浙 江 省 硕 士 学 位 论 文

论文题目：基于深度学习的交通标志检测与识别研究

专业领域名称及代码：控制工程 085210

论文编号：

基于深度学习的交通标志检测与识别研究

摘要：随着人工智能的发展，无人驾驶成为当前研究的热点之一。由于在道路信息当中，提供最多信息的是交通标志，所以无人驾驶的关键之一便是建立对交通标志进行检测与识别的驾驶辅助系统。其中的难点便是如何快速、准确的检测并识别交通标志。得益于计算机性能的提升以及大数据的威力，深度学习在目标检测以及物体分类方面的性能得到了大幅度的提升。如何使得检测的精度更高以及如何优化使得识别率更高，实时性更强，是本文研究的重点。本文从检测精度以及实时性方面进行研究，主要的研究内容如下：

（1）提出了一种优化目标检测框精度的算法。本文主要结合目标检测算法和视觉显著性算法。利用目标检测算法对图片进行检测，得到图像中交通标志的位置。由于考虑到算法的实时性，本文采用实时性更强的 YOLO V3 目标检测算法，但是其检测的精度并不理想。因此，在得出检测结果的基础上结合视觉显著性算法，使目标框得到再次修正。实验表明，结合后的算法将 YOLO V3 的 IoU 提高了 3%-10%，并且提高了后续不同交通标志分类的精确率和召回率。

（2）为了进一步提高算法的实时性以及准确性，本文提出了结合卡尔曼滤波的跟踪算法。现有算法有以下缺点，一方面，目标检测算法会出现丢帧或目标被遮挡的情况，每次丢帧或遮挡就要对目标进行重新识别是相当耗费资源的；另一方面，持续的进行检测以及识别不仅对算法的时间性能产生影响而且会提高算法识别的错误率。因此，本文提出在检测到交通标志后，对固定帧长（10-20 帧）进行检测以及识别，通过对当前检测识别结果投票来确定交通标志的正确类别。在确定类别之后，无需对后续检测到的交通标志再次分类识别，只是对已经得到的结果进行跟踪。实验表明，结合后的算法在识别准确率上提升了 15%，在时间性能上提升了约 50%，并且能够克服目标被遮挡和检测过程中掉帧的情况。

关键词：交通标志；深度学习；目标检测；视觉显著性；卡尔曼滤波

分类号：TP391.4　　UDC:004.8

Research on Traffic Sign Detection and Recognition Based on Deep Learning

Abstract： With the development of artificial intelligence, self-driving has become the focus of the new phase of research. Because traffic signs can provide the most information while driving on the road, one of the key aspects of self-driving is the driver assistance system, which has the function of detecting and identifying traffic signs. The difficulty lies in how to quickly and accurately detect and identify traffic signs. Thanks to the improvement of computer performance and the power of big data, the performance of deep learning in target detection and object classification has been greatly improved. How to improve the detection accuracy, optimize the recognition rate and improve the real-time performance of the algorithm are the research focus of this paper. This paper will study the detection accuracy and real-time performance, the main contents are as follows:

(1) An algorithm for improving the accuracy of target detection is proposed. It mainly combines the target detection algorithm and the visual saliency algorithm. The target detection algorithm is used to detect the image to obtain the location of the traffic sign in the image. Due to the real-time nature of the system, this paper uses the more real-time YOLO V3 target detection algorithm, but the detection accuracy is not ideal. Therefore, a visual saliency algorithm is added based on the detection result, thereby correcting the detected position again. Experiments show that the combined algorithm increases the IoU of YOLO V3 by 3%-10% and improves the accuracy and recall rate of subsequent traffic classifications.

(2) In order to further improve the real-time performance and accuracy of the algorithm, a tracking algorithm combined with Kalman filter is proposed. The existing algorithm has the following disadvantages. On the one hand, the target detection algorithm will have a frame loss situation, and it is very resource-intensive to re-identify the target every time the frame is dropped.

On the other hand, continuous detection and identification not only causes loss in real-time performance but also increases the error rate of recognition. Therefore, the algorithm first detects and identifies the fixed frame length (10-20 frames) and determines the correct category of the current traffic sign by voting on the current recognition result. After the category is determined, the subsequently detected frames are no longer identified, the results are tracked by the tracking algorithm. Experiments show that the combined algorithm improves the recognition accuracy by 15%, improves the time performance by 50%, and overcomes the situation where the target is occluded and dropped during the detection process.

Keywords： traffic sign; deep learning; target detection; visual saliency; Kalman filter

Classification： TP391.4 UDC:004.8

目　次

图清单

表清单

1 绪论

1.1 研究背景与意义

随着我国近年来科技的蓬勃发展，经济水平的飞速提升，人们的生活质量也日益增加。而汽车作为代步工具，已经基本成为每个家庭必不可少的一部分。截止到 2018 年底，我国汽车的保有量达到了 2.4 亿辆，比 2017 年增加 2285 万辆，增长比率为 10.51%。连年来汽车的保有量一直以 10%以上的速率在增加。但是随着汽车的普及，不可避免的暴露了许多弊端，比如说汽车尾气对大气产生污染、城市的交通系统瘫痪，堵车成为了司空见惯的事情，除此之外，车祸的发生概率也越来越高。因为车祸而丧生的人数也在连年增加，据世界卫生组织报告，每年在全球中，因为道路交通而死亡的人数高达 135 万人。每一次事故的发生，都对一个家庭产生了难以挽回的损失。车祸带来的不仅是财产的损失，更是对整个社会造成了人才的流失。通过分析所发生的交通事故，很大一部分是因为司机对道路信息的判断不准确，从而做出错误的驾驶行为。如何能有效的避免交通事故的发生成为了每个国家亟待解决的问题。高级驾驶辅助系统(Advanced Driver Assistance System, ADAS)成为了诸如美国、日本、欧洲等国家研究的热点，大量的科研经费以及人力物力的投入使得研究的成果一路提升。ADAS 由实时交通系统、车道偏移报警系统、行人保护系统、交通标志识别系统等系统构成，在众多系统统一完善后，最终将会达到无人驾驶的阶段。

科学是第一生产力，科学技术的发展，使得人们对以往只有在科幻电影中才能见到的某些技术产生了憧憬。得益于计算机硬件的提升，使得计算力得到大幅度提高，再加上互联网的蓬勃发展，使得获取大量数据成为了可能。二者相得益彰，使得相关科研项目如雨后春笋般蓬勃发展。而近几年来，发展最为火热的无疑是无人驾驶。并且随着 5G 技术的逐步发展，似乎无人驾驶离我们的距离又再近了一步。

人眼是人观察万物的唯一途径，如何让计算机拥有和人眼一样的功能是研究的热点之一。由于在相当多的情况下，机器犯错的概率是小于人类

的，将计算机视觉应用于辅助驾驶方面，通过计算机去分析当前的驾驶路况，从而做出当前最优的判断，辅助驾驶员完成驾驶任务，这样将会大大减少交通事故的发生。随着深度学习的发展，计算机视觉成为近几年研究的热点，其中目标检测、目标分类、目标跟踪以及行为分析等方向成为了领域研究的热点，并且均取得了较好的结果。无人驾驶的关键一步便是有效的识别路况信息，而道路中蕴含信息最多的则是交通标志。交通标志可以提供有效的限速信息、禁令信息、指示信息等，如果可以准确的检测并识别交通标志的内容，那么对于降低车祸的发生有着至关重要的作用，并且高效的交通标志检测与识别也是无人驾驶的重要一步。

由此可见，交通标志的检测与识别是很有研究价值和现实意义的，并且完善的交通标志识别系统有很好的应用价值。

1.2 国内外研究现状

国内外对交通标志的检测与识别研究主要是围绕驾车辅助系统以及无人驾驶的发展而展开的[1,2]。由于辅助驾驶的重要意义，最早的科学研究可以追溯到上世纪八十年代。在早期的研究当中，主要受限于计算机的计算力以及数据量，研究的方法基本是利用形状、颜色等人为总结的信息，这种方法比较依赖于个人经验，并且耗费人力、物力和财力。虽然取得了一定的成果，但需要改进的地方也有很多。近些年来，随着深度学习的发展，使得对交通标志的检测以及识别的结果取得了显著地进步。

在传统算法的方面，由于交通标志的形状以及颜色都有很明显的特征。所以有很多学者便根据此特征对交通标志进行检测与识别。其中，在利用颜色检测的方面，H. Kamada 等人提出了一种根据特定的 RGB 颜色分量的强度与 RGB 的强度之和的颜色比的方法对交通标志进行检测[3]。L. E. Moreno 等人提出了一种通过 RGB 相关性阈值进行分割的方法[4]。J. Miura 提出基于 YUV 的方法[5]。P. Amoul 提出了一种通过 HIS 彩色空间进行交通标志检测的方法[6]。在利用交通标志的形状进行检测的方面，S. Maldonado Bascon[7]，H. Liu[8]提出了基于径向对称的算法。Hu M K[9]提出了一种根据 Hu 矩进行检测的算法，提出图像的 7 个不变矩，通过提取这些特征从而对图像中交通标志进行检测。由于霍夫变换能够检测简单的图形，而交通标志便是由简单图形构成，Garcia 提出了使用霍夫变换对圆形

和三角形交通标志进行检测[10]。而 Boumediene[11]通过灰度图的思想，获取到交通标志的角点信息，通过检测其对称线，从而实现对三角形标志的检测。

如果只是单纯的使用颜色信息或者形状信息进行交通标志的检测不可避免有各自的优缺点，因此有部分学者通过结合两者信息来对算法进行进一步的提升。其中，X. W. Gao 提出了一种中心凹陷注视模型，可以同时提取到交通标志的形状信息和颜色特征信息[12]。为了完善 Garcia[10]的算法，Ruta[13]通过向其算法增加颜色信息的方法，从而提高了算法的性能。

为了提高检测的效率，一般的检测过程会分为两个部分：首先是分割，其次是检测[14]。算法会先通过颜色的信息对图片进行分割，其次对分割区域进行检测。由于通过颜色分割可以得到更可能存在交通标志的区域，而排除大量无意义的区域，这样会大大减少检测所需要的时间，使在保持精度的前提下，减少无关区域对检测的影响，提高运行的速度。

对于交通标志的识别部分，主要是通过对检测得到的目标进行图像特征的提取。基于传统的学习方向，主要是提取图像的特征，从而再将特征结合机器学习分类器算法如：支持向量机、随机森林等分类器从而完成交通标志的识别。其中，黄志勇[15]通过提取交通标志图像的彩色信息，边缘信息等众多特征，将得到的众多特征结合支持向量机从而实现了交通标志的识别。Zaklouta F 则提取了图像的 HOG 特征并将之结合 SVM 从而实现交通标志的识别[16]。Takaki M [17]则利用了 SIFT 特征与 SVM 的结合从而达到识别交通标志的目的。除此之外，为了提高性能，文献[18,19]采用稀疏特征的方法来检测识别交通标志。诸如此类的方法虽然在精度和准确率上有所提高，但是选取到的特征的鲁棒性并不是很高，而且由于车载视频得到的每帧图片分辨率极高，使用传统的特征提取方法是很耗费时间的。从而使得其识别速度较慢，并不能达到实时性。

深度学习是通过大数据的思想，将大量的数据提供给计算机，使用合理的网络结构，使得机器自行去学习其中的特征信息，这样学习到的信息鲁棒性更强，并且只要有足够的数据和合适的网络模型，就能达到相当可观的效果。深度学习在计算机视觉的成功离不开卷积神经网络(Convolutional Neural Network, CNN)的发展。得益于此，交通标志检测和识别的效果也在逐年提升。其中，Schmidhuber[20]首次运用卷积神经网络的思想在交通标志数据集上进行交通标志的识别，取得了 0.56%的错误率，

使得神经网络的结果首次超越了人类。Dan Ciresan[21]也设计了一种神经网络来进行交通标志识别，并在 GTSRB 数据集上取得了 99.15%的准确率。Pierre Sermanet[22]使用多尺度卷积神经网络使得识别准确率达到 99.17%。Junqi Jin[23]等提出了一种使用铰链损失梯度下降的方法对反向传播进行优化来提升检测的效果。此外，Qian[24]等人提出了一种基于区域的深度卷积神经网络来达到同样的检测效果。随着目标检测算法的持续完善，越来越多研究人员直接使用现有的检测框架，在优化后进行交通标志的检测并取得了很好的结果。比如通过与 Fast R-CNN 结合进行交通标志的检测[25]，通过与 Fast R-CNN 的提升算法 Faster R-CNN 的结合达到同样的目的[26]，除此之外，还有学者通过结合 YOLO 算法来完成对交通标志的检测[27]。

1.3 本文研究内容

交通标志的有效检测以及识别是发展无人驾驶的重要一步，如何快速准确的检测并识别交通标志是难点之一。本文主要分析了目前一些算法存在的缺点，提出了一些相应的改进措施。

(1)详细对比分析了常见的目标检测算法如：Faster R-CNN、YOLO、SSD 并通过结合不同的骨干网络训练新的检测模型。对 TT100K 交通标志数据集进行检测，通过从检测时间，召回率等方面对比分析，从而确定实时性强并且召回率高的目标检测模型。实验结果表明，YOLO V3 算法在实时性、召回率以及小目标检测等方面有明显的优势。

(2)YOLO V3 算法虽然在实时性上有所保证，但是在中大型目标检测精度上有一定的欠缺。为了弥补 YOLO V3 算法在精度上的欠缺，本文提出了结合视觉显著性检测的算法。由于交通标志在颜色上是与周围的环境有明显的区别，对检测得到的交通标志区域进行扩大之后，再结合使用视觉显著性检测的算法对扩大后的区域进行交通标志检测，对得到的检测框进行二次修正，获得更为完整的交通标志。通过实验表明，结合视觉显著性的算法能够提升检测的 IoU, 并且提升了后续的分类性能。

（3）由于最终的运行现实条件是基于视频的检测与分类，而视频基本都是由每秒 25 帧组成的，如果持续的检测和分类对时间性能是有影响的，不仅如此，检测算法会出现短暂的掉帧以及目标被遮挡的情况，如果

遇到此类问题就重新识别是很耗费资源的。因此，本文提出通过结合跟踪算法来优化当前的检测与识别算法。首先，算法对固定帧长(10-20 帧)进行识别。通过对当前识别结果投票确定目标种类，当类别确定之后，不再对后续的每一帧进行识别，只是对当前检测到的目标进行跟踪。这样能有效提高识别的准确性，减少因重复识别而造成的资源浪费，并且结合后的算法能够应对短暂的掉帧以及目标被遮挡问题的发生。

1.4 各章内容简介

该论文共由五章组成，各章节的具体内容描述如下：

第一章：主要介绍了本文的研究背景以及研究意义，并详细讨论了该课题的国内外研究现状，在最后则提出了本文对现有算法的一些提升之处。

第二章：详细介绍本文研究所涉及到的相关知识，具体包括：交通标志的简单介绍、卷积神经网络的介绍以及基于深度学习的目标检测算法的介绍。

第三章：首先，对常见的基于深度学习的目标检测算法进行性能分析，从而选取效果最好的模型。其次，对算法存在检测框不准确的情况，提出结合视觉显著性算法对检测框进行二次修正的方法。最后，验证结合算法对后续分类的影响。通过实验表明，结合后的算法可以进一步提升检测框的精度，提升了后续的分类性能。

第四章：通过第三章得到交通标志检测与识别模型，对现实条件下的车载视频进行交通标志的检测与识别。通过实验发现模型存在以下问题：第一，视频的每一秒由 25 帧组成，包含的冗余信息过多，持续对每一帧进行检测与分类对时间性能带来损耗；第二，在检测过程中，算法会遇到目标物体被遮挡以及掉帧的情况；第三，持续的分类会提高识别算法的识别错误率。综上所述，采用对结果投票以及结合跟踪算法来优化现有算法。通过实验表明，优化后的算法提高了识别的准确率、提升了算法的时间性能以及缓解掉帧和目标被遮挡的情况。

第五章：对本文研究的进一步总结，提出存在的问题，以及对未来研究的展望。

2 相关知识

2.1 交通标志简介

我国所实行的道路交通标志依照国家标准《道路交通标志和标线》中的有关规定。主要可以分为：警告标志、禁令标志、指示标志、指路标志、旅游区标志、作业区标志、告示标志以及辅助标志八种。部分交通标志图如图 2.1 所示。

由于车载辅助系统发展的必要，国外出现了一系列的交通标志数据集，比较常用的有德国交通标志检测基准数据集（GTSDB）、德国交通标志识别数据集（GTSRB）[28]、KUL 数据集[29]、STS 数据集[30]、RUG 数据集[31]等等。其中德国交通标志数据集成为了国内外进行交通标志检测以及识别算法评判的标准数据集。但是，上述的所有数据集都是主要针对国外的交通标志，其中某些标志与我国的并不相同。除此之外，数据集虽然收集于现实条件下，但是已经人为将其处理，使得图中交通标志的情况得以改善，背景更简单，噪声更小。因此，利用这些数据集进行国内的交通标志检测识别研究就有一定的不妥之处。

图 2.1 部分交通标志

由于我国开始对行车辅助系统的重视，由清华大学与腾讯共同合作，收集了一批在我国真实道路上拍摄所得的数据集 Tsinghua-Tencent 100K[32]，这也是迄今为止我国最大的交通标志数据集，其中包含了 30000 个交通标志实例的 100000 张图像，并且这些图像涵盖了不同光照和天气条件下的交通标志，有很强的实际意义。

2.2 卷积神经网络

近几年来，计算机视觉无论是在物体分类、目标检测还是目标跟踪方向都取得了巨大的成功。其核心的思想便是卷积神经网络(Convolution Neural Network, CNN)[33]，无论其模型有多么优秀，参数如何调优，其骨干网络必然是由 CNN 的各种变种组成。CNN 使用权值共享的思想，能极大减少参数的个数，极大的加快了训练的时间，促进了科学研究的进程。一般来说，卷积神经网络由以下几部分构成，卷积层(convolutional layer)、池化层(pooling layer)以及全连接层(fully connected layer)。

(1) 卷积层

卷积层是卷积神经网络的核心部分，卷积的数学本质便是两段序列翻转移位相乘。一维离散的卷积公式如下所示：

$$c(n) = f(n) \otimes g(n) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(n-\tau) \tag{2-1}$$

其中 $f(n)$与 $g(n)$为两个离散信号，$\otimes$表示卷积操作，$c(n)$为卷积结果。其卷积可以理解为计算一个滑动的加权总和，使用 $g(n-\tau)$为加权函数对 $f(\tau)$取加权值。同理二维数据的卷积公式如下所示，其数学本质是一个矩阵翻转后和另一个矩阵移位相乘。

$$\begin{gathered} \mathrm{C}(s,t) = \sum_{m=0}^{r_A-1} \sum_{n=0}^{c_A-1} \mathrm{A}(m,n)\mathrm{B}(s-m,t-n) \\ s.t. 0 \le s < r_A + r_B - 1, 0 \le t < c_A + c_B - 1 \end{gathered} \tag{2-2}$$

其中矩阵 **A** 的行数和列数分别为 r_A 和 r_B。矩阵 **B** 的行数和列数分别为 r_B 和 c_B。**C** 为在当前坐标下的卷积得到的结果。

图像的卷积操作的核心是权值共享。可以理解为通过一个固定大小的模板以一定的步长在原始图片上进行滑动，从而提取到图片的信息。这样做的结果有一个弊端，那就是单一模板提取到的特征可能并不完善。解决该问题的方法就是使用多个模板，体现在深度学习中就是会有很多的通道数。它的效果就是用多个不同的模板从而提取到图像更多地特征。卷积神

经网络训练学习的核心过程便是如何取得合适的模板权值，使之具有更强的代表性。在深度学习当中，这个模板被称为卷积核。其中，图片与当前卷积核进行卷积得到下一步的特征图大小的计算公式如(2-3)(2-4)所示：

$$w = (n + 2p - f) / s + 1 \tag{2-3}$$

$$h = (m + 2p - f) / s + 1 \tag{2-4}$$

其中，w 和 h 为卷积之后得到的特征图的宽和高，n 为卷积前图片的长度，m 为卷积前图片的宽度，p 为扩充大小，当卷积核的范围超过了当前特征图的大小，则需要设置 p 的值以满足正常的卷积操作，一般对扩充的特征图数值置为 0。f 为卷积核的大小，s 为卷积的步长。

(2) 池化层

池化层的作用就是对卷积得到的图像进行降采样，并不需要学习什么参数，只是对上一步得到的特征进行一个聚合统计。常见的池化操作有最大值池化，就是取当前的池化模板中的最大值；平均池化，取当前池化模板的平均值；随机池化[34]，随机的在当前模板中取值。池化主要能起到以下几个作用：

首先，池化拥有平移不变的特性。因为，在池化的选择过程中，是在一个范围内通过不同池化模板得到一个池化后的值，由于这个范围的存在，使得图像可以接受一定程度上的平移。

其次，池化可以达到增大感受野的作用。感受野其实是特征图上一个点映射回原图得到的大小。如果两个卷积层之间添加了池化层，因为池化是经过下采样操作的，那么下一步即将卷积的特征图所代表的原始图像的感受野更大。

此外，池化层还降低了优化难度，较少了参数个数。因为池化层的参数是不需要训练学习，但是由于其可以缩小当前特征图的大小，使得下一次卷积得到的参数也更少。

(4) 全连接层

通过卷积以及池化操作可以将原始的输入数据映射到特征的隐层空间，而全连接层的作用，则是将已经映射到特征空间的数据重新映射回样本的标记空间。之后根据分类的要求，一般都在最后一层使用 softmax 函数，从而实现多分类。全连接层实际上是十分占用存储空间的，因为其含有大量的参数。所以，现在大部分学者都会使用 1x1 的卷积来代替全连接层，这样做的目的可以极大的减少参数，从而缓解全连接层带来的巨大参

数量。

2.3 基于深度学习的目标检测算法

目标检测，即在一张图中快速准确的让计算机定位到你想要寻找的物体。

随着人工智能和深度学习的发展，计算机视觉成为从中受益最大的方向。早期的目标检测算法主要是利用传统图像的算法。该类算法需要很强的先验知识，往往需要耗费很大的人力与物力才有很少的提升。这一阶段使用较多的有 HOG[35]、SIFT[36]、SURF[37]、Haar[38]等特征，对目标区域进行上述特征的提取。将得到的特征送入到诸如 SVM[39]、决策树[40]、随机森林[41]等分类器中。在检测的过程中，主要通过模板匹配或者固定框暴力搜索的思想，去目标图中和模板进行匹配。这种方案的缺点也很明显，受到光照、尺寸、环境的影响大，而且相对来说比较耗时。

在 2013 年之后，目标检测算法基本开始从传统的检测算法转移向深度学习的方法，并且在很多方面深度学习的方法都已经超过了大量的传统的算法，并且深度学习在特征选择方面没有很强的先验知识，大部分都取决于机器的自主学习。随着深度学习的发展，也出现了一系列优秀的目标检测算法，大体可以分为两大类，一类是根据候选区域的目标检测算法，主要代表如 R-CNN 系列；另一类则是单级式目标检测算法，主要代表作为 YOLO、SSD 系列，现对其发展进行简单的介绍。

2.3.1 基于候选区域的目标检测算法

基于候选区域的目标检测算法，先利用算法获得可能的候选区域，然后将这些候选区域送入到已经训练好的神经网络当中，从而检测当前的目标物体。

基于候选区域的检测算法，最早的想法其实很简单，就是使用暴力框搜索的方式。由于不同物体在图像中有可能有不同的大小，那么就分别使用不同比例大小的框在图片上进行滑动，分别再将得到的框中的信息送入卷积神经网络中对其进行检测识别。此方法的缺点就是需要大量的搜索框，并且这些被框出的候选框很多都没有任何有价值的信息，只是增大了计算量。

为了解决暴力搜索的大量冗余框的问题，提出了一种选择性搜索的方法[42]，该方法考虑到了图像的颜色相似度(color similarity)、纹理相似度(texture similarity)、尺寸相似度(size similarity)以及交叠相似度(shape compatibility measure similarity)。基本的算法可以分为四步。第一，对当前的图片生成区域集；第二，通过上述的相似度之和，计算区域集当中每个相邻区域的相似度；第三，找出当前计算的相似度中，相似度最高的两个将之合并成为新的集合；第四，重复以上步骤，直至初始集合为空。该方法的主要优点是通过将相似度高的区域合并，往往就能得到图像中可能是目标物体的子块，直接将这些子块送给训练好的神经网络做后续的检测与识别。相比于暴力搜索，选择性搜索通过相似度的结合，能够极大的减少候选框的个数，提高了目标检测的性能。

最早的目标检测框架 R-CNN[43]便是使用这种思想。首先，算法通过选择性搜索的方式，在整张图片产生 2000 个左右的刚兴趣区域(region of interest, ROI)，通过缩放转化为固定的尺寸，随后再使用以 AlexNet[44]作为骨干网络，SVM 作为分类器的预训练好的分类模型，从而完成后续的目标检测。

对于 R-CNN 的改进算法 Fast R-CNN[45]，其主要思想也还是通过选择性搜索的方式产生目标候选框，但由于 R-CNN 方法是重复的将原始的 ROI 送入到神经网络，使得神经网络会重复地提取特征，这样必然产生了资源的浪费。于是，作者提出了先对原始图片进行卷积，通过卷积网络提取到对应的特征图，之后对特征图再使用选择性搜索的方式，得到特征图上的目标区域，将通过卷积特征图提取到的候选区域直接送入分类网络，这样一张图片只要进行一次卷积提取，便可以达到目标检测的效果。在此期间，通过选择性搜索的方式得到的特征图的大小并不固定，通过直接裁减或者缩放，都会产生一定的损失。因此，提出了通过 RoIPooling[46]的思想将特征图转为后续操作所需的固定尺寸的特征图。算法主要是通过分割池化的思想，使得得到的图像尽可能包含完整的信息。除此之外，为了进一步提升算法的性能，损失函数由 R-CNN 的平方损失变为 $smooth_{L1}$ 损失，计算公式如(2-5)所示：

$$smooth_{L_1} = \begin{cases} 0.5x^2, |x| > 1 \\ |x| - 0.5, otherwise \end{cases} \tag{2-5}$$

通过上式可知，当 x 的绝对值大于 1 时，其损失为二次损失，但是当 x 的绝对值小于等于 1 时，其损失则为线性损失。这样做可以防止因为预测值与目标值偏差很大时，L_2 损失导致的梯度爆炸。

而 R-CNN 系列表现最好的则属于 Faster R-CNN[47]，其完善了前两版算法的不足，直接使用一个卷积神经网络结构就可以完成目标检测，并且进一步提升了产生候选区域的效率。充分利用计算机 GPU 的计算力，直接将复杂耗时的提取候选区域的部分放在 GPU 上完成，并利用深度学习的特性，将之整合为一个网络层，并取名为区域候选网络(Region Proposal Networks, RPN)，其大大的加速了产生候选区域的速度。Faster R-CNN 的结构如图 2.2 所示。

Faster R-CNN 主要的原理可简单的概括如下，其前期的工作和前述版本的目标检测框架相似。图像都要经过骨干的卷积神经网络进行特征的提取，从而提取到对应的特征图。但是，在得到特征图后，Faster RCNN 算法将特征图一方面作为 RPN 网络的输入，另一方面将特征图和后续的 RoIPooling 层相连。RoIPooling 层结合 RPN 网络的结果，再进行全连接层和 softmax 计算每个候选框的类别以及位置的偏移量。

其中，Faster R-CNN 的核心部分便是 RPN 网络，其大概的思想分为以下几步：

(1)将通过前期卷积网络得到的特征图传入 RPN 网络，RPN 网络将之分为上下两部分，一部分用来做前景与后景的识别，另外一部分则用来做边框的回归，从而预测出物体所在的位置。

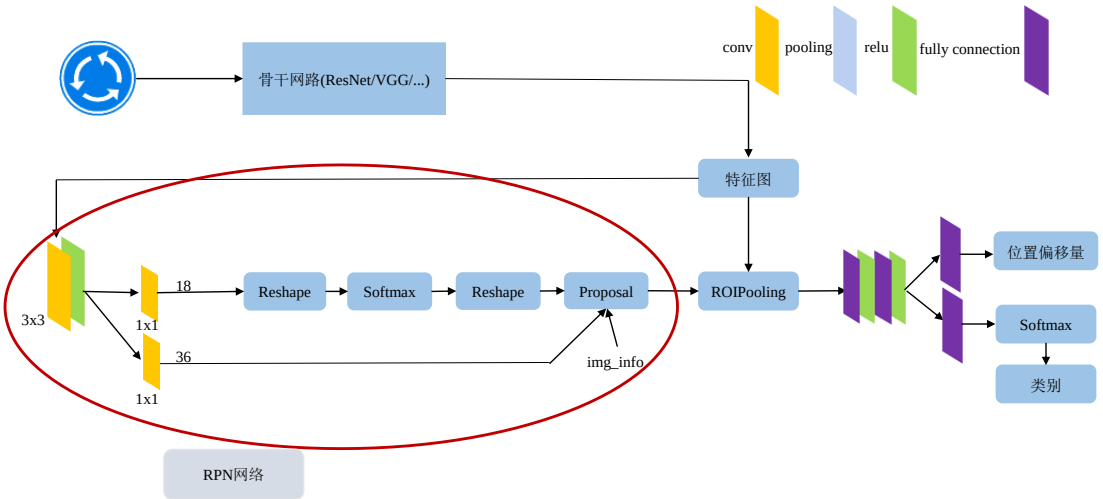


图 2.2 Faster R-CNN 网络结构

(2)为了模拟多尺度、多长宽比(一般的长宽比为 1:2、2:1、1:1)的情况，RPN 网络使用 Anchor 的思想，即在遍历特征图像的每一个点时，使用不同的 anchor 对其进行特征提取，这样就相当于在原图上做了不同尺度的检测，具体如图 2.3 所示。

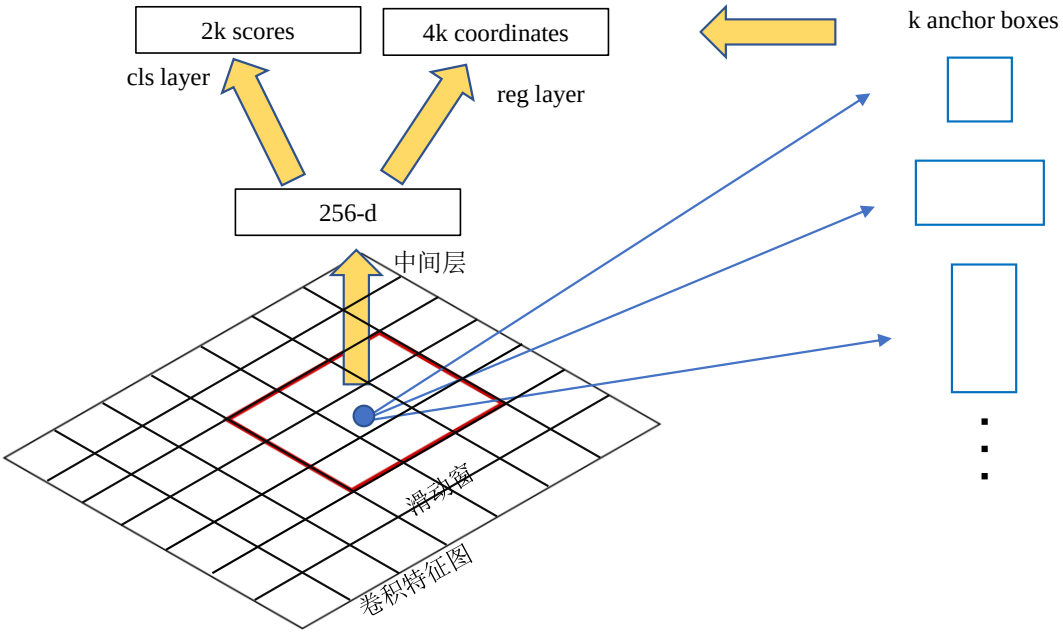


图 2.3 anchor 机制

(3)通过观察图 2.2 可知，RPN 网络上下两个部分均出现 1x1 的卷积核，其主要的目的便是通过 1x1 卷积使得网络根据我们所设置的 anchor 数量完成后续的计算。比如 anchor 的个数为 9，由于上层我们是通过 softmax 函数判断所检测的目标属于前景还是背景，因此使用 1x1x18(2x9：前后景两类以及 9 个 anchor)的卷积网络进行卷积；而下层的网络主要是通过回归的方法得到最终坐标的平移量和变换尺度，所以使用 1x1x36(4x9：一个 anchor 对应一个中心点坐标和该 anchor 的宽和高 4 个变量，共有 9 个不同的 anchor 框)的卷积核进行卷积操作。

(4)通过前景与背景的判断以及对当前目标区域回归所得到的偏移量，得到相应的候选框(Proposal)并传送给后续的网络结构。

由于整个 RPN 网络不仅判断当前位置是前景和背景，又要计算当前目标为前景，其坐标的偏移量是多少。因此，整个 RPN 网络的 Loss 公式如(2-6)所示。

$$L\left(\{p_i\},\{t_i\}\right) = 1 / N_{cls}\sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda / N_{reg}\sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \quad (2\text{-}6)$$

其中，公式(2-6)中 i 表示 anchor 的索引，p_i 表示预测的概率，p_i^*表示真实的概率，其中只取概率 p_i 小于 0.3 或 p_i 大于 0.7 的部分，其余的均不参与训练，这样可以更好的训练难训练的部分。t_i 表示预测得到的坐标，t_i^*表示真实的坐标。N_{cls} 表示种类的个数，N_{reg} 表示边框的个数。L_{cls} 和 L_{reg} 分别对应分类和回归的损失。λ相当于一个调整因子，避免 N_{cls} 和 N_{reg} 相差过大。

2.3.2 单级式目标检测算法

基于候选区域的检测算法，其优点是检测框的位置准确，检测的精度更高，但其缺点是因为产生候选框要花费额外的时间，从而使得其检测速度并不理想。而随着计算机视觉的发展，对于目标检测来说，有更高的实时性也成为考虑的重点。因此研究者提出了检测速度更快的 SSD[48]、YOLO [49-51]这样的单级式目标检测算法。

单级式目标检测算法的主要思想是直接通过一次卷积提取操作就同时完成目标的检测以及位置的预测。其思想是通过多层的卷积，得到了较小尺寸的多通道特征图。比如，最终得到的特征图为 7x7x512，将之映射到原图就相当于将原始图片分割成 7x7 网格的形状,之后判断物体标注的边界框中心点落在哪个方格当中，这个网格就负责当前物体的检测。为了能够检测该目标,需要额外添加一个卷积层并学习结合之前 512 个特征图的核参数信息，以得到一个对应于包含目标的网格单元的激活。相应的，如果一张图片中有多个物体，那么将会有多个网络单元被激活。

每个激活描述检测目标的参数由以下几部分构成：

1、当前网格包含目标的可能 P_{obj}；

2、该目标属于哪一个类别 c_1,c_2,c_3...；

3、边界框点的横坐标和纵坐标以及宽和高四个标定量，可以描述为 (t_x, t_y, t_w, t_h)。

综上所述，对每一个网络格的卷积通道数应该是 $5+N_c$,其中 5 表示类别的可能性和边界框的描述量，N_c 则表示总共的类别个数。以上的表示是当前网格只有一个物体，如果当前的网格中有多个目标的时候，当前的卷积通道数为 $B(5+ N_c)$,其中 B 为当前网格的目标个数。

通过上述操作后，由于每个网格都会进行多个 anchor 框的预测，因此，一个目标物体可能会出现多个预测框，对最终的结果产生干扰。解决该情况的方法就是对此进行非极大值抑制(Non-Maximum Suppression, NMS)。其思想可以简单的分为两步，第一，选取当前所有预测框当中拥有最高置信度的框；第二，计算出所选框和其余预测框的 IoU，当 IoU 超过预设的阈值时，丢弃到这些冗余框。通过以上这两步，便可以解决一个目标被多个预测框同时预测的情况。

以上便是单级目标检测的核心思想，在此基础上，出现的比较有代表性的算法有 SSD 系列和 YOLO 系列。

其中，SSD 算法的骨干网络采用了在 ImageNet 上经过预训练的 VGG16 模型。其后再添加了卷积层等网络结构，从而达到目标检测的目的。由于添加了卷积层，不可避免会使得图像的分辨率再次降低，所以为了解决这个问题，SSD 对骨干网络后续添加的每一卷积层得到的特征图都进行独立的目标检测。

对于 YOLO 系列算法，其主要的骨干网络是 DarkNet 网络，DarkNet 网络的设计思想是通过多层小尺度卷积核从而实现参数优化的作用。DarkNet 53 网络结构模型主要由 3x3 和 1x1 的卷积核构成，并且借鉴了 ResNet 的跳过连接的思想。除此之外，DarkNet 53 相比与 ResNet 有更低的十亿次浮点数运算，但是能够以两倍的速度获得相同的分类精度。

YOLO 系列算法与 SSD 算法的不同点主要在于以下几点。第一，YOLO 算法的 anchor 的选取是通过聚类算法 k-means 生成的，而并非人为的选定；第二，由于 YOLO 的作者考虑到重叠标签的情况，使用了 sigmoid 进行多标签的分类；第三，YOLO 和 SSD 最大不同的地方是 YOLO 运用了特征金字塔网络(feature pyramid network, FPN)[52]的方式实现了多尺度的预测，使之能应对小目标的检测。

构建特征金字塔主要分为两条路径，其中一条路径是由下到上的路径，使用卷积操作对原始图片进行提取特征，层数越高其包含的语义特征越高而图像的分辨率则越低；另一条路径则是由上到下的路径，从上向下重新构建高分辨率层，通过融合当前层和经过上采样的顶层信息，并且通过 3x3 卷积操作消除可能会存在的混叠效应。图中放大的矩形框区域为横向连接部分，1x1 的卷积核用来减少特征图的个数，上方的路径是对得到的上层图进行上采样。通过上述方法，从而达到对小目标的有效检测，具体

的 FPN 结构如图 2.4 所示。

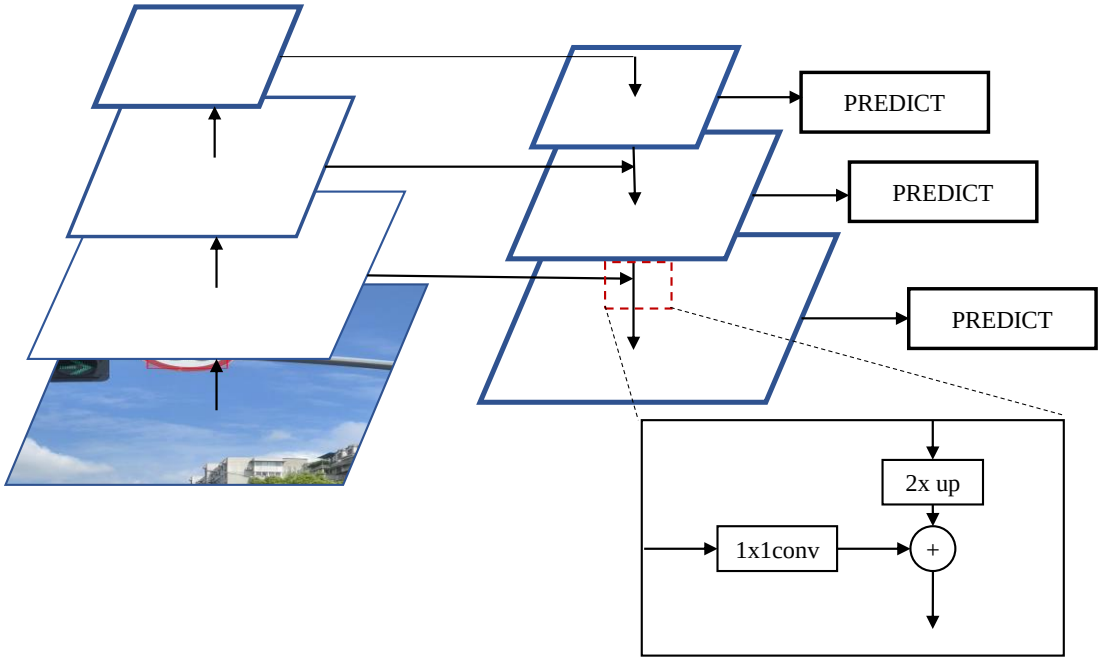


图 2.4 特征金字塔网络(FPN)

2.4 本章小结

本章主要介绍了本文研究所需的一些基础知识。首先，对交通标志进行了简单的介绍；其次，对深度学习的核心卷积神经网络的组成部分进行了介绍；最后，介绍了基于深度学习的目标检测算法，该类算法主要分为两个大类，一类是基于候选区域的目标检测算法，另一类则是单级式目标检测算法。

3 结合视觉显著性算法的交通标志检测与识别

基于深度学习的目标检测算法在现实中应用众多，比如车道检测、行人检测、车牌检测等。本章将通过对比现阶段的目标检测算法在交通标志检测方面的效果，通过召回率、识别效率等评价指标，获取最优的检测算法。并在此基础上结合显著性算法对检测框进行二次优化。除此之外，只是对交通标志检测是不足以满足实际应用的，还需要确定具体的交通标志类别。因此，实验对 TT100K 数据集进行数据增强，训练得到相关的分类模型。分析对比了优化后的检测结果对分类的提升。最终通过权衡模型大小和准确率等信息，确定第四章在基于视频对交通标志进行检测与识别的初步算法。

3.1 TT100K 数据集介绍

Tsinghua-Tencent 100K[32]数据集是由清华大学和腾讯共同合作，专门针对我国的道路而收集到的一系列交通标志。该数据集相对于德国交通标志检测数据集(GTSDB)，图片的总数量是 GTSDB 数据集的 111 倍，并且每张图片的分辨率是 GTSDB 的 32 倍。数据集主要来自中国的五个不同城市的 10 个区域，全部的图片均来源于真实的环境，其中包含不同光照条件下，不同的天气条件下和某些遮挡情况下的交通标志。数据集中共包含 151 个类别，基本涵盖了所有的交通标志类别。

其中，TT100K 数据集中对含有交通标志的图片分别进行了边界框的标记和类别的标记，这样便于用来训练模型。本实验使用的训练集共有 6105 张，测试集共有 3071 张，数据的特点是高分辨率，每张图片的分辨率为 2048x2048。数据集中部分照片如图 3.1 所示。但是，这个数据集的交通标志的主要特点就是小尺寸的交通标志的涵盖比率很高，具体的分布如表 3.1 所示。

由表 3.1 可知，对于 2048 x 2048 高分辨率的图像，交通标志的分布主要集中在 160 x 160 分辨率以下，可见小目标的含占比极高，这对目标检测算法造成了很大的挑战，因为目标检测算法在小目标检测方面的表现

并不理想。

由图 3.1 可观测到，红色区域为图中交通标志的区域，大部分的交通标志在图片中只占了很小的比例，使得检测的难度增加。

表 3.1 交通标志尺寸分布

尺寸	占比 /(%)
16 x 16 – 25 x 25	17.68
25 x 25 – 38 x 38	30.1
38 x 38 – 56 x 56	23.17
56 x 56 – 78 x 78	14.24
78 x 78 – 160 x 160	12.9
160*160 以上	1.91

图 3.1 部分数据集图片

3.2 目标检测算法的对比

现阶段基于深度学习有众多目标检测算法，本章节主要是通过分析对比不同算法在不同评价指标上的性能，从而确定性能最优的检测算法。

3.2.1 评价指标

在分析不同的目标检测算法的优劣时，主要通过以下指标来评判。

(1) 召回率(Recall)

召回率为检测算法正确检测到目标个数占本应该为交通标志的总个数的概率，其计算公式如下式所示。

$$recall = TP / (TP + FN) \tag{3-1}$$

其中，*TP* 为检测算法正确识别为交通标志的个数，*FN* 为本应该是交通标志，但是被识别成为不是交通标志的个数。

(2) 精确率(Precision)

精确率为正确的交通标志个数占所有被识别成交通标志的概率，其计算公式为：

$$precision = TP / (TP + FP) \tag{3-2}$$

其中 *TP* 表示检测算法检测为真正为交通标志的个数，*FP* 表示本应该不是交通标志但是被识别成交通标志的个数。

(3) 检测耗时

检测耗时主要去评判不同检测算法在检测速度上的性能差异，计算其处理一张图片平均所需要的时间。

(4) IoU(Intersection-over-Union，IoU)

IoU 即交并比，如图 3.2 所示，用来评价目标检测算法的预测精度，通过计算预测框和真实框相交面积与预测框和真实框的相加面积之比，即区域 C 与 G 交的面积以及 C 与 G 总面积的比值，其值越接近 1，说明预测精度越高。计算公式如下所示

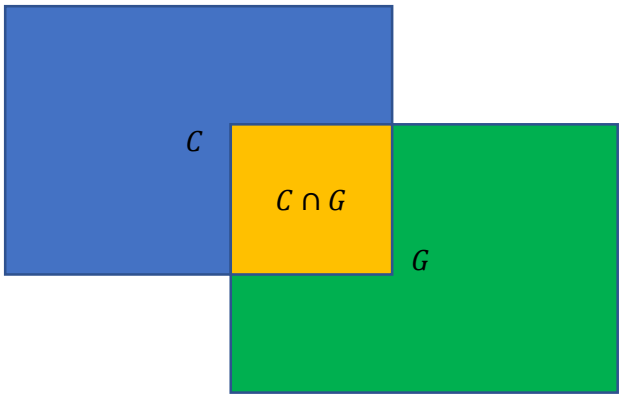


图 3.2 IoU 示意图

3.2.2 网络结构

深度学习的特征提取的好坏有一定程度上取决于网络结构设计的好

坏。本实验通过结合对比不同的网络模型，来验证网络模型对实验结果的影响。常见的网络模型主要分为以下几种。

ResNet[53,54]系列模型，其主要思想是加深神经网络的层数学习到更多语义信息更加丰富的特征。由于不断地加深网络模型的深度，会使得模型产生退化，ResNet 系列提出了残差结构来解决该问题，具体的结构如图 3.3 所示。网络的输入为 x，期望的映射输出为 $H(x)$，直接去拟合 $H(x)=x$ 是很困难的，因此将网络模型设计成为 $H(x)=F(x)+x$，这样我们就可以将学习的过程转化成为一个残差的过程即 $F(x)=H(x)-x$。使得学习的过程变得容易。

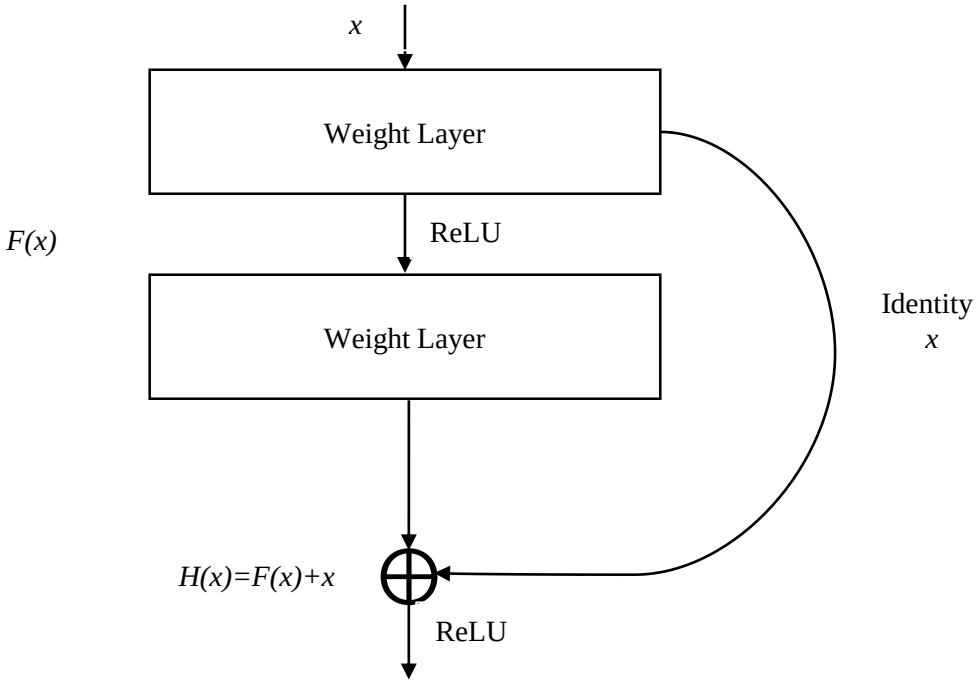


图 3.3 残差网络结构

Inception[55-58]系列模型，该模型主要的思想是在所有模型都在研究如何使得网络更深，从而提取更高等级特征的时候，Inception 提出了卷积核的并行合并(也称为 Bottleneck Layer)。通过不同大小(1x1、3x3、5x5)的卷积核同时学习到当前特征图的稀疏特征和非稀疏特征，最后再通过将上述学习到的信息拼接起来，实现了对特征图的提取，具体的 Inception 模块如图 3.4 所示。其后的不同版本的提升也是对 Bottleneck Layer 进行不同

的结构调整，但总体的思路还是相同的。

DarkNet 系列模型，DarkNet 模型为了缓解参数问题也是由多层的 3x3 以及 1x1 的卷积核组成的，并且借鉴了 ResNet 的跳过连接的思想。DarkNet53 相比于 ResNet 有更低的十亿次浮点数运算，能够以比 ResNet 快两倍的速度获得其相似的准确率。

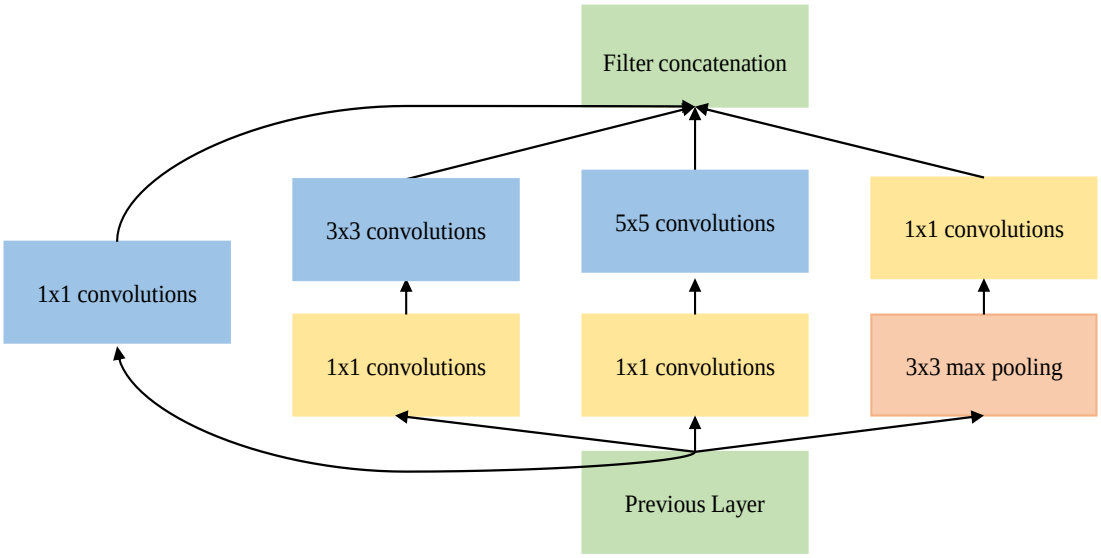


图 3.4 Inception 网络

3.2.3 算法对比

由于要使用到行车辅助系统，其实时性是考虑的重点之一，本小结将通过对比 Faster R-CNN、SSD 以及 YOLO V3 算法结合不同的骨干模型处理图片的效率。通过对测试集的分析，判断其平均处理一张图片所耗费的时间，从而判断算法的时间性能。

基于候选框的目标检测算法结合不同的骨干网络模型得到的结果如表 3.2 所示，实验分别对比了 Faster RCNN 结合 ResNet、Inception、以及 Inception ResNet 模型。统计其处理 3071 张测试集所用的平均时间。

同理，本实验也对流行的单级式系列目标检测算法结合不同的骨干网络进行实验。得到的检测结果如表 3.3 所示，实验分别对比了 SSD、RFCN[58] 结合 Mobilenet V1[60]、Mobilenet V2[61]、Inception ResNet V2 等模型的表现效果。

表 3.2 基于候选框检测算法时间性能对比

检测算法	网络结构	检测时间 /(ms)
Faster R-CNN	Inception ResNetV2	1082
Faster R-CNN	Inception ResNetV2 LOW	514
Faster R-CNN	Inception V2	213
Faster R-CNN	ResNet 50	216
Faster R-CNN	ResNet50 LOW	118
Faster R-CNN	ResNet101	224
Faster R-CNN	ResNet101_LOW	170

表 3.3 单级式检测算法时间性能对比

检测算法	网络结构	检测时间 / (ms)
SSD	MobileNet V1	87
SSD	MobileNet V2	86
SSD	Inception V2	92
SSDLite	MobileNet V2	97
RFCN	ResNet 101	178
YOLO V3	DarkNet 53	82

仅仅对比不同目标检测算法结合不同模型在检测时间上的性能是不够的，能够有效的检测到图片中的交通标志也是评价标准之一。因此，实验通过对比不同算法在不同的置信度下，计算其召回率，从而判断检测算法的性能。

实验的置信度从 0.5 开始选取并每递增 0.1 做对比实验。选择从 0.5 开始是因为只有当预测的概率在大于该置信度下才可以确定目标的有效性。基于候选区域的目标检测算法的结果如表 3.4 所示。

同理，使用同样的策略，对单级目标检测算法也做了上述的实验对比，其结果如表 3.5 所示。

表 3.4 基于候选区域检测算法在不同置信度下的召回率

检测算法与模型	置信度	召回率/ (%)
Faster R-CNN	0.5	71.9
(ResNet 101)	0.6	68.9
	0.7	66.1
	0.8	63.3
	0.9	58.7
Faster R-CNN	0.5	72.7
(Inception ResNet V2)	0.6	70.5
	0.7	67.6
	0.8	64.4
	0.9	59.4
Faster R-CNN	0.5	87.9
(ResNet 50)	0.6	85.9
	0.7	82.8
	0.8	78.4
	0.9	73
Faster R-CNN	0.5	92.2
(Inception V2)	0.6	90.4
	0.7	88.2
	0.8	85
	0.9	79

表 3.5 单级式目标检测算法在不同置信度下的召回率

检测算法与模型	置信度	召回率 / (%)
SSD (MobileNet V1)	0.5	78.9
	0.6	74.4
	0.7	69.7
	0.8	63.3
	0.9	53.5

检测算法与模型	置信度	召回率 / (%)
SSDLite (MobileNet V2)	0.5	83.8
	0.6	80.4
	0.7	75.3
	0.8	68.2
	0.9	56.5
SSD (MobileNet V2)	0.5	83.8
	0.6	80.9
	0.7	77.4
	0.8	72.3
	0.9	63.8
YOLO V3 (DarkNet 53)	0.5	98.4
	0.6	98
	0.7	97.4
	0.8	96.5
	0.9	95

3.3 实验结果及存在问题

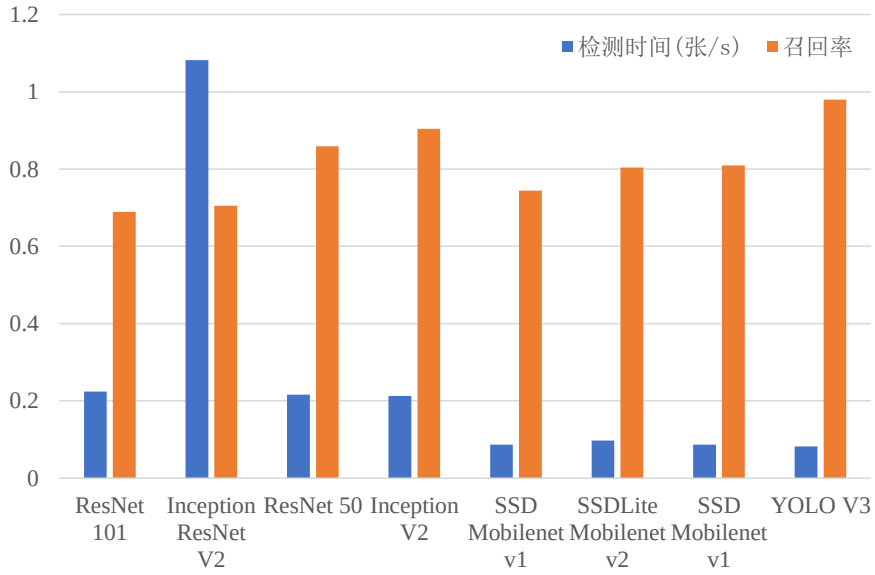
3.3.1 实验结果分析

本实验在 Linux Ubuntu16.04 环境下，使用 Python 3.6 和 TensorFlow 开源框架。在训练阶段，使用 4 块 Titan X 显卡服务器。通过迁移学习的方法，首先取得在 COCO 数据集上预训练的权重，之后将得到的训练权重结合交通标志训练集再次进行 30000 次训练，其中 Dropout 设置为 0.8，Batch size 为 32，使用 RMSprop 优化器，初始学习率为 0.004，Momentum 中 β 为 0.9。

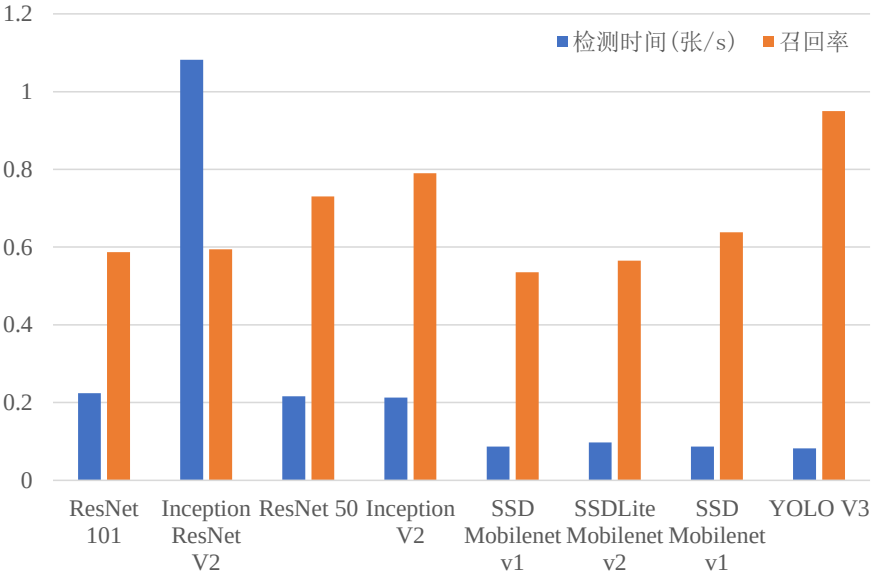
通过表 3.2 以及表 3.3 分析可得，在结合不同的网络模型下，YOLO V3 和 SSD 有很好的时间性能，处理一张图片的时间基本需要 80ms 左右。但是，对于 Faster R-CNN 算法来看，基本模型的处理时间为 200ms 左右，其中更有骨干模型为 Inception ResNet V2 模型，所耗时间达到了 1000ms。对比可得，SSD 和 YOLO 算法的检测速度基本上比 Faster R-CNN 算法快

3 倍左右，最多的快 12 倍。因此，从实时性来看，SSD 和 YOLO V3 更适合实时要求。

通过分析表 3.4 和 3.5 可知，在不同的置信度下，YOLO V3 的检测召回率遥遥领先，平均比 SSD 高出 20%，比 Faster R-CNN 高出 10%，由此可见，YOLO V3 在小目标的检测上优势更加明显。为了直观的表现不同算法的性能，将之汇总于图 3.5。



(a) 置信度为 0.6 时不同算法效果



(b) 置信度为 0.9 时不同算法效果

图 3.5 不同算法的在不同置信度下的性能对比

综上所述，YOLO V3 算法无论在实时性以及检测召回率方面，都有着优越的表现，实验中不同算法的召回率总体对比结果如图 3.6 所示，图中 FRCNN 表示 Faster RCNN 结合 ResNet50，FRI 表示 Faster RCNN 结合 InceptionV2。

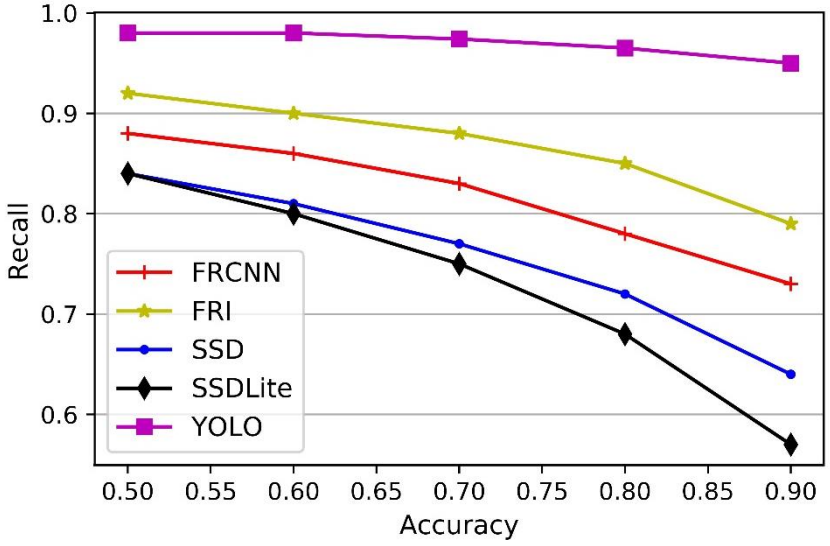


图 3.6 不同算法准确率与召回率的对比

3.3.2 存在问题及解决方案

通过权衡召回率以及检测速度，最终选取 YOLO V3 模型。虽然在小目标检测以及实时性上 YOLO V3 模型都有着极大的优势。但是由于速度上的提升以及利用 FPN 网络对小目标检测上的优化导致其在中大型目标的定位精度上就有了一定的损失。不同于 Faster R-CNN 系列，算法可以有两次调整检测框的机会，单级目标检测算法只能够一次完成。因此，如果前期检测定位不准确，那么就会影响后续分类的效果。部分定位不准确的检测图如图 3.7 所示。

由图可知，交通标志在图中占据了较大的比例，由于 YOLO V3 算法在中大型图标检测的预测框的缺陷，导致了 YOLO V3 算法在预测框性能上的损失，为了能够进一步提高检测框的检测精度，以提高对后续分类效果的提升，提出了结合视觉显著性检测的算法，从而对检测框进行二次修正，以提高算法的检测精确性，具体的方法将在下一节描述。

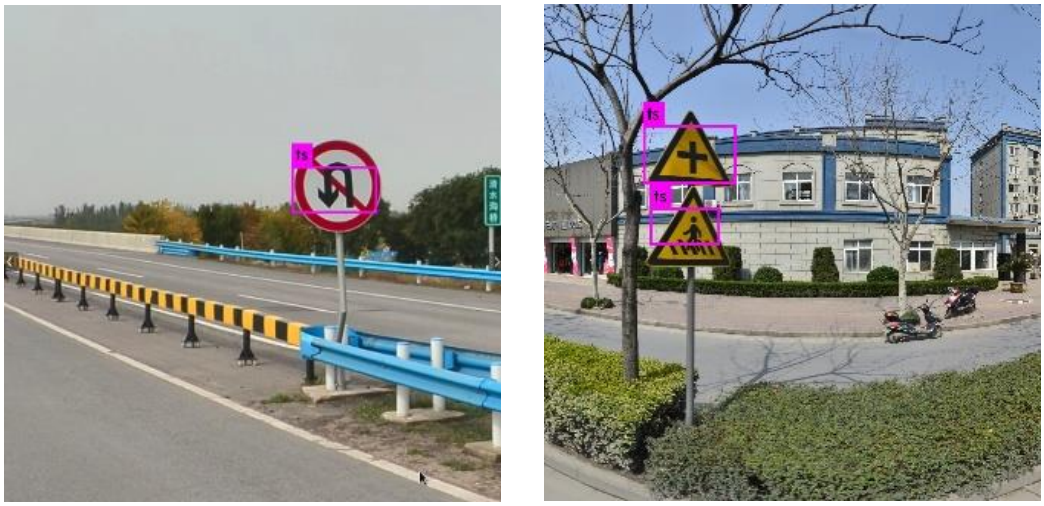


图 3.7 部分定位不准确的检测图

3.4 目标检测框的改进

在使用 YOLO V3 算法作为目标检测的基础框架后，能够有效的解决小目标检测的问题。但是却发现其对中大型的目标的检测效果并不理想，因此，本文提出了结合视觉显著性检测算法 Ranking Saliency[62]算法，从而对得到的目标检测框进行再次优化，二次调整其检测框的精度。使得其在后续分类的性能上得到优化。

3.4.1 视觉显著性算法

显著性检测算法是通过算法模拟人的视觉特点，提取图像中的显著区域。由于交通标志一般都为比较醒目的颜色，算法选取自底向上基于数据驱动的注意力机制。主要思想是仅通过感知数据的驱动，将人的视点指导到场景中的显著区域。正因为交通标志会与周围环境有较强烈的反差，所以通过图像的颜色、亮度、边缘等特征判断它与周围区域的差异，从而计算图像的显著性。算法的主要思想是根据以下公式计算结点的显著性。

$$f^{*}=(\mathbf{D}-\alpha\mathbf{W})^{-1}\boldsymbol{y} \tag{3-3}$$

公式(3-3)中，$\mathbf{D}$ 表示图的度矩阵，$\mathbf{W}$ 表示权重矩阵，$\boldsymbol{y}$ 表示指示向量，当前结点是种子结点 $\boldsymbol{y}$ 取 1，否则 $\boldsymbol{y}$ 取 0，f^{*}表示当前计算所得到的显著性值。通过(3-3)计算每个结点的显著性从而得到排名分数。

主要的算法思想可以分为以下几步。首先，通过超像素分割算法得到

种子节点；其次，由于一般图像的边界部分为背景部分，所以，分别以上、下、左、右四个边界分别作为初始种子点，使用公式(3-3)对其余点进行显著性计算，从而得到相关的显著性图；最后，为了进一步提升显著图的准确性，改变自适应阈值的二值分割方法，而是通过上一步得到的显著图的平均显著性作为新的阈值进行二值分割。

本实验主要的过程是，在基于 YOLO V3 目标检测算法的基础上，对检测到的交通标志再次进行显著性检测。首先，对得到的目标区域首先进行扩充，即对检测得到的长和宽扩充 20%从而使得扩充后的区域包含完整的交通标志，在此基础上，通过 Ranking Saliency 算法对扩充区域进行显著性检测，从而对检测框再次修正，确定交通标志更准确的位置。

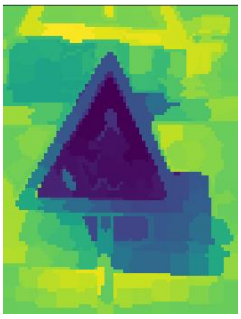
3.4.2 算法改进后实验结果及分析

通过结合视觉显著性的算法，其总体操作如图 3.8 所示。首先，通过目标检测算法得到的初始图如图 3.8(a)所示，扩充后的图如图 3.8(b)所示。之后，对其进行显著性检测，效果如图 3.8(c)所示。在得到了显著图后，对其进行阈值分割，得到如图 3.8(d)所示，对得到的阈值图进行形态学操作进一步优化。之后，分别对阈值图进行垂直投影，如图 3.8(e)所示以及水平投影，如图 3.8(f)所示，从而分割得到最终的交通标志图如图 3.8(g)所示。

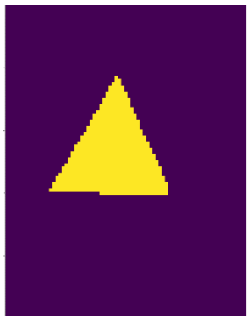
通过实验对比可得，结合视觉显著性检测算法确实对交通标志的预测框有一定的提升。通过 IoU 评判标准，对比单纯的使用 YOLO V3 目标检测算法和结合视觉显著性算法的效果。可以发现结合显著性算法后，对 IoU 有所提升，具体结果如表 3.6 所示。

(a)检测算法得到图

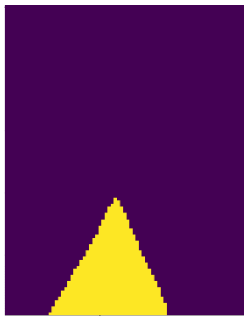
(b)经过扩充后的图



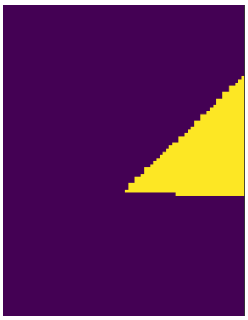
(c)显著性图



(d) 阈值分割图



(e)垂直投影



(f)水平投影

(g)结果图

图 3.8 结合视觉显著性算法检测效果

表 3.6 结合显著性检测算法的 IoU

检测算法	**IoU / (%)**
YOLO V3	79.57
YOLO V3 + Ranking Saliency	82.27

分析表 3.6 可知，结合了显著性算法后，对 IoU 的性能有所提升，平均提升了 3%。如果去除掉测试集中检测 IoU 已经大于 0.7 的图片，留下剩余的交通标志，再次使用显著性检测的算法，可以发现对于这些标志，结合后的算法在检测得到的平均 IoU 上，大约提升了 20%，得到结果如表 3.7 所示。

表 3.7 结合显著性检测算法对 IoU（<0.7）的表现

检测算法	**IoU / (%)**
YOLO V3	53.59
YOLO V3 + Ranking Saliency	74.28

仅仅检测到交通标志而不知道具体的类别是不能达到实用的。因此，实验利用 TT100K 数据集训练后续的分类网络。

通过分析 TT100K 数据集，其有效类别共有 151 类，分别对应了不同的交通标志。通过统计不同类别的个数，去除掉个数少于 100 的类别，得到最终的有效类别为 33 类。通过分析所得到的类别，由于类别个数的不均衡性，因此成比例的对不同的类别进行数据扩充。本实验为了尽可能保证交通标志的完整性，数据增强的方法主要有对图片进行加噪、对图片设置不同的对比度以及对图片设置不同的亮度。

本实验的运行环境为 Linux 系统下的 Ubuntu 16.04，使用 GTX 1060 3GB 内存显卡，电脑内存为 8G。使用迁移学习的思想，先取得在 ImageNet 预训练好的权重，之后，将通过数据增强得到的 33 类交通标志再次进行 10000 次训练。通过上述操作之后，分别对比不同网络模型结构下的分类的性能。具体的结果如表 3.8 所示。

除此之外，在确定分类模型后。实验还验证了目标检测定位精度的提升，对后续分类性能的影响。

表 3.8 不同网络模型的权重大小和准确率

分类网络	模型大小(M)	准确率(%)	分类网络	模型大小(M)	准确率(%)
ResNet V1 50	102.7	95	ResNet V2 101	179.2	92.6
Inception V1	26.8	90.5	PNASNet-5	345.7	77.8
Mobile V1 025 160	2.1	90.3	Mobile V1 050 224	5.6	94.1
Mobile V1 075 224	10.6	93	Mobile V1 100 224	17.2	95.9
Mobile V2 035 192	7	93.3	Mobile V2 050 224	8.2	94.4
Mobile V2 075 224	10.9	94.2	Mobile V2 100 224	13.4	94.7
Mobile V2 130 224	22	91.6	Mobile V2 140 224	24.9	91.7
NASNet-A	356.8	83.5	NASNet-A(Mobile)	22.1	79.7

通过上表对比可以发现，大部分网络结构均能取得较好的分类结果。因为最终的模型需要应用到驾驶辅助系统，那么模型的大小也是衡量的一个标准。因此，去除掉权重偏大的训练模型，最终选用准确率高、模型小的 MobileNet_V2_100_224 模型。使用该模型对得到的 33 类交通标志进行分类识别，验证结合算法的表现性能。分别统计模型对不同类别的召回率

以及精确率，并和文献[32]中所提到的算法进行比较，具体结果如表 3.9 所示，其中 FR 表示 Fast RCNN 算法，Zhu R 是文献[32]提出的算法，OR 和 OP 分别对应本文算法的召回率和精准率。

表 3.9 本文算法与文献[32]算法的对比

类别	**I2**	**I4**	**I5**	**IL60**	**IL80**	**IO**	**IP**	**P10**	**W55**
FR	0.32	0.61	0.69	0.8	0.39	0.65	0.67	0.51	0.50
Zhu R	0.82	0.94	0.95	0.97	0.94	0.89	0.92	0.95	0.72
OR	**0.87**	**0.95**	**0.99**	0.88	**0.94**	**0.91**	**0.99**	0.90	**0.97**
OP	0.87	0.99	0.99	0.95	0.85	0.88	0.98	0.84	0.95
类别	**P11**	**P12**	**P23**	**P26**	**P3**	**P5**	**PG**	**PH 45**	**W57**
FR	0.44	0.48	0.70	0.60	0.60	0.72	0.89	0.83	0.56
Zhu R	0.91	0.89	0.94	0.93	0.91	0.95	0.91	0.88	0.79
OR	**0.96**	**0.94**	0.93	0.9	0.88	**0.97**	**1**	**0.95**	**0.98**
OP	0.92	0.83	0.94	0.94	0.82	0.93	1	0.90	0.99
类别	**PL100**	**PL 120**	**PL30**	**PL40**	**PL5**	**PL50**	**PL 60**	**PL 70**	**W59**
FR	0.82	0.57	0.43	0.48	0.65	0.29	0.42	0.23	0.67
Zhu R	0.98	0.96	0.94	0.96	0.94	0.94	0.93	0.93	0.82
OR	0.96	**0.97**	0.67	0.92	**0.96**	0.90	0.89	**0.97**	**0.97**
OP	0.98	0.95	0.83	0.95	0.98	0.81	0.78	0.58	0.97
类别	**PL80**	**PM 20**	**PN**	**PNE**	**PO**	**PR40**			
FR	0.40	0.53	0.59	0.77	0.29	0.98			
Zhu R	0.95	0.88	0.91	0.93	0.67	0.98			
OR	0.84	0.82	**0.98**	**0.99**	**0.75**	**1**			
OP	0.97	0.78	0.99	0.99	0.85	0.97			

通过分析表 3.9 可知，以 YOLO V3 作为骨干网络并结合显著性检测的目标检测算法，在 I2、I4、I5、PG 等类别的交通标志的召回率方面有一

定的提升。除此之外，结合算法在交通标志检测的精确率方面也有不错的表现。

3.5 本章小结

本章首先通过在检测时间以及检测召回率方面，对比了目前比较流行的基于深度学习的目标检测算法，通过对比发现，YOLO V3 在实时性以及检测召回率等方面，均取得了优异的表现。通过实验发现，YOLO V3 算法虽然使用特征金字塔网络的思想对小目标的物体检测进行了优化，但是对于中大型物体，算法便会出现定位不准确的情况。为了一定程度上弥补这一缺点，提出了结合视觉显著性的算法，对检测算法得到的目标检测框进行二次优化。除此之外，仅仅检测到交通标志而不知道具体类别是不能达到实用的。因此，本章使用 TT100K 数据集，首先分析其数据构成，为了满足实验的需求，对数据集进行数据增强，训练得到后续的分类网络。其次，通过权衡不同模型的准确率以及模型大小等指标性能，从而选取最适合本实验的模型。最后，将检测到的结果送入后续的分类网络进行识别，通过分析算法的召回率和精确率确定算法的有效性。

4 结合跟踪算法的交通标志检测与识别的进一步优化

通过上一章节的论述，确定了实验所需的相关网络模型。实验采取 YOLO V3 作为基础的检测模型，除此之外，使用 MobileNetV2_100_224 作为后续的分类网络，并取得了很好的检测以及分类效果。但是，毕竟所有的实验只是基于单张图片的检测识别。作为辅助驾驶系统的一部分，整个操作应该是基于视频流的处理。虽然视频也是由多张图片所组成，但是归根结底，仍然有许多不同之处，使得检测与识别又有了新的挑战。如何能有效利用视频的信息，对算法进行优化是本章的研究重点。

众所周知，为了适应人眼的观察需求，视频基本都是由每秒 25 帧组成。这样就带来一个问题，视频其实是包含很多冗余信息的，连续的视频帧其实都包含了重复的信息。而对于本文的交通标志检测识别算法来说，对检测到的每一帧如此往复的进行识别，必然会使得实时性下降，影响整个系统的时间性能。并且，连续的分类识别会提高错误分类发生的概率。除此之外，目标检测算法在检测的过程中会存在掉帧或者目标物体被短暂遮挡的情况，如何在这些情况下仍然能够很好的完成交通标志的检测与识别是需要考虑的问题。

因此，本章节对交通标志检测识别算法进行进一步优化，使得其在时间性能和识别准确率上能够进一步提升。算法的主要思想是充分利用视频的多帧相似的信息。在检测到标志的初期，通过对固定帧长(10-20 帧)的检测识别结果进行投票确定当前的交通标志属于哪一类别。当确定类别之后，只使用跟踪算法对当前检测得到的目标进行跟踪，而不再对其进行目标分类。这样就可以优化因持续分类所带来的时间性能损耗，并且跟踪算法对每个类别分别保存了一定时间长度的帧信息，从而能够很好的应对掉帧、目标被遮挡等情况的发生。

4.1 算法的准备工作

4.1.1 数据集构建

为了测试真实路况下算法的性能，本实验所使用的视频数据集是通过

车载摄像机的方式，真实的收集于杭州的某路段之上，数据集中包含白天以及夜晚不同条件下的视频，其中每段时长约为 10 分钟左右。

由于深度学习的效果一定程度上取决于数据量，为了尽可能使模型满足当地的路况信息，需要对已有的数据集进行扩充，对检测算法进行进一步优化。扩充的方法是对当前收集到的视频数据集进行采样，通过 Opencv 提取到数据集当中包含交通标志的片段，之后对视频片段进行分帧处理，即每隔 10 帧截取一张图片。在得到相关图片之后，利用标注软件对当前的图片标注交通标志的位置。标注软件会记录交通标志左上角以及右下角位置的坐标，图中红色框以及蓝色框为标记框，如图 4.1 所示。

图 4.1 使用标注软件标注交通标志示意图

通过上述方法得到相关坐标后，并不能直接用来用作 YOLO V3 的训练样本，因为 YOLO V3 有其自己的训练数据格式。YOLO V3 将标记软件所得到的 ROI 框进行进一步处理，获取其 ROI 中心点相对于图片大小的比例坐标和 ROI 框宽和高相对于图片大小的比例。具体的计算公式如下所示。

$$d_w = 1 / s_0 \tag{4-1}$$

$$d_h = 1 / s_1 \tag{4-2}$$

$$x = (b_0 + b_2 / 2) * d_w \tag{4-3}$$

$$y = (b_1 + b_3 / 2) * d_h \tag{4-4}$$

$$w = b_2 * d_w \tag{4-5}$$

$$h = b_3 * d_h \tag{4-6}$$

在上述公式中，d_w 表示图像宽度的倒数，d_h 表示图像高度的倒数，s_0 表示原始图像的宽度，s_1 表示原始图片的高度，b_0 和 b_1 分别代表 ROI 左上角的横坐标和纵坐标，b_2 和 b_3 则分别代表 ROI 右下角的横坐标和纵坐标，x 和 y 对应中心点相对原始图片的宽和高的比例，w 和 h 对应宽和高相对原始图片的比例。通过上述公式，得到最终训练所需的数据格式。

4.1.2 前期算法存在问题及优化

通过之前章节的工作，得到交通标志检测与识别算法，算法具体的流程图如图 4.2 所示。通过对已有的视频数据集进行分析发现，算法能够有效的完成交通标志的检测以及识别工作，在不同环境下的检测效果如图 4.3 所示。通过分析识别结果可以发现，算法的在白天表现效果明显优于夜间的表现，主要原因是白天的光照比较充分，并且干扰的光源比较少，对检测算法造成的影响并不大。反观夜间，由于本身的数据集就对夜间环境的收集量相对较少，使得其测试集的分布和训练集的分布产生了一定的不同。而作为深度学习，获取到高性能的条件之一便是训练集和测试集尽可能服从同一分布，即训练集尽可能包含各种现实可能发生的情况。

除此之外，由于算法使用检测完成后对检测得到的标志再次进行分类的思想，这样便会导致要对每一帧图片进行先检测再分类。对每一帧都进行分类处理并没有有效利用到视频连续多帧的信息冗余的特点，导致因连续分类而产生时间性能下降；其次，对连续的每一帧都进行分类，不可避免会造成错误率的增加；最后，在检测过程中会出现目标物体被遮挡以及掉帧的情况，算法在此类情况下表现的并不是很理想。因此，本章节通过有效利用视频的冗余信息来提升算法的性能。首先，利用对固定帧长(10-20 帧)检测与识别，将识别的结果进行保存到字典当中，之后通过对当前结果投票确定交通标志的类别，该做法能够有效提高分类的准确率；其次，在结合 Deep SORT[63]跟踪算法后，当确定交通标志类别之后，只是持续对检测到的目标进行跟踪而停止后续的分类操作，从而减少算法在时间上的损耗；最后，该算法还能一定程度上克服目标物体被遮挡以及掉帧等情况的发生。

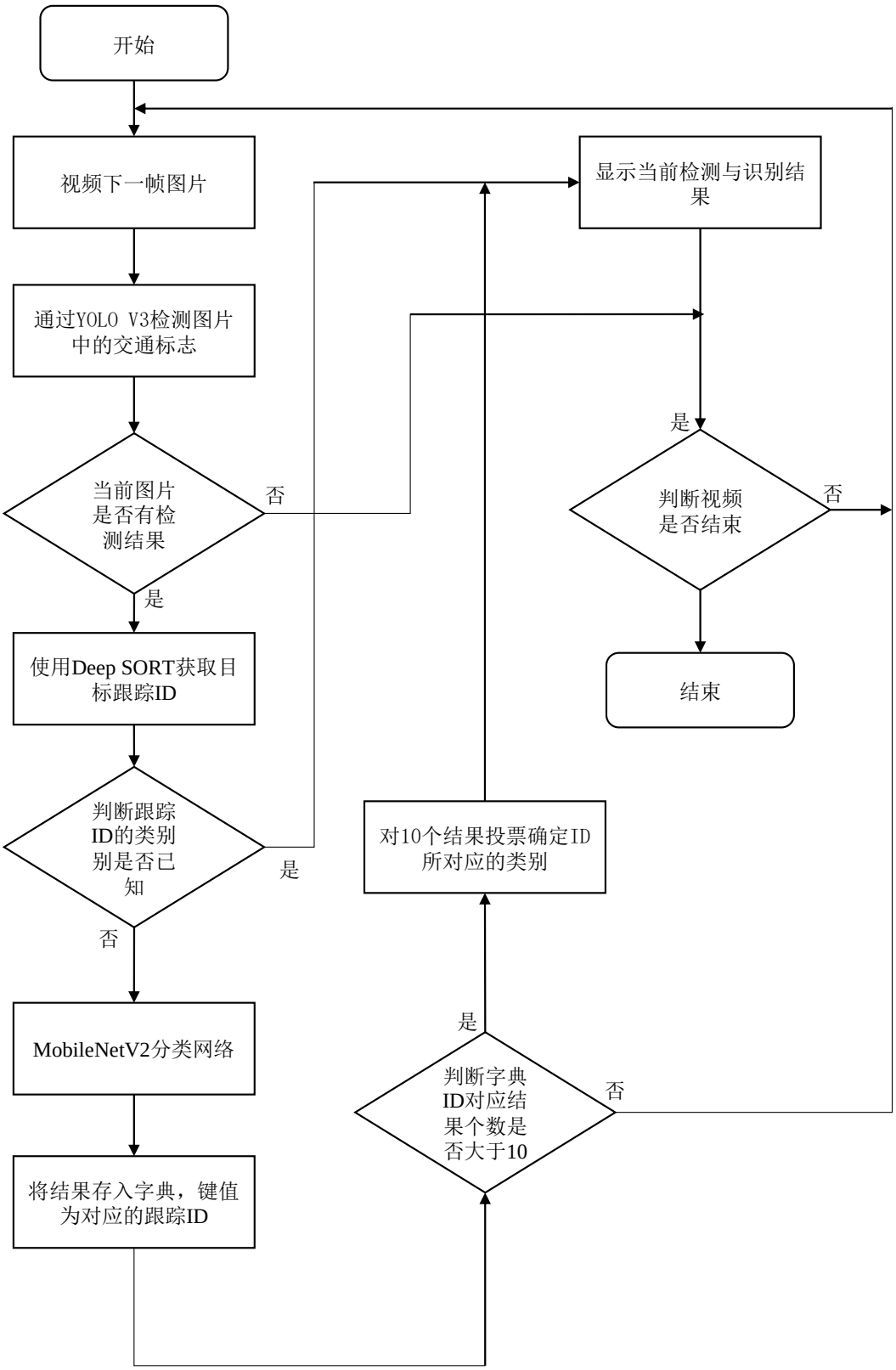
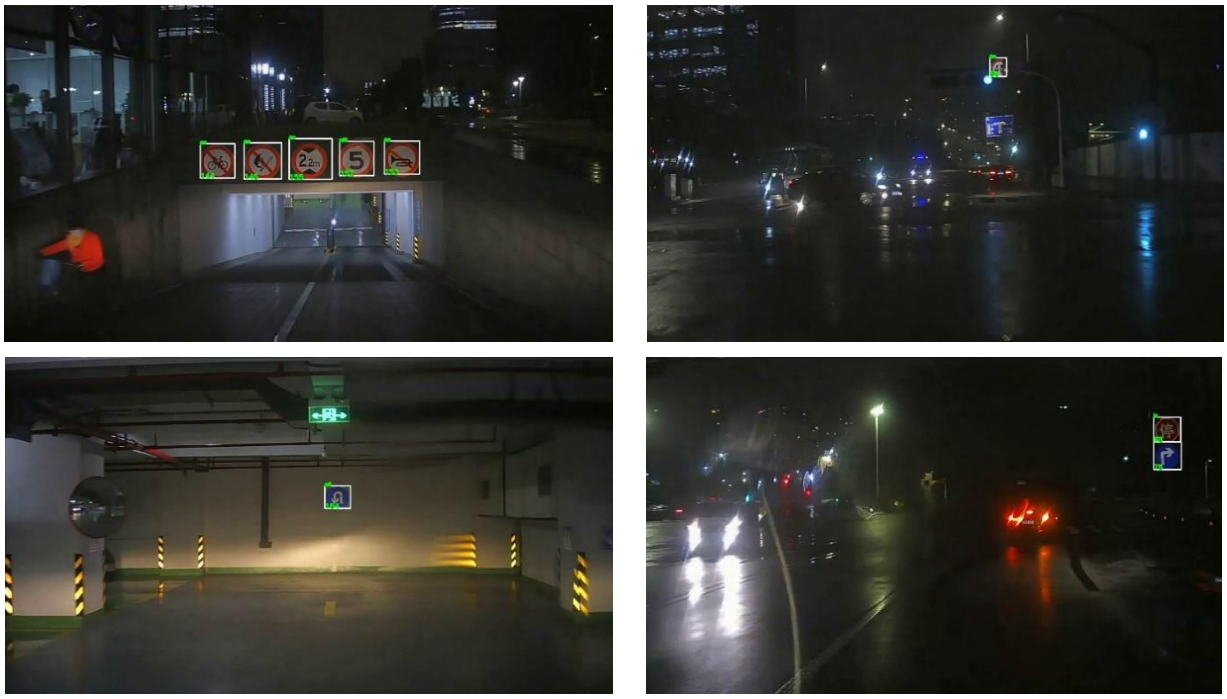


图 4.2 结合跟踪算法的交通标志检测与识别算法流程图

(a) 白天检测与识别结果



(b) 夜间检测与识别结果

图 4.3 不同条件下交通标志检测与识别结果

4.2 Deep SORT 算法介绍

Deep SORT 算法是在 SORT 目标跟踪算法基础上的改进，改善 SORT 算法提取特征的方式，通过用深度学习完成提取特征的提取。使得得到的特征更具有代表性，从而提升后续的跟踪性能。

本实验通过目标检测算法和跟踪算法的结合，先通过目标检测算法检测到相关的交通标志的信息，之后将检测到的信息传递给 Deep SORT 算法。Deep SORT 已经有预训练好的模型权重，算法将检测得到的交通标志图片传递给 Deep SORT 模型后，通过模型的处理，将当前的图片转化为后续目标跟踪所需的 128 维向量。由于深度学习在特征提取方面的优势，使得相似的物体在通过深度学习进行特征提取后，有着极其相似的特征向量。跟踪算法通过不同方法比较获得前后帧的关系来对物体实现跟踪。

当 Deep SORT 获取到当前的标志位置以及当前的交通标志的特征向量之后，将会通过以下方法对目标进行跟踪。

首先，算法会使用一个 8 维的空间去刻画当前轨迹在某时刻的状态，可以使用 $(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ 来表示，其中 u 和 v 表示当前目标的中心点的坐标，γ 表示纵横比，h 表示图片的高度，$\dot{x}, \dot{y}, \dot{\gamma}, \dot{h}$ 则对应于在不同指标上的运动速度。

其次，通过使用一个基于常量速度模型和线性观测模型的标准 Kalman 滤波器进行目标运动状态的预测，对当前的目标检测框预测出新的可能目标位置。之后，计算当前的目标框和通过卡尔曼滤波预测到的目标框的马氏距离，当此距离小于设置的阈值时，便证明预测到的框是正确的，从而对其进行跟踪，具体的公式如下所示。

$$d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \tag{4-7}$$

其中，d_j 表示第 j 个检测框的位置，y_i 表示第 i 个追踪器对目标的预测位置，$\mathbf{S_i}$ 表示检测得到的位置与平均追踪位置的协方差矩阵。$d^{(1)}(i, j)$ 就表示检测位置和预测位置的马氏距离。

由于运动的不确定性，单纯的使用马氏距离进行卡尔曼滤波预测成功与否的判断是不准确的。因此，除了使用马氏距离之外，引入了新的判断标准，即对当前检测到的所有图片的特征向量建立一个“博物馆”。当有新的目标被检测到之后，通过计算当前目标的特征向量和“博物馆”中所

有特征向量的余弦距离，取其中最小的余弦距离。如果发现得到余弦距离小于某个阈值距离，那么就证明当前检测到的特征向量可以被关联跟踪。计算公式如(4-8)所示。

$$d^{(2)}(i, j) = min\left\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in R_i\right\} \tag{4-8}$$

其中，$\boldsymbol{r_j}$表示当前检测的特征向量，$\boldsymbol{r_k^{(i)}}$表示特征向量“博物馆”中的不同特征向量，$\mathbf{R}$ 表示特征向量矩阵，$d^{(2)}(i, j)$为计算得到的最小余弦距离。最后，整个算法在确定是否能够成功关联匹配时，使用两种距离的加权和，如公式(4-9)所示。通过调整λ的值来权衡两种度量方式的关系。

$$c_{i,j} = \lambda d^{(1)}(i, j) + (1 - \lambda)d^{(2)}(i, j) \tag{4-9}$$

除此之外，由于部分遮挡问题的发生，遮挡的时间越长，会使得由于长时间未更新位置信息，导致协方差矩阵变大。在计算通过卡尔曼滤波器预测的框与当前检测得到的框的马氏距离的时候，因为计算要使用协方差的倒数，便会使得其更可能匹配到长时间未出现的目标，而非直观上间隔时间最短的目标（由于间隔时间短导致位置信息的不断更新，使得获得的协方差矩阵小）。导致的问题便是，在长时间丢帧后，算法不会匹配到距离丢帧时间最短的目标，而是匹配到很久之前丢失的目标。因此，使用级联匹配的思想，即按照消失时间的顺序，从消失时间最短到消失时间最长分别对之进行匹配，这样便能够保证对最近出现的目标赋予最大的优先权。

4.3 实验结果及分析

通过上述章节论述，整体的交通标志检测流程可以分为如下的步骤。首先，使用 YOLO V3 和 Deep SORT 算法进行交通标志检测与跟踪；其次，使用分类网络对固定帧长的检测结果进行分类；最后，不再对已经确定类别的检测结果进行持续分类，只是对检测得到的结果进行跟踪。

之所以采用这类方法，主要有以下的考虑。第一，对目标进行持续的检测以及分类过程中，由于当前的车辆是持续运动的，随着运动的远近，会导致检测到的目标的尺寸发生改变，并且也会出现检测不够准确的情况。这样便会一定程度上增加后续分类的难度，使得分类的准确率下降。第二，持续的对每一帧进行检测以及后续识别是比较耗时的，并且持续的分类会调高算法的分类错误率。第三，在使用检测算法的过程中，会遇到掉帧以

及目标物体被遮挡的情况，影响了算法的性能。

通过实验表明，结合后的算法有以下优点。第一，结合跟踪算法后，只需在检测到交通标志后，对固定的帧长进行识别，无需持续识别，这样就提升了算法的性能；第二，通过对连续的 10 帧进行识别，将识别结果存到字典当中，只有当字典键对应的结果超过 10 时，通过投票返回当前 10 个结果中占比最多的结果，这样会提升识别算法的准确率；第三，由于跟踪算法对每一个类别会保存一定帧长的数据，从而可以优化目标检测算法遇到掉帧以及遮挡等问题。

本实验的运行环境为 Linux 系统下的 Ubuntu 16.04，使用 GTX 1060 3GB 内存显卡，电脑内存为 8G。实验初期，先对已有的测试数据集进行相关数据的统计。测试数据集收集于真实的路况之下，分别有白天以及夜间两段视频。先将数据集中的交通标志统计如表 4.1 所示，经统计此路段有 10 个交通标志并未包含在现有的训练集内，影响了算法的识别率。

表 4.1 测试集中不同交通标志的统计

类别	**PL30**	**PN**	**I5**	**P11**	**P23**	**PL40**	**I2**
个数	2	17	2	3	2	3	12
类别	**PM20**	**P26**	**P12**	**P10**	**W59**	**PL80**	**未包括**
个数	2	3	5	5	1	13	10

4.3.1 在检测效果上的对比

为了评价模型在真实视频下的检测效果，测试集使用两段视频。一段是白天的行车视频，另一段是夜晚的行车视频。首先通过人工收集的方式，收集两段视频中总共出现交通标志的个数，具体的结果如表 4.1 所示；其次，通过计算当前目标检测算法检测得到的交通标志数，计算其召回率和精确率，得到的结果如表 4.2 所示。

通过观察表 4.2 可得知算法在白天达到了一定的实用效果。部分未检测到和误检的图如图 4.4 所示，图中每一行的左侧为原始图，右侧为局部放大图。通过观察图 4.4 上行可知，车上的图标与交通标志有十分类似的信息，导致了算法的误检。通过观察图 4.4 下行可知，夜间的光照条件太差，影响了交通标志的有效检测。分析算法的召回率和精确率可知，算法

在白天的整体性能要优于夜间的表现。主要的原因有以下几点，第一，夜间的情况比较复杂，不同的车辆的刹车灯、大灯、雾灯都产生了不可控的光照条件，误检也多发生在这种情况之下。比如在夜间将大车的尾灯检测为交通标志或者是将红灯检测为交通标志，但是，白天的光照充足，干扰条件少，类似这种情况在白天是极少发生的。第二，夜间的数据集相对较少，使得模型针对夜间的训练并不充分，导致的结果就是夜间的表现效果并不理想。为了解决这一情况，除了我们自己收集更多数据集之外，希望大型机构可以收集更多有关夜间的交通标志数据集，以提高完善后续的实验性能。第三，部分标志的信息与交通标志的信息太过类似，算法难以区分发生了误判。

表 4.2 不同条件下算法检测结果

时间	精确率	召回率
白天	87.8	94.93
夜晚	56.2	76.6

4.3.2 在时间性能上的对比

为了体现结合跟踪算法的交通标志识别系统在时间性能上的提升，对比算法在一秒中处理的帧数(Frame per Second, FPS)。分别对比结合跟踪算法与不结合跟踪算法在处理含有交通标志的图像时一秒中平均处理帧数和不含有交通标志的图像时一秒中平均处理的帧数。具体的结果如表 4.3 所示。

表 4.3 是否结合跟踪算法的 FPS 对比

是否含有交通标志	不结合跟踪算法（FPS）	结合跟踪算法(FPS)
含有	2.7	6.5
不含有	7.2	7.2

通过对比可以发现，在不含有交通标志的区域，由于检测算法并没有检测到交通标志，因此也无需分类和跟踪，时间性能上是没有区别的。但是，在有交通标志的区域，不结合跟踪算法需要持续对每一帧进行检测与分类，处理速度只有每秒 2.7 帧。而结合跟踪算法后，由于只是检测前期对固定帧长进行分类识别，后续并不需要持续的识别，每秒处理的帧数提

升了 4 帧左右，提升幅度达到 50%，基本达到了在本实验环境下的最高性能。除此之外，由于跟踪算法会保存一定时间长度的帧数，建立对应的数据“博物馆”。所以，在碰到短暂的掉帧以及遮挡情况时，并不需要重新识别，而是与已有的类别“博物馆”数据集进行余弦距离的比较，从而确定相应的类别。这样对比于不使用跟踪算法，能进一步提升算法的时间性能。因此，通过结合跟踪算法，确实能够在算法的时间性能上做出一定的改进。

4.3.3 在识别效果上的对比

在识别方面，由于夜间的外界干扰较多，算法在夜间的表现并不理想。因此，实验主要针对白天的行车视频进行分析。为了进一步加强识别的效果，只有当检测框的分辨率大小超过 22x22 后才对其进行识别。为了验证跟踪算法对识别性能的提升，分别对比分析了算法结合跟踪算法与不结合跟踪算法对分类效果的影响。具体的结果如表 4.4 所示。通过对结果视频进行分析，总结识别错误的类别，部分错误分类的结果如图 4.5 所示，其中每一行左侧为原始检测图，右侧为局部放大图。识别错误主要有以下几方面的原因。第一，由于固有数据集的缺陷，本地路段上某些交通标志的类别并未在已有的交通标志数据集内，导致了分类的失败，如图 4.5 下行所示，“停”字并未在已有的数据集内。第二，某些类别的相似性太大，在测试集中，存在将限速 80 误分类为限速 30 等情况，如图 4.5 上行所示。通过表 4.4 可知，结合跟踪算法后，对识别的准确率有明显提升，提升幅度约为 15%。

表 4.4 是否结合跟踪算法的准确率对比

评价标准	未结合跟踪算法	结合跟踪算法
准确率	73.48	89.67

图 4.4 漏检以及误检图

图 4.5 部分识别错误图

4.4 本章小结

尽管交通标志检测以及识别算法在现有的交通标志数据集的表现相对来说比较优异。但是，最终的应用场景是基于真实环境下的，并且数据是以视频流的形式传递的。如何能更好的适应真实环境，优化算法对视频处理的性能是更有意义的研究。

本章为了提高算法的性能。首先分析现有算法的缺点：第一，由于视频连续帧的冗余信息多，连续的检测以及分类会对算法的性能产生影响；第二，连续的分类也会使得最终的识别错误率提高；第三，检测算法遇到掉帧以及目标物体被遮挡的情况表现并不好。

为了更好的利用视频中的多帧冗余的信息，使用识别算法在检测前期进行固定帧长的识别(10-20 帧)，对识别结果进行投票来确定该目标的类别。当确定标志种类之后，停止分类识别，只是对当前的目标进行跟踪。这样的做法有以下优点：第一，通过投票得到的交通标志种类更准确，减少算法的识别错误率；第二，当确定类别后，不再持续分类，从而提高算法的实时性；第三，结合跟踪算法后可以一定程度上克服目标被遮挡或掉帧带来的影响。

最后，通过分别分析算法的时间性能和识别性能，来确定结合跟踪算法的有效性。

5 总结与展望

5.1 研究总结

随着人工智能的蓬勃发展，让计算机协助人类完成更多的事情已成为趋势。近几年来，得益于计算机计算力的提升和大数据的威力，无人驾驶技术成为了热点研究方向。由于交通标志在道路当中能够提供最多的信息，因此对交通标志的检测与识别成为无人驾驶的重要一环。本论文研究的重点则是如何更加快速、准确的识别交通标志。从而作为辅助驾驶的一部分，帮助驾驶者做出更正确的判断，以此也能加快无人驾驶的发展。本文的研究工作主要有以下几部分。

第一，由于深度学习的飞速进步，出现了众多基于深度学习的目标检测算法。本文分别介绍了基于候选框的目标检测算法以及单级式目标检测算法。通过对比不同目标检测算法结合不同骨干网络的检测时间以及检测的召回率，确定了实时性更强、检测召回率更高的基于 YOLO 的检测算法。

第二，虽然基于 YOLO 的检测算法在时间性能上表现较好。但是，算法对于中大型目标会出现检测框不精确的情况，这样会对后续的分类性能造成影响。为了缓解这一情况，提出了结合视觉显著性检测的算法，对检测得到的目标进行二次矫正，提高其检测的精度。

第三，对现有交通标志数据集进行数据增强，训练合适的分类网络。将优化后的 YOLO 算法得到的检测结果送入后续的分类网络。通过权衡不同模型的权重大小、识别的召回率与精确率等指标选取最合适的模型。

第四，在构建交通标志检测与分类算法后。对现实条件下的视频进行测试时发现有以下弊端。首先，由于是基于视频处理的，算法对视频的每一帧进行持续检测与识别，这种做法会影响算法的时间性能；其次，连续的识别会提高算法识别出错的概率；最后，目标检测过程中会出现目标被遮挡或者掉帧的情况，已有算法表现并不理想。

第五，为了解决上述问题，考虑到视频多帧冗余的特点，本文提出了结合卡尔曼滤波的跟踪算法。在检测初期对连续帧(10-20 帧)进行识别，

通过对结果进行投票从而确定最终类别。当确定类别后，后续算法则只对当前结果跟踪而无需持续分类，从而提高算法的时间性能。并且，该算法能够有效解决掉帧以及目标被遮挡的情况。

5.2 工作展望

虽然本文使用了现阶段优秀的目标检测算法以及分类网络，并且对存在的不足进行了适度的改进，提高了其在分类精度以及实时性方面的性能。但是，仍然还有值得去进一步研究的内容。

第一，本文通过结合算法来对目标检测框进行二次修正。希望在后续研究过程中，能够通过设计更好的网络模型一步完成这一目标。

第二，本文的交通标志识别系统在夜间的表现并不理想。由于夜间的情况更复杂，尤其是光照的不确定性，使得检测的难度提升。如何能进一步克服夜间的不足是下一步研究的重点。

第三，数据样本不足。虽然有 TT100K 数据集作为训练数据集，但最终有效样本类别只有 33 类，远远没有达到总的交通标志类别。对于系统训练来说也还不够充分，尤其是夜间样本的缺乏，导致夜间效果的下降，希望可以通过更多途径收集到更多的交通标志。

第四，虽然通过结合跟踪算法可以解决遮挡、掉帧等情况，并且能够一定程度上提高算法的时间性能。但是对于实际应用，应当进一步优化算法的性能，使其在时间性能方面达到更好的效果。

参考文献

[1] Levinson J, Askeland J, Becker J, et al. Towards fully autonomous driving: Systems and algorithms[C]. 2011 IEEE Intelligent Vehicles Symposium (IV), 2011. 163-168.

[2] Timofte R, Prisacariu V A, Gool L V, et al. Combining Traffic Sign Detection with 3D Tracking Towards Better Driver Assistance[M]. Emerging Topics In Computer Vision And Its Applications, 2011.

[3] H. Kamada, S. Naoi and T. Gotoh. "A compact navigation system using image processing and fuzzy control[C]" IEEE Proceedings on Southeastcon, New Orleans, LA, USA, 1990. 1:337-342.

[4] A. de la Escalera, L. E. Moreno, M. A. Salichs and J. M. Armingol, "Road traffic sign detection and classification[C]" in IEEE Transactions on Industrial Electronics, 1997, 44(6):848-859.

[5] J. Miura, T. Kanda and Y. Shirai, "An active vision system for real-time traffic sign recognition[C]" ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.00TH8493), Dearborn, MI, USA, 2000. 52-57.

[6] P. Arnoul, M. Viala, J. P. Guerin and M. Mergy, "Traffic signs localisation for highways inventory from a video camera on board a moving collection van[C]" Proceedings of Conference on Intelligent Vehicles, Tokyo, Japan, 1996. 141-146.

[7] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, et al. "Road-Sign Detection and Recognition Based on Support Vector Machines[C]" in IEEE Transactions on Intelligent Transportation Systems, 2007, 8(2):264-278.

[8] Han Liu, Ding Liu and Jing Xin, "Real-time recognition of road traffic sign in motion image based on genetic algorithm[C]" Proceedings. International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, (1):83-86.

[9] Ming-Kuei Hu, "Visual pattern recognition by moment invariants[J]" in IRE Transactions on Information Theory, vol. 8, 1962, 8(2):179-187.

[10] GarcíaGarrido, Miguel Ángel, Sotelo, et al. ″Fast road sign detection using Hough transform for assisted driving of road vehicles[C]″. International Conference on Computer Aided Systems Theory. Springer Berlin Heidelberg, 2005. 543-548.

[11] Boumediene M, Cudel C, Basset M, et al. Triangular traffic signs detection based on RSLD algorithm[J]. Machine Vision and Applications, 2013, 24(8):1721-1732.

[12] Gao X W, Podladchikova L, Shaposhnikov D, et al. Recognition of traffic signs

based on their colour and shape features extracted using human vision models[J]. Journal of Visual Communication and Image Representation, 2006, 17(4):675-685.

[13] A. Ruta, Y. Li and X. Liu, "Detection, Tracking and Recognition of Traffic Signs from Video Input[C]," 2008 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, 2008. 55-60.

[14] Gomez-Moreno H, Maldonado-Bascon S, Gil-Jimenez P, et al. Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition[J]. IEEE Transactions on Intelligent Transportation Systems, 2010, 11(4):917-930.

[15] 黄志勇. 基于支持向量机的交通标志识别系统的研究[D]. 北京工业大学，2004.

[16] F. Zaklouta and B. Stanciulescu, Real-time traffic sign recognition using spatially weighted HOG trees[C]. 2011 15th International Conference on Advanced Robotics (ICAR), Tallinn, 2011. 61-66.

[17] Takaki M, Fujiyoshi H. Traffic Sign Recognition Using SIFT Features[J]. EEJ Transactions on Electronics, Information and Systems, 2009,129(5):824-831.

[18] Lu K, Ding Z, Ge S. Sparse-Representation-Based Graph Embedding for Traffic Sign Recognition[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4):1515-1524.

[19] Bin Wang, Bin Kong. A novel traffic sign recognition algorithm based on sparse representation and dictionary learning[J]. Journal of Intelligent & Fuzzy Systems 2017, 32(5): 3775-3784.

[20] Dan Cireşan, Meier U , Masci J , J. Schmidhuber. Multi-Column Deep Neural Network for Traffic Sign Classification[J]. Neural networks: the official journal of the International Neural Network Society, 2012, 32:333-338.

[21] D. Cireşan, U. Meier, J. Masci and J. Schmidhuber, A committee of neural networks for traffic sign classification[C]. The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011, 1918-1921.

[22] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale Convolutional Networks[C]" The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011, 2809-2813.

[23] J. Jin, K. Fu and C. Zhang. Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5):1991-2000.

[24] Rongqiang Qian, Bailing Zhang, Yong Yue, et al. Robust chinese traffic sign detection and recognition with deep convolutional neural network[C]. 2015 11th International Conference on Natural Computation (ICNC), Zhangjiajie, 2015. 791-796.

[25] R. Qian, Q. Liu, Y. Yue, F. Coenen et al. Road surface traffic sign detection with

hybrid region proposal and fast R-CNN[C]. 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, 2016. 555-559.

[26] Z. Zuo, K. Yu, Q. Zhou, X. Wang, et al. Traffic Signs Detection Based on Faster R-CNN[C]. 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, 2017. 286-288.

[27] Jianming Zhang, Manting Huang, Xiaokang Jin, et al. A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2[J]." Algorithms, 2017, 10(4):127.

[28] Stallkamp J, Schlipsing M, Salmen J, et al. The German Traffic Sign Recognition Benchmark: A multi-class classification competition[C]. The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011. 1453-1460.

[29] R. Timofte, K. Zimmermann and L. V. Gool. Multi-view traffic sign detection, recognition, and 3D localization[J]. Machine Vision and Applications, 2014, 25(3):633-647.

[30] Larsson F, Felsberg M. Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition[J]. Scandinavian Conference on Image Analysis, 2011, 6688:23 8-249.

[31] Grigorescu C, Petkov N. Distance sets for shape filters and shape recognition[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2003, 12(10):1274-1286.

[32] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li and S. Hu. Traffic-Sign Detection and Classification in the Wild[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, 2110-2118.

[33] Lécun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[34] Dhiraj, Rohit Biswas, Nischay Ghattamaraju. An effective analysis of deep learning based approaches for audio based feature extraction and its visualization[J]. Multimedia Tools and Applications, 2018.1-24.

[35] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection[C]. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, 1:886-893.

[36] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.

[37] Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision and Image Understanding, 2008, 110(3):346-359.

[38] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object

detection. Proceedings. International Conference on Image Processing, Rochester, NY, USA, 2002. 1-1.

[39] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297.

[40] Quinlan J R. Induction of decision tree[J]. Machine Learning, 1986, 1(1):81-106.

[41] Breiman L. Random Forests[J]. Machine Learning, 2001, 45(1): 5-32.

[42] Felzenszwalb P F, Huttenlocher D P. Efficient Graph-Based Image Segmentation[J]. International Journal of Computer Vision, 2004, 59(2):167-181.

[43] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014: 580-587.

[44] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[M]. Communications of the ACM, 2017, 60(6):84-90.

[45] Girshick R. Fast R-CNN[C]. IEEE International Conference on Computer Vision(ICCV). 2015:1440-1448.

[46] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9):1904-1916.

[47] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.

[48] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C]. European Conference on Computer Vision. 2016. 9905: 21-37.

[49] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. Computer Vision & Pattern Recognition. 2016:779-788.

[50] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]. IEEE Conference on Computer Vision & Pattern Recognition. 2017. 7263-7271.

[51] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[52] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]. IEEE Conference on Computer Vision & Pattern Recognition. 2017. 2117-2125.

[53] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 770-778.

[54] He K, Zhang X, Ren S, et al. Identity Mappings in Deep Residual Networks[C].

European Conference on Computer Vision, 2016. 630-645.

[55] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. The IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1-9.

[56] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. Proceedings of the 32nd International Conference on Machine Learning (PMLR), 2015, 37: 448-456.

[57] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. 2818-2826.

[58] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. arXiv preprint arXiv:1602.07261, 2016.

[59] Dai J, Li Y, He K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[C]. NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016. 379-387.

[60] 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.

[61] Sandler M, Howard A, Zhu M, et al. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation[J]. 2018: 66-74.

[62] Yang C, Zhang L, Lu H, et al. Saliency Detection via Graph-Based Manifold Ranking[C]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013. 3166-3173.

[63] N. Wojke, A. Bewley and D. Paulus. Simple online and realtime tracking with a deep association metric[C]. 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017. 3645-3649.