改变，与本系统的可行性并无直接关联。但摄像头的清晰度确实会影响目标检测模型的识别能力。一般而言建议使用稳定 60 帧 1080p 的摄像头，同时配置在室内高处，方便后续的拍摄和坐标转换。

5.2.2.1 训练目标检测模型

由于本实验需要一个只识别人物类别的目标检测模型，因此选取第3章优化后的 mobilenet v3 $small_2$ 模型作为目标检测的骨干模型，同时引入第4章的剪枝策略对其进行进行卷积核的删减，使其检测速度进一步提升。头部网络使用 SSD，因此整体模型就是通过方案 C 剪枝后的 mobilenet v3 $small_2$-SSD 结构。

本系统在数据集选取方面选择 openimage[66] 数据集。openimage 是 google 发布的目前最大的含标注的目标检测数据集，数据集中有大约 600 个类别[①]，不同类别中的数据数量不一致，每个类别数据大小一般为 3G-100G 不等。除了用于训练和测试的图像之外，类别所属的框选范围由另外的文件给出，为了方便校对和训练。由于本实验只需要检测人物类别，因此在实验中下载所有类别的 openimage 2018_04 数据集，并将其人物类别作为正例，其它所有类别均匀采样使其整体数量大约是正例的 3 倍。初始学习率 $\alpha = 0.01$，训练轮数为 100。其中训练过程如图 5-2所示。

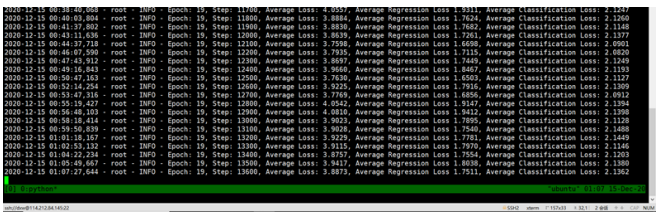


图 5-2: mobilenet v3 $small_2$ - SSD 训练截图

训练结束后获得了 mobilenet v3 $small_2$-SSD 模型，训练效果如图 5-3所示。后续实验以此模型展开。

[①]具体类别及标号见 https://storage.googleapis.com/openimages/2018_04/class-descriptions-boxable.csv.

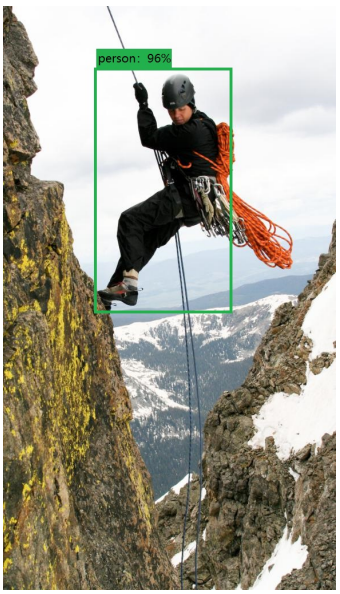


图 5-3: mobilenet v3 small$_2$-SSD 在图片上的测试结果

5.2.2.2 计算仿射变换矩阵

在有了可靠的单类别目标检测模型后，下一步需要的是确定室内空间的实际坐标。由于摄像头往往配置在一间屋子的角落处，其照射的地板区域不是一个均匀的矩形。实际上，靠近摄像头区域两个像素点间的距离较近，远离摄像头区域的两个像素点间距离较远。因此需要利用一定的几何形变知识将摄像头拍摄的不均匀四边形转换成为一个垂直向地面照射的均匀矩形，在本系统中采用仿射变换实现形变的转换。

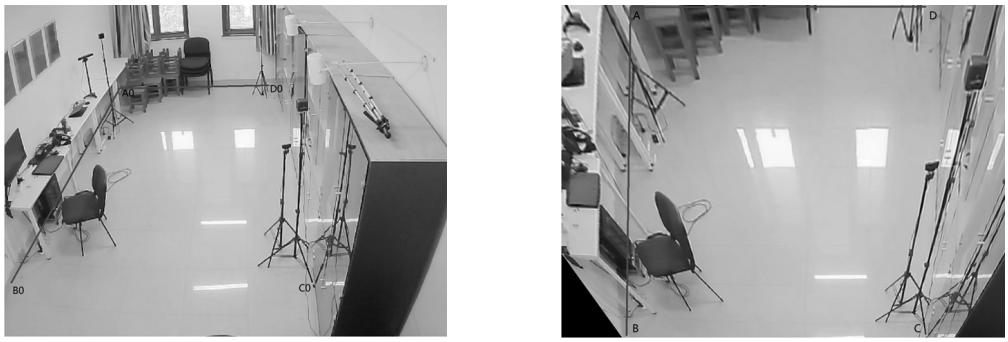
从数学的角度来讲，仿射变换指的是将坐标在原本的向量空间中进行一次平移和一次线性变换，使其转换到另一个向量空间的过程。具体而言，假设原坐标为 (x,y)，转换后的坐标为 (x',y')，那么仿射变换如式 5-4所示：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{5-4}$$

其中，c_1, c_2 起到平移的作用，a_1, b_2 起到缩放的作用，b_1, a_2 影响旋转。仿射变换具有很多优良的性质，其中主要包括二维图形之间的相对关系在仿射变换前后保持不变，同时相互平行的直线在仿射变换后依旧平行。因此通过这种方式，可以将原图中的形状转换成更容易处理的几何模型。重要的是在本实验系

统中会让坐标的表示更加精确。仿射变换的操作条件是必须提供转换前的非共线三点坐标以及转换后的非共线三点坐标。

如图5-4(a)所示为本实验初始的摄像头拍摄图像，左上角 $A0$，左下角 $B0$，右上角 $D0$ 为实验所选取的仿射变换之前非共线三个点，像素坐标分别为 $(351,256),(800,239),(20,845)$。由于摄像头输入的图像较大，像素本身不适合作为实际的坐标，因此设定仿射后的像素坐标与实际坐标相当，后续通过进一步线性变换转化成最终输出的实际坐标。每个点仿射后对应的点位为图5-4(b)中 $A(200,0),B(1100,0),D(200,1000)$，通过 opencv [67] 计算出仿射变换矩阵 M，从而实现摄像头拍摄图片竖直于地面。可以看出两条墙面与地面的交线 AB 和 AD 在仿射变换的作用下已经基本垂直。



(a) 仿射变换前摄像头采集到的图像　　(b) 仿射变换后摄像头采集到的图像

图 5-4: 仿射变换对拍摄图像的影响

在仿射变换操作后，系统已经可以将图像中的像素坐标 (x,y) 转化成仿射变换后的像素坐标 (x',y')。要想输出实际坐标还需要进行最后一步线性转换，也就是人为规定房间的原点位置，横纵坐标方向以及单位距离。在这里实验将图5-4(b)中的 A 点设定为原点，实际坐标为 $(0,0)$，B,C,D 的实际坐标分别为 $(0,20),(15,20),(15,0)$。从而像素坐标到实际坐标 (x_m,y_m) 的转换由式5-5来表示：

$$(x_m,y_m)=\left(\frac{x'-x'_0}{x'_1-x'_0}(x_{1m}-x_{0m}),\frac{y'-y'_0}{y'_1-y'_0}(y_{1m}-y_{0m})\right),(x'_1\neq x'_0,y'_1\neq y'_0) \tag{5-5}$$

其中，(x',y') 表示我们预测的目标仿射变换后的像素坐标，(x'_0,y'_0) 为原点仿射变换后的坐标，(x_{0m},y_{0m}) 为原点实际坐标 $(0,0)$。(x'_1,y'_1) 表示除了原点之外的任意我们已知的实际坐标点，实际坐标表示为 (x_{1m},y_{1m})。注意其中 $x'_1\neq x'_0,y'_1\neq y'_0$。因此我们选择图5-4(b)中的 C 点作为我们 (x'_1,y'_1) 的参考点，这里需要注意的是实际坐标与图像中的像素坐标相互颠倒，因此将对应的仿射变

换后的像素坐标 (x', y') 与 C 点仿射后像素坐标及 C 点实际坐标带入可推出最终计算公式，如式5-6所示。

$$(x_m, y_m) = \left(\frac{y'-0}{1000-0} \times 15, \frac{x'-200}{1100-200} \times 20\right) = \left(\frac{3y'}{1000}, \frac{x-200}{45}\right) \tag{5-6}$$

如此整个系统就完成了坐标转换。在目标检测模型预测的过程中，输出的预测框包含人物边界的左上角 (x_{lt}, y_{lt}) 和右下角 (x_{rb}, y_{rb})。由于系统的最终输出结果是室内中人物的位置，需要输出人物脚掌所在地面相对于室内的坐标，因此选择整个框的下方正中心为人物脚掌所在位置，即 $(x, y) = (\frac{x_{lt}+x_{rb}}{2}, y_{rb})$。进一步通过计算得出的仿射变换矩阵 M 将 (x, y) 转化为仿射后的像素坐标 (x', y')，最后根据式 5-6将 (x', y') 转化成实际坐标 (x_m, y_m) 最终输出。

5.2.3 系统效果反馈

目前整个系统已经成功在实验室部署完毕，并已参与使用。经过一段时间的实测发现整个系统运行稳定，无明显缺陷。重要的是，整个系统只需要在确定摄像头位置之后计算一次仿射变换矩阵 M 即可，同时训练模型也可以重复利用，具有很大的迁移空间。

整个系统由屏幕实时输出房间内的情况，其中人物位置和预测概率由目标检测模型给出，并在屏幕上实时显示。具体测试图片可如图5-5所示。

从图5-5中可以看出，模型精确地预测了房间内的人物并给出预测概率，同时相比于人物在室内的位置 (左上角为 $(0, 0)$，右下角为 $(15, 20)$) 也有一个较为准确的判断。整体而言，模型具有预测精度高，预测效果良好，可实时显示，迁移性强操作方便等优点。在本实验中所选用的计算机处理器为 cpu i7 4700，算力普通更加凸显了本系统在检测速度上的优势。

长期测试表明，在无明显遮挡或者多人重叠的情况下，错检漏检率可以降低到 1% 左右，完全符合室内定位系统准确性的标准。在精度方面误差在 2 米以内，也好于大多数现有的室内定位系统。可以说本文所述的基于单类目标检测模型的室内定位系统基本解决了文中最开始提出的检测速度和精度的问题，体现了第3章与第4章中优化算法的有效性和实用性。

图 5-5: 室内定位系统效果图

5.3 本章小结

本章介绍了一个基于单类别目标检测模型设计的室内定位系统，并与现今主流的其它室内定位系统对比，突出了该系统的优势。同时该室内定位系统模型使用了第3章和第4章的优化算法，证实了之前文章中算法及优化策略的有效性和实用性。该模型的检测精度较高，实用性强，可以为商场、工地、教室等公共场所提供技术支持。

第六章　总结与展望

目标检测技术是计算机视觉领域的热门研究问题，但对于实时性单类目标检测问题研究较少。本文针对单一类别的目标检测问题，从现有的网络模型出发，试图将其改进得到一个可以在普通算力计算机上能够实时性运行的单类别目标检测网络模型。本文提出两个优化策略，分别从特征提取和网络剪枝的角度减少网络模型的参数量，最终成功得到了可以快速完成单类别目标检测的网络模型，并使用该网络模型搭建了一个基于单类别目标检测模型的室内定位系统，且成功投入使用。本文的主要贡献如下：

1. 面对单类别检测问题的 mobilenet v3 优化。本文提出了一种基于 mobilenet v3 网络模型的优化策略，修改了 depthwise、SE 模块、和 pointwise 的实现和它们之间的连接顺序，并加入 BatchNorm 模块使网络表现更加稳定。与 mobilenet v3 small[9] 相比，优化后的 mobilenet v3 small 模型参数量减少约 7%，且在图像分类任务上的表现更好。同时在实验中使用 warmup 训练方法，使模型的特征提取能力进一步提升，最终获得 mobilenet v3 $small_2$ 网络模型，成为下一步工作的基础。
2. 面向单类别目标检测问题的网络剪枝算法。本文提出了一种针对单类别目标检测的网络剪枝方式，该方法减少了目标检测模型中骨干网络最后一个卷积层的卷积核数量，适用于目前所有骨干——头部类型的目标检测网络。利用这种方式可以使得剪枝后的网络具有更快的检测速度，同时与剪枝前网络相比，提升了目标类别上的检测精度。本文还从可解释性和可视化的角度出发讨论了该剪枝策略的合理性。实验表明在该剪枝算法的作用下，新的网络模型无论是作为目标类别图像分类还是作为目标检测骨干网络都有速度和精度上的提升。
3. 基于单类别目标检测模型的室内定位系统。本文在上述两个工作的基础上，搭建了一个只识别人物类别的室内定位系统，该系统可以检测室内人的位置，且能满足室内监测的精度需求，更重要的是，该系统可以在普通算力的计算机上实时性运行。

按照本文的研究路线，可以继续深入展开研究。首先，对剪枝的理解可以

进一步向可解释性方面延伸，比如具体探究哪些卷积核用于识别纹理，哪些卷积核用于识别边角临界区域。在这个思路下可以针对不同的目标类别直接剪枝，而不需要人为计算偏导数和梯度。其次，优化后的 mobilenet v3 $small_2$ 模型还有优化空间，尤其是卷积核之间运算的 Squeeze and Excitation 区域如何更加合理设计；最后，基于单类别的目标检测网络精度还可以进一步提升。具体而言可以根据最终实用场景的不同在数据集上进行扩充和修改，针对每一个特殊场景训练一个特定的模型。甚至抛弃仿射变换思维，设计一个另外的网络自动识别墙角来进行坐标的较定，进一步做到全自动室内检测。

参考文献

[1] IRWIN D E. Visual memory within and across fixations[G] // Eye movements and visual cognition. [S.l.] : Springer, 1992 : 146 – 165.

[2] YILMAZ A, JAVED O, SHAH M. Object tracking: A survey[J]. Acm computing surveys (CSUR), 2006, 38(4) : 13 – es.

[3] VIOLA P, JONES M. Rapid object detection using a boosted cascade of simple features[C] // Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 : Vol 1. 2001 : I – I.

[4] FREUND Y, SCHAPIRE R, ABE N. A short introduction to boosting[J]. Journal-Japanese Society For Artificial Intelligence, 1999, 14(771-780) : 1612.

[5] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C] // 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) : Vol 1. 2005 : 886 – 893.

[6] FELZENSZWALB P, MCALLESTER D, RAMANAN D. A discriminatively trained, multiscale, deformable part model[C] // 2008 IEEE conference on computer vision and pattern recognition. 2008 : 1 – 8.

[7] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2014 : 580 – 587.

[8] YOUNIS A, SHIXIN L, JN S, et al. Real-time object detection using pre-trained deep learning models MobileNet-SSD[C] // Proceedings of 2020 the 6th International Conference on Computing and Data Engineering. 2020 : 44 – 48.

[9] HOWARD A, SANDLER M, CHU G, et al. Searching for mobilenetv3[C] // Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019 : 1314 – 1324.

[10] AARTHI R, HARINI S. A survey of deep convolutional neural network applications in image processing[J]. Int. J. Pure Appl. Math, 2018, 118 : 185 – 190.

[11] ZHANG Y, PEZESHKI M, BRAKEL P, et al. Towards end-to-end speech recognition with deep convolutional neural networks[J]. arXiv preprint arXiv:1701.02720, 2017.

[12] OTTER D W, MEDINA J R, KALITA J K. A survey of the usages of deep learning for natural language processing[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020.

[13] CASTRILLÓN M, DÉNIZ O, HERNÁNDEZ D, et al. A comparison of face and facial feature detectors based on the Viola–Jones general object detection framework[J]. Machine Vision and Applications, 2011, 22(3) : 481 – 494.

[14] FARSHBAFDOUSTAR M, HASSANPOUR H. A locally-adaptive approach for image gamma correction[C] // 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010). 2010 : 73 – 76.

[15] SAID Y, ATRI M, TOURKI R. Human detection based on integral Histograms of Oriented Gradients and SVM[C] // 2011 International Conference on Communications, Computing and Control Applications (CCCA). 2011 : 1 – 5.

[16] REN X, RAMANAN D. Histograms of sparse codes for object detection[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2013 : 3246 – 3253.

[17] GIRSHICK R. Fast r-cnn[C] // Proceedings of the IEEE international conference on computer vision. 2015 : 1440 – 1448.

[18] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. arXiv preprint arXiv:1506.01497, 2015.

[19] ZHONG Y, WANG J, PENG J, et al. Anchor box optimization for object detection[C] // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020 : 1286 – 1294.

[20] FRIEDEN B R. Image enhancement and restoration[G] // Picture processing and digital filtering. [S.l.] : Springer, 1975 : 177 – 248.

[21] SUDOWE P, LEIBE B. Efficient use of geometric constraints for sliding-window object detection in video[C] // International Conference on Computer Vision Systems. 2011 : 11 – 20.

[22] DAI J, QI H, XIONG Y, et al. Deformable convolutional networks[C] // Proceedings of the IEEE international conference on computer vision. 2017 : 764 – 773.

[23] ZHU X, HU H, LIN S, et al. Deformable convnets v2: More deformable, better results[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019 : 9308 – 9316.

[24] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector[C] // European conference on computer vision. 2016 : 21 – 37.

[25] NEUBECK A, VAN GOOL L. Efficient non-maximum suppression[C] // 18th International Conference on Pattern Recognition (ICPR'06) : Vol 3. 2006 : 850 – 855.

[26] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016 : 779 – 788.

[27] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017 : 7263 – 7271.

[28] REDMON J, FARHADI A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[29] BOCHKOVSKIY A, WANG C-Y, LIAO H-Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[30] LAW H, DENG J. Cornernet: Detecting objects as paired keypoints[C] // Proceedings of the European conference on computer vision (ECCV). 2018 : 734 – 750.

[31] ZHAO S, GUO Y, SHENG Q, et al. Heatmap3: an improved heatmap package with more powerful and convenient features[J]. BMC bioinformatics, 2014, 15(10) : 1 – 2.

[32] DUAN K, BAI S, XIE L, et al. Centernet: Keypoint triplets for object detection[C] // Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019 : 6569 – 6578.

[33] ZHOU X, ZHUO J, KRAHENBUHL P. Bottom-up object detection by grouping extreme and center points[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019 : 850 – 859.

[34] ZOU Z, SHI Z, GUO Y, et al. Object detection in 20 years: A survey[J]. arXiv preprint arXiv:1905.05055, 2019.

[35] LIN T-Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[C] // Proceedings of the IEEE international conference on computer vision. 2017 : 2980 – 2988.

[36] ELSKEN T, METZEN J H, HUTTER F, et al. Neural architecture search: A survey.[J]. J. Mach. Learn. Res., 2019, 20(55) : 1 – 21.

[37] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018 : 4510 – 4520.

[38] AVENASH R, VISWANATH P. Semantic Segmentation of Satellite Images using a Modified CNN with Hard-Swish Activation Function.[C] // VISIGRAPP (4: VISAPP). 2019 : 413 – 420.

[39] ZHANG X, ZHOU X, LIN M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018 : 6848 – 6856.

[40] DUMOULIN V, VISIN F. A guide to convolution arithmetic for deep learning[J]. arXiv preprint arXiv:1603.07285, 2016.

[41] YAROTSKY D. Error bounds for approximations with deep ReLU networks[J]. Neural Networks, 2017, 94 : 103 – 114.

[42] CHOLLET F. Xception: Deep learning with depthwise separable convolutions[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017 : 1251 – 1258.

[43] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018 : 7132 – 7141.

[44] FRANKLE J, SCHWAB D J, MORCOS A S. Training batchnorm and only batchnorm: On the expressive power of random features in cnns[J]. arXiv preprint arXiv:2003.00152, 2020.

[45] MASTERS D, LUSCHI C. Revisiting small batch training for deep neural networks[J]. arXiv preprint arXiv:1804.07612, 2018.

[46] SHETTY S. Application of convolutional neural network for image classification on Pascal VOC challenge 2012 dataset[J]. arXiv preprint arXiv:1607.03785, 2016.

[47] LIN T-Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C] // European conference on computer vision. 2014 : 740 – 755.

[48] TORRALBA A, FERGUS R, FREEMAN W T. 80 million tiny images: A large data set for nonparametric object and scene recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2008, 30(11) : 1958 – 1970.

[49] TORRALBA A, EFROS A A. Unbiased look at dataset bias[C] // CVPR 2011. 2011 : 1521 – 1528.

[50] 周志华. [M] // 机器学习. [S.l.] : 清华大学出版社, 2016 : 29 – 33.

[51] GOUTTE C, GAUSSIER E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation[C] // European conference on information retrieval. 2005 : 345 – 359.

[52] DALLY W J. The end of denial architecture and the rise of throughput computing[C] // Keynote speech at Desgin Automation Conference. 2010.

[53] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[54] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016 : 2818 – 2826.

[55] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[56] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C] // 2009 IEEE conference on computer vision and pattern recognition. 2009 : 248 – 255.

[57] ZHOU Y, CHEN S, WANG Y, et al. Review of research on lightweight convolutional neural networks[C] // 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). 2020 : 1713 – 1720.

[58] MOLCHANOV P, MALLYA A, TYREE S, et al. Importance estimation for neural network pruning[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019 : 11264 – 11272.

[59] ZHOU B, KHOSLA A, LAPEDRIZA A, et al. Learning deep features for discriminative localization[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016 : 2921 – 2929.

[60] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient convnets[J]. arXiv preprint arXiv:1608.08710, 2016.

[61] YANG C, SHAO H-R. WiFi-based indoor positioning[J]. IEEE Communications Magazine, 2015, 53(3): 150–157.

[62] BRYMAN A. Triangulation and measurement[J]. Retrieved from Department of Social Sciences, Loughborough University, Loughborough, Leicestershire: www. referenceworld. com/sage/socialscience/triangulation. pdf, 2004.

[63] BEKKELIEN A, DERIAZ M, MARCHAND-MAILLET S. Bluetooth indoor positioning[J]. Master's thesis, University of Geneva, 2012.

[64] ROBERTS C M. Radio frequency identification (RFID)[J]. Computers & security, 2006, 25(1): 18–26.

[65] WEISSTEIN E W. Affine transformation[J]. https://mathworld. wolfram. com/, 2004.

[66] ZHANG C, KAESER-CHEN C, VESOM G, et al. The iMet collection 2019 challenge dataset[J]. arXiv preprint arXiv:1906.00901, 2019.

[67] JOSHI P. OpenCV with Python by example[M]. [S.l.]: Packt Publishing Ltd, 2015.

简历与科研成果

基本信息

董学文，男，汉族，1996 年 1 月出生，黑龙江齐齐哈尔人。

教育背景

2018 年 9 月 — 2021 年 6 月 南京大学计算机科学与技术系 硕士
2014 年 9 月 — 2018 年 6 月 南京大学计算机科学与技术系 本科

攻读硕士学位期间完成的学术成果

1. 申富饶, 董学文, 赵健, 李俊 “一种基于深度学习的室内人物定位方法” 专利, Dec. 2020.

攻读硕士学位期间参与的科研课题

1. 国家自然科学基金面上项目 “基于深度感知增量式联想记忆神经网络的信息融合系统研究，Information fusion system based on deep perception and incremental associative memory neural networks” （课题年限 2019.01～2022.12），负责神经网络模型相关研究。

致　　谢

毕业论文至此已接近尾声，也意味着作者的三年研究生生涯即将结束。岁月如梭，不知不觉我在这个校园已经生活了七年。值此离别之际纵然有些许不舍和遗憾，但大多被一路下来收获的喜悦和激动掩盖。感谢那些曾经在科研之路给予我帮助的老师、同学们，没有你们的教导和帮助我很难有今天的成绩。

首先要感谢的是我的导师申富饶教授。申老师经常在作者个人讨论时问询目前的科研现状，为当前的问题和解法提供新的思路和技巧。不但如此，申老师经常在教导我们的过程中强调算法要以解决问题为主，不要为了解答而提问，要做对社会有帮助的算法研究。在课堂教学之外，申老师是一个有很高社会责任感的人，对任内课程和教学事务呕心沥血，为学生的前途和未来着想。一个老师的思想会潜移默化地影响学生后续的价值取向和人生轨迹，因此成为老师不但要有过硬的学习、科研能力，更为重要的是要有一个甘于奉献的精神和强烈的责任感，这种价值取向会慢慢的感染学生逐渐成为对社会有用的人。在这两方面，我认为申老师所做的一切都无可挑剔。

其次要感谢的是作者身边的同学们，是他们的经验和演算能力间接激发了作者。在此特别感谢王绪冬同学在相关论文的推导技巧和插图绘制的过程中对我提供的帮助，以及刘小亮师兄和许翔师弟对整个组内服务器和程序运行环境系统配置方面所付出的心血，你们的维护保障为本文的算法提供了实践可行的沃土。感谢李雪健、刘雅辉同学的帮助，在科研过程中你们对我提出的问题进行了逐一详尽的解答，哪怕是在休息之余。

最后要感谢我的家人，没有你们这么多年来辛勤的付出，就没有我的成绩。某种程度上，是你们的鼓励和帮助支持我走到了今天。我知道你们不求回报，但我仍然想对你们说声谢谢，毕竟这是我现在所能做的一切。

再见了我的校园，你会迎来更多、更优秀的学子，我也将会在不久踏入充满未知的社会。但我们彼此都不必感伤，因为真正的朋友不在于是否形影不离，而在于是否常忆心间。我的举手投足早已刻上了你的烙印，即使以后不再相见我们也都深知曾经的我属于这里的一切。海内存知己，天涯若比邻，相信未来你可以在更大的平台看到我的身影，为有我这样一个朋友而感到骄傲。

《学位论文出版授权书》

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：

2021年5月29日

论文题名	单类别实时目标检测算法与系统研究				
研究生学号	MF1833017	所在院系	计算机科学与技术系	学位年度	2021
论文级别	☐ 硕士　☑ 硕士专业学位 ☐ 博士　☐ 博士专业学位 （请在方框内画勾）				
作者 Email	1004510839@qq.com				
导师姓名	申富饶教授				

论文涉密情况：

☑ 不保密

☐ 保密，保密期 (_____年_____月_____日 至 _____年_____月_____日)

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。