基于车载视频的交通标志检测与识别系统

摘要：随着人工智能的发展，无人驾驶成为了新一阶段研究的重要一步。由于在道路信息当中，能够提供最多信息的是交通标志，所以无人驾驶的关键环节之一便是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。优秀的驾驶辅助系统可减少事故的发生概率，也能加快无人驾驶的发展。本文将从检测精度以及实时性方面进行研究，主要内容如下：

（1）提出了一种提高目标检测精度的算法。主要是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，检测到图像中的交通标志位置。由于考虑到系统的实时性，本文采用了实时性更强的单级式目标检测算法，但是其检测的精度并不理想。因此，在得到检测结果的基础上添加视觉显著性算法，使预测框得到再次修正。实验表明，结合后的算法能够有效的提高目标检测框的精度，并且提高了后续分类的精度。

（2）为了进一步提高整个系统的实时性以及准确性，提出了结合卡尔曼滤波的跟踪算法。考虑到视频是由多帧组成，并且视频的多帧的信息是冗余的。一方面，目标检测算法会出现丢帧的情况，每次丢帧就要对目标进行重新识别是很耗费资源的；另一方面，持续的进行检测以及识别不仅对实时性方面带来损失而且会提高识别的错误率。因此，算法首先对固定帧长（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当前交通标志的正确类别。当确定类别之后，通过结合卡尔曼滤波的跟踪算法，不再对后续检测到的帧进行分类识别，只是对已经得到的结果进行持续跟踪，这样能够解决丢帧问题以及降低其误检率。

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：TP181

**Traffic Sign Detection and Recognition System based on Vehicle Video**

**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 and optimization to make the recognition rate higher and more real-time is the focus of this paper. Excellent driver assistance systems reduce the incidence of accidents and accelerate the development of self-driving. This paper will study the detection accuracy and real-time, 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 a more real-time single-level 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 predicted position again. Experiments show that the combined algorithm can effectively improve the positional accuracy of the target detection and improve the accuracy of subsequent classification.

（2）In order to further improve the real-time and accuracy of the whole system, a tracking algorithm combined with Kalman filtering is proposed. The video is composed of multiple frames, and the information of the multiple frames of the video is redundant. 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 detection recognition result. After the category is determined, the subsequently detected frames are no longer identified, but the results are tracked by a combined Kalman filter tracking algorithm. This can effectively solve the frame loss problem and reduce its false detection rate.

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

**Classification：**TP181

目 次

摘要……....…......…..………………………………………………………………. Ⅰ

目次…….........….………..……………………………………………….……….. Ⅳ

图和附表清单……...……………………………………………………….………Ⅵ

[**1** 绪论 1](#_Toc4350303)

[1.1 研究背景和意义 1](#_Toc4350304)

[1.2 国内外研究现状 2](#_Toc4350305)

[1.3 交通标志数据 4](#_Toc4350306)

[1.4 本文研究内容 5](#_Toc4350307)

[1.5 各章内容简介 6](#_Toc4350308)

[2 理论基础 7](#_Toc4350309)

[2.1 卷积神经网络 7](#_Toc4350310)

[2.2 基于深度学习的目标检测算法 9](#_Toc4350311)

[2.2.1 基于候选区域的目标检测 10](#_Toc4350312)

[2.2.2 单级式目标检测 14](#_Toc4350313)

[2.3 本章小结 16](#_Toc4350314)

[3 基于深度学习的交通标志检测研究 17](#_Toc4350315)

[3.1 TT100K数据集介绍 17](#_Toc4350316)

[3.2 目标检测算法的对比 18](#_Toc4350317)

[3.2.1 评价指标 18](#_Toc4350318)

[3.2.2 网络结构 20](#_Toc4350319)

[3.2.3 算法对比 21](#_Toc4350320)

[3.3 实验结果分析 24](#_Toc4350321)

[3.4.1 结果分析 24](#_Toc4350322)

[3.4.2 存在问题及解决 26](#_Toc4350323)

[3.4 结合视觉显著性的改进 26](#_Toc4350324)

[3.4.1 Grabcut算法 27](#_Toc4350325)

[3.4.2 视觉显著性算法 27](#_Toc4350326)

[3.4.3 实验结果及分析 28](#_Toc4350327)

[3.5 本章小结 32](#_Toc4350328)

[4 基于视频的交通标志检测与识别 33](#_Toc4350329)

[4.1 相关工作 33](#_Toc4350330)

[4.1.1 准备工作 33](#_Toc4350331)

[4.1.2 算法问题结果及分析 33](#_Toc4350332)

[4.2 结合跟踪的交通标志识别算法 34](#_Toc4350333)

[4.2.1 Deep SORT算法 34](#_Toc4350334)

[4.2.2 基于跟踪的识别系统 34](#_Toc4350335)

[4.3 实验结果及分析 34](#_Toc4350336)

[4.3.1 在时间性能上的对比 34](#_Toc4350337)

[4.3.2 在检测效果上的对比 34](#_Toc4350338)

[4.3.3 在识别效果的对比 34](#_Toc4350339)

[4.4 本章小结 34](#_Toc4350340)

[5 总结与展望 36](#_Toc4350341)

[5.1 研究总结 36](#_Toc4350342)

[参考文献 37](#_Toc4350343)

1. 绪论
2. 研究背景和意义

随着我国近年来的蓬勃发展，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比不可少的一部分。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。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。ADAS(Advanced Driver Assistance System)高级驾驶辅助系统成为了诸如美国、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，在众多系统同意完善后，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，使得大数据成为了可能。二者相得益彰，必然会使得科研课题如雨后春笋般蓬勃发展。而近几年来，发展最为火热的就是无人驾驶。随着2018年底5G技术的提出，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。通过计算机去分析当前的驾驶路况，在相当多的情况下，机器犯错的概率是小于人类的，通过计算机的辅助完成驾驶员的驾驶任务，从而做出当前最优的判断，从而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、目标跟踪、以及行为分析等成为了领域研究的热点，并且均取得了极好的结果。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。交通标志可以有效提供限速信息、禁令信息、指示信息等，如果可以准确的检测并识别到交通标志的内容，那么对于降低车祸的发生率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，并且完善的交通标志识别系统有很好的应用价值。

1. 国内外研究现状

国内外对交通标志的实时性检测与识别主要是围绕驾车辅助系统以及无人驾驶发展而展开的[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，主要受限于计算力以及数据，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，但需要改进的地方更多。近些年来，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。文献[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进行识别。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型构成，Garcia提出了使用霍夫变换对圆型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，从而实现对三角型标志的检测。

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。因此，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][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，这样会大大减少样本的维数，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，主要是提取其HOG[15]、SIFT、LBP等特征，从而再将特征结合机器学习如：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，[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]。[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，[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，机器自行去学习其中的信息，这样学习到的信息鲁棒性更强，并且只要有足够的数据，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（CNN）的发展，得益于此，目标检测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，并且取得了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。除此之外，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优化后进行交通标志的检测能取得更好的结果。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. 交通标志数据

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1.1所示。

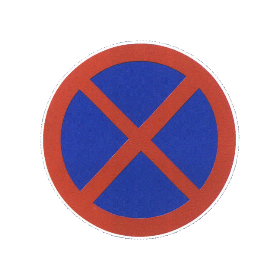
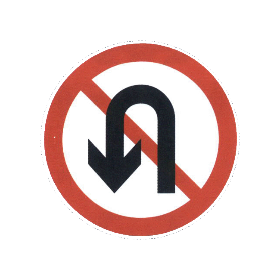


图1.1 部分交通标志图

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[31]数据集等等。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。

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。

1. 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。主要的创新如下：

（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检测精度高的目标检测模型。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，但是在检测精度上有一定的欠缺。为了弥补YOLO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区别的，对检测得到的目标框进行扩大之后，再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，从而对得到的检测框进行二次修正，获得更为完整的交通标志，从而提升后续的分类性能。

（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性能是有影响的。因此，本文提出通过结和跟踪算法来优化当前的检测系统，在前几帧确定了当前的目标种类之后，不再对后续的每一帧再此进行识别，而只是对当前已经识别到的目标进行跟踪，这样能有效的提高识别的性能，减少因重复识别而造成的资源浪费。

1. 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的理论基础，具体包括：卷积神经网络（CNN）、基于深度学习的目标检测算法。

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，对算法会存在检测框不准确的情况，提出结合视觉显著性检测的算法而对检测框进行二次修正的方法。通过实验表明，结合后可以进一步提升检测框的精度，从而提升后续的分类性能。

第四章：通过第三章得到的模型，对车载视频进行交通标志的检测与识别。首先，考虑到视频的重复信息过多，每一秒有25帧，但大部分都是冗余的信息，因此，对前几帧进行识别，当目标信息位置确定后便停止识别模块，而只对其进行跟踪，这样能够因为减少了识别模块的重复调用而提升整个系统的实时性。

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

1. 理论基础

在深度学习条件下计算机视觉的发展，除了得益于计算力以及大数据的魅力之外，其特殊的网络结构才是真正使其可以进一步模仿人类观察世界的钥匙。通过对猫的多次实验，生物学家发现了“感受野”的概念，从而得知了人类在观察事物的时候，大脑中枢其实只有部分的神经元在兴奋，其余的则是处于抑制状态。计算机科学家从这种特性得到了灵感，提出了卷积神经网络的思想。卷积神经网络最重要的思想便是模仿“感受野”从而实现权值共享，大大的减少了权值参数。

除此之外，深度学习在计算机视觉能成功应用的另一特点便是其多层网络的结构。这种结构恰恰模拟了人眼在观察到物体时候的大脑的迭代处理过程。具体来说，人眼在看到一个东西的时候会首先摄入图像的像素，接着对其进行初步处理来发现图像的边缘和方向，之后进行抽象判断出当前物体的形状，最后再次抽象判断出物体的类别。深度学习的每一层就相当于去抽象相关的特征，通过多层的叠加，抽象出最高的层次的特征，从而达到识别分类的目的。

1. 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（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。由于其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，一个卷积神经网络大概由以下几部分构成，卷积层（conv）、池化层（pooling）以及全连接层（FC）。

（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：

(2-1)

其中与为两个离散信号，表示卷积操作，为卷积结果。其卷积可以理解在满足的条件下，所有的总和。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。

(2-2)

其中矩阵的行数和列数分别为和。矩阵的行数和列数分别为和。为在当前坐标下的卷积结果。

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图像大小的计算公式如下所示：

(2-3)

其中，为卷积之后得到的尺寸，为原始图片的长度，*m*为原始图片的宽度，为扩充大小，如果卷积核的范围超过了原始图像的大小，则需要设置的大小以满足正常的卷积，一般填充的数值为0。为卷积核的大小，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，由于这个区域的存在，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。感受野其实是一个像素对应回原图得到的大小。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，池化层还降低优化难度和参数。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。

（3）全连接层

顾名思义，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损失函数，从而实现多分类，达到分类的目的。全连接层实际上是十分占用存存储空间的，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，从而缓解全连接层的参数量。

1. 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的寻找的物体。

随着人工智能和深度学习的发展，计算机视觉成为了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。将得到的传统特征送入到诸如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；另一类则是单次目标检测，主要代表作为YOLO、SSD系列，现对其发展进行简单的介绍。

1. 基于候选区域的目标检测

基于候选区域的算法就是，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。

基于候选目标区域的检测算法，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ROI区域，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的检测。

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的平方损失变为损失，计算公式如下所示：

(2-4)

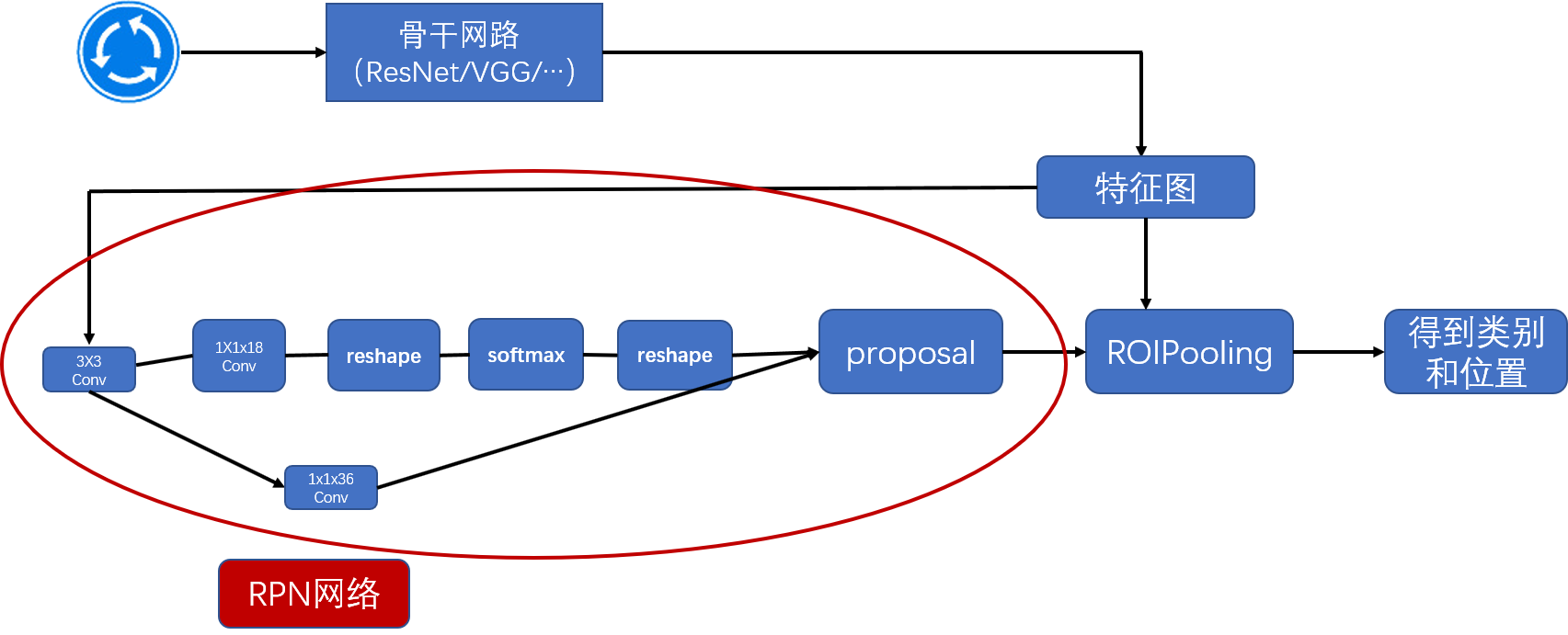
通过上式可知，当其损失为二次损失，但是当时，其损失则为线性损失。这样做可以防止因为预测值与目标值偏差很大时，L2损失导致的梯度爆炸。

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RPN(Region Proposal Networks)网路，其大大的加速了产生候选区域的速度。Faster R-CNN的结构如图2-1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2.1 Faster R-CNN网络结构图

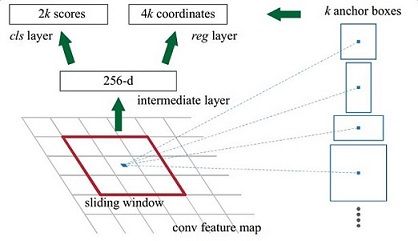
（2）为了模拟多尺度、多长宽比（一般的长宽比为）的情况，RPN提出使用Anchor的思想，即在遍历特征图像的每一个点时，使用不同的anchor对其进行特征提取，这样就相当于在原图上做了不同尺度的检测。具体如图2.2所示：

图2.2 anchor机制原理图

（3）通过观察图2.1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的宽和高共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）传送给后续的网络结构。

由于整个RPN网络是即考虑该位置是前景和背景，又要考虑到如若为前景，其坐标的偏移量是多少。因此，整个RPN网络的Loss公式如下：

（2-5）

其中，公式中表示anchor的索引，分别表示预测的概率以及真实的概率，其中只取概率的部分，其余的均不参与训练，这样可以更好的训练难训练的部分。则分别表示预测得到的坐标和真实的坐标。分别表示种类的个数以及边框的个数。相当于一个调整因子，避免相差过大。

1. 单级式目标检测

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，但其缺点也比较明显，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][50][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,之后判断物体的中心点落在哪个方格当中。如果中心点落在该网格内，这个方格就负责这个物体的检测。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个通道的信息，使得最终包含该位置的网络单元格被选中，之后再通过得到的坐标信息确定最终的目标位置。相应的，如果一张图片中有多个物体，那么将会有多个网络格被选中。

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；

2、该目标属于哪一个类别；

3、边界框的点的坐标（如：左上角坐标）以及宽和高四个标定量可以描述为。

综上所述，对每一个网络格的卷积通道数应该是,其中5表示类别的可能性和边界框的描述量，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,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抑制（Non-Maximum Suppression）。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，便可以解决出现多个预测框的情况。

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上经过与训练的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网络的设计思想也是通过多层小尺度卷积核从而实现参数优化的作用。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（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。

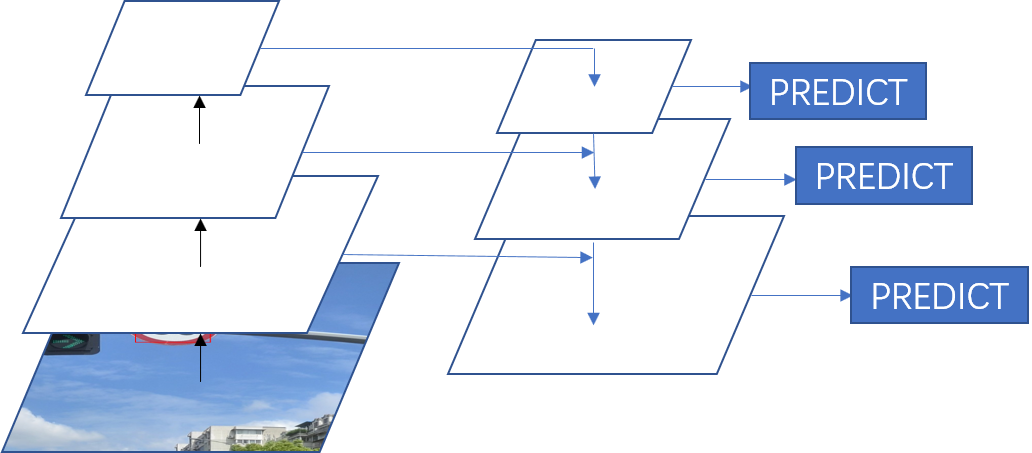
构建特征金字塔主要分为两条路径，其中一条路径是由下到上的路径，使用卷积操作对原始图片进行提取特征，层数越高其包含的语义特征越高而其分辨率则越低；另一条路径则是由上到下的路径，从上向下重新构建高分辨率层，通过融合当前层和经过上采样的顶层信息，并且通过3x3conv消除可能会存在的混叠效应。通过上述方法，从而达到对小目标的有效检测，具体的FPN结构如图2.4所示。

图2.4 金字塔网络图（FPN）

1. 本章小结

本章主要介绍了本论文后续中用到的算法的理论基础，对深度学习的核心卷积神经网络的结构进行了介绍。除此之外，本文主要是依赖目标检测，所以深入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，另一类则是单级目标检测算法。

1. 基于深度学习的交通标志检测研究

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结果进行优化。除此之外，还分析对比了不同模型在后续分类性能上的优劣。

1. TT100K数据集介绍

Tsinghua-Tencent 100K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），总图片的数量有GTSDB的111倍，并且每张图片的分辨率是GTSDB的32倍。图片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，不同的天气条件和某些遮挡情况下的交通标志。数据集中共包含221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。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。具体的数据集如图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

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 |

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。

图3.1 部分数据集图片

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。

1. 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不同算法在不同评价指标上的对比从而找到性能最优的检测算法。

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。

（3-1）

其中，为检测算法正确识别为交通标志的个数，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）

精确率表示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：

(3-2)

其中表示检测算法检测为真正为交通标志的个数，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），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交集与预测框与真实框的并集之比，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

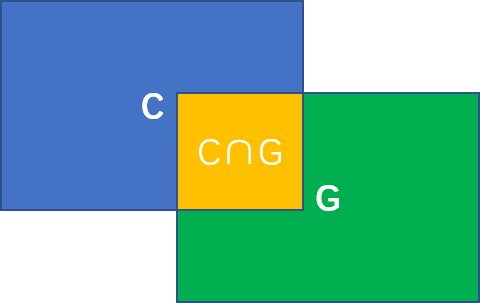
 (3-3)

图3.2 IoU示意图

1. 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。

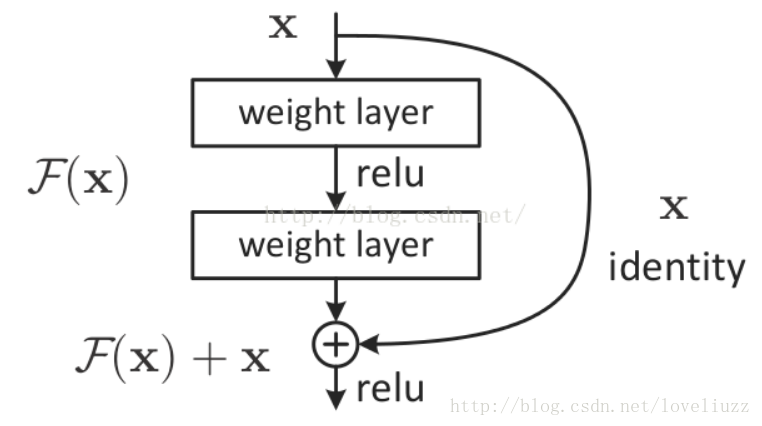
ResNet[53][54]系列模型，其主要思想是加深神经网络的层数学习到更多语义信息更加丰富的特征。由于不断地加深网络模型的深度，会使得模型产生退化，ResNet系列提出了残差结构来解决该问题，具体的结构如图3.3所示。网络的输入为，期望的映射输出为 ，直接去拟合是很困难的，因此将网络模型设计成为，这样我们就可以将学习的过程转化成为一个残差的过程即。使得学习的过程变得容易。

图3.3 残差网络结构图

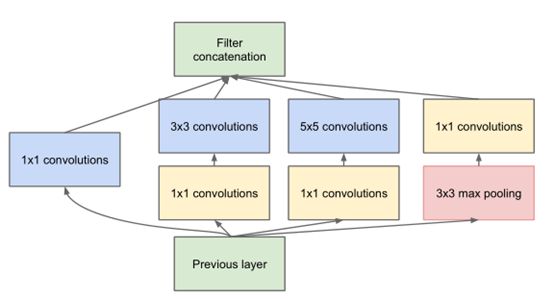
Inception[55][56][57][58]系列模型，该模型主要的思想是在所有模型都在研究如何更深提取更高等级特征的时候，Inception提出了卷积核的并行合并（Bottleneck Layer）。通过不同大小（1x1、3x3、5x5）的卷积核同时学习到当前特征图的稀疏特征和非稀疏特征，最后再通过将上述学习到的信息拼接起来，实现了对特征图的提取，具体的Bottleneck Layer结构如图3.4所示。其后的不同版本的提升也是对Bottleneck Layer进行不同的结构调整，但总体的思路还是相同的。

图3.4 Bottleneck Layer结构图

DarkNet系列模型，DarkNet模型的也是由多层的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BFLOP（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。

1. 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，从而判断其实时性能。

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。

表3.2 基于候选框检测算法速度

|  |  |  |
| --- | --- | --- |
| 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 |

单级式系列目标检测算法结合不同的骨干网络得到的检测结果如表3.3所示，实验分别对比了SSD、RFCN[59]结合Mobilenet V1[60]、Mobilenet V2[61]、Inception ResNet V2等模型的表现效果

表3.3 单级式检测算法速度

|  |  |  |
| --- | --- | --- |
| 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，从而判断该检测算法的性能。

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。

表3.4 基于候选区域检测算法在不同置信度下的检测率

|  |  |  |
| --- | --- | --- |
| 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 |

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3.5 单级式目标检测算法在不同阈值下的检测率

|  |  |  |
| --- | --- | --- |
| 检测算法与模型 | 置信度 | 召回率 / (%) |
| SSD | 0.5 | 78.9 |
| (MobileNet V1) | 0.6 | 74.4 |
|  | 0.7 | 69.7 |
|  | 0.8 | 63.3 |
|  | 0.9 | 53.5 |
| SSDLite | 0.5 | 83.8 |
| (MobileNet V2) | 0.6 | 80.4 |
|  | 0.7 | 75.3 |
|  | 0.8 | 68.2 |
|  | 0.9 | 56.5 |
| SSD | 0.5 | 83.8 |
| (MobileNet V2) | 0.6 | 80.9 |
|  | 0.7 | 77.4 |
|  | 0.8 | 72.3 |
|  | 0.9 | 63.8 |
| YOLO V3 | 0.5 | 98.4 |
| (DarkNet 53) | 0.6 | 98 |
|  | 0.7 | 97.4 |
|  | 0.8 | 96.5 |
|  | 0.9 | 95 |

1. 实验结果分析
2. 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有很高的实时性，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的速度基本上比Faster R-CNN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的有效检测率遥遥领先，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，检测效果如图3.5所示。为了直观的表现不同算法的性能，将之汇总于图3.5。

综上所述，YOLO V3算法无论在实时性以及有效检测率方面，都有着优越的表现，实验的总体对比结果如图3.6所示。

1. 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

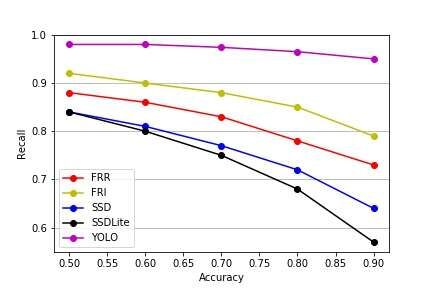
图3.5 不同算法的在不同置信度下的性能对比

图3.6 不同算法准确率与召回率的对比

1. 存在问题及解决

通过权衡检测率以及检测速度，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，它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。

图3.6 部分定位不准确的检测图

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，提高算法的检测精确性，具体的方法将在下一节描述。

1. 结合视觉显著性的改进

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，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

1. Grabcut算法

Grabcut[63]算法利用了图像中的纹理信息和边界信息的一种交互式图切割技术，一般来说Grabcut算法需要通过用户预先框定矩形区域，矩形区域为前景除此之外的区域为背景区域，之后再使用K个高斯分量的全协方差混合高斯模型（GMM）来对目标和背景区域进行建模。最后，通过迭代更新优化GMM的参数，使得图像分割更优。

通过熊昌镇[64]等人提出的Faster R-CNN结合Grabcut算法，为了解决检测不准确的情况，该算法先将检测区域扩大20%，目的是让扩大后的范围包含整块交通标志区域，之后将该区域代替Grabcut算法用户的交互式输入，迭代一次实现交通标志的分割。

1. 视觉显著性算法

显著性检测算法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，我们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、边缘等特征表示，判断它与周围的差异，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。

(3-6)

假设一张图为*G(V, E)*。公式（3-6）中***V***表示数据集***X***，***E***由隐射矩阵***W***组成，***D***表示图*G*的度矩阵，***W***表示权重矩阵，***y***表示指示向量,当前结点是种子结点*x*取1，否则取0。通过（3-6）计算每个结点的显著性从而得到排名分数。

主要的算法思想是通过基于图的流形排序对图像元素（像素或区域）与前景线索或背景先验的信息进行排序。图像元素的显著性是基于给定种子（queries）的相关性来定义的。通过流行排序的方法将此先验扩散到并增加得到前景的比较可靠的估计。这些节点基于背景和前景种子的相似性来排序（通过关联矩阵（affinity matrices））。随后将前景的信息用类似的流行排序的方法加强，显著性检测以两阶段方案进行，以有效地提取背景区域和前景显著对象。

本实验主要的过程是，在基于YOLO V3目标检测算法的基础上，对检测到的图标首先进行尺寸的判断，如果其尺寸大于了一定的比例，我们对得到的目标区域首先进行扩充，即对得到的长和宽扩充20%从而达到包含到完整目标检测区域的情况，在此基础上，由于交通识别的特殊性，通过Ranking Saliency算法对其进行显著性检测，从而确定交通标志更准确的位置。

1. 实验结果及分析

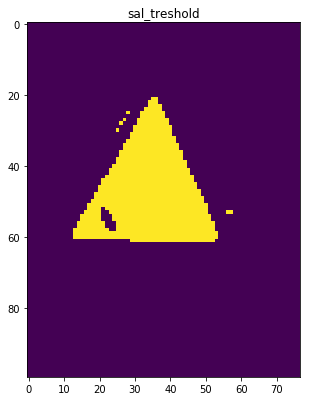
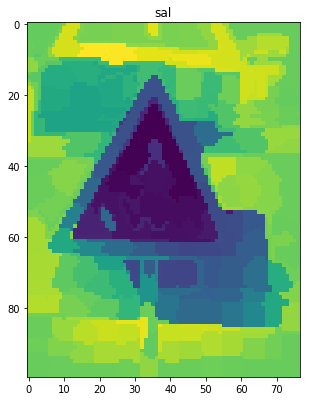
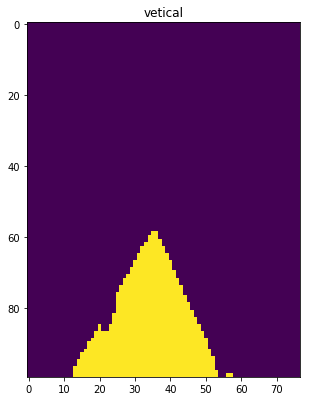
为了验证结合视觉显著性算法对目标检测算法的预测框的提升。我们分别使用Grabcut算法以及视觉显著性的算法对检测得到的交通标志进行实验。

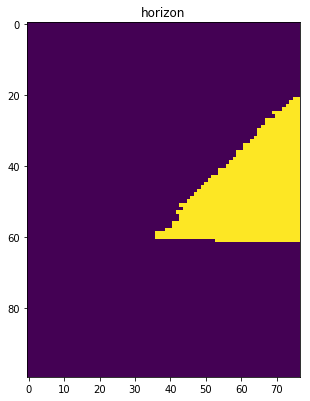
首先得到通过YOLO V3检测算法得到的交通标志，如图3.7（a）所示。之后，通过对其进行区域扩充20%得到的结构如图3.7（b）所示，对该区域进行Grabcut算法的操作，得到的结果如图3.7(c)所示，由图可以直观的发现，Grabcut算法的效果并不好。

（a） 检测算法得到图 （c）经过Grabcut算法图

（b）经过扩充后的图

图3.7 Grabcut算法总体流程图

通过结合视觉显著性的算法，其总体流程效果如图3.8所示，其前期的扩充如图3.7（b）所示。之后，对其进行显著性检测，其效果图如图3.8（a）所示，在得到了显著图后，对其进行阈值分割，得到如图3.8（b）所示，分别对其进行水平投影以及垂直投影，从而分割得到最终的交通标志图。

（a）显著性图 （b）阈值分割图 （c）垂直投影

（d）水平投影 （e）结果图

图3.8 结合视觉显著检测结果图

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的二次修正有一定的提升。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性后检测算法。可以发现结合显著性算法后，对平均IoU有所提升。具体结果如表3.5所示。

表3.5 不同算法的IoU表现

|  |  |
| --- | --- |
| 检测算法 | IoU / (%) |
| YOLO V3 | 79.57 |
| YOLO V3 + Ranking\_saliency | 82.27 |

分析上表可知，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3.6所示。

表3.7 不同算法对IoU（<0.7）的表现

|  |  |
| --- | --- |
| 检测算法 | IoU / (%) |
| YOLO V3 | 53.59 |
| YOLO V3 + Ranking\_saliency | 74.28 |

除此之外，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。在得到检测目标后，将之送入后续的分类网络，验证其分类性能。实验分别对比常见的模型结构，比如ResNet系列、MobileNet系列以及Inception系列。

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，得到最终的类别为33类。由于类别个数的不均衡，因此成比列的对不同的类别个数进行数据扩充。数据增强的方法主要有对图片进行加噪、设置不同的对比度以及设置不同的光照情况，通过数据增强后尽可能使每一类别的数目均衡。

进行完上述操作之后，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。

表3.8 不同网络模型的分类准确率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | 95 | 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 | 14.4 | 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，最终选用准确率更高的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，分别统计模型对不同类别的召回率以及精确率，并和[32]中所提到的算法进行比较，具体结果如表3.9所示。

表3.9 不同算法的识别效果对比

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **标志类别** |  | **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 | | |
| **标志类别** |  | **P11** |  | **P12** |  | **P23** |  | **P26** |  | **P3** |  | | **P5** |  | | **PG** |  | | **PH45** |  | | **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 | | |
| **标志类别** |  | **PL100** |  | **PL120** |  | **PL30** |  | **PL40** |  | **PL5** |  | | **PL50** |  | | **PL60** |  | | **PL70** |  | | **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 | | |
| **标志类别** |  | **PL80** |  | **PM20** |  | **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等交通标志的检测召回率方面有一定的提升，除此之外算法在交通标志检测的精确率方面也有不错的表现。

1. 本章小结

本章首先通过在实时性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。但是，由于其使用FPN的思想虽然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，对其得到的目标检测框进行二次优化。除此之外，本章通过对比不同的模型，将检测到的结果进行后续的分类识别，并从检测精确率、召回率以及模型大小方面，判断后续分类的各项指标性能，从而选取最合适本实验的模型。

1. 基于视频的交通标志检测与识别

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，视频信息其实是包含很多冗余信息的，许多视频帧都包含了重复的信息。而对于本文的交通标志检测识别系统来说，如此往复的进行检测和识别，必然会使得实时性下降，影响整个系统的性能。除此之外，目标检测算法会存在掉帧或者被短暂遮挡的情况，如何在这些情况下啊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系统进行进一步优化，使得其在实时性上能够进一步提升。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，我们通过对前10帧的检测识别结果进行投票，确定当前的交通标志到底属于哪一类别。当通过投票确定好类别之后，只是单纯使用跟踪算法对当前识别得到的目标进行跟踪，这样就可以优化后续因持续检测分类的所带来的性能损耗。并且跟踪算法采用级联的思想保存了部分帧的信息，从而能够很好的应对掉帧、遮挡等情况的发生。

1. 相关工作
2. 准备工作

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的当地的路况信息，需要对已有的数据集进行扩充。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，之后对之进行分帧处理。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区域的位置。标注软件会记录交通标志左上角以及右下角位置的坐标，具体如图4.1所示。

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的宽和高相对于图片大小的比例。具体的计算公式如下所示。

(4-1)

(4-2)

(4-3)

(4-4)

(4-5)

(4-6)

其中表示原始图像的宽度，表示原始图片的长度，和分别代表ROI左上角的X坐标和Y坐标，和则分别代表ROI右下角的X坐标和Y坐标。通过上述公式，得到最终训练所需的数据格式。

1. 算法问题结果及分析

通过之前章节的准备工作，我们可以设计出交通标志检测识别系统，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，由于系统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。对每一帧的处理并没有使用到视频连续多帧的信息冗余的特点。另外，对每一帧都连续的进行分类，不可避免会造成错误率的增加。因此，通过有效利用到视频的冗余信息来提升算法的性能。主要思想是使算法结合Deep SORT[65]跟踪算法，检测确定交通标志的种类之后，便只是持续检测而停止后续的分类操作，从而减少算法的时间损耗。具体流程将在下一节展开。

1. 结合跟踪的交通标志识别算法
2. Deep SORT算法

Deep SORT算法是在SORT目标跟踪基础算法上的改进，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DEEPSORT算法。Deep SORT已经有预训练好的模型权重，将检测得到的交通标志信息传递给Deep SORT模型，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。

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，可以使用来表示，其中表示当前目标的中心点的坐标，表示纵横比，表示图片的高度，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](https://www.baidu.com/s?wd=%E6%BB%A4%E6%B3%A2%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)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。对当前的目标框和通过卡尔曼滤波预测到的目标框计算其马氏距离，当小于固定阈值时，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。

(4-7)

其中，表示第个检测框的位置，表示第个追踪器对目标的预测位置，表示检测得到的位置与平均追踪位置的协方差矩阵。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。计算公式如（4-8）所示。如果发现其余弦距离小于某个距离，那么就证明当前检测到的特征向量可以被关联跟踪。

（4-8）

其中，表示当前检测的特征向量，对应表示特征向量“博物馆”中的不同特征向量。最后，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。通过调整的值来权衡二者的关系。对于本实验来说，由于相机的移动，所以采用余弦距离的匹配更为合适，即设置=0。

(4-9)

除此之外，由于部分遮挡问题的发生，遮挡的时间越长，会使得其由于长时间未更新位置信息，导致其协方差矩阵变大。而通过卡尔曼滤波器预测的框与当前检测得到的框计算马氏距离的时候，因为计算要使用协方差的倒数，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框（间隔时间短由于位置信息的不断更新，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。

1. 基于跟踪的识别系统

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进行交通标志检测后；其次，使用分类网络对检测结果进行分类；最后，根据视频的连续信息，对确定的分类结果进行跟踪。

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，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，这样便会一定程度上增加后续分类的难度，因为要尽可能多的考虑到不同的缩放尺度，这就对数据集的要求很高，要尽可能包含更多不同尺度的数据集。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，如果只是对其进行检测，当确定当前类别后，后续只是跟踪，那么在算法的时间性能山有一定的优化。

综上所述，使用结合跟踪的交通标志检测识别系统是有意义的，具体的算法流程图如图4.x所示。

1. 实验结果及分析

本实验的运行环境为Linux系统下的Ubuntu16.04，使用GTX 1060显卡，8G内存。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端视频。先将数据集中的交通标志统计如表4.1所示。

表4.1 测试集中不同交通标志的分布

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **标志类别** |  | **PL30** |  | **PN** |  | **I5** |  | **P11** |  | **P23** |  | **PL40** |  | **PL60** |  |
| **个数** |  | 2 |  | 17 |  | 2 |  | 3 |  | 2 |  | 3 |  | 1 |  |
| **标志类别** |  | **PM20** |  | **P26** |  | **P12** |  | **P10** |  | **W59** |  | **PL80** |  | **I2** |  |
| **个数** |  | 2 |  | 3 |  | 5 |  | 5 |  | 1 |  | 13 |  | 12 |  |

1. 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，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一秒中平均处理的帧数。具体的结果如下表所示。

表4.2 不同情况下的fps

|  |  |  |
| --- | --- | --- |
|  | 不结合跟踪算法（fps） | 结合跟踪算法(fps) |
| 含有交通标志 | 3.3 | 6.5 |
| 不含有交通标志 | 7.2 | 7.2 |

通过对比可以发现，在不含有交通标志的区域，由于检测算法并不能检测到交通标志，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，由于只是前期的10帧进行分类识别，后续并不需要持续的识别，每秒处理的帧数提升了3帧左右。因此，通过结合跟踪算法，确实能够在算法的性能上做出一定的改进。

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，测试集主要使用两段视频。一段是白天的行车视频，另一端是夜晚的行车视频。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。得到的结果如下表所示。

表4.3 算法检测结果

|  |  |  |
| --- | --- | --- |
|  | 精确率 | 召回率 |
| 白天 | 87.8 | 89.5 |
| 夜晚 | 56.2 | 47.6 |

通过观察上表可得知算法在白天达到了一定的实用效果。部分未检测到的图如图4.x所示。通过分析可知，算法的在白天的整体性能要优于夜间的表现。主要的原因有以下两点，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，比如说将打车的尾灯检测为交通标志或者是将红灯检测为标志，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，使得其不能有效的训练，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。

1. 在识别效果的对比

在识别方面，为了进一步加强识别的效果。只有当识别框大小超过32x32后才对其进行识别。并且对连续的10帧进行识别，识别过后，返回当前10帧占比最多的结果。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。分别通过召回率和精确率对之进行分析。具体的结果如表4.4所示。

表4.4

|  |  |  |
| --- | --- | --- |
|  | 精确率 | 召回率 |
| 白天 |  |  |
| 夜晚 |  |  |

通过分析可知，结合跟踪算法后，对识别的性能有所提升。对于行车辅助系统来说，正确的类别尤为重要，因为要通过该方法来辅助驾驶者做出正确的判断，如果识别准确率不够，做出错误的判断，后果是不可估计的。

1. 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，最终的应用应该是基于真实环境下的，并且是以视频流的形式传递的。如何能更好的适应真实环境，是更有意义的研究。

本章首先通过分析视频的信息，发现视频中包含大量的冗余信息，这样对检测和识别会带来一定的影响，为了更好的利用视频中的信息，使用检测识别算法在检测前期进行检测识别，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。

除此之外，由于连续的检测以及分类会对算法准确性产生一定的影响，增加很多的不确定型。因此，在视频的前面几帧识别投票确定种类之后，不再对后续进行分类识别，直接使用跟踪算法对其进行跟踪。从而提高检测识别的准确率。

最后，通过分别分析算法的时间性能、检测性能以及识别性能，来确定结合算法的有效性。

1. 总结与展望
2. 研究总结

随着人工智能的蓬勃发展，越来越多的领域可以使用

参考文献

1. Levinson J , Askeland J , Becker J , et al. Towards fully autonomous driving: Systems and algorithms[C]// Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011.
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. Kamada H , Naoi S , Gotoh T . A compact navigation system using image processing and fuzzy control[C]// Southeastcon 90, IEEE. IEEE, 2002.
4. Auturo D L E . Road Traffic Sign Detection and Classification[J]. IEEE Transaction on Industrial Electronics, 1997, 44(6):848-859.
5. Miura J , Kanda T , Shirai Y . An active vision system for real-time traffic sign recognition[C]// Intelligent Transportation Systems. IEEE, 2002.
6. Arnoul P , Viala M , Guerin J P , et al. Traffic signs localisation for highways inventory from a video camera on board a moving collection van[C]// Intelligent Vehicles Symposium, 1996. Proceedings of the 1996 IEEE. IEEE, 1996.
7. Maldonadobascon S , Lafuentearroyo S , Giljimenez P , et al. Road-sign detection and recognition based on support vector machines[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(2):264-278.
8. Liu H , Liu D , Xin J . Real-time recognition of road traffic sign in motion image based on genetic algorithm[C]// Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on. IEEE, 2002.
9. Hu M K . Visual Pattern Recognition by Moment Invariants[J]. Information Theory, IRE Transactions on, 1962, 8(2):179-187.
10. GarcíaGarrido, Miguel Ángel, Sotelo, Miguel Ángel, MartínGorostiza, Ernesto. 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.
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. Ruta A , Li Y , Liu X . Detection, Tracking and Recognition of Traffic Signs from Video Input[C]// Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on. IEEE, 2008.
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. Zaklouta F . Real-time traffic sign recognition using spatially weighted hog trees[C]// International Conference on Advanced Robotics. IEEE, 2011.
17. Takaki M, Fujiyoshi H. Traffic Sign Recognition Using SIFT Features[J]. EEJ Transactions on Electronics, Information and Systems, Volume 129, Issue 5, pp. 824-831 (2009)
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. Journal of Intelligent & Fuzzy Systems 32 (2017) 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. Ciresan D , Meier U , Masci J , et al. A committee of neural networks for traffic sign classification[J]. 2011.
22. Sermanet P , Lecun Y . [IEEE 2011 International Joint Conference on Neural Networks (IJCNN 2011 - San Jose) - San Jose, CA, USA (2011.07.31-2011.08.5)] The 2011 International Joint Conference on Neural Networks - Traffic sign recognition with multi-scale Convolutional Networks[C]// International Joint Conference on Neural Networks. IEEE, 2011:2809-2813.
23. Jin J , Fu K , Zhang C . Traffic Sign Recognition With Hinge Loss Trained Convolutional Neural Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5):1991-2000.
24. Qian R , Zhang B , Yue Y , et al. Robust Chinese Traffic Sign Detection and Recognition with Deep Convolutional Neural Network[C]// International Conference on Natural Computation. 0.
25. Qian R , Liu Q , Yue Y , 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). IEEE, 2016.
26. Zuo Z , Yu K , Zhou Q , et al. Traffic Signs Detection Based on Faster R-CNN[C]// 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW). IEEE Computer Society, 2017.
27. Zhang, J.; Huang, M.; Jin, X.; Li, X. A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2. 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]// Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011.
29. Timofte R , Zimmermann K , Gool L J V . Multi-view traffic sign detection, recognition, and 3D localisation[C]// IEEE Workshop on Applications of Computer Vision (WACV 2009), 7-8 December, 2009, Snowbird, UT, USA. IEEE, 2009.
30. Larsson F , Felsberg M . Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition[J]. 2011.
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.
32. Zhu Z , Liang D , Zhang S , et al. Traffic-Sign Detection and Classification in the Wild[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2016.
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. Zeiler M D, Fergus R. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks[J]. Eprint Arxiv, 2013.
35. Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 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. Lienhart R , Maydt J . An Extended Set of Haar-like Features for Rapid Object Detection[C]// Image Processing. 2002. Proceedings. 2002 International Conference on. IEEE, 2002.
39. Hearst M A. Support Vector Machines[J]. IEEE Intelligent Systems & Their Applications, 1998, 13(4):18-28.
40. Quinlan J R. Induction on decision tree[J]. Machine Learning, 1986, 1(1):81-106.
41. Breiman L. Random Forests[J]. Machine Learning, 2001.
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[C]//Advances in neural information processing systems. 2012: 1097-1105.
45. Girshick R. Fast R-CNN[C]// IEEE International Conference on Computer Vision. 2015.
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.
49. Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. 2016.
50. Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2017.
51. Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 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.
53. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015.
54. He K, Zhang X, Ren S, et al. Identity Mappings in Deep Residual Networks[J]. 2016.
55. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[J]. 2014.
56. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015.
57. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]// Computer Vision & Pattern Recognition. 2016.
58. Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. 2016.
59. Dai J , Li Y , He K , et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[J]. 2016.
60. Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.
61. Sandler M, Howard A, Zhu M, et al. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation[J]. 2018.
62. Yang C , Zhang L , Lu H , et al. [IEEE 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Portland, OR, USA (2013.06.23-2013.06.28)] 2013 IEEE Conference on Computer Vision and Pattern Recognition - Saliency Detection via Graph-Based Manifold Ranking[J]. 2013:3166-3173.
63. ROTHERC, KOLMOGOROV Y, BLAKE A. GrabCut: interactive foreground extraction using iterated graph cuts [J]. ACM Transactions on Graphics, 2004, 23(3): 309-314.
64. 熊昌镇, 王聪. 图切技术和卷积网络的交通标志数据集构建方法[J]. 计算机应用, 2017(z1):183-186.
65. N. Wojke, A. Bewley, D. Paulus, "Simple online and realtime tracking with a deep association metric" in ICIP, pp. 3645-3649, 2017.