



南京大學

研究生畢業論文

(申請工程碩士學位)

論文題目 基于功能安全的自動駕駛圖像數據擴增技術

作者姓名 徐彬桐

學科、專業名稱 工程碩士(軟件工程領域)

研究方向 軟件工程

指導教師 陳振宇 教授, 馮洋 助理研究員

2021年5月20日

学 号 : MF1932204
论文答辩日期 : 2021 年 5 月 20 日
指 导 教 师 : (签 字)



Automatic Driving Image Data Augmentation Technology Based on Functional Safety

By

Bintong Xu

Supervised by

Professor Zhenyu Chen, Assistant Researcher Yang Feng

A Thesis

Submitted to the Software Institute

and the Graduate School

of Nanjing University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Engineering

Software Institute

2021 May

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于功能安全的自动驾驶图像数据扩增技术

工程硕士（软件工程领域）专业 2019 级硕士生姓名：徐彬桐

指导教师（姓名、职称）：陈振宇 教授，冯洋 助理研究员

摘 要

随着人工智能技术和传感器技术的发展，自动驾驶技术得到了广泛的应用。自动驾驶汽车可以通过各种传感器的协同配合，实现对周围环境的全方位感知。在众多的传感器中，图像传感器作为应用最多，范围最广的传感器，在探测周围环境的任务中起到重要作用。然而，作为汽车电子电器系统的图像数据采集系统组成部分和所处的环境非常复杂。在实际工作过程中，会因为各种异常的场景或设备本身故障等问题导致采集的图像数据发生异常。这些异常的数据可能会造成严重的交通事故。为了保证自动驾驶汽车的功能安全，国际标准化组织颁布了 ISO26262 标准，为汽车电子电器产品的开发、设计和测试过程制订出一套可行的指导规范。但是，出于安全考虑处在开发阶段或发生电子电器系统故障情况下的自动驾驶汽车无法在现实道路上进行路况图像数据采集的工作。并且在自动驾驶汽车的安全性测试中需要使用这些异常图像数据。于是，这就产生了自动驾驶汽车异常图像数据如何获取的问题。目前对自动驾驶图像数据扩增领域研究只局限于驾驶场景变换的阶段，图像数据扩增方法并未涵盖自动驾驶汽车图像数据采集通路可能发生故障的各个功能点。因此，本文将汽车工业开发测试领域的功能安全分析技术与深度学习测试领域的图像扩增技术相结合，使用图像处理的手段来模拟自动驾驶汽车图像数据采集通路中各种异常情况发生时的图像数据。为了更为完整全面的发现自动驾驶汽车图像数据采集系统内存在的问题，同时总结出因为这些问题导致的异常图像数据类型实现数据扩增，本文基于 ISO26262 标准中功能安全的分析方法，针对自动驾驶汽车图像数据采集系统发生异常时自动驾驶汽车无法获取图像数据集的问题，提出了基于功能安全的自动驾驶图像数据扩增技术。

本文研究工作和成果主要包括以下几个方面：

1. 对自动驾驶汽车图像数据采集系统进行功能安全分析。本文采用 ISO26262 中推荐的失效模式与影响分析方法 (FMEA)，将自动驾驶汽车的图像数据采集系统的整体输入流程进行切片分析，并对图像采集过程中各个部分中可能产生的故障和异常失效形式进行分类汇总，总结出图像采集系统中最容易发生故障的异常类型。

2. 总结异常数据扩增规律并通过代码进行模拟实现。本文对图像数据流通路中最容易发生故障的异常类型进行分析，分析产生异常图像数据的原因。本文研究了场景异常的规律和图像数据采集系统中硬件或软件模块发生问题的机理以及这些故障发生对图像数据产生的影响和变化。根据这些原因和机理，本文总结归纳出这些异常发生时数据扩增的规律，并使用 Python 语言和 OpenCV 库进行扩增数据的实现。
3. 训练深度学习模型并对扩增数据进行验证分析。本文对 KITTI 数据集进行收集和修改，并使用修改后的 KITTI 数据集训练深度神经网络。随机选取 KITTI 数据集的测试集数据进行数据扩增，将原图像和经过扩增处理后的图像一同输入训练好的神经网络中，观察深度学习模型的识别结果，分析总结扩增图像对深度学习模型带来的影响。

对扩增数据进行测试后本文发现，使用基于功能安全的自动驾驶图像数据扩增技术生成的图像数据会对深度学习系统的判断产生不同程度的影响。这些图像数据可以导致系统对目标检测的误判和漏判以及目标检测置信度的下降。利用该技术生成的图像可以检测到自动驾驶汽车异常情况发生时深度学习系统中的不足与漏洞，从而反映出系统安全性和鲁棒性问题。

关键词：数据扩增，自动驾驶，功能安全，图像处理，深度学习

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Automatic Driving Image Data Augmentation Technology Based on Functional Safety

SPECIALIZATION: Software Engineering

POSTGRADUATE: Bintong Xu

MENTOR: Professor Zhenyu Chen, Assistant Researcher Yang Feng

Abstract

With the development of artificial intelligence technology and sensor technology, autonomous driving technology has been widely used. Self-driving cars can use a variety of sensors to achieve a full range of awareness of the environment. Among many sensors, image sensor, as the most widely used sensor, plays an important role in the task of detecting the surrounding environment. However, the image data acquisition system as an automotive electronic and electrical system, its components and the environment is very complex. In the actual working process, abnormal image data will occur due to various abnormal scenes or equipment failures. These abnormal data can cause serious traffic accidents. In order to ensure the functional safety of autonomous driving vehicles, the International Organization for Standardization (ISO) issued the ISO26262 standard, which establishes a set of feasible guidelines for the development, design and testing process of automotive electronic and electrical products. However, due to safety concerns, self-driving vehicles in the development stage or in the case of failure of electronic and electrical systems cannot collect road condition image data on real roads. And these abnormal image data need to be used in the safety test of autonomous driving vehicles. Thus, the problem of how to obtain abnormal image data for autonomous vehicles arises. At present, the research in the field of autonomous driving image data augmentation is only limited to the stage of driving scene transformation, and the image data augmentation method does not cover each function point where the image data acquisition pathway of autonomous driving vehicles may fail. Therefore, this paper combines the functional safety analysis technology in the automotive industry development and testing field with the image augmentation technology in the deep learning testing field, and uses the means of image processing to simulate the image

data when various abnormal situations occur in the image data acquisition path of the autonomous driving vehicle. In order to find the problems existing in the image data acquisition system of autonomous driving vehicles in a more complete and comprehensive way and summarize the abnormal image data types caused by these problems to achieve data augmentation, this paper is based on the functional safety analysis method in ISO26262 standard. Aiming at the problem that the automatic driving vehicle can't acquire the image data set when the image data acquisition system of the autonomous driving vehicle is abnormal, an autonomous driving image data augmentation technology based on functional safety is proposed.

The research work and achievements of this paper mainly include the following aspects:

- The function safety analysis of the image data acquisition system of autonomous driving vehicle is carried out. In this paper, the failure mode and impact analysis method (FMEA) recommended in ISO26262 is adopted to slice and analyze the overall input process of the image data acquisition system of autonomous driving vehicles, and the possible failures and abnormal failure forms in each part of the image acquisition process are classified and summarized. Summarizes the abnormal types that are most likely to fail in the image acquisition system.
- Summarize the law of abnormal data augmentation and carry out simulation through the code. In this paper, the abnormal types that are most likely to fail in the image data flow path are analyzed, and the causes of abnormal image data are analyzed. This paper studies the law of the scene anomaly and the mechanism of the hardware or software module problems in the image data acquisition system as well as the influence and change of the image data caused by these failures. According to these reasons and mechanisms, this paper summarizes the rules of data augmentation when these anomalies occur, and uses Python language and OpenCV library to implement the augmentation data.
- The deep learning model was trained and the augmented data was validated and analyzed. In this paper, we collect and modify the KITTI data set, and use the modified KITTI data set to train the deep neural network. In this paper, the test set data of KITTI data set were randomly selected for data augmentation, and the original image and the augmented image were input into the trained neural network to observe the recognition results of the deep learning model, and the

impact of the augmented image on the deep learning model was analyzed and summarized.

After testing the augmented data, this paper finds that the image data generated by the functional-safety-oriented autonomous driving image data augmentation technology will have varying degrees of impact on the judgment of the deep learning system. These image data can lead to the misjudgment and omission of target detection and the decrease of target detection confidence. The images generated by this technology can detect the deficiencies and loopholes in the deep learning system when abnormal situations of autonomous vehicles occur, so as to reflect the security and robustness of the system.

Keywords: Data Augmentation, Automatic Driving, Functional Safety, Image Processing, Deep Learning

目录

表 目 录	xi
图 目 录	xiv
第一章 绪论	1
1.1 研究背景及意义	1
1.2 图像数据扩增国内外研究现状	3
1.3 论文研究内容及组织结构	5
1.4 本章小结	6
第二章 相关理论综述	7
2.1 功能安全的概念与标准	7
2.1.1 功能安全的概念	8
2.1.2 功能安全标准 ISO26262	9
2.2 数字图像处理技术	9
2.2.1 图像的基本表示方法	11
2.2.2 图像的几何变换	11
2.2.3 图像的数值处理	14
2.2.4 图像的色彩空间类型	17
2.3 摄像头相关技术	18
2.3.1 摄像头工作原理	19
2.3.2 图像信号处理器相关技术	21
2.4 卷积神经网络	24
2.4.1 卷积层	25
2.4.2 池化层	26
2.4.3 激活函数	27
2.4.4 全连接层	27
2.5 本章小结	28

第三章	基于功能安全的数据扩增方法	29
3.1	图像采集通路硬件架构及错误分析	29
3.1.1	图像输入流程切片分析	29
3.1.2	图像输入异常分类	30
3.2	图像采集通路功能安全分析	32
3.2.1	异常图像 FMEA 研究	32
3.2.2	摄像头 FMEA 分析结果	47
3.3	异常（故障）数据扩增方法	47
3.3.1	光线场景异常	47
3.3.1.1	雾场景异常	48
3.3.1.2	雨场景异常	49
3.3.2	摄像头异常	50
3.3.2.1	镜头异常	50
3.3.2.2	对焦马达异常	51
3.3.2.3	传感器异常	51
3.3.2.4	ISP 校正异常	52
3.3.3	板级模块异常	55
3.3.3.1	主控芯片时序异常	55
3.3.3.2	图片缓存异常	55
3.3.3.3	线束传输异常	56
3.4	数据扩增的实现	56
3.4.1	光线场景异常数据扩增	57
3.4.2	摄像头异常数据扩增	58
3.4.3	板级模块异常数据扩增	64
3.5	本章小结	66
第四章	扩增数据的准备与检验	67
4.1	数据准备与处理	67
4.1.1	KITTI 数据集简介	67
4.1.2	数据集标签类别转换	68
4.1.3	数据集 VOC 格式及转换	69

4.2	扩增数据有效性检验	71
4.2.1	YOLO V3 环境搭建及配置文件调试	71
4.2.2	扩增数据生成及有效性检验	71
4.2.2.1	研究问题	72
4.2.2.2	实验对象	72
4.2.2.3	实验设计	73
4.2.2.4	评价指标	74
4.3	实验结果与分析	75
4.4	本章小结	80
第五章	总结与展望	83
5.1	本文研究内容总结	83
5.2	本文工作内容局限与展望	84
	参考文献	85
	简历与科研成果	89
	致谢	91

表 目 录

3.1	摄像头 FMEA 分析结果表	48
4.1	标注文件含义示意表	68
4.2	系统软硬件环境表	71
4.3	扩增前后图像目标检测结果表	76

插图

1.1	自动驾驶的核心技术与相关设备示意图	1
1.2	DeepTest 处理前后对比图	3
1.3	DeepRoad 合成的驾驶场景图像	4
2.1	失效模式与后果分析方法 (FMEA) 示意图	8
2.2	功能安全标准 ISO26262 架构示意图	10
2.3	平移前后对比图	12
2.4	缩放前后对比图	13
2.5	旋转前后对比图	14
2.6	卷积运算过程示意图	16
2.7	OV3640 核心功能框架图	18
2.8	图像传感器光电转换原理示意图	19
2.9	Bayer 滤光片工作原理示意图	20
2.10	Bayer 格式图像示意图	21
2.11	感光阵列读出电路示意图	22
2.12	卷积神经网络基本结构图	25
2.13	卷积操作示意图	25
2.14	平均池化操作示意图	26
2.15	最大池化操作示意图	26
2.16	常见的激活函数图像	27
2.17	全连接层网络结构示意图	28
3.1	典型自动驾驶视频流通路示意图	29
3.2	摄像头组成示意图	30
3.3	大气散射模型示意图	33
3.4	入射光衰减模型示意图	33
3.5	环境光模型示意图	34
3.6	挡风玻璃上雨滴形状示意图	37

3.7	相机相对位置示意图	38
3.8	凸透镜成像示意图	39
3.9	弥散圆示意图	39
3.10	激光损坏 COMS 后的图像	40
3.11	ISP 工作流程及数据通路示意图	41
3.12	Raw Image 和经过 ISP 差值后的图像	43
3.13	时钟时序关系图	45
3.14	图像数据在内存中的组织形式示意图	46
3.15	OV3640 输出格式示意图	47
4.1	KITTI 数据集 Lable 文件格式示意图	67
4.2	各种扩增方法生成的图像数据	72
4.3	YOLO V3 识别效果图	73
4.4	扩增数据识别结果示意图	74
4.5	扩增前后目标识别置信度变化示意图	76
4.6	扩增前后目标识别判断错误示意图	77
4.7	扩增前后目标识别漏判断示意图	77
4.8	扩增算法导致系统漏判统计图	78
4.9	扩增算法导致系统误判统计图	79
4.10	扩增数据前后目标识别数目比统计图	80
4.11	扩增数据前后目标识别置信度总和比统计图	81

第一章 绪论

1.1 研究背景及意义

我国汽车销量水平已经连续多年保持全球市场第一，截止到 2020 年 4 月，我国汽车市场的保有量已达到 2.6 亿辆 [1]。与此同时，近年来随着人工智能技术和信息通信技术的快速发展，传统汽车正在加速向智能化和网联化转型，自动驾驶汽车已经走入人们的日常生活。

目前，几乎所有的汽车制造公司公司和互联网科技企业都成立了专门的智能车研究部门并纷纷研发了自己的自动驾驶系统。他们所开发的自动驾驶系统几乎都是基于传感器探测，目标检测等深度学习等技术 [2]。在基于深度神经网络的自动驾驶系统中，汽车将各种传感器接采集的数据作为输入，经过神经网络的计算和控制系统的处理后输出各种控制信号从而驱使控制单元做出相关的驾驶行为。自动驾驶的核心技术与相关设备如图 1.1 所示。

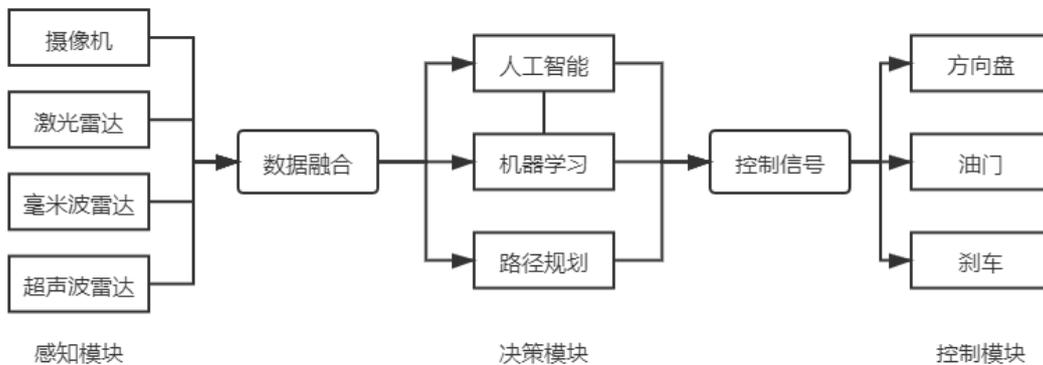


图 1.1: 自动驾驶的核心技术与相关设备示意图

如图 1.1 所示，其中环境感知模块包括各种用于环境探测感知的传感器，它为决策模块提供了自动驾驶汽车所处驾驶环境的数据。为了提高感知的准确性和可靠性，自动驾驶汽车会使用多传感器数据融合技术 [3]，将不同类型的传感器信息相互融合，提高环境信息数据的准确程度。决策模块依据感知模块获取的各种数据利用人工智能技术进行目标识别和行路径规划，同时将这些规划决策信息转换成相应的控制信号向执行模块发送。在执行模块中，控制信号被执行单元转化为成转向、刹车或加速等汽车行为，从而控制自动驾驶汽车的行驶 [4]。

可见，自动驾驶汽车驾驶行为的表现完全取决于其所采集数据的质量，质量良好的数据经过自动驾驶汽车深度神经网络的计算会得出更加准确安全的驾驶行为，反之自动驾驶汽车可能无法根据质量较差的异常数据做出正确的判断，从而导致交通事故的发生。因此，作为自动驾驶汽车收集和处理数据最重要部分的电子电器系统，在自动驾驶汽车的行驶安全中就显得举足轻重。在异常驾驶场景或电子电器系统发生故障的情况下，自动驾驶汽车根据异常采集数据做出的判断直接反应了自动驾驶系统的安全性和鲁棒性。

在自动驾驶汽车各种各样的传感器中，图像传感器作为自动驾驶汽车数据采集应用最多，应用范围最广的传感器，其工作稳定性和异常状态下采集数据的质量成为了自动驾驶汽车需要研究的重点 [5]。

智能化汽车相对于传统汽车而言，其电子电器化系统的复杂度越来越高。例如，感知系统和决策系统的数据会先转换成相关信号，然后这些信号传递给控制器的处理芯片，最终驱动相关的执行器来执行。又或者发动机控制器 ECU，在它其中包含了大量复杂的电路和电子器件。由于，智能化汽车的电器化的程度先比传统汽车高，所以汽车的安全性在很大的程度上取决于电子器件工作的稳定性。正是因为电子系统的复杂性，在自动驾驶汽车的数据流通的通路中，各部分组件和模块失效的可预见性变得非常低 [6]。导致电子控制器失效的原因非常多，比如外界干扰、电子器件发生故障、工作模块发生异常等。因此，人们无法估计自动驾驶汽车的电子电器系统将在什么时间发生什么样的状况，这些状况的发生会导致哪些问题的产生。所以，自动驾驶汽车必须考虑当这些异常发生时汽车应该如何工作的问题。然而，出于安全考虑，处在开发阶段或发生电子电器系统故障情况下的自动驾驶汽车无法在现实道路上进行路况图像数据采集的测试工作 [7]。这就产生了摄像头在异常工作情况下的图像数据获取困难的问题，尤其是那些在电子器件发生随机故障的异常情况下采集的这类数据更是难以获得。而自动驾驶汽车需要使用这异常情况发生时采集的数据对汽车的表现情况进行测试，才可以验证自动驾驶系统的安全性和鲁棒性。因此，这些异常情况下所采集的数据是研究自动驾驶汽车可靠工作的重点研究对象 [8]。目前，对自动驾驶图像数据扩增领域研究只局限于驾驶场景变换的阶段，而且图像数据扩增方法并未涵盖自动驾驶汽车图像数据采集通路可能发生故障的各个功能点。所以，研究这些异常情况下数据的产生原因和生成规律，使用相应的手段对已有图像进行修改，模拟生成自动驾驶汽车在这些特殊情况下采集的数据，并将这些数据用于自动驾驶汽车的测试，对确保自动驾驶汽车在这些情况下正确表现具有重要的安全意义。因此，本文将汽车工业开发测试领域的功能安全分析技术与深度学习测试领域的图像扩增技术相结合，使用图像处理的方法来模拟图

像数据采集通路中各种异常情况下的图像数据。本文主要考虑通过上述手段模拟获取自动驾驶汽车在这些故障发生时所采集的数据，并将这些数据作为输入，检测自动驾驶汽车在这些故障或异常发生时的表现，从而验证自动驾驶系统是否满足功能安全的相关要求，达到检验自动驾驶系统的安全性和鲁棒性的目的。

1.2 图像数据扩增国内外研究现状

图像数据扩增主要可以分两种类型。第一种是通用领域的图像数据扩增方法，常见的图像扩增方法有图像旋转、随机添加噪声、对图像做一些弹性畸变以及截取原始图像的一部分等 [9]。在这些数据扩增的方法中，对图像随机添加噪声可以快速大量的获取到扩增后的数据，相较于其它通用领域的图像数据扩增方法，想图像中添加噪声更加简单。同时，通过这样的方式生成的扩增数据更加随机适用性更强。另一种是专业领域的图像数据扩增方法，这种方法需要结合相关领域的专业知识。在这些专业领域知识的指导下总结归纳出相关的图像数据扩增规则，进而通过将这些规则作用于原始数据以产生新的数据 [10]。从而增加图像数据的数量，改变原有图像数据的质量，检测深度学习模型的鲁棒性与安全性，发现其漏洞。以上所述的图像数据扩增方法，能够在实际的实验中实现对数据规模快速的扩展，从而利用不同类型的数据分析深度学习模型中存在的各种问题。为了进一步了解图像数据扩增技术的研究发展情况，本文将对该技术的国内外研究现状进行简要介绍。



(a) 原图像自动驾驶系统方向转角

(b) 处理后自动驾驶系统方向转角

图 1.2: DeepTest 处理前后对比图

Tian 等人设计和实现了一种自动驾驶数据集扩增的自动驾驶系统测试工具 DeepTest，并对其进行测试对测试结果进行了评估 [11]。该工具利用添加图层和修改线条等方式实现了对自动驾驶汽车驾驶环境场景的改变。例如将自动驾驶汽车的场景由原先的晴朗天气切换为雨、雾等异常天气场景，从而实现生成自

自动驾驶汽车测试用例的目的。其中添加雨的前后对比如图 1.2所示。DeepTest 通过模拟不同现实驾驶条件激活了自动驾驶系统中不同神经元。DeepTest 通过数据扩增的手段,在各种异常天气的驾驶条件中(雨、雾等)找出了自动驾驶系统的多种错误行为,其中许多扩增图像的驾驶条件可以造成自动驾驶汽车的产生潜在的致命事故。

Krizhevsky 等人通过数据扩增的手段来增加图像的数据集 [12]。他们通过从原始图像数据集中随机选取并裁剪成 224×224 大小的图像,然后水平翻转它们。作者还通过使用图像处理中颜色变化的手段来改变图像的 RGB 通道的数值。这种数据扩增的办法可以减少训练深层神经网络时的过度拟合。作者声称,他们的使用扩增后的数据对深度学习模型进行训练,使深度学习模型的错误率减少了 1%。

Taylor 和 Nitschke 使用几何变换以及颜色空间变换的方式实现了对图像数据的扩增 [13]。作者在文章中使用了翻转、 -30° 到 30° 旋转和修剪的几何变换方式对图像数据进行处理。同时,作者还在文章中使用了包括颜色抖动、随机颜色处理、边缘增强等颜色空间变换的图像处理方式对图像数据进行扩增。他们对 Caltech101 数据集使用上述的图像扩增方式进行处理,最终得到了 8421 张 256×256 大小的图像。



(a) DeepRoad 合成的雪天场景图像

(b) DeepRoad 合成的雨天场景图像

图 1.3: DeepRoad 合成的驾驶场景图像

Zhang 等人于 2018 年提出了将深度学习技术应用在自动驾驶测试数据集上从而进行数据的扩增,并以此技术为主干提出了 DeepRoad 框架 [14]。DeepRoad 采用深度学习技术中的对抗生成网络技术来合成相不同驾驶场景下的图像数据(如雨雪等非正常驾驶场景),DeepRoad 使用了对抗生成网络中 UNIT 框架来进行路况图像的合成。如图1.3所示,DeepRoad 使用对抗生成网络技术合成了雪天和雨天的驾驶场景图像。实验结果表明 DeepRoad 合成的图像较 DeepTest 合成

的图像与真实的驾驶场景路况更为接近。DeepRoad 首先是将不同天气情况下的图像数据作为训练集输入对抗生成网络，从而对 UNIT 框架进行训练。经过训练后，对抗生成网络就可以对输入的图像数据进行驾驶场景转换，对图像进行合成 [15]。作者最后将通过对抗生成网络生成的不同驾驶场景的图像以及未做变换的原始图像数据输入给自动驾驶系统，并观察自动驾驶汽车的驾驶行为是否存在差异。通过这种数据扩增的方法，作者测试了自动驾驶系统的稳定性和鲁棒性。

通过平均像素值将图像混合在一起是一种违反现实常理的数据扩增方法。这样所产生的图像人们来说并不符合实际的规则，也不像是一个有用的转换。然而，Ionue 就使用了这种方式，对图像数据做出了有效的扩增 [16]。在实验中，他们首先将两幅图像从 256×256 随机裁剪到 224×224 大小，并进行水平翻转。然后通过计算两幅图像的每个 RGB 通道像素值的平均值来混合这些图像。这就产生了一个用于训练分类模型的混合图像。新图像的标签与第一随机选择的图像相同。在 CIFAR-10 数据集上，Ionue 的测试报告显示，在使用平均像素值混合的数据扩增技术时，深度学习模型的错误率从 8.22% 降低到 6.93%。研究人员发现，当测试一个较小的数据集时，结果甚至更好。对于小的数据集，平均像素值混合技术扩增数据使错误率从 43.1% 降低到 31.0%。简化后的 CIFAR-10 数据集测试结果证明了该技术在有限数据应用中效果更强。

Perez 等人提出了一项关于使用多种数据扩增技术来提高模型训练和测试的有效性的研究 [17]。文章中测试的这些数据扩增方法包括颜色增强、旋转、剪切、缩放、翻转、随机裁剪、随机擦除、弹性、混合以及技术之间的组合。实验显示扩增后的图像数据在模型训练和测试中可以表现出更高的性能。Perez 等人还证明了使用几何变换（如旋转、平移、缩放和翻转）扩增后的数据对模型进行训练可以使模型对皮肤损伤分类更为精准。

1.3 论文研究内容及组织结构

本文将汽车工业开发测试领域的功能安全分析技术与深度学习领域的图像扩增技术和计算机视觉领域的图像处理技术相结合，通过建立自动驾驶汽车图像数据传输的通路来分析自动驾驶汽车图像传感器采集数据中可能出现的相关问题。本文对每种异常场景产生数据的原因、机理以及这些异常对图片的影响规律进行分析，从而模拟生成异常场景下的图像数据来测试深度学习系统。本文的主要结构如下：

第一章为绪论部分，介绍论文研究的背景和意义，图像数据扩增技术的发展、现状和存在的问题。并且简要叙述了本文主要的研究工作，最后介绍了本文

的内容与结构安排。

第二章为相关技术和理论综述，主要介绍了本文所涉及的技术和相关的知识要点。主要内容为：功能安全的概念和标准，图像处理的相关技术和理论依据，最后介绍了目标识别和深度学习相关方面的知识，指出了各技术要点在文章当中的用途。

第三章主要介绍了基于功能安全的数据扩增方法，本章首先采用功能安全分析方法中的失效模式与影响分析 (FMEA) 方法对自动驾驶图像数据的传输流程进行分析，并根据其传输路径建立模型。其次对图像传输流通路进行切片，分析每个片段内可能出现的问题和产生原因并对其分类。然后对图像数据的处理过程和异常图像的修正方法进行详细分析。最后对每种问题进行详细讨论，分析这些问题出现时将会对图片产生哪些影响，并对这些影响进行规律性总结，从而通过图像处理技术模拟出问题出现时的图像。

第四章首先介绍了图像数据的准备和处理工作。然后，我们使用计算机图像处理的方法将第三章所总结的问题发生时图像变化的规律作用于原始图像数据，进而得到不同问题场景下的异常数据。最后，我们针对扩增后图像对于目标识别模型的影响进行了实验设计和结果分析。我们对目标识别模型进行训练，并将扩增后的图像数据与原始图像数据输入给训练好的框架，观察并分析实验结果，总结扩增图像对深度学习模型带来的影响。

第五章主要对实验进行总结分析，分析实验结果的数据并对实验进行总结，总结实验中的问题。最后论文的局限性及未来进行规划展望。

1.4 本章小结

本章为绪论部分，首先介绍论文研究的背景和意义，分析了实际生产过程中出现的问题并提出了相关的解决思路。其次介绍了图像数据扩增技术的发展、现状和存在的问题。最后简要叙述了本文主要的研究工作，介绍了本文的内容与结构安排。

第二章 相关理论综述

2.1 功能安全的概念与标准

早在上个世纪六十年代，航空领域和核电领域便首先提出了功能安全的概念。到了上世纪七十年代，由于电子、电器可编程控制系统及相关产品自身安全功能的不完善，导致了当时在全球各地范围内发生多起石油化工爆炸与核电领域的污染泄露事件。而这些事件发生的原因正是由于控制系统相关安全功能失效而造成的。此时，为了解决这些控制系统安全功能不完善的问题，同时为了提高电子电器控制器的安全性能，国际电工委员会便计划制定一套可行的安全技术标准。经过多个领域内专家的积极参与和讨论，最终在上个世纪末期国际电工委员通过并颁布了 IEC61508-功能安全标准 [18]。

现在很多领域的国际基础安全标准的制订基础都是 IEC61508-功能安全标准。但是，由于不同领域研究深度的增加和研究方向的的分化，该标准逐渐不能适用于各个领域的要求。因此，IEC61508-功能安全标准根据不同领域的特点和要求发生了进一步的衍化和应用。例如，过程控制领域的 IEC61511 标准就是由 IEC61508-功能安全标准发展衍化而来的，在核电领域的 IEC61513 标准也是如此。除此以外，汽车领域也结合自身行业的特点和发展要求指定出了专门针对道路车辆功能安全相关标准。其中，ISO26262 就是 IEC61508 标准在汽车工业领域中的具体应用。ISO26262 车辆功能安全标准于 2009 年发布初稿，在进行了讨论和修改后在 2011 年 11 月正式颁布 [?]。ISO26262 车辆功能标准包含了汽车行业内与功能安全相关的全套开发过程，它为汽车安全的设计和开发过程提供了一个生命周期概念，并在汽车开发的生命周期内的每个阶段提供了相关技术支持和问题分析方法。

ISO26262 车辆功能安全标准可以在汽车的研发流程和开发过程中的管理设计阶段发挥其巨大优势。ISO26262 标准为汽车安全的开发提供了一套指导规范。使用该标准对汽车设计开发的整个流程进行设计和管理，可以大大提高汽车电子电器产品安全性，从而保证车内乘客和车外路人的安全。使用 ISO26262 当中推荐的汽车开发流程和方法对汽车电子电器系统以及汽车中的硬软件的模块开发，可以保证汽车整体的功能安全。这样如果与汽车安全功能相关的电子电器系统在面对各种严酷条件发生异常时，车辆整体就不会发生严重的事故 [19]。

在 ISO26262 功能安全标准中，提到了多种方法进行功能安全分析。这些功能安全分析方法为我们提供了一系列完整成熟的措施来发现汽车电子电器中

存在的安全隐患，使用功能安全分析方法可以有效的分析、检测和解决汽车电子电器系统中存在的问题。同时，功能安全分析方法的质量很重要的决定了功能安全项目的成败。因此，本文基于 ISO26262 功能安全标准中所推荐的汽车电子电器系统可靠性分析和风险问题分析方法，使用失效模式与后果分析方法 (Failure Modes and Effects Analysis, FMEA) 对自动驾驶汽车图像数据采集通路中各种可能的风险评估分析，揭示出图像数据采集通路中可能出现的故障。同时基于该方法进行故障问题分析可以预测该故障对于整体系统的影响，从而寻找相应的解决方法对故障发生时所出现的问题进行处理。基于功能安全标准的失效模式与后果分析方法 (FMEA) 对自动驾驶图像数据采集系统进行分析并进行数据扩增和测试的主要步骤如图2.1所示。

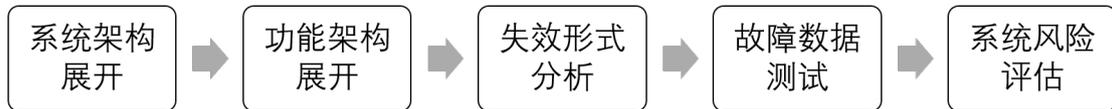


图 2.1: 失效模式与后果分析方法 (FMEA) 示意图

2.1.1 功能安全的概念

为了在汽车的研发和管理过程中提前找出汽车内电子电器系统中所潜在风险及隐患，国际标准化组织提出了功能安全的概念。在功能安全的相关规范和指导下，汽车电子配件的制造厂商以及汽车安全相关系统设计企业就可以使用相关的方法分析预估汽车安全相关的软硬件系统内部存在的潜在隐患，并通过进一步研究判断模拟出故障或异常发生时汽车的表现。这样，汽车生产制造企业就可以设计出相关的安全措施来减少故障发生时的风险，使汽车电子电器产品满足功能安全的要求，确保汽车的安全系统在异常情况下可以正常运行，保证乘客的生命财产安全。同时，在这样的开发过程中，汽车制造企业也可以避免因设计不当而导致的成本增加。

但是，在现实世界中绝对的安全是很难实现的，实际的生产过程中所说的安全指的是不存在不可接受的风险。也就是说，在汽车的实际生产中的车辆行驶的安全就从绝对安全变成了相对安全。只要风险可以被相关的评测机构接受，那么这种风险就是安全的。这就，汽车制造企业就可以把对汽车安全的控制转变为对汽车内部存在的风险控制。因此，汽车功能安全所要保证的就是与汽车

内安全相关所有功能不存在不可接受的风险 [?]。

总的来说，功能安全的关注点是当汽车内部系统发生故障或遇到异常情况时，汽车安全系统所作出的保护反应，而不是汽车电子电器系统内的原有功能或相关零配件的性能。因此，功能安全的可以保证当汽车内部系统发生故障或遇到异常情况时，汽车的相关安全系统进行及时的介入，确保汽车在安全可控的条件下继续运行，从而避免事故发生对人身、财产造成伤害。

2.1.2 功能安全标准 ISO26262

(1) 功能安全标准 ISO26262 的适用领域

ISO26262《道路车辆功能安全》国际标准是第一个适用于大批量量产汽车电子电器系统的国际标准。它的适用对象是总体质量不超过 3.5 吨八座及以下的乘用车上所装配的所有与安全相关电子电器系统 [?]。功能安全标准 ISO26262 仅针对安全相关电子电器系统，包含电机、电子硬件设备以及相关的软件系统，不应用于非电子电器系统（如机械、液压等）。

(2) 功能安全标准 ISO26262 架构

功能安全标准 ISO26262 整体结构与内容如图 2.2 所示，它一共包含了十个部分。其中最主要的是第四五六部分。该标准将汽车电子电器系统的设计和开发分为三个阶段，对每个开发阶段中的功能安全管理过程进行了定义。

在汽车电子电器产品开发的三个阶段中，该标准也对功能安全分析方法进行了推荐与规范。它对汽车功能安全的生产管理流程进行规定，形成了一套成熟化，标准化的汽车安全开发参考模型。在概念阶段汽车制造厂商需要对整体安全进行管理，确定安全管理的目标；在开发阶段该模型根据开发对象的不同又将其细分为系统层面、硬件层面和软件层面的三个层面分别进行开发。产品开发阶段同样需要进行概念阶段和产品开发过程中的安全管理；如图 2.2 所示，产品经过开发阶段并通过了相关的验证测试，产品便进入了生产运行的阶段，处于此阶段的产品仍需要进行系统安全管理。

2.2 数字图像处理技术

图像数据是进行各模块之间实现信息高效交互和自动驾驶汽车感知周围环境的最重要的载体之一。近年来，几乎所有行业领域中都可以看到图像处理技术应用的影子。作为深度学习技术中图像识别和分析的基础，图像处理技术尤为重要。随着图像处理技术的完善和成熟，它在空间探测、生物医学、人工智能以及工业检测等众多领域都得到了广泛的应用，同时结合这些领域各自的特点产生了新的发展 [20]。

第二章 相关理论综述

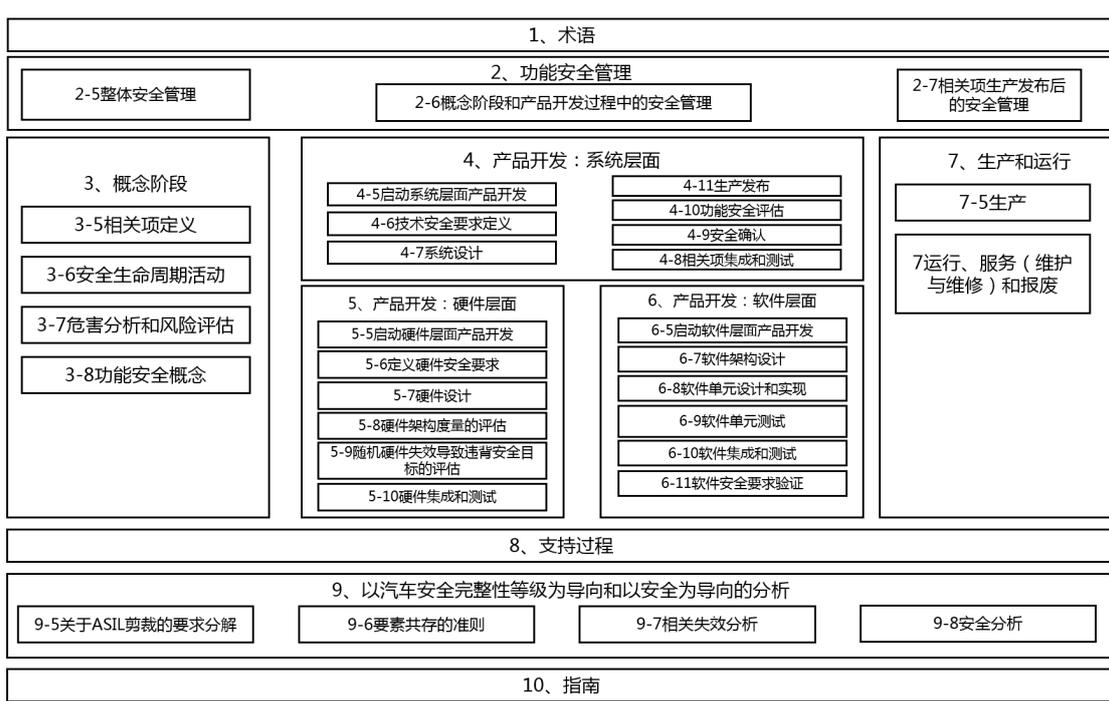


图 2.2: 功能安全标准 ISO26262 架构示意图

目前，人们把图像处理技术分成两类：模拟图像处理和数字图像处理。其中，数字图像处理技术是指将传感器采集到的图像信息转换成数字信息，然后利用计算机对数字图片进行处理的过程。数字图像处理技术既可以改善图片的视觉效果突出图像中的某些特征，又可以对图像的内容和进行修改，生成新的图像数据用于测试或训练深度学习模型 [21]。数字图像处理的优势在于：

(1) 处理精度高。在实际的研究和计算过程中计算机及相关硬件计算设备是图像处理的主要工具，这些设备可以快速精确的完成人类无法完成的计算工作。经过数字化后的图像以多维矩阵的形式表现，人们可以通过相关的计算设备对多维矩阵进行操作。

(2) 再现性好。数字化后的图像不会在图像的存储、传输或展示等环节，因为某些变换操作而导致图像质量发生退化。

(3) 应用面广。数字图像处理技术被广泛应用于各个领域。数字图像处理技术的应用领域主要包括：生物医学工程、通信工程、自动驾驶、军事国防、机器人产业等。

(4) 灵活性高。数字图像处理将模拟图像转换为数字图像后，就可以使用数学思想、手段和工具对其进行运算处理。

数字图像处理技术主要包括：几何处理、数值处理、图像增强、图像重建、

图像编码、图像识别等。

2.2.1 图像的基本表示方法

数字图像是由像素点（像元）组成的，每个像素点亮度的深浅用数值来进行衡量。其中，像素点的属性包括：位置和灰度，像素点间相互组合就形成了多维矩阵型的数字图像。在实际操作中为了准确的描述像素点所在的位置，一般以图像左上角为原点，以单个像素点的长度为单位，建立直角坐标系 [20]。像素点的横坐标与纵坐标分别是在该像素点在图像矩阵中所处的列数与行数。其中根据图像的通道数和灰度级差异可以将数字图像分为二值图像，灰度图像，彩色图像 [22]。

(1) 二值图像：顾名思义，二值图像的像素点灰度属性只能为 0 或 1 两种数值，没有中间的过渡数值。其中像素点为 0 代表该像素点为黑色，反之该像素点为白色。因此，二值图像又被称为黑白图像。

(2) 灰度图像：二值图像中仅有黑白两种颜色，这种方法表示的图像内容和细节丢失严重。为了得到质量更高，内容更为丰富的图像人们提出了灰度图像。灰度图像也使用矩阵进行表示，矩阵中元素的值表示亮度的深浅。区别于二值图像，灰度图像像素点的颜色在黑色与白色之间有许多等级。通常人们用八位二进制来保存一个像素点的灰度值。因此，每个像素点的颜色有 256 个等级，其中 0 表示纯黑色，255 表示纯白色。

(3) 彩色图像：上述的二值图像和灰度图像均为单通道图像，与此相比彩色图像为三通道图像。由于各种色光都可以通过三基色（红绿蓝）（或其他表示方法）按照不同比例搭配形成。因此，彩色图像使用三维矩阵进行表示，彩色图像中每个像素点由 R、G、B（或其他各种分量）分量构成，每种分量使用灰度图像进行描述。

2.2.2 图像的几何变换

图像几何变换又称为图像空间变换，是指将一幅图像空间映射到另一幅图像空间内的操作 [23]。几何变换操作是指将原图像中像素的坐标位置映射到新图像中像素对应的新坐标位置，这个过程中图像的像素灰度值不会发生变化，而像素的位置属性发生改变。图像进行几何变换的前提和核心就是要确定变换前后像素点间位置的映射关系和映射过程中像素点位置的变换参数。几何变换通常可以分为两部分运算：1、确定输出图像与生成图像相关像素点之间位置的映射关系。2、使用特殊的计算方法得到变换后图像对应像素点的灰度值。有些图

像在几何变换时，输出图像的像素会产生非整数坐标。此时需要用到一些特殊的计算方法得出图像像素点灰度值。图像的几何变换主要分为以下几类：

(1) 图像的平移：

图像平移变换是指对原图像的像素坐标进行水平或垂直方向的算数操作。图像的像素点的集合构成一个多维的矩阵，每一个像素都有一个坐标位置。假设原图像中的某一像素点的位置坐标为 (x_0, y_0) ，经过平移量 $(\Delta x, \Delta y)$ 后，坐标变为 (x_1, y_1) 其中 $x_1 = x_0 + \Delta x$ ， $y_1 = y_0 + \Delta y$ ，该计算过程的矩阵形式表示为：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

为了适应像素、拓展适应性，本文使用三维向量进行表示，(以下部分均采用这种形式)，其中矩阵：

$$\begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix}$$

称为平移变换矩阵(因子)，经过平移变换后的图像如图 2.3 所示，其中左图为平移前图像，右图为平移之后的图像。

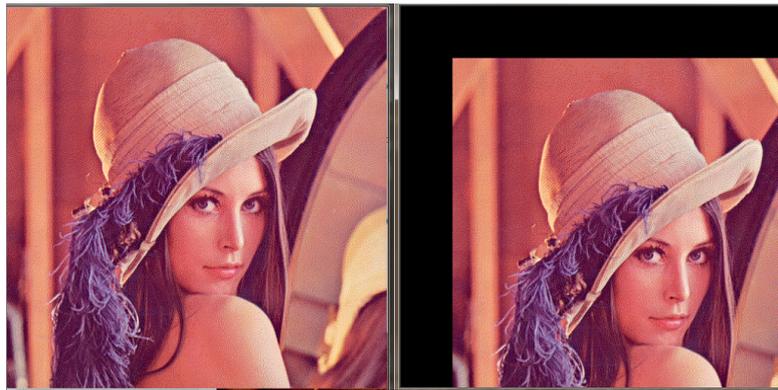


图 2.3: 平移前后对比图

(2) 图像的缩放：

图像缩放是指将图像的两个坐标轴按一定比例缩放生成新图像的过程。如果在图像缩放过程中，两个坐标轴的缩放比例比例一致，那么这种图像的缩放

称为全比例缩放。如果在图像缩放过程中，两个坐标轴的缩放比例不同，会使图像发生畸变，导致图像失真。经过平移变换后的图像如图2.4所示，其中左图为原始图像，右图为缩小后的图像。



图 2.4: 缩放前后对比图

在进行图像缩放时，既需要用到像素位置的映射操作，又需要用到像素值的计算操作。当图像进行缩小时，图像中像素点的个数也会减少。此时，要通过一定的规则选择图像内需要保留的像素点，使图像内容的基本信息特征得到保留。常用的保留像素点选取规则有等间隔采样法和局部均值法。等间隔采样法首先生成一个大小固定的数值作为间隔数，然后在图像中每过一段间隔就选取像素点作为保留的点。局部均值则首先要将图像分解为特定大小的子矩阵。然后计算每个子矩阵中像素点的平均值，用平均值作为新像素点的灰度值代替子矩阵作为保留点。图像的放大需要生成新的像素点来补充放大后图像内部的空白像素区域。图像放大后空白区域内像素的计算需要采用最近邻域法或线性插值法。最近邻域法是与该像素点最近的像素值来填充放大后的空白像素区域。线性插值法是通过被插入像素点附近邻域的像素灰度值进行线性插值，从而得出待填充区域像素点的灰度值。图像缩放计算的矩阵表示形式为：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ S_y & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

其中矩阵：

$$\begin{bmatrix} S_x & 0 & 0 \\ S_y & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

称为缩放矩阵（因子）， S_x 、 S_y 分别为 x、y 方向上的缩放因子。根据缩放公式和相关计算方法对原图像进行操作，可以得到缩放操作后的图像矩阵。

(3) 图像的旋转：

一般情况下，图像的旋转中心就是图像的几何中心。图像的旋转就是原图像中所有像素点都围绕旋转中心同时转动一个固定大小的角度。图像的旋转变换也可以用矩阵变换表示。设点 $P_0(x_0, y_0)$ 旋转 θ 角后的对应点为 $P(x_1, y_1)$ ，则图像旋转的矩阵计算形式为：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

变换公式为：

$$x_1 = x_0 \cos\theta + y_0 \sin\theta$$

$$y_1 = -x_0 \sin\theta + y_0 \cos\theta$$

图像绕中心点旋转 180° 后的图像表现如图2.5所示，左图为原始图像。



图 2.5: 旋转前后对比图

(4) 图像的错切：

平面景物在投影平面上得到的非垂直投影叫做图像的错。错切使图像中的图形产生扭变，错切也被称为图像的剪切或错位变换。图像错切包括水平错切和垂直错切，这种图像处理技术一般被用于产生弹性变形的图像内容处理。

2.2.3 图像的数值处理

图像的算数处理一般可以分为以下几种方法：图像的逻辑运算、图像的算术运算、图像的卷积运算和图像的位平面运算 [22]。

(1) 图像的逻辑运算：图像处理过程中经常需要图像的像素值按照位进行逻辑运算，逻辑运算一般情况下主要用于二值图像的处理。两幅图像的逻辑运算实际上就是两幅图像汇总相应像素并进行按位逻辑运算，最终将逻辑运算后的结果填充到输出图像的相应像素位置。常见的图像逻辑运算有图像的按位与运算，按位或运算和按位非运算。其中按位与运算操作将两幅图像的对应位置的像素数值转换为二进制数值，并进行按位与操作。图像的按位与运算主要用于求两幅二值图像的交集。按位或运算时和与运算操作方法类似，只是对每个像素点进行进行按位或运算。图像的按位或运算主要用于求两幅二值图像的并集。对图像进行按位非运算操作时，对于二值图像来说，就是对图像的色彩进行翻转，原先的黑色变为白色，白色变为黑色。对于灰度图像来说，就是将图像的对应像素转换为二进制并进行按位取非操作。

(2) 图像的代数运算：对输入图像的像素点进行加（减）乘（除）运算称为图像的算术运算。图像之间进行相加操作，可以计算出不同图像的平均效果。当图像所处的场景相似时，对不同的图像进行相加操作，可以有效地降低图像内由于各种异常所导致的随机噪声。需要注意的是，当两幅图像进行加法操作时，如果输出图像的类型与原始图像一致，那么最终得出的图像结果可能会超过图像数据类型所支持的最大值，从而导致图像结果发生异常。这种由于数值溢出导致的异常情况在 `uint8` 类型的图像中更加常见。一般对于数据值溢出这种情况，OpenCV 所提供的函数将会取为原图像数据类型所支持的最大值作为计算的最终结果。人们把在计算过程中发生数值溢出的情况称之为饱和。为了避免出现饱和现象，在进行加法计算前，通常要对图像进行转换，将其转换数据范围较宽的图像类型。图像减法又叫做图像的差分，利用该种计算方法可以对图像变化及运动物体进行检测。在同一场景下对两幅图像进行减法运算，可以达到减去背景的效果。

图像的掩模操作就是通过图像之间的乘法运算得到的。使用掩膜操作可以对图像不关心的区域进行屏蔽，从而得到图像的兴趣区域（ROI）。除此以外，图像与常数之间也可以进行乘法操作，这种操作能够在保持图像的相关对比度不变的情况下对图像进行处理。图像的除法运算经常被用于成像设备非线性影响的校正过程，同时图像的除法运算也可以用于不同图像之间的区别检测。

(3) 图像的卷积运算：卷积操作是指选取特定的卷积核（kernel）并在原图像上移动卷积核同时进行相关计算，从而得到新的图像矩阵的过程。也就是说，图像的卷积是原图像矩阵 $f(x)$ 和卷积核 $g(x)$ 进行一系列的运算。在卷积运算的过程中，卷积核 $g(x)$ 在原图像 $f(x)$ 上按特定的规律进行移动。卷积核每移动到新的位置，就把原图像 $f(x)$ 与卷积核 $g(x)$ 相重合的元素进行乘积求和运算，从

而得到被卷积后的图像矩阵中的像素。卷积核与图像一样都可以使用矩阵的形式进行表现。卷积运算的具体运算过程如图2.6所示。

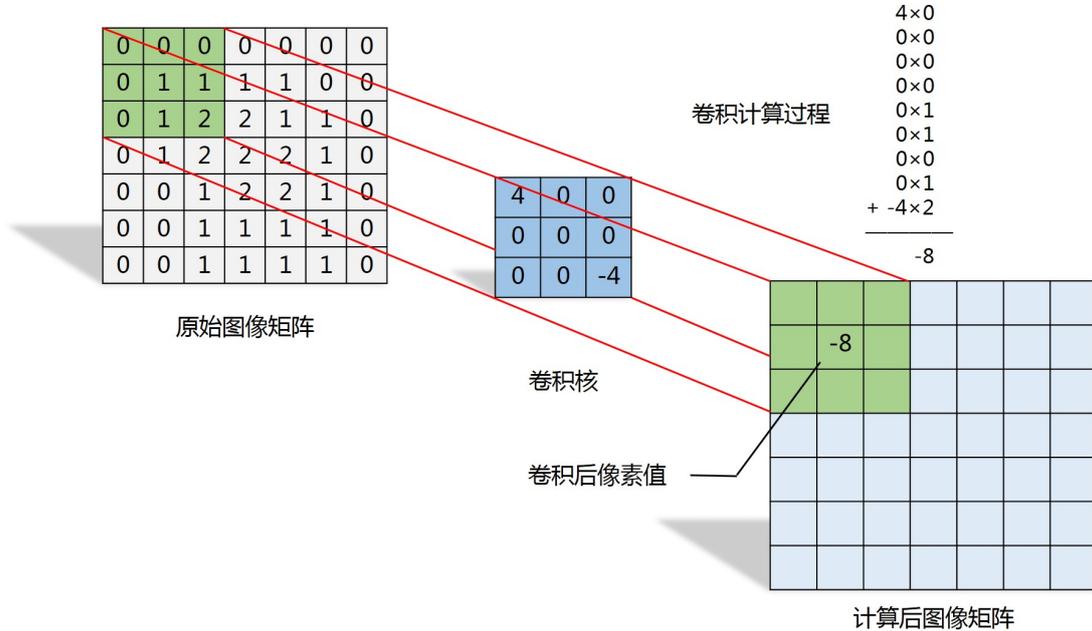


图 2.6: 卷积运算过程示意图

如图2.6所示，图像中最左部分为待处理的图像，它是一个二维矩阵。图像的中间位置是一个大小为 3×3 卷积核矩阵，它是卷积操作最为核心的部分。图像卷积运算的结果很大部分取决于卷积核的选取。卷积运算的像素值的计算过程如图2.6右半部分所示。卷积核在原图像上按照一定的规律进行滑动，同时不断对各个位置的像素进行相乘求和，当卷积核滑动到图像的右下角时整个卷积过程结束。如图2.6所示，目前卷积核处于图像左上角的位置，我们把图像上的9个值与卷积核的9个数值按照对应位置相乘再相加的方式进行计算，最终得到的总和就是我们得到的卷积矩阵中一个像素点的值。然后卷积核继续向右移动一个像素，在对应位置执行相同的相乘再相加操作得到第二个卷积值。之后的操作以此类推，当把所有像素遍历完成之后我们得到的结果就构成了一幅图像。

图像的滤波操作就广泛运用了图像的卷积运算操作。常见的滤波操作有：均值滤波，高斯滤波和双边滤波。在均值滤波中，卷积核矩阵的每个元素大小为矩阵面积分之一。通过均值滤波后，输出图像的每个像素值是卷积核与原图像重合区域内像素的平均值。均值滤波可以使像素点之间的梯度值下降，图像过渡更为平顺。与均值滤波相比，高斯滤波的卷积核则以卷积核的中心为核心元素，矩阵元素的大小成正态分布（越靠近中心元素的值越大，越远离中心元素的值

越小)。由于卷积核中心的权重比较高，经过高斯滤波的图像相对于均值滤波可以更加完整的保留图像的特征。除此以外，高斯滤波在计算图像边缘区域时效果也比较好。在高斯滤波的基础之上，高斯双边滤波对高斯滤波进行进一步优化和修改，高斯双边滤波可以完整的保留图像边缘和轮廓。经过这种滤波方式处理后的图像可以进一步减小边缘信息丢失，从而清晰的保留图像内容的基本轮廓。

(4) 图像的阈值处理：图像的阈值处理指的是对图像内符合条件的的所有像素进行改变或删除。例如，二值化阈值处理就可以设置阈值大小为 127，并将图像内所有高于该值的像素点置为 255，同时将图像内所有低于该值的像素点置为 0。通过这样的方法，我们可以将一副灰度图像变为二值图像进行输出。一般常见的阈值处理主要包括：(反)二值化阈值处理、截断阈值化处理、超/低阈值零处理和自适应阈值处理等。

2.2.4 图像的色彩空间类型

色彩空间指的是图像对色彩的组织方式。使用不同的色彩空间类型，可以用不同的表现形式来表示相同的色彩从而达到对整张图片的数字化描述。常见的色彩空间类型主要有 RGB 色彩空间、GRAY（灰度图像）色彩空间、YCrCb 色彩空间、HLS 色彩空间和 Bayer 色彩空间等 [24]。下面我们对这几种色彩空间进行简单的介绍。

1、GRAY（灰度图像）色彩空间

GRAY 色彩空间也被称为灰度图像，一般情况下，灰度图为单通道图像，且图像中的每个像素点由八位二进制组成。每个像素点的像素范围是 0 255。当图像由 RGB 色彩空间转换为 GRAY 色彩空间时，其具体算法如下：

$$Gray = 0.299 * R + 0.587 * G + 0.114 * B$$

2、RGB 色彩空间

作为最为常用的色彩空间，RGB 色彩空间由 R（红色）、G（绿色）、B（蓝色）三个通道组成。每个通道代表三基色的一种，它们之间通过不同程度的叠加，可以表示不同种的颜色。

3、YCrCb 色彩空间

在 YCrCb 色彩空间中包含三个分量，“Y”表示光源的明亮度，也就是灰度值；而“Cr”和“Cb”表示的则是色度，“Cr”代表红色分量的信息而“Cb”代表蓝色分量的信息。YCrCb 色彩空间主要用于电视以及模拟视频的数据传输领

域，它将亮度信息 (Y) 与色彩信息 (Cr, Cb) 分离，在没有色彩信息，也可以做到可以显示完整的黑白图像，这样的设计可以解决彩色电视机与黑白电视传输信号的兼容问题。同时，将 YCrCb 格式用于数据的传输可以有效的降低数据传输过程中信号占用的频宽，因为这种格式的数据不像 RGB 格式数据那样要求三个独立的视频信号同时传输。

4、HLS 色彩空间

HLS 色彩空间是一种主要面向视觉感知的色彩空间，它从视觉角度出发，指出人感知物体主要包含三种要素：色调 (Hue)、明度 (Lightness)、饱和度 (Saturation)。在进行视觉相关要素操作时，可以将色彩空间转换为 HLS 色彩空间从而对图像数据直接处理。

5、Bayer 色彩空间

Bayer 色彩空间又称为 Bayer 模型，该色彩空间主要用于 CCD 和 CMOS 相机中，它能够从单通道的 Bayer 格式图像中通过 ISP 中的 CAF 插值算法计算得出每个像素点的 R、G、B 值，从而获取到彩色图像。单通道 Bayer 格式图像如图3.12的第一张图像所示。

2.3 摄像头相关技术

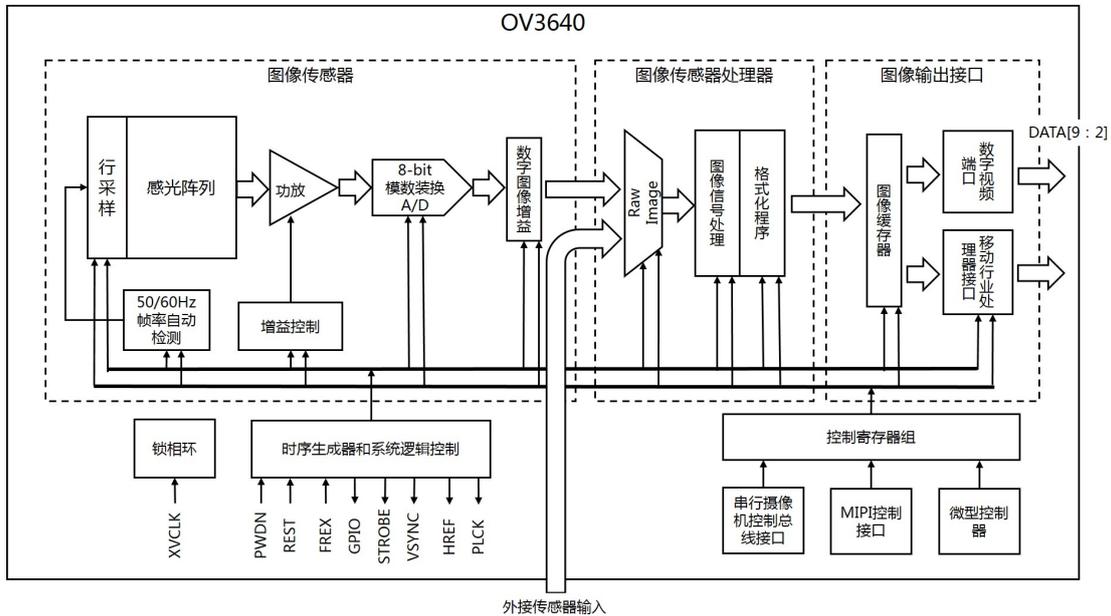


图 2.7: OV3640 核心功能框架图

为了对图像数据进行有效扩增，我们首先需要了解摄像头的组成部分以及各个组件的工作过程，我们以 OmniVision 公司的 OV3640 图像传感器为例，对

典型的摄像头传感器的各部分核心功能进行分析。OV3640 的核心功能框架图如图2.7所示。我们查阅了 OV3640 图像传感器的规格手册来明确各个芯片及其各个引脚的功能。本文将电路板上的各部分电子元器件，按照所负责的功能分为各个功能模块分别进行分析，最终得到 OV3640 传感器的核心功能框架图2.7。

2.3.1 摄像头工作原理

图2.7中左半部分的图像传感器是摄像头的核心部件。图像传感器又被称为感光元件，它是一种半导体芯片，其表面排布了众多的光电二极管，光电二极管受到光照就会产生电荷，将光线转换成电信号 [25]。因此，图像传感器性能的好坏将直接影响到摄像头采集数据的质量，该部分是本论文在数据扩增过程中关注的一点。如图2.8所示，当光照到光电二极管上面，外界物体反射的光照就转换成了微弱的电流。电流通过电容存储起来，就转换成了对应的电压。这时，如果将电容两端的电压通过模数转换装置转换成数字信号，就可以得到一个确定大小的电信号值。这就是感光元件工作的基本原理。

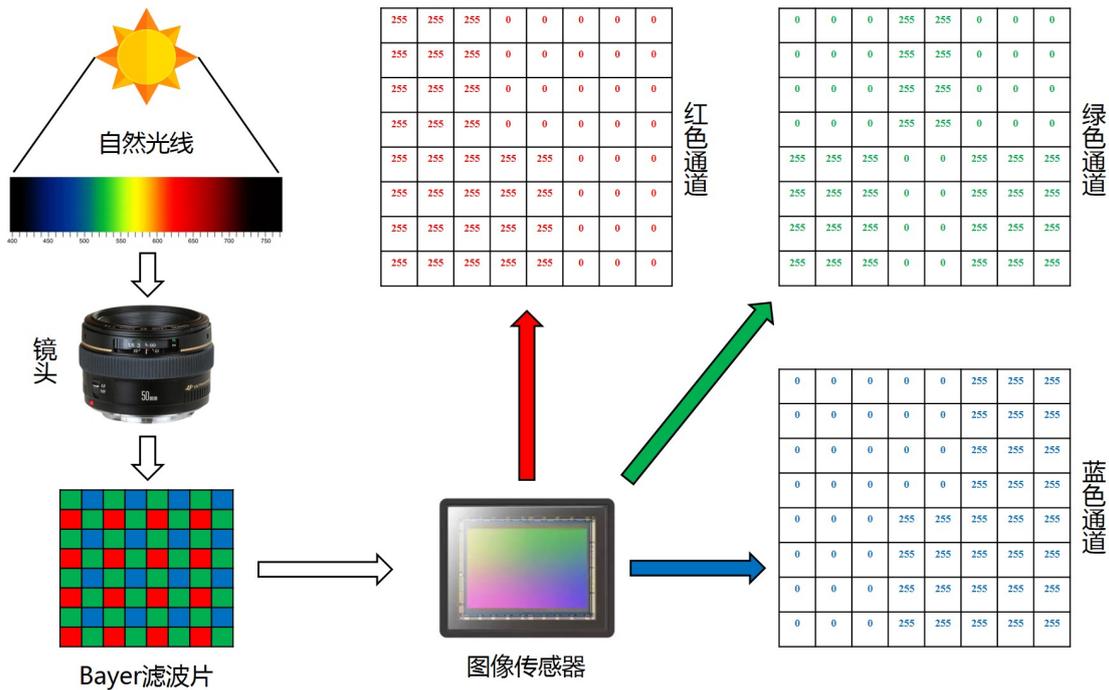


图 2.8: 图像传感器光电转换原理示意图

如果对整个光电二极管组成的矩阵阵列进行模数转换，可以得到一个数字化的图像矩阵。这个数字化的图像矩阵还需要按照某些规定进行进一步的标准化处理。人们对矩阵内数字的范围进行了规定，并使用最大阈值表示白色，最

小阈值表示黑色。在最大阈值和最小阈值之间的数值就表示了不同强度的光线。这个标准化处理后的图像矩阵就是 Raw Image，它是一副单通道的灰度图像。为了将灰度图像转换成彩色图像，相机的设计人员需要用到一种巧妙的方法。参考光学中的基本知识：任何颜色都可以由三基色（红、绿、蓝）以一定的比例混合而来。因此，在图像数据中只要记录下每个像素点红、绿、蓝三种颜色的比例就可以得到彩色图像。但是，由于技术原因，感光元件阵列中的每个元件只能记录光的亮度，不能分辨光的颜色。这就导致每个像素点只能记录一种颜色光的强度。因此，为了得到彩色的图像，设计人员在相机的采光阶段使用滤光片将进入镜头内的自然光进行分解，将其分解为红、绿、蓝三种颜色。这样每种颜色的光线强度就可以使用一个感光元件记录下来，最后通过相关的算法对感光元件记录的数据进行处理，就可以得到一幅彩色的图像。目前，应用最为广泛的滤光片是 Bayer 滤光片，Bayer 滤光片的工作原理如图2.9所示。光线经过 Bayer 滤光片时，只有与滤光片相对应位置允许通过的色光一致的光线才可以穿透滤光片，这样光线就在特定的位置被分解为指定的颜色。

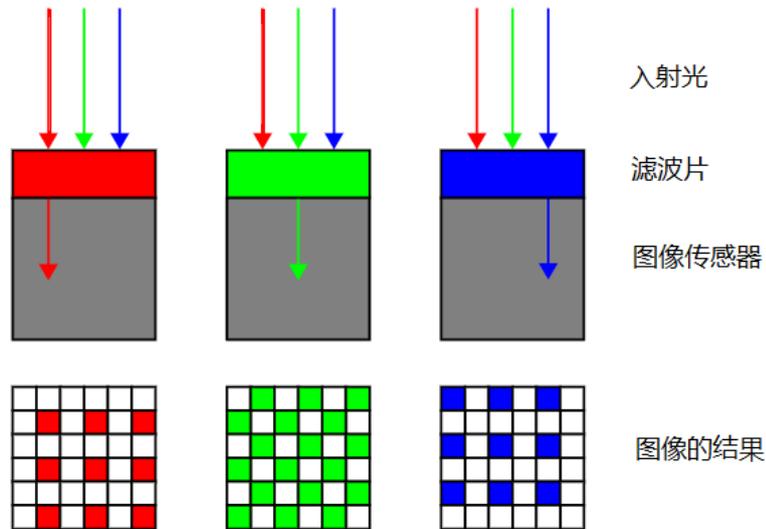


图 2.9: Bayer 滤光片工作原理示意图

在相机内部的图像处理过程中，为了更加规范统一的将 Raw Image 转换为彩色图像，Bayer 格式 (pattern) 将二极管的感光阵列 (image array) 的每四个点定义为一个整体进行计算。由于，人眼对绿色较为敏感。因此在 Bayer 滤光片中，有两个位置能通过绿色光，而能通过红色和蓝色光分别只有一个位置。Bayer pattern 如图2.10所示。图像中 RGGB 子矩阵经过重复的排列组合最终构成了 Bayer 格式图像。

感光芯片除了包含感光阵列以外，功率放大器 (AMP)、增益控制器 (Gain con-

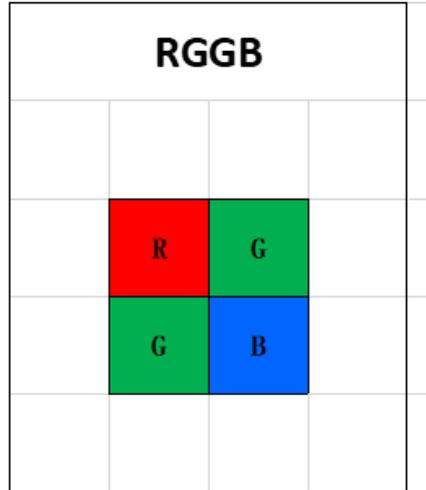


图 2.10: Bayer 格式图像示意图

控)、模数转换器 (ADC) 和数字图像增益模块 (Digital gain) 也是其主要的构成部分。其中, 功率放大器负责对感光阵列的输出值进行操作。感光阵列上的电压越大表示图像越亮, 反之图像越暗。功率放大器的作用就是结合增益控制器对感光阵列每个像素点的电压进行乘法运算。增益控制器负责确定这个乘法运算的相关参数, 当外界环境黑暗的时候加大增益图像就会变亮, 反之就将图像调暗。经过增益处理的模拟图像被送入模数转换器中, 在这里模拟图像就会转换成为 8-bit 的数字图像并传输给给后续流程。8-bit 表示输出图像的每个像素点所占的内存大小为 8-bit, 这种类型图像的像素点的解析力为 0~255。输出的图像接下来将进入数字图像增益模块, 该模块主要负责对数字图像做进一步的增益处理, 使数字图像矩阵中的元素值在范围内合理分布。

在对图像数据进行扩增时还需要对图像对外输出的规律进行研究, 因此, 本文对感光阵列中的读出电路做了简单介绍并进行分析。感光阵列的电路结构如图 2.11 所示, 主控芯片每次选择一行像素输出给后续的处理, 所以图像都是按行逐行输出。

2.3.2 图像信号处理器相关技术

经过图像传感器采集并进行简单处理后图像的数据质量仍然没有达到可接受的水平因此, 图像接下来将会进入图像传感器处理器, ISP 根据图像数据的质量对图像完成进一步的处理过程, 这些处理过程如下。

- 1、黑电平校正 (Black Level Calibration, BLC)。在数字图像矩阵中像素点的值为零时的信号电平叫做黑电平。在相机工作时, 由于存在干扰电流, 导致图

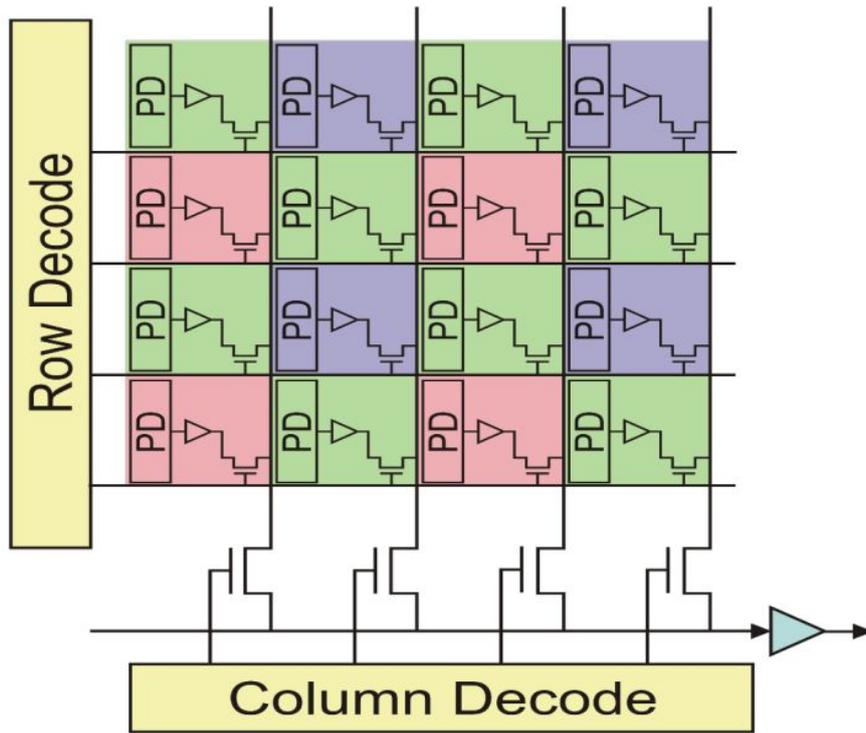


图 2.11: 感光阵列读出电路示意图

像传感器所记录的电压与理想情况下的结果并不一致。导致输出的数字图像矩阵中实际数据应该为零的像素点并不是不为零。对于这种情况，我们可以使用黑点平校正算法对数字图像矩阵进行处理从而减少干扰电流对数字图像矩阵的影响。黑点平校正算法通常的处理过程是将图像矩阵与参考干扰电流信号值矩阵做差。为了得到参考干扰电流信号值的大小，可以考虑使用感光矩阵的边缘像素点进行计算。在相机的设计中，为了得到有效且完整的图片，相机的感光矩阵中实际像素点会多于有效像素点。感光矩阵的边缘区域由于接受到的光线照射量较少，因此该区域又被称为不感光区。在黑电平校正的实际计算过程中常常使用该区域像素点的平均值作为校正值进行运算。在数字图像矩阵中用所有的像素点的值与此校正值做差，就可以得到黑电平校正后的图像。

2、镜头校正 (Lens Calibration): 在相机的取景过程中，光线并不一定都以垂直于镜头的方向进入相机。这时，进入相机镜头组的斜射光线会随着视场角增大而减少。这种情况的发生会导致采集的图像的中心位置的亮度较高，而越靠近图像边缘的地方亮度越低。这种现象在光学成像中被称为渐晕。图像的视角越大，渐晕现象越为明显。在这种情况下采集图像的亮度会发生明显的不均匀现象，这会导致图像边缘内容细节的丢失，造成图像质量的下降。因此，图像

需要使用镜头校正算法来消除渐晕现象。镜头校正算法可以调整图像边缘位置像素与图片中心像素亮度的不同，从而还原图像边缘内容的细节，提升图像数据的质量。

3、自动对焦 (Auto Focus): 由光学知识可知，景物反射光线经过镜头组的折射最终应该将最清晰像投射在图像传感器的合焦平面上，这样才可以得到清晰，高质量的图像。因此，想要获取到轮廓清晰的图像，就需要使用自动对焦模块算法操作镜头组的位置，使得景物将清晰的成像投射在图像传感器上。

4、坏点校正 (Bad Pixel Correction, BPC): 由于制造工艺的缺陷，在生产的过程中，图像传感器上的像素点阵列存在不同程度的损坏或异常。这些异常的像素点在进行光电信号转换时会出现各种各样的错误，从而造成图像内像素点信息出现偏差，导致图像在某些细节上的表现不准确。图像传感器中这些有缺陷的像素点被称为图像坏点。坏点校正算法通过对图像局部的子矩阵进行滤波处理来消除这些坏点对图像的影响。

5、颜色插值 (Color Interpolation): 如上文所述，光线经过 Bayer 滤光片的过滤和图像传感器的记录后形成 Raw Image, Raw Image 为单通道图像，该通道只记录一种颜色。因此，若将图像还原成彩色图像，必须计算出像素点上其它两个缺失通道的色彩信息。通过 Raw Image 计算另外两个缺失的色彩通道的信息的过程叫做颜色的插补。颜色插值算法可以通过 Bayer Pattern 来计算像素点的另外两个通道的信息。由于现实世界中景物是连续的，所以在数字图像矩阵中任何一个位置像素点数值的梯度都不会太大。因此，想要确定一个像素点的 R、G、B 的值，可以考虑使用周围像素点的其他两通道记录的数值通过某种算法进行计算后获得。在 OpenCV 的 Bayer2RGB 图像格式转换算法中利用待计算像素点周围像素值的平均值来计算该点的其他两通道的数值。

6、自动白平衡 (Automatic White Balance, AWB): 由于环境光色温的影响以及图像传感器自身的原因，现实世界中白色的物体，在图像中不一定可以显示为白色，导致图像内容色彩的真实性和准确性出现异常。与图像传感器感受光线的原理不同，人类的视觉感受器官可以对不同光源下的物体颜色进行最大程度的还原。因此，人类的视觉感受器官具有颜色恒常性的特点。但是，图像传感器在不同色温光源下采集到的图像，会因为光源颜色造成图像内容的真实性发生异常。例如，在晴朗天气这种高色温的条件下进行图像的采集会导致图像内容颜色偏蓝，而在高色温的情况下采集的物体颜色偏红。当这种情况发生时，需要使用自动白平衡算法对图像进行处理。自动白平衡可以改变不同颜色光线的比例，在不同的场景下还原物体本来的颜色。为了消除光源对于图像传感器成像的影响，自动白平衡算法通过调节图像中各种色彩的比例还原物体真实的

颜色。

7、颜色校正 (Color Correction)。光线经过 Bayer 滤光片的过滤后，最理想的情况是感光阵列的每个像素点都只会采集到相应单色光光线的强度。但是在实际过程中经过过滤的光线之间会发生颜色渗透，这会导致生成的图像产生颜色误差。颜色校正就是修复这种由颜色渗透产生色彩偏差的过程。颜色校正算法将采集到的图像颜色与标准比色卡进行比较，通过计算后可以得到一个颜色校正矩阵。在颜色校正的过程中，利用颜色校正矩阵和图像传感器所拍摄的图像矩阵进行计算，可以进一步还原图像中物体的颜色，使其与真实情况下的颜色更为接近。

8、图像降噪 (Noise Reduction)。使用 CMOS 图像传感器获取的图像数据会由于传感器本身的问题使输出图像中包含大量的噪声。同时，模拟图像经过图像增益以及模数转换时，这些模块也会由于各种不确定的因素引入其他噪声。这些噪声最终会反应在图像上，这就导致了图像细节的丢失以及图像质量的下降。图像数据中的某些内容特征也会因为这些情况的发生而不能被后续流程识别处理。所以图像降噪算法可以使用滤波的手段对图像进行降噪处理从而减少上述情况对图像质量的影响。

9、图像锐化 (Sharpen)。由上文可知在 ISP 处理流程中存在降噪处理过程。在使用滤波的方法给图像降噪处理的后，图像内容的细节会发生丢失，导致图像模糊。为了减小降噪过程中图像内容细节的损失，使物体边缘轮廓更加清晰，需要对图像进行锐化处理。图像锐化算法可以提高图像局部的清晰程度，使图像色彩内容更加鲜明。

经过 ISP 处理后的图像数据进入输出出层的相关接口、引脚和数据传输线束将图像数据对外进行输出。至此，图像数据完成采集、处理和格式化，可以作为其他模块的输入数据进行分析 and 计算。

2.4 卷积神经网络

20 世纪 80 年代末，Yann Lecun 将 CNN (卷积神经网络) 结构用于手写数字的识别，并取得了不错的表现，确立了 CNN 在图像信息内容处理处理和目标识别领域的地位。从此之后，CNN 就逐渐开始被广泛应用于解决图片分类检索和目标分割等各种问题。卷积神经网络的基本结构如图 2.12 所示。卷积神经网络主要包含输入层、卷积层、池化层、全连接层和输出层几个部分 [26]。卷积神经网络通过输入层接收图片，经过卷积层和池化层进行特征提取和压缩，最后交给全连接层和输出层进行分类和识别。下面本文将对卷积神经网络的主要结构进行介绍。

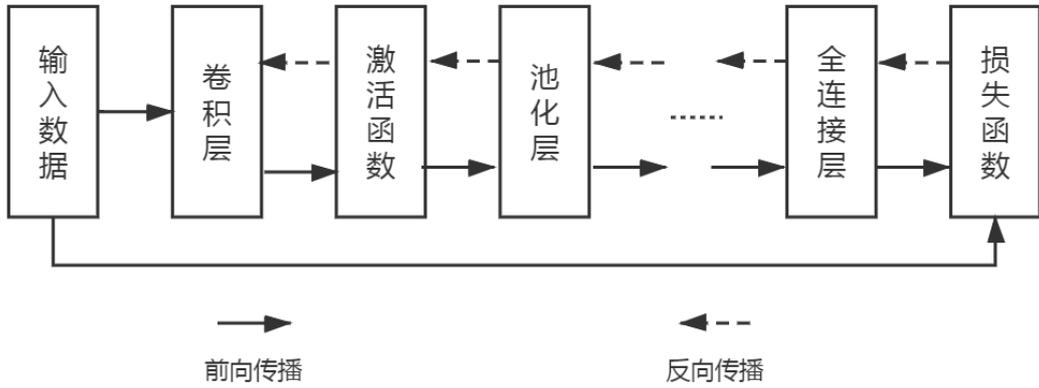


图 2.12: 卷积神经网络基本结构图

2.4.1 卷积层

卷积层是卷积神经网络重要的组成部分之一，卷积神经网络也由此得名，该部分的主要作用是对图片特征信息进行提取。图像矩阵在卷积核的作用下在该部分进行卷积操作。卷积层中图像的卷积操作如图2.13所示。在卷积操作中，卷积核与图像先进行卷积计算，然后再滑动一定步长继续卷积。该过程重复循环最终得到具有高语义特征的卷积特征图 (Feature Map)。对图像进行卷积操作相当于对原图像的局部区域按照卷积核排列做线性组合最后加和得到输出的图像特征值。

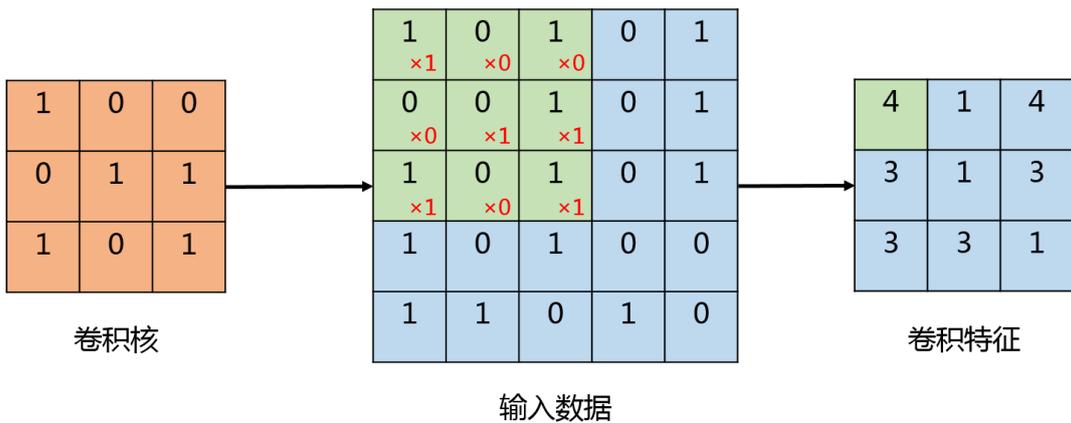


图 2.13: 卷积操作示意图

2.4.2 池化层

卷积层之后往往要对高语义特征的卷积特征图进行池化操作，池化操作可以在保持特征图的特征不变的情况下，进一步对卷积特征图进行降维，相当于对特征图进行降采样操作。该操作可以以极高的效率减少卷积运算参数的数量，缩短模型计算的时间复杂度。池化操作一般可以分为两种，第一种为平均池化运算如图2.14所示。平均池化运算是将待池化窗口中所包含的全部元素进行平均值计算，并将其作为池化后的一个子元素，然后移动池化窗口重复以上操作。另一种为最大池化运算，如图2.15所示。最大池化运算选取待池化窗口中最大元素作为池化后矩阵的子元素，然后移动池化窗口重复以上操作。

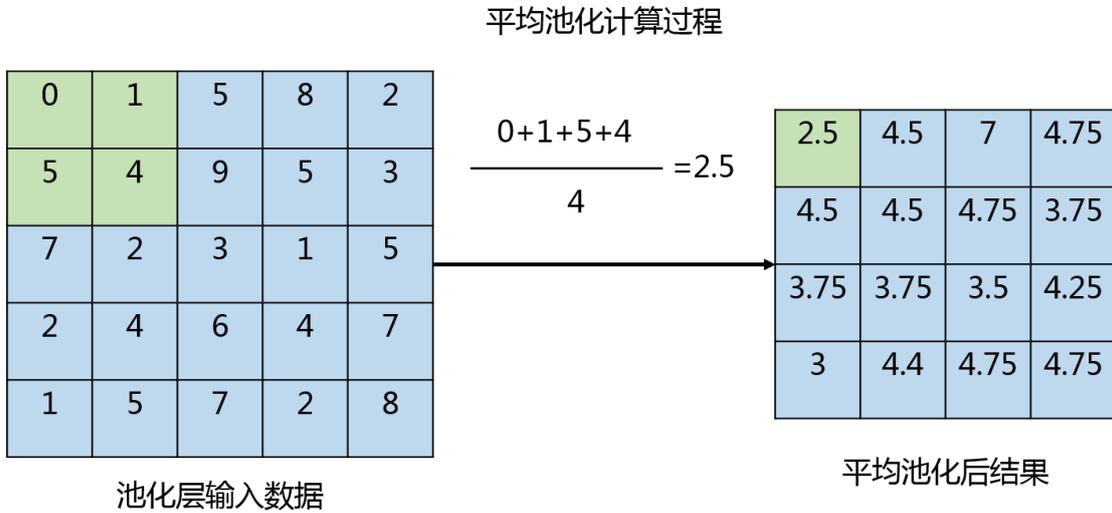


图 2.14: 平均池化操作示意图

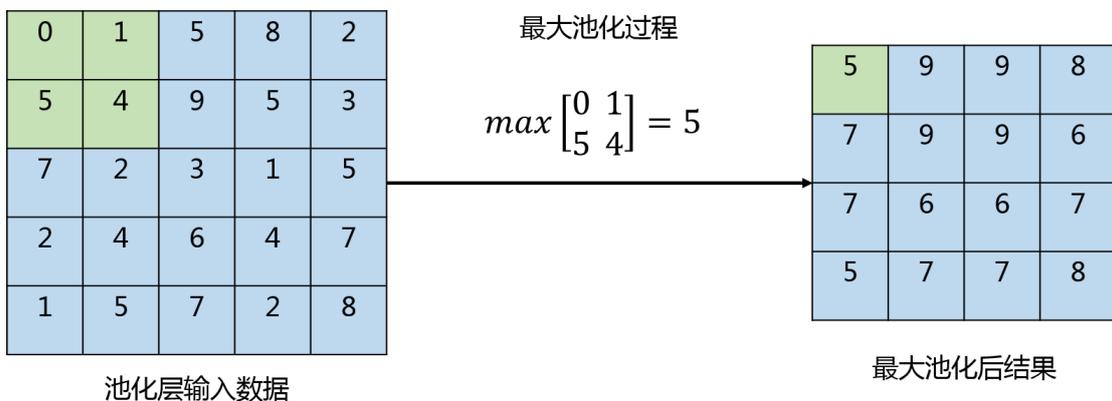


图 2.15: 最大池化操作示意图

2.4.3 激活函数

激活函数对卷积神经网络模型的学习，理解有着相当重要的作用。在卷积神经网络中，单纯的卷积操作只是一种线性变换，尽管网络中有许多层卷积层，但是最终的运算结果仍然是矩阵相乘。为了增加学习模型的非线性，可以在完成卷积操作后安排相应的激活函数进行进一步计算，从而给模型引入非线性因素，这样在非线形模型中也可以运用神经网络处理相关问题。常用的激活函数有 (a) Sigmoid 函数。(b) Tanh 函数。(c) ReLU 函数。他们的图像如图2.16所示。

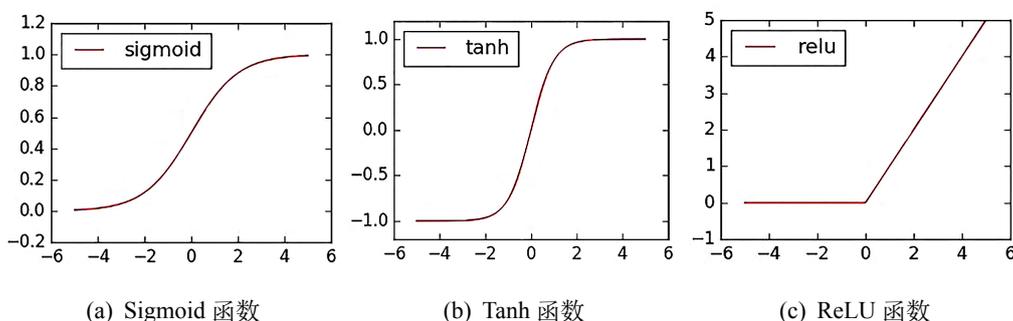


图 2.16: 常见的激活函数图像

2.4.4 全连接层

卷积神经网络中，在完成若干次的卷积与池化计算后，会得到图片提取的相关特征。为了综合这些特征，将其送入 1 个或 1 个以上的全连接层。全连接层的结构如图2.17所示。

在全连接层中，每一层的神经元都会与其前一层的每个神经元相互连接。全连接层的计算过程为前一层的输出与权重系数 W_{ij} 相乘，然后与偏置系数 b 相加，全连接的数学计算式如下所示。

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

其中， W_{ij} 为每个节点的权重， x_1, x_2, x_3 分别表示第一层第二层第三层的输入， b_1, b_2, b_3 分别表示第一层第二层第三层的偏置。

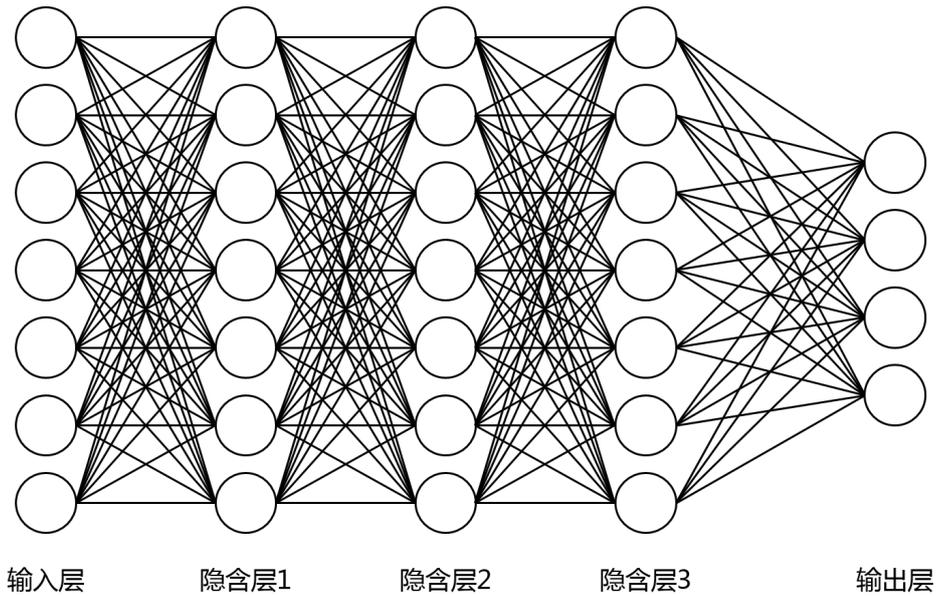


图 2.17: 全连接层网络结构示意图

2.5 本章小结

本章主要介绍了本文所用到的相关技术、概念和理论基础。首先介绍了功能安全的概念及相关标准，这些标准将为下文的分析过程提供重要的支撑。然后介绍了图像处理的相关技术，主要包括图像表示和相关的计算处理技术，这些技术理论将为本文的数据扩增提供主要支撑。其次，为了对图像数据进行有效扩增，本文以 OmniVision 公司的 OV3640 图像传感器为例，介绍了摄像头传感器所用到的相关技术。包括摄像头的组成部分的工作原理以及各个组件的工作过程。以这些理论知识为基础，我们可以进一步理解摄像头异常图像数据形成原理和形成过程，从而更加准确的进行异常图像数据扩增。最后，本章介绍了深度学习中卷积神经网络的相关知识，本文对卷积神经网络的结构进行了简要的介绍，这些理论知识为扩增图像有效性的检测提供了重要支撑。

第三章 基于功能安全的数据扩增方法

3.1 图像采集通路硬件架构及错误分析

3.1.1 图像输入流程切片分析

为了更好的分析和理解自动驾驶汽车的异常图像数据产生原因和过程，我们首先对自动驾驶汽车图像数据采集过程进行切片并对整个图像数据的采集通路建立模型，绘制出图像数据的流转过程，建立数据流的传播线路，如图3.1所示。

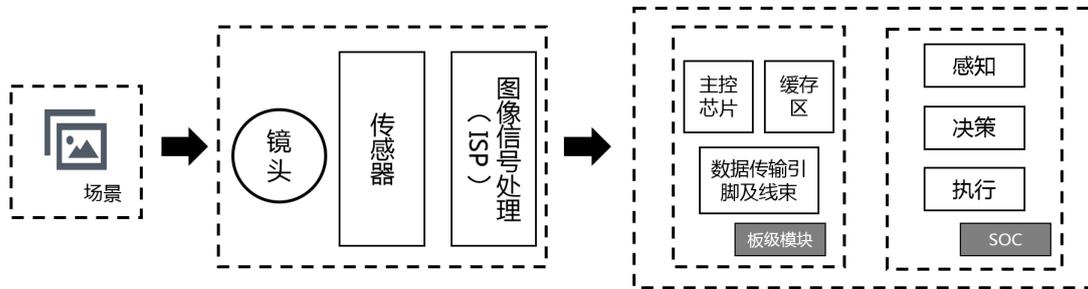


图 3.1: 典型自动驾驶视频流通路示意图

如图3.1所示，图像数据是自动驾驶汽车在具体的驾驶场景场景下采集获得。驾驶场景的反射光线首先被车载摄像头采集，车载摄像头主要包括镜头 (Lens)、对焦马达 (Focus motor)、滤波片 (Filter)、图像传感器 (Image Sensor Processor, sensor 模块的组成部分) 以及主控芯片 (Main Control Chip) 几个部分 [27]，如图3.2所示。自动驾驶汽车车载摄像头采集数据的工作流程如下：首先，物体反射的光线进入相机镜头并通过镜头组的折射进行光线的聚集。经过汇集的光随后通过滤波片进行过滤，去除光线中不需要的部分，从而消除这些光线对后续的 ISP 的校正工作的影响。经过过滤的光线最终汇聚到图像传感器 (CMOS 或 CCD) 上。镜头组对光线的折射程度通过对焦马达调整，经过调整后的折射光线可以将物体的像清晰的呈现在图像传感器上。图像传感器是摄像头的核心，负责将通过镜头汇聚的自然光信号转换为电信号。由于，图像传感器的每个光电二极管代表了图像的一个像素点，它只能感受红、绿、蓝三种光的其中一种。因此，每个像素点中存放的数据代表了红、绿、蓝其中之一的单色光的强度。图像传感器存放的数据经模数转换输出的最原始的数据称为 Raw Data。Raw Data

数据为单通道数据，为了将其转化为彩色图像，还需要经过图像传感器处理器的处理。图像传感器处理器（ISP）处理流程包含了许多模块。经过 ISP 加工处理后的图像最终转换成 YUV 等格式图像信号，经过高速线束的传输对外进行传输。



图 3.2: 摄像头组成示意图

为了形成连续的视频流数据，摄像头驱动程序将图像传感器采集的数据及时快速地传递到图像缓存区。这时，图像传感器就可以进行刷新，从而获得下一帧图像。这些图像数据经由相机内部的电路板和传输线束进行传输，最终形成稳定的视频流数据。随后，这些连续的视频数据通过相关的数据传输总线输入给自动驾驶系统进行处理。自动驾驶系统根据所摄像头采集的数据进行决策，这些决策信息最终会转变为控制转向和油门刹车等执行器的电信号。最终，在这些电信号的驱使下自动驾驶汽车执行相关操作，控制汽车在道路上安全行驶。

3.1.2 图像输入异常分类

失效模式与后果分析（FMEA, Failure Mode and Effects Analysis）是汽车开发过程中一种常用的功能安全归纳分析方法。失效模式与后果分析通过分析汽车电子电器系统中软硬件的失效模式、后果及原因，可以发现系统中存在的缺陷。对这些缺陷和漏洞进行原因分析，可以减少汽车电子电器系统失效的发生，减低异常发生所带来的风险。失效模式与后果分析首先需要技术人员详细了解被分析系统中的每一项功能，寻找各项功能的相关资料，然后运用头脑风暴的方法列举出系统中潜在的所有失效情况。最后技术人员需要对系统内部所有因失效情况而引起的后果进行分析，同时确定并列举出因各种失效情况导致的相应后果。

根据上节对图像输入进行的切片分析，我们了解到自动驾驶汽车内图像采集系统通路中的各个环节，以及各环节负责的主要供。因此，我们可以依据自动驾驶汽车图像采集系统通路中各个节点内部可能存在的问题的按照所对应的功能模块进行分类。根据图3.1中所示的内容，本文使用 ISO26262 所推荐的 FMEA 分析问题的思想，基于问题在图像采集系统通路中发生的位置，将自动驾驶汽车采集图像的异常分为三大类：

1、光线场景异常。在这种异常中本文重点考虑非正常天气因素给图像传感器带来的影响。在该类问题中本文主要考虑两种非正常的天气状况：雨和雾。雨天条件下场景的亮度偏低，同时空气中的雨滴也会影响光线的传播。而在雾天的情况下，由于空气中充满了悬浮的颗粒或液滴大大削减了光线传播的有效长度，造成摄像头探测范围变短，进而产生相关异常影响自动驾驶汽车的判断。

2、摄像头异常。摄像头作为自动驾驶汽车图像采集的核心电子元器件，在工作时可能会发生各种类型的故障。根据上文所述和相机内部元器件负责的功能，本文将该类型异常细分为四种不同类型：镜头异常、对焦马达异常、传感器异常、ISP 校正异常。(1) 在镜头成像异常中本文主要考虑物体遮挡的情况，主要为水滴遮挡。(2) 在对焦马达异常中，本文主要考虑该元器件失效的情况下引起相机不能对焦的故障对图像的影响。(3) 在感光芯片异常中，本文主要考虑激光等强光源对图像传感器的损害，一般这种情况下图像会出现明显的色块，色斑，干扰线条等。(4) 在 ISP 校正异常中，本文依据 ISP 所负责的功能对其异常情况进行分类。根据上述 ISP 负责功能，本文将这种类型的异常情况分为九类分别为：黑电平校正异常、镜头校正异常、坏点校正异常、颜色插值异常、颜色校正异常、自动白平衡校正异常、后处理模块异常。该类的异常情况的发生都是由于摄像头内部核心电子元器件或功能模块失效所引起的。

3、板级模块异常。该类型异常主要由于相机内部 PCB 电路板上焊接器件、连接线束及其所负责功能失效所引起的。在查阅相关资料后，本文将板级模块异常细分为三种不同类型的异常。(1) 主控芯片时序异常：图像传感器需要外界对其提供电源和驱动时钟才可以正常工作。本文主要考虑时序逻辑电路失效引起的异常。(2) 图片缓存异常：在相机的图像传感器进行图像采集后，大量的图像数据进入缓存，最后以视频流的格式对外进行输出。在该异常模式下本文主要考虑图片缓存器失效引起的异常。(3) 线束传输异常：图片数据对外传输需要经过高速连接线束，在这个过程中如果线束连接不稳定或受到相关干扰，图像数据也会出现异常情况。

至此，本文根据功能安全推荐的失效模式与后果分析方法，对自动驾驶汽车图像数据采集通路上常见的的基本异常均进行了考虑。为了分析异常图像形成

的原因并进行相关异常发生后图像数据扩增的实现，本文对异常图像的形成原因，形成规律和异常发生时图像数据受到的影响进行了详细分析。

3.2 图像采集通路功能安全分析

3.2.1 异常图像 FMEA 研究

根据 FMEA 的分析方法，本文可总结出以上的异常分类结果。在本节本文将深入探讨每一种异常问题的形成原因、失效形式(规律)和失效后图像的结果。

1、**光线场景异常**：在该异常类型下本文重点考虑非正常天气因素给图像传感器带来的影响。

(1) **雾场景异常**：当湿度达到饱和状态时，空气中的水蒸气就会依附于凝结核析出并发生凝结形成的水滴。这些悬浮在空气中的水滴可以导致能见度的下降，这就产生了雾。通常悬浮在空气中的水滴分布较为均匀，因此，雾天气本质上是气溶胶现象所造成的。当自动驾驶汽车遇到大雾天气时，其采集的图像质量会严重下降，导致图像内容的丢失。这种情况的发生会使自动驾驶系统发生误判，最终造成严重的交通事故。因此，本节将重点研究雾场景下光线传播的规律，以及雾对图像质量的具体影响。

在雾天气条件下，光线在传播过程中会与悬浮在空气中的水滴遭遇。这时光线的传播线路会因为空气中悬浮的水滴折射和散射发生改变，最终导致摄像头接收的光减弱。光线的折射和散射通常与空气中悬浮物的大小、类型、密度有关。为了分析雾天情况对图片的影响，本文参考了大气散射模型。该模型以 Mie 米氏散射理论为基础，于 1975 年被 McCartney 首次提出 [28]。McCartney 的大气散射模型示意图，如图 3.3 所示。

大气散射模型参考了相关的光学知识，着重研究了图像在雾天条件下成像的规律和图像质量下降的机理。模型指出，由于雾的原因光线在传播过程中会发生不同程度的散射。这种情况的发生将直接影响相机对于光线的接收。参考大气散射模型，本文将图像传感器接收到的光线分为两个部分：1、场景中物体反射光。外界光源照射在物体上发生反射，这部分光线在从物体到相机镜头传播的过程中，会与悬浮在空气中的水滴相遇。物体反射光线经过这些水滴的吸收和散射到达相机镜头内部会发生严重的衰减，导致光线强度降低。2、场景内的环境光。在光线到相机镜头的传播过程中，最终被相机采集的光线也包含了一部分环境光。同样，这部分光线在传播的过程中也会遭遇空气中悬浮的水滴发生散射，导致镜头的进光量发生衰减。[29]McCartney 在大气散射模型中分别对相机接收到的这两部分光线进行详细分析，并根据相关知识建立了数学模型。

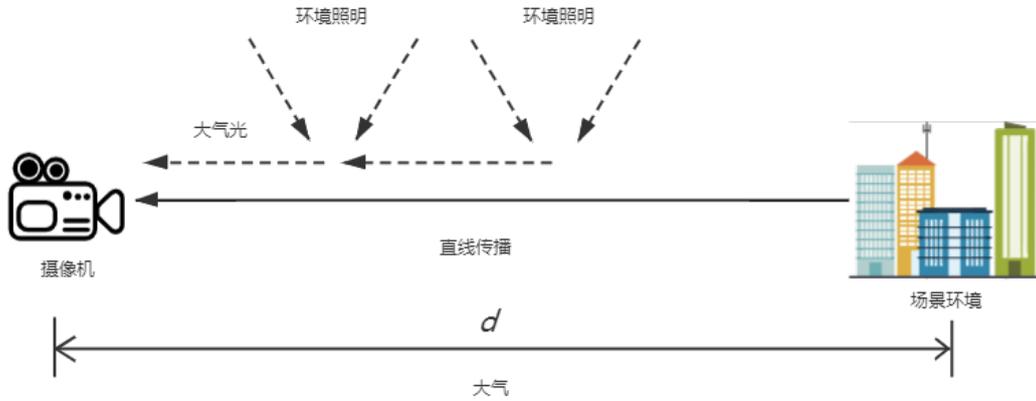


图 3.3: 大气散射模型示意图

最终，进入相机内部的两部分光线可以分别用入射光衰减模型和环境光模型进行描述。但是如果想要使用该模型对入射光线进行描述，自动驾驶汽车所在的环境还需要满足以下两个前提条件：1、图像传感器与采集物体的距离不能过远，一般规定两者之间的距离小于 3 千米。2、空气中悬浮的微粒的分布均匀的。结合自动驾驶汽车采集图像的实际环境和雾的形成特点，大气散射模型的这两个前提条件均得到满足。因此，本文将使用大气散射模型对图像数据进行修改从而生成雾场景条件下的图像数据。下面，本文将对大气散射模型进行详细的介绍和分析。

入射光的衰减模型示意图如图3.4所示。

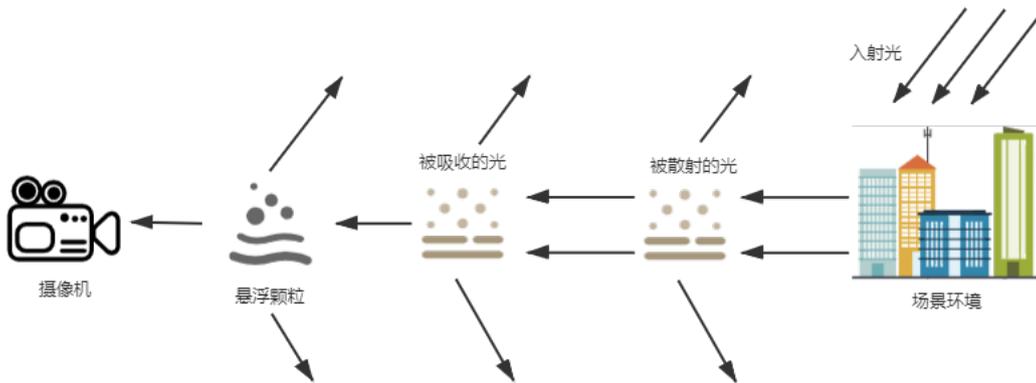


图 3.4: 入射光衰减模型示意图

通过该示意图可以发现，物体反射光在从目标物体到进入相机镜头的传播

过程中，会与空气中悬浮的微粒相接触，导致光线传播路径发生改变。目标物体的反射光线在遇到空气中悬浮的水滴后发生散射，这一过程造成进入相机镜头光线的强度产生衰减，仅有少部分光线参与相机的成像过程。参考相关资料，本文使用数学表达式的方法定量的对雾天场景下入射光传递衰减过程进行建模，得到式 3.1。

$$E_r(d, \lambda) = k_0 * e^{-\beta(\lambda)d} \quad (3.1)$$

公式中 λ 表示入射光线的波长， d 表示物体与镜头之间的距离， $E_r(d, \lambda)$ 代表目标物体的反射光线经过空气中水滴的散射后最终达相机镜头的强度， k_0 表示入射光线在传播距离为零时的能量强度，即光线未经衰减的初始能量， $\beta(\lambda)$ 表示单位体积内空气中的微粒对光线散射的能力，即散射系数 [30]。 $\beta(\lambda)$ 的值越大，散射能力越强。观察公式可知，由于光线散射导致的入射光线强度的衰减曲线为指数型衰减，且进入镜头的光线强度与物体和相机镜头之间的距离密切相关。

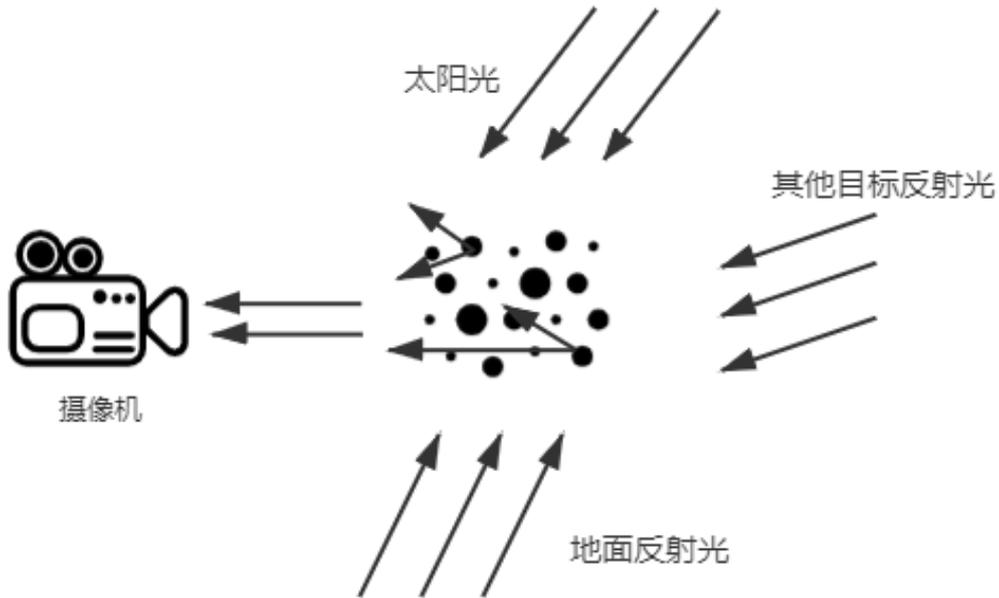


图 3.5: 环境光模型示意图

除了物体的反射光，场景内部的环境光也会进入到相机镜头中参与成像。环境光模型示意图如图3.5所示。在相机采集图像数据的过程中，除了目标物体反射的光线以外，其他所有光线都属于环境光。常见的环境光包括：太阳光、地面反射的光、天空散射的光。这些光线在传播过程中也会与空气中悬浮的微粒接

触，从而使光线的传播路径发生变化，最终导致进入相机内部的环境光发生衰减。环境光和物体的反射光线在传播过程中会同时发生散射最后被相机镜头采集。但是，场景内部环境光的传播过程与物体的反射光线传播过程又存在不同。物体的反射光在从目标物体到相机镜头的传播过程中不断发生光线强度的衰减，而环境光的强度却会随着光源与相机镜头的距离增大而增加。因为环境光传播过程中，随着传播距离的增加会有越来越多的环境光加入成像过程 [30]。为了定量的描述进入镜头的环境光强度，本文参考相关资料，得到式 3.2。

$$E_a(d, \lambda) = k_1 * (1 - e^{-\beta(\lambda)d}) \quad (3.2)$$

公式中 $E_a(d, \lambda)$ 代表环境光经过空气中悬浮的微粒散射后最终达成像设备的光照强度， k_1 表示环境光在距离为无穷大时的能量强度，即环境光亮强度的最大值。

参考 McCartney 的大气散射模型可知，在雾天场景下，进入相机镜头参与成像的光线既包括物体反射光又包括环境光。这两种光线在传播过程中会发生散射和汇聚现象，导致不同光线强度的变化。因此最后进入相机镜头内部的光线强度是衰减后物体反射光和环境光强度的总和，即：

$$E(d, \lambda) = k_0 * e^{-\beta(\lambda)d} + k_1 * (1 - e^{-\beta(\lambda)d}) \quad (3.3)$$

对公式 3.3 进行合并化简，同时使用 $I(x)=E(d, \lambda)$, $J(x)=k_0$, $A=k_1$, $t(x)=e^{-\beta(\lambda)d}$ 进行替换最终得到公式 3.4：

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (3.4)$$

经过对公式的替换化简，本文得到了雾天图像的成像退化数学模型（公式 3.4 所示）。公式中 $I(x)$ 代表雾天场景下采集的图像， $J(x)$ 表示天气晴朗时摄像头采集的图像， $t(x)$ 表示透射率， A 代表环境光。透射率表示进入相机镜头光线的透射雾的能力， $t(x)$ 值越大表示光线的透射力越强。对公式进行分析可以发现：目标图像透射雾的能力与距离成反比，大气光透射雾的能力与距离成正比。

参考相关资料，同时对上述模型进行分析，本文将雾天条件下图像的成像特征进行归纳。具体结论如下：

（一）随着物体反射光线传播距离的增加，物体反射光线的强度会发生不断的衰减，这将造成图像中的白色噪点急剧增加，导致图像色彩失真。

（二）图像中物体的亮度和对比度随光线传播距离增加而减小。在图像中具体表现为距离镜头位置越远，物体亮度和对比度越低，轮廓越模糊。

(三) 在空气悬浮微粒分布均匀的情况下, 雾的浓度与光线传播距离呈正比。在图像当中可以表现为距离相机镜头位置近的地方的雾气浓度小, 反之雾气浓度大。

为了对雾天场景的图像进行有效扩增, 本文还查阅相关资料, 最终总结出晴天条件下的无雾图像与有雾图像的区别, 具体区别包括以下几点:

(一) 与无雾图像相比, 雾天场景下图像的像素灰度值在散射的作用下发生改变。因此有雾图像的内容细节较为模糊, 特征难以分辨, 且图像的清晰度和分辨率降低, 可见度和对比度降低, 动态范围变小等。

(二) 与无雾图像相比, 雾天场景下的图像会出现明显的色彩偏差。图像整体呈现灰白化, 视觉上图片色彩饱和度下降。

(三) 在频域特性上, 雾天场景下的图像高频分量减少, 低频分量增加。

(四) 在空域特性上, 雾天场景下的图像的直方图灰度级趋向于平均值, 且集中分布, 动态范围小。

(2) 雨场景异常: 雨滴的直径要远远大于雾天场景下空气中悬浮水滴的直径, 并且空气中的雨滴的分布情况也不同于雾天场景。在雨天场景下空气中的雨滴分布趋向于离散随机的状态。因此, 在雨天场景下所采集的图像与雾天场景的图像有很大的区别。雨天场景下, 光线传播主要受到折射现象的影响, 因此, 物体的反射光线在传播的过程中大部分会发生巨大的变化, 仅有极少部分光线会到达镜头。物体的反射光线会因为雨滴的折射导致光线穿透量大大降低, 在图像内主要表现为对光线遮挡作用。同时, 研究表明雨滴距离相机位置越近, 对于光线的遮挡作用就越加明显 [31]。因此, 在研究雨天场景对于图像的影响时, 本文主要考虑雨滴对于光线传播的阻挡作用。

为了更为真实的模拟雨天情况下图像的特征, 本文对雨滴的形态特征也进行了分析和研究。雨滴在空气中下落的过程会受到地球引力和空气阻力的共同作用, 从而导致雨滴的形态很不稳定。首先, 云层中悬浮的细小水珠, 通过不断聚合, 在表面张力的作用下形成球形水滴。在水滴聚合到一定程度时, 在重力的作用下雨滴开始下降。在下降的过程中不同的水滴不断撞击合并成更大的不规则雨滴。这时, 在空气阻力的作用下, 雨滴的下表面逐渐扁平。由于雨滴下落过程中速度不断增加, 导致空气阻力变大, 这时雨滴的下表面向内凹陷并开始拉伸, 最终分解成更小的雨滴。雨滴在下落途中, 这种过程往复循环。因此, 雨滴在下降的过程中大部分表现为细长的纺锤形态, 这就是雨天图像内所观察到的细长型白色雨线噪声的原因。由于图像传感器在采集图像数据时具有瞬时性特点, 在本文的数据扩增中, 我们将图像中雨滴形态描述为细长的纺锤状。

结合雨天情况下光线的传播规律和雨滴形态学特征, 本文对雨天情况下图

像的特征进行归纳，具体结论如下：

(一) 相对于雾天场景，雨天场景下光线的传播距离较远，图像内容的轮廓清晰。

(二) 光线在传播过程中，主要受到雨滴的遮挡作用。因此，雨天图像中雨滴通常表现为大量的白色雨线噪声，且雨滴距离相机位置越近，对于光线的衰减作用越明显。

(三) 瞬间降雨量决定了图像中雨线噪声的数量。

(四) 雨线噪声下落的姿态受到重力、空气阻力以及风力等影响。在图像中随机分布，无具体规律。

2、摄像头异常：在该异常类型中我们主要考虑以下四种类型的异常：镜头成像异常、对焦马达异常、感光芯片异常、ISP 校正异常。

(1) 镜头成像异常：自动驾驶汽车装配的摄像机通常位于车辆前挡风玻璃后面的后视镜后方。安装在这个位置的摄像头的视角最佳，可以在正常的情况下准确的采集路况信息。但是，如果挡风玻璃上存在物体，那么自动驾驶汽车采集的图像会因视线遮挡导致图像数据质量下降，从而引发各种问题。本文主要考虑挡风玻璃上雨滴遮挡引起的采集图像异常。这种异常会导致基于图像数据的自动驾驶汽车在行驶过程中发生错误。为了研究雨滴在玻璃上的形态，本文参考了 Garg 和 Nayar 提出的雨滴模型。该模型描述了静止雨滴在挡风玻璃上的形态以及雨滴对传播光线的折射作用 [32]。结合流体表面张力等相关知识，以及实际观测的情况，Garg 和 Nayar 提出的雨滴模型表明雨滴可以在挡风玻璃表面呈现复杂的二维形状，但大部分的雨滴形状可以近似为椭圆球形。现实中挡风玻璃上雨滴的形态如图3.6所示。



图 3.6: 挡风玻璃上雨滴形状示意图

由于雨滴在挡风玻璃上的形态为凸椭球状，且表现为透明状态，因此可将雨滴视为透明材料制成的光学透镜。参考基础光学知识可知，当光线以不垂直

于折射平面的路径通过不同密度的透明介质时会产生折射现象。这种现象的发生是由于光在不同介质的传播速度不一致造成的。在汽车挡风玻璃上存在雨滴遮挡时，我们对自动驾驶汽车采集图像时光线的传播路径进行分析，最终得到物体反射的光线的传播路径依次为：空气、雨滴、挡风玻璃、空气、镜头。因此，物体反射光线在传播过程中会发生折射。经雨滴折射后摄像头采集的图像类似于经小的凸透镜折射后的图像。在实际情况下场景中物体与光心的距离大于两倍焦距，而摄像头一般距离光心在一倍焦距和两倍焦距之间。其位置关系如图3.7所示。

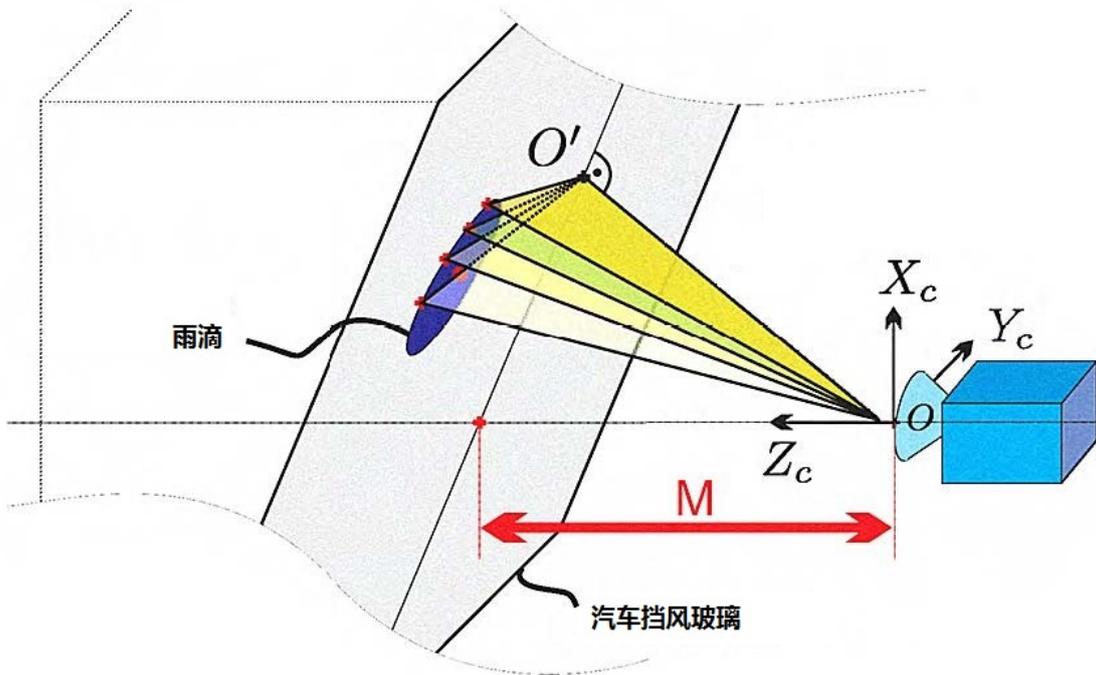


图 3.7: 相机相对位置示意图

如图3.8所示，根据凸透镜成像规律可知：当物距大于两倍焦距而成像设备在一倍焦距和两倍焦距之间时，相机内部会呈现倒立、缩小的实像。此时像距小于物距，物体的像小于物体的大小。考虑到凸透镜对光线的汇聚作用，水滴内的图像亮度要比原始亮度高。

因此，本文将挡风玻璃上雨滴带来的成像异常的图像特征归结为以下三点：

- (一) 雨滴落在挡风玻璃上，在表面张力和重力的作用下通常成椭圆球状。
- (二) 雨滴相当于凸透镜，根据成像物体、雨滴和摄像头相对位置关系，光线经过水滴的折射，最终成倒立、缩小的实像。
- (三) 光线经过水滴汇聚后，水滴内的图像亮度要比原始亮度高。

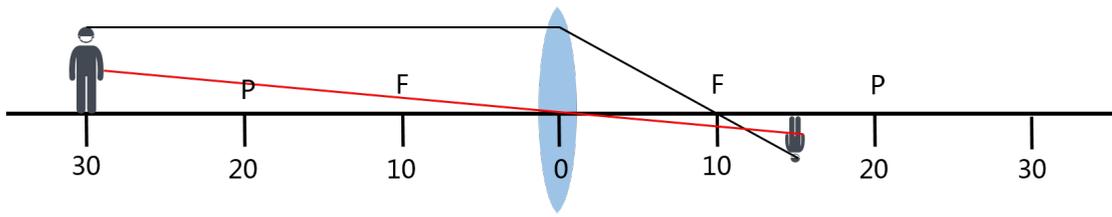


图 3.8: 凸透镜成像示意图

(2) 对焦马达异常: 对焦马达可以在信号的控制下将电能转化为机械能驱动镜头组进行运动。通常摄像头的自动对焦功能就是由对焦马达和相关的控制电路完成的。在对焦马达的驱动下镜头组可以改变相对于图像传感器的位置, 从而实现对焦功能。当其失效后, 会导致图像失焦, 图像内容变的模糊。为了进一步理解失焦图像的形成原理, 本文引入弥散圆的概念: 光线通过镜头的折射会发生汇聚与扩散, 在这个阶段, 景象的折射光线经历了从圆到焦点再扩散到圆的过程。焦点前面和后面的圆就叫做弥散圆。若这个圆形像的直径足够小, 那么采集的图像就会足够清晰。如果圆形变大, 图像会变模糊, 此时这个圆形像由清晰到模糊的临界直径被称为容许弥散圆直径。弥散圆如图3.9所示。因此, 在相机对焦时通过对焦马达的调节, 可以将景物的像呈现在两个弥散圆之间, 这时采集图像就是清晰的。两个容许弥散圆之间的区间称为景深。

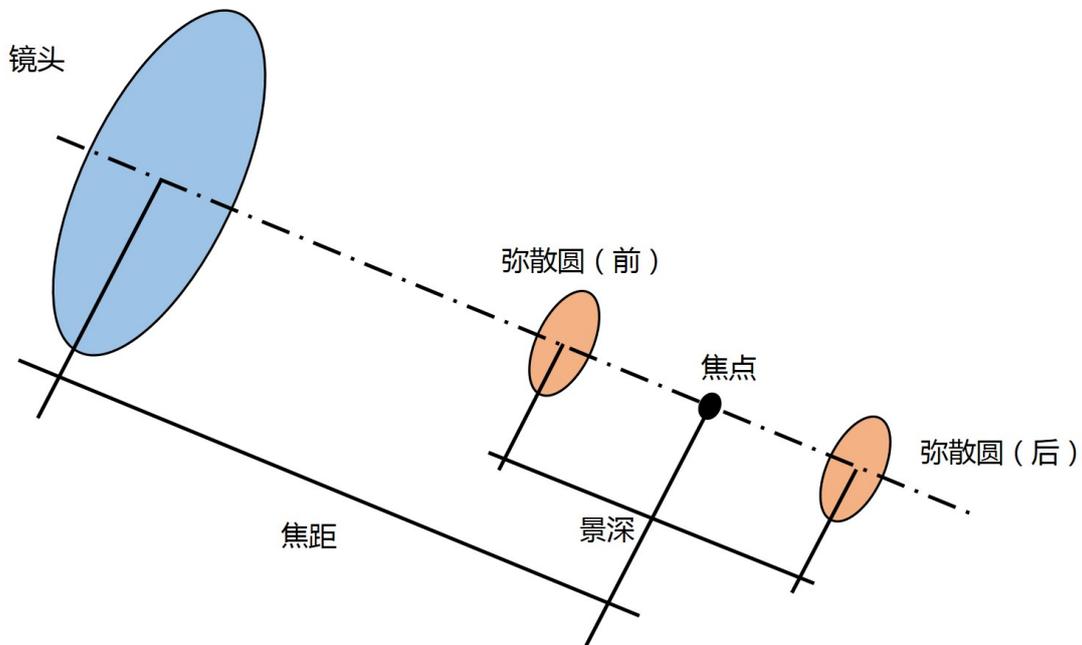


图 3.9: 弥散圆示意图

由此可知, 对焦马达的作用就是通过调整镜头组, 使图像传感器位于景深

范围内。若图像传感器不在该范围内，图像将变得模糊。因此，当对焦马达失效时，不能确定图像传感器与景深区间的相对位置。因此，图像传感器可能不在景深范围内，在这种情况下采集的图像将变的十分模糊，质量大大下降。

(3) 感光芯片异常：随着激光技术的发展，生活中激光的应用范围也越来越广泛。激光对图像传感器器件的损伤的案例也越来越多。因此，本文主要考虑激光照射造成图像传感器故障导致的采集图像数据异常。图像传感器的构成和电路十分精密且复杂，如图2.11所示，这种结构会使感光阵列中的像元之间串扰很少。所以，小面积激光照射图像传感器并不会对其造成大面积损伤。当图像传感器遭受到激光照射后，在激光照射的范围内感光元件特别容易发生损伤。由于感光阵列的设计，这些损伤会影响到与之有关的其他像元，从而发生串扰现象。激光照射导致图像传感器损伤的程度主要由激光的能量密度、重复频率、辐射时间以及图像传感器的材料等因素有关 [33]。

根据激光对图像传感器的损伤程度，工业界将激光对图像传感器的损伤分为两种情况：硬损伤，软损伤。硬损伤指的是图像传感器经过激光照射后结构发生变化。这种损伤通常是破坏性的，并且是不可逆的。软损伤是指图像传感器的功能特性产生了暂时性的变化。例如图像传感器暗电流增加或部分像元的暂时性失效等。 [34] 激光损伤图像传感器后的图像效果如图3.10所示。

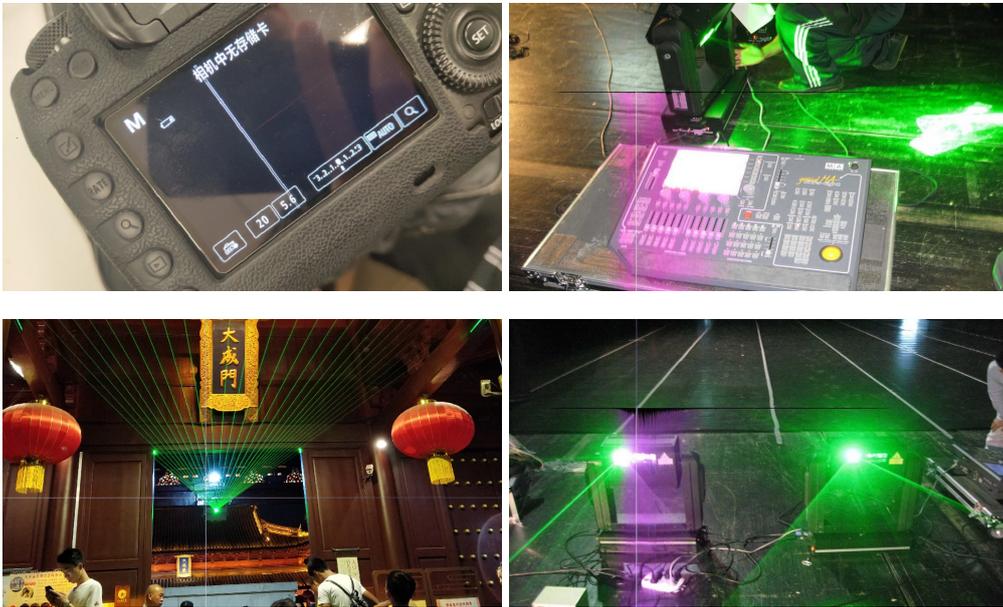


图 3.10: 激光损坏 CMOS 后的图像

本文经过研究调查发现，相机在激光照射下出现的暂时性和永久性损伤表现会有以下几种不同类型。例如，相机在连续激光照射下采集的图像数据会出

现饱和亮斑、串扰亮线、过饱和全黑屏和在重复脉冲激光照射下的漂移亮点、主光斑丢失等 [35]。因此，在进行感光芯片异常数据扩增时，本文主要考虑对图像添加这些效果从而模拟图像传感器因激光照射导致的异常图像数据。

(4) ISP 校正异常：图像信号处理器主要负责对图像信号传感器输出的数字图像进行校正修改。它是相机系统重要的组成部分，因此可以将图像信号处理器视为一个独立的系统级芯片。在图像信号处理器的内部运行着用于处理图像的各种算法程序。根据这些算法程序对于图像的不同作用，研究人员将这些算法程序依次进行串联，形成了一条有着明确先后关系的图像信号数据处理的流水线。在该流水线中，负责处理图像数据的不同算法模块被安排在不同位置互不干扰。这样由图像传感器输入的每一帧数字图像数据就可以依次流经每个算法模块，每个算法功能模块接受上游数据处理并输出给下游模块，最终图像经过各种算法的处理对外输出。其工作流程及数据通路如图3.11所示。

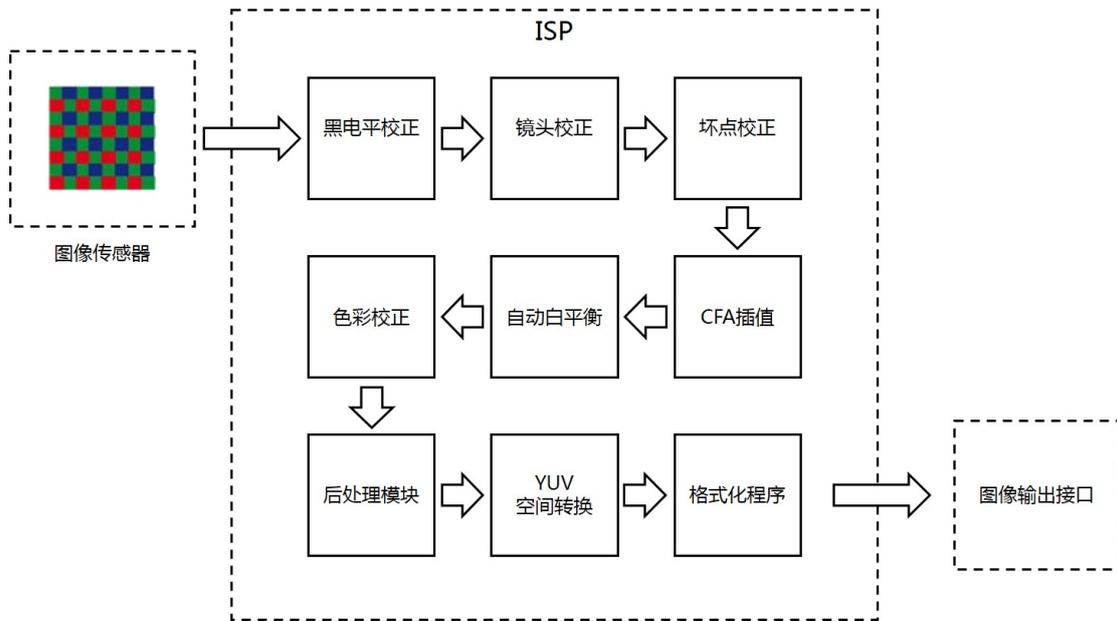


图 3.11: ISP 工作流程及数据通路示意图

由图可知，ISP 工作流程主要包括：黑电平校正 (Black Level Calibration)、镜头校正 (Lens Calibration)、坏点校正 (Bad Pixel Correction)、CFA 插值 (Color Filter Array Interpolation)、自动白平衡校正 (AWB Correction)、色彩校正 (Color Correction)、后处理模块 (Post Process Module) 以及 YUV 空间转换 (RGB2YUV) 等。本文结合功能安全的分析方法，参照每个模块的功能责任与相应算法，分别对各模块失效情况进行分析，同时研究了这些异常对图像质量的具体影响。

a. 黑电平校正 (BLC): 由于受到传感器内部干扰电流的影响, 数字图像中的黑色与标准色黑色存在差异, 这会导致图像的对比度的下降。为了解决这样的问题, 必须使用黑电平校正算法对图像进行进行处理。黑电平校正的基本原理如下:

$$\begin{bmatrix} R'_{i,j} \\ G'_{i,j} \\ B'_{i,j} \end{bmatrix} = \begin{bmatrix} R_{i,j} \\ G_{i,j} \\ B_{i,j} \end{bmatrix} - \begin{bmatrix} O^R_{i,j} \\ O^G_{i,j} \\ O^B_{i,j} \end{bmatrix}$$

公式中, $R_{i,j}$, $G_{i,j}$, $B_{i,j}$ 分别表示待处理图像中坐标为 (i, j) 的点的 RGB 的分量值, 而 $R'_{i,j}$, $G'_{i,j}$, $B'_{i,j}$ 分别表示经过校正后的该点的 RGB 的各分量值。 $O^R_{i,j}$, $O^G_{i,j}$, $O^B_{i,j}$ 则分别表示用于黑电平校正的校正参数。如上文所述, 校正参数需要通过图像传感器边缘的像素点进行计算。若该模块失效, 暗电流的干扰信息将影响后续 ISP 各模块的处理, 导致自动白平衡校正不准, 出现画面整体偏绿或者整体偏红, 对比度下降等现象。

b. 镜头校正 (Lens Correct): 由于受到渐晕效应的影响, 当相机视角扩大时, 进入相机镜头的边缘斜光线将急剧下降。这会使得采集的图像中间区域较亮, 边缘区域比较暗。渐晕效应会造成图像内部亮度分布不均匀, 这会大大降低图像质量, 影响自动驾驶系统对目标识别的准确性。因此, 为了补偿四周的亮度, 需要使用相关算法进行镜头校正。

为了根据原始图像模拟出未做镜头校正的图像数据, 我们首先需要了解镜头校正算法的具体工作原理。镜头校正算法首先需要划分待调整区域和不处理区域。不处理区域内像素点亮度值分布均匀, 该区域范围内的像素不需要做任何调整。图像中这个区域以外的所有像素点为待调整区。以不处理区域为中心, 计算出待调整区域各像素点灰度值衰减速度。这样就可以得到每个像素点相应的补偿因子 [36]。将补偿因子与待调整区域内各像素点进行运算, 可以调整图像的整体亮度, 使其分布更加均匀。该模块失效会导致采集图像中间区域亮度较高而边缘亮度不足。

c. 坏点校正 (BPC): 由于生制造工艺等原因, 在生产过程中图像传感器上的像素点阵列存在不同程度的损坏或异常。在图像传感器的加工过程中, 由于制造工艺的原因, 这些坏点呈现集中分布。这些存在坏点的图像在进行光电信号转换中某些像素点信息将会出现偏差。为了消除图像传感器中这些坏点带来的影响, 坏点校正算法对图像子矩阵进行滤波处理。当该模块失效后, 图像中的某区域会产生若干异常像素点, 这些像素点与周围像素点的灰度值相差极大。

d.CFA 插值 (Color Filter Array Interpolation)：由于技术条件的原因，图像传感器的每一个像素点只能记录一种色光的强度。这样记录的图像为单通道图像，图像中将会有 2/3 的色彩信息丢失。为了还原图像的完整色彩信息，需要对 Bayer 图像的其他两个缺失通道进行补充，这个过程称为 CFA 插值。当该模块失效时，图像会以 Raw Image 的形式进行后续的处理工作。因此，要模拟该模块异常时图像的状态，需要对对应像素点位置的 R、G、B 通道进行修改。如图 3.12 所示，左侧是 Raw 原始图像，右侧是经过插值之后的图像。

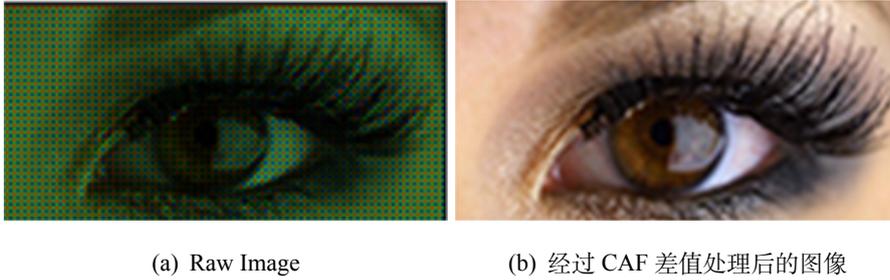


图 3.12: Raw Image 和经过 ISP 差值后的图像

e. 自动白平衡校正 (AWB Correction)：白平衡校正是在不同的场景色温下调整图像中红、绿、蓝的色彩比例来修正图像偏色的过程，经过白平衡校正图像的色彩更接近于人类观察的习惯。在正常状态下感光元件对这三种颜色的放大比例是相同的，白平衡通过改变对三种颜色增益的比例从而修正光源色温给图像真实颜色带来的影响，使图像中白色恢复正常。如果在图像中的蓝、绿、红色光的比例关系是 2：1：1，那么自动白平衡校正模块会将这三种颜色的增益比例设置为 1：2：2。这样通过白平衡校正后所采集的图像中红、绿、蓝色彩的比例才会相同，图像颜色接近真实状态。白平衡的基本原理可用下式来表达：

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \begin{bmatrix} W_R \\ W_G \\ W_B \end{bmatrix}$$

其中 W 为权重参数，由 AWB Monitor 统计采集到的图像的信息后经过计算得到。当该模块失效后，图像色彩会出现明显的偏差，使图像数据的质量严重下降。

f. 色彩校正 (Color Correction)：色彩校正是指在不改变白色的情况下使用相关的算法修正图像内其他颜色的过程，经过该算法处理后的图像可以还原图像内物体的真实颜色。相机进行图像采集时，由于感光元件、白平衡、环境光照

以及人为因素的影响，最终获取的图像往往与真实效果有很大差别。色彩校正算法通过调整数字图像的 RGB 分量值，从而还原图像内物体真实的颜色。色彩校正的基本原理如下：

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

其中，矩阵

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

称为颜色校正矩阵 (Color Correction Matrix, CCM)， M_{ij} 表示颜色校正矩阵内各个分量的值， M_{11} 表示分量 R' 对 R 的影响，而 M_{12} 和 M_{13} 分别表 G' 和 B' 分量对 R 的影响。假设图像传感器完全符合人眼观测图像的习惯，那么显然 $M_{11}=1$ ， $M_{12} = M_{13}=0$ 。由于自动白平衡后图像中的白色已经恢复正常，所以 CCM 不能改变图像内部的白色，因此我们可以得出： $M_{11} = M_{22} = M_{33}$ 。该矩阵一般利用标准色卡的对比实验来得到，它可以在不改变白色的情况下修正图像的其他颜色。为了模拟该模块失效时图像的状态，我们只需在不改变白色和黑色的情况下修改图像中其他颜色。本文通过修改图像的饱和度实现该功能。

g. 后处理模块 (Post Process Module)：后处理模块内部主要包括边缘增强和去除噪声两部分操作。为了实现这两种效果，需要使用图像锐化和图像平滑处理技术。图像锐化处理的目的是突出图像的边缘轮廓，使图像更加清晰。而图像平滑处理的主要目的是修复图像采样系统中各种不确定因素产生的噪声。为了模拟出该模块失效时图像的状态，本文使用图像平滑处理技术来模糊图像的边缘细节，同时在图像的不同位置随机添加噪声。

3、板级模块异常：该类型异常主要由于 PCB 电路板上焊接器件、连接线束及其所负责功能模块失效所引起的异常。在该异常类型中本文主要考虑以下几种异常：主控芯片时序异常、图片缓存异常、线束传输异常给图像采集带来的影响。

(1) 主控芯片时序异常：要得到稳定连续的视频数据流，需要相机内部的各种操作信号的相互协调，从而驱使相机各部分协同工作。对操作信号的产生时间、稳定时间、撤销时间进行控制称为相机的时序控制。相机系统只有在严格的时序控制下才可以采集到理想的数据结果。在时序控制中锁相回路

(Phase Locked Loop, PLL) 提供统一的时钟信号。其他相关的时钟信号可以通过分频技术进行实现。参考相关资料，在相机的时序控制逻辑中常用到 PCLK, VSYNC 和 HERF 这三个信号。其中 PCLK 是像素点的同步时钟信号 [37]。相机通过 PCLK 信号可以确定某个像素点。HERF 是行同步信号，该信号用来确定某行像素，HERF 信号的一个方波内形成的所有 PCLK 信号为同一行像素。VSYNC 是场同步信号，VSYNC 信号一个方波内形成的所有 HERF 信号数据组成一帧画面 (Frame)。例如在一张 640 x 320 的图像中它们之间的关系公式如下：

$$VSYNC = HERF * 320$$

$$HERF = PCLK * 640$$

这三种信号的时序关系图如图3.13所示。

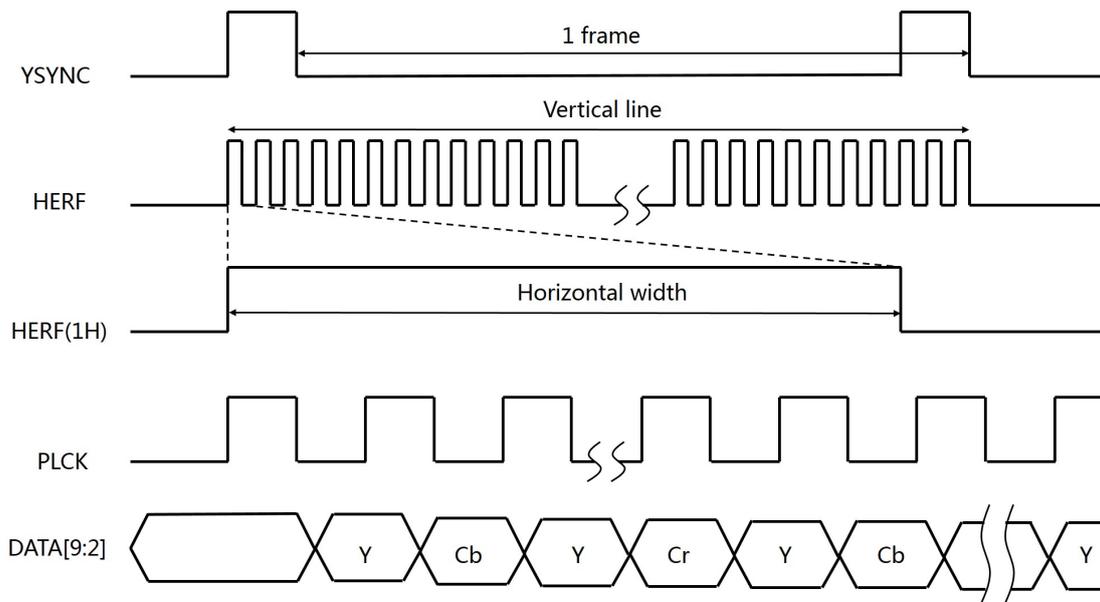


图 3.13: 时钟时序关系图

当图 PCLK、VSYNC 和 HERF 这三个信号发生异常时，会导致图像相位不同步。最终图像数据采集发生错位，导致图像数据显示异常。

(2) 图像缓存异常：在相机图像缓冲区的配合下摄像头采集数据可以以视频流的形式对外输出。相机在采集到一帧一帧的图像后，若要将图像组合成为视频就需要相机将拍摄过程中的图像信息存储到相关的缓存器中。当收到相关指令时，这些被暂存的数据就可以通过相关的接口连续的传递给后端设备进行处理。一般选用随机存取存储器 (Random Access Memory, RAM) 作为板级缓存

器。与其他精细的半导体元件一样，随机存取存储器对外部环境中的静电荷非常敏感。静电会干扰存储器内电容器的电荷，导致数据损坏等故障。本文主要考虑因存储器数据丢失引起图片异常现象。

要准确的理解存储器数据丢失对图像数据的影响，首先需要了解图像数据在内存中的组织形式。在查阅相关资料后，本文对图像数据在内存中的组织形式进行介绍。对于批量 2D 图像数据，图像的维度为：图像数量 (N) × 通道数 (C) × 图像高度 (H) × 图像宽度 (W)。显然，像素点的位置可以通过四维坐标进行精确定位。如坐标为 (n, c, h, w) 的像素点，其中 n 表示图像序号 (0 ~ N-1)，c 表示图像的通道序号 (0 ~ C-1)，h 表示图像行坐标 (0 ~ H-1)，w 表示图像列坐标 (0 ~ W-1)。了解了图片数据在内存中的组织形式，就可以根据这个坐标找到对应的数据。图像数据存储方式如图3.14所示：

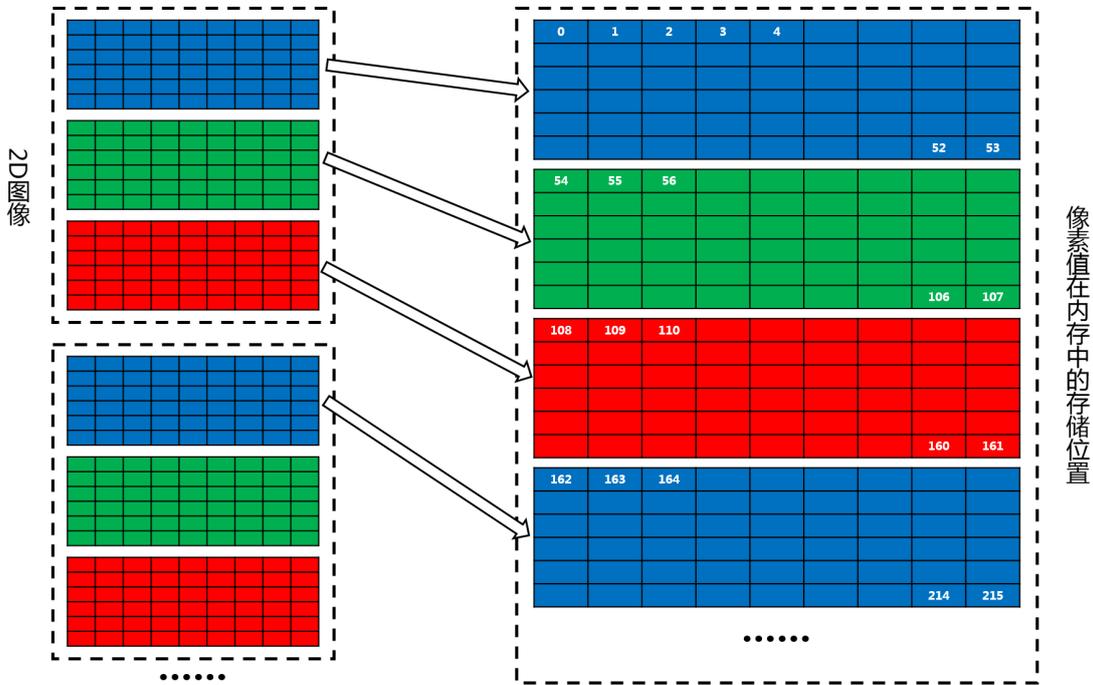


图 3.14: 图像数据在内存中的组织形式示意图

在内存中，第一张图像的第一个像素值存储在内存的某一个位置。以这个位置为起点，图像数据按照从左到右，从上到下的顺序，逐列、逐行、逐通道、逐张图像的将每个像素值存入内存。因此，在存储器发生故障时，主要考虑图像 RGB 矩阵元素丢失情况引起的图像数据异常。

(3) 线束传输异常：图像数据若要对外输出需经过图像输出接口 (Image output Interface) 的 DATA[9: 2] 的 8 个引脚对外进行。查阅 OV3640 的使用手册

可知，相机以 YUV 格式的图像数据对外进行输出 [38]。OV3640 相机数据输出的标准如图3.15所示。但是在实际过程中，这些高速传输的连接线束中会存在错连、虚焊、连接不稳定等问题。这些问题会导致输出图像信息的严重丢失。因此，本文主要考虑线束传输中虚焊、断路或连接松弛这些情况引起的图像数据异常丢失问题。当这种故障发生时，图像内部会大面积色偏、水波纹等状况。

format	YUVV	DATA[9 : 2]	First pixel	First pixel	Second pixel	Second pixel	Third pixel	Third pixel	
		even		Y[7:0]	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]
		odd		Y[7:0]	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]

format	UYVY	DATA[9 : 2]	First pixel	First pixel	Second pixel	Second pixel	Third pixel	Third pixel	
		even		U[7:0]	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]
		odd		U[7:0]	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]

format	YVUY	DATA[9 : 2]	First pixel	First pixel	Second pixel	Second pixel	Third pixel	Third pixel	
		even		Y[7:0]	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]
		odd		Y[7:0]	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]

format	VUYV	DATA[9 : 2]	First pixel	First pixel	Second pixel	Second pixel	Third pixel	Third pixel	
		even		V[7:0]	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]
		odd		V[7:0]	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]

图 3.15: OV3640 输出格式示意图

3.2.2 摄像头 FMEA 分析结果

对车载相机电路进行 FMEA 分析，本文首先对自动驾驶汽车图像数据数据采集通路进行切片并建立模型。本文绘制出了图像数据在自动驾驶汽车内部的流转过程，建立了图像流数据通路并将其分成了不同的功能模块。本文对各个模块的失效模式和失效原因进行分析，并且得到了各故障的发生给采集图像数据带来的不同影响。分析的结果如表3.1所示。

3.3 异常（故障）数据扩增方法

3.3.1 光线场景异常

在该部分本文主要讨论天气因素给图像数据带来的影响以及在不同场景下图像扩增的方法。

表 3.1: 摄像头 FMEA 分析结果表

	失效模式	所属模块	失效原因	故障导致结果
1	雾场景失效	光线场景	摄像头进光量减少	图像灰白化
2	雨场景失效	光线场景	光线传播受阻	亮度下降, 白色雨线
3	镜头水滴遮挡	镜头	光线折射	出现水滴并呈现倒立的的像
4	感光芯片失效	传感器	激光照射发生故障	图像出现亮线亮斑
5	黑电平校正异常	ISP	黑电平校正模块失效	对比度下降, 色偏
6	镜头校正异常	ISP	镜头校正模块失效	图像出现渐晕效果
7	坏点校正异常	ISP	坏点校正模块失效	图像出现异常像素点
8	CFA 插值异常	ISP	CFA 插值模块失效	像素点 RGB 值丢失
9	自动白平衡校正异常	ISP	自动白平衡校正模块失效	颜色出现偏差
10	色彩校正异常	ISP	色彩校正模块失效	除白色外颜色出现偏差
11	后处理模块异常	ISP	后处理模块失效	图像模糊且有噪点
12	主控芯片时序异常	板级模块	时序逻辑信号异常	图像行像素丢失
13	图片缓存异常	板级模块	图像缓存区损坏	图像矩阵数据丢失
14	线束传输异常	板级模块	数据连接线束损坏	图像信息丢失

3.3.1.1 雾场景异常

根据上文对雾的形成原因、光在雾天情况下的传播规律和大气散射模型, 本文分析总结了雾天图像成像降质退化的机理。因此, 本文定性的总结了雾天图像成像的规律。具体规律如下:

1、在雾天条件下, 由于光线在传播中发生散射, 目标物体的反射光线强度在传播过程中发生明显衰减。且目标物体距离相机镜头越远光线衰减越明显, 最终进入镜头的光线强度越弱。

2、在雾天情况下, 环境光(太阳光、地面反射的光、天空散射的光等)也参与图像的成像过程。当光线传播距离较长时, 会有越来越多的环境光参与进来。因此距离镜头越远, 环境光越强。

3、由于入射光在传播过程中产生衰减。因此, 图像中的内容细节将会变得更加模糊, 特征更加难分辨, 且图像的清晰度下降。

4、光线经过空气中悬浮的水滴折射和散射会导致图像整体呈现出灰白化的效果。在视觉上图像表现为色彩饱和度不足。

5、在频域特性上, 图像中的细节边缘信息属于高频分量。在雾天场景下图像较为模糊, 因此, 图像的高频分量减少, 低频分量增加。

6、在空域特性上, 雾天场景下的图像的直方图灰度级趋向于平均值, 且集中分布, 动态范围小。因此, 图像整体对比度下降。

根据雾天场景下的这些成像规律和大气散射模型, 本文对雾场景图像数据

的扩增方法进行定量分析并总结出以下几点扩增规则：

1、对图像进行高斯滤波，使图像细节模糊化，降低其清晰度和边缘细节信息。降低图像中的高频分量，同时增加图像的低频分量。

2、降低图像的饱和度，降低图像色彩的鲜艳程度，使图像色彩变得暗淡。

3、根据摄像头安装位置，可以大致计算出每个像素区域与镜头位置的距离。由于自动驾驶摄像头采集数据一般在公路上进行，因此，图像内自上而下与镜头距离逐渐变近。同时根据成像规律可知，一般情况下图像内像素点代表的场景区域与相机镜头间的距离从中间向两边逐渐增加。因此，本文可以近似计算出图像的雾化中心和像素点位置与镜头的相对距离。

4、根据大气散射模型公式在图像中大致计算出雾化中心、散射系数，透射率，大气光强度。将其代入公式得到物体反射光线强度随距离的变化规律，并将其作用于图像的每个像素点，最终得到图像雾化以后的结果。

3.3.1.2 雨场景异常

根据上文分析，在雨天场景下，降落的雨滴在成像范围内呈现随机分布状态，且雨滴对光线主要表现为遮挡作用，同时雨滴下落时的形状通常为细长的纺锤状。因此，雨天场景下图像中经常出现细长型白色雨线噪声。结合相关资料，我们定性的总结了雨天图像成像的规律。具体规律如下：

1、在大气清澈透明（无雾气、大型尘埃颗粒等）的前提下，雨天图像的能见度较远、图像内物体的轮廓较为清晰。

2、雨天图像中，由于雨滴对光线的遮挡作用，成像范围内的雨滴呈现为白色效果，即大量的白色雨线噪声。图像内单条雨线噪声的遮挡面积与雨滴与相机间的距离呈反比，因此，距离较近的雨滴对光线主要起到了遮挡作用。

3、雨天图像中，雨线噪声的数量与瞬时降雨量呈正比，同时雨线噪声的形态会受到外力（风力、空气阻力等）影响。雨线噪声在图像内部呈现为随机分布，无具体规律。

4、在频域特性上，图像中由于雨线噪声的出现，使雨线噪声区域的灰度值变化较为明显（噪声边缘与背景的过渡区域）。这将导致雨天场景下图像的高频分量增加。

5、由于雨滴和乌云对光线的遮挡和散射作用，雨天场景下图像的亮度降低。

根据雨天情况下的这些成像规律，本文对雨天场景的图像数据扩增方法总结如下：

1、降低原图像亮度，使图像变得暗淡。

2、由于，雨滴在空中呈随机分布，无具体规律，因此我们可以根据雨量的大小生成单通道雨线噪声图像。

3、将单通道雨线噪声图像扩展为三通道图像，以便与原图像进行混合。

4、将雨线噪声图像与原图像加权合成，最终得到雨天场景后的图像结果。

3.3.2 摄像头异常

根据上文所述，本文依据摄像头的结构，对图像采集通路中相机可能发生异常的点进行分类。在该部分本文主要讨论各个故障点发生异常时对图片产生的影响，以及各种异常状态下图像数据的扩增方法。

3.3.2.1 镜头异常

由于自动驾驶汽车的相机通常安装在车辆前挡风玻璃后面，因此，汽车挡风玻璃上的物体会干扰相机的视线导致自动驾驶汽车发生错误。因此，在镜头异常方面，本文主要考虑挡风玻璃上的水滴对图像产生的干扰。

根据上文所述，我们定性的总结了镜头异常图像成像的规律。具体规律如下：

1、根据相关文献和流体表面张力和粘度的相关知识，水滴在挡风玻璃上的二维图像表现为近似的椭圆状。

2、受表面张力的影响，水滴的表面为凸椭球状。同时，由于水滴是透明的光介质，因此可将水滴视为透明材料制成的光学凸透镜。

3、由基础光学知识可知，场景中物体与水滴光心的距离大于两倍焦距，而自动驾驶汽车相机的安装位置一般距离光心在一倍焦距与两倍焦距之间。因此物体的反射光线经过折射在图像内最终呈现倒立缩小的像。

4、考虑凸透镜对光线的汇聚作用，水滴内的图像亮度要比原图像的亮度高。

5、在频域特性上，由于图像中水滴的出现，使水滴区域内的亮度、灰度值与周围区域变化明显。因此，挡风玻璃水滴图像内的高频分量增加。

根据挡风玻璃上水滴的这些成像规律，本文对镜头异常的图像数据扩增方法总结如下：

1、根据图像内水滴的数量，随机选择若干个水滴中心。

2、以水滴为中心，在原图上截取合适大小的图像作为水滴成像区域。

3、依据水滴的形状和相关的光学知识，将水滴成像区域的图像截取为椭圆状，同时将其缩小为合适大小，并进行 180° 旋转。

4、将调整好的水滴按照水滴中心坐标插入原图，得到挡风玻璃水滴图像。

3.3.2.2 对焦马达异常

对焦可以保证照片清晰。当对焦马达失效后，图像将会失焦，导致图像模糊。根据上文所述，本文定性的总结了失焦图像成像的规律。具体规律如下：

1、在失焦状态下，图像传感器并不在景深范围之内。因此，在这种状态下物体的像并不能清晰的呈现在图像传感器上。

2、在未合焦状态下，由于整个画面处于虚焦状态，类似于对图像进行整体滤波后的结果，所以图像内各像素之间的灰度值分布更为均匀。

3、在频域特性上，图像失焦，导致图像中各个像素点之间的颜色变化趋于平缓，图像内整体色彩较均匀。因此图像的低频分量较高。

据此，根据失焦图像的这些成像规律，本文对对焦马达异常的图像数据扩增方法总结如下：

1、对图像采用平滑处理，分别使用高斯滤波和均值滤波的方式模拟失焦图像效果。

2、使用不同大小的卷积核作用于原图像，得到镜头组在不同位置失效时失焦图像的结果。

3.3.2.3 传感器异常

感光芯片是相机的核心组成部分，它利用光电器件的光电转换功能。在传感器异常方面，本文主要考虑由激光照射引起图像传感器故障问题。

根据上文所述，本文分析了激光对传感器损伤的图像以及传感器的电路原理。根据这些知识，本文定性的总结了由激光照射引起图像传感器故障后采集图像成像的规律。具体规律如下：

1、由于图像传感器的电路结构，在激光损坏传感器后，经过损坏的图像传感器采集的图像数据一般会出现串扰亮线。因此，我们在图像中随机选取一个像素点，对该像素点所在行列的像素进行颜色处理，生成随机颜色的十字线。

2、在激光损坏传感器后，经过损坏的图像传感器采集的图像内部会出现饱和光斑。因此我们在图像中随机选取若干个中心，并根据该中心对亮斑区域进行填充。

3、经过损坏图像传感器采集的图像会出现色彩偏移，因此，我们以串扰亮线相交所在位置为中心生成竖条状渐变色，并将其与原图像融合。

4、当图像传感器发生严重损坏（如融化）后会出现区域性损伤，造成该区域无法成像等问题。在参考了相关异常图像后，本文发现融化区域以串扰亮线

相交点为中心呈现类抛物线型。因此，本文在串扰亮线交点处随机生成类似抛物线形状融化区域，并将该区域设置为黑色。

3.3.2.4 ISP 校正异常

在对图像信号处理器的整体处理流程进行分析后，本文根据各个功能模块在 ISP 流水线上的位置和所负责功能，将不同模块的失效图像归纳为以下六种异常：黑电平校正异常、镜头校正异常、CFA 插值异常、自动白平衡校正异常、色彩校正异常、后处理模块异常。在该部分，本文重点讨论 ISP 模块中各个功能点发生异常时，异常数据的数据扩增方法。

a、黑电平校正异常

黑电平校正就是为了消除由于相机内部存在潜在电流引起的感光芯片异常成像而采取的手段。根据上文所述，本文分析了黑电平校正的基本原理，定性的总结出黑电平校正异常图像成像的规律。具体规律如下：

- 1、由于该流程在 ISP 流水线中处于 CFA 插值前，首先将其转变为 Raw 图像。
- 2、对原图像进行阈值处理，对像素点 RGB 值高于限定值的像素点统一设置为限定值，再对所有像素点的灰度值进行统一计算，从而抵消黑电平校正的作用。
- 3、黑电平校正模块的失效会影响后续 ISP 各模块的处理，导致画面整体出现偏绿或者偏红，对比度下降的效果。因此我们生成与图像相同大小的红绿两个蒙版，并将其与原图像进行混合，使图像颜色呈现偏红或偏绿两种状态。
- 4、将 Raw 图像转换为标准的 RGB 图像。

b、镜头校正异常

由于镜头本身和光线传播的规律，当相机视场角的增大时会导致进入镜头的斜光束减少，造成图像的边角失光。这样采集的图像中间区域亮度较高，边缘区域亮度较低。当 ISP 流水线中镜头校正模块失效后图像就会出现渐晕效果。本文分析了其成像的基本原理后，定性的总结了因镜头校正模块失效导致的图像成像异常的数据扩增规律。具体规律如下：

- 1、由于该流程在 ISP 流水线中处于 CFA 插值前，首先将其转变为 Raw 图像。
- 2、根据图像的尺寸生成阴影区域的内半径大小，在本文中我们将半径设置为图像长宽差值的二分之一。

3、生成阴影颜色深度比例矩阵，对小于阴影区域内半径的图像区域不做处理，对大于阴影区域内半径的部分，根据像素点坐标位置与图像中心的位置距离设置该点的颜色深度比例，距离越远颜色越深，比例系数越小。

4、将原图像与阴影颜色深度比例矩阵做乘法，模拟生成由于镜头校正模块失效后引起的异常图像。

5、将 Raw 图像转换为标准的 RGB 图像。

c、坏点校正异常

因制作工艺的原因，图像传感器上像素阵列存在缺陷，导致光电信号转换的过程中出现错误，从而会引起图像像素信息的异常。根据上文分析，坏点校正异常数据的具体扩增方法如下：

1、由于该过程在 ISP 流水线中处于 CFA 之前，因此先将图像变为 Raw Image 进行处理。

2、选取图像中的 ROI，在该区域内随机选取若干像素点，并将其灰度值设置为零，以模拟该位置像素点损坏时的效果。

3、将 Raw 图像转换为标准的 RGB 图像。

d、CFA 插值异常

为了获得完整色彩信息的图像，需要对 Bayer 图像进行插值。这个插值的过程就被称为 CFA 插值。当该模块失效时，图像会以 RAW 图像的形式进行后续的处理工作。在分析了 Bayer 图像及 Bayer 滤波的基本工作原理后，本文定性的总结了因 CFA 插值模块失效导致的图像数据的扩增规律。具体规律如下：

1、将图像长度宽度摄者为四的倍数。因为一般情况下图像传感器为了符合最小的拜耳单元，它的长宽都是四的倍数。

2、由于每个像素点只能记录 RGB 中单一通道的值，因此本文假设图像格式为 Bayer RG 格式。根据 Bayer RG 图像的格式的特点设置该位置像素点数值。

3、若当前位置为 Bayer 滤光片中 R 位置点，则把 G 和 B 两个通道的数值都设置为零；若当前位置为 Bayer 滤光片中 G 位置点，则把 R 和 B 两个通道的数值设置为零；若当前位置为 Bayer 滤光片中 B 位置点，则把 R 和 G 两个通道的数值都设置为零。

4、将图像的通道数设置为一，得到 Raw 图像。

e、自动白平衡校正异常

在理想情况下图像传感器内感光元件对颜色的放大比例是相同的。但是在不同的光源场景下，图像传感器采集的数据会产生色彩差异，导致红蓝绿三种光的比例发生偏差。因此，本文模拟自动白平衡失效时的图像只需随机调整图

像内红绿蓝三种色光的比例成分即可。自动白平衡校正异常时图像数据的扩增方法如下所示：

1、由于要对图像内三种色光的比例进行调整，因此首先对原图像进行通道分离处理，将图像的 RGB 通道进行分离提取。

2、改变图像内三种色光的比例关系。本文分别对原图像的 R 通道、G 通道、B 通道进行随机调整，将各通道的数值随机改变为原来的 0.7-1 倍，从而使图像内三种色光的比例不一致。

3、将改变后的 RGB 通道进行合并，得到自动白平衡校正失效后的图像。

f、色彩校正异常

色彩校正矩阵 CCM 通常在自动白平衡校正 (AWB) 之后。AWB 把白色校正了，图像内其他色彩也跟着发生改变，这会导致图像内物体的色彩发生偏差。CCM 可以在保持白色不变的情况下，把其他色彩还原到真实的状态。由上文可知，若想实现在保持图像白色不变的情况下修改图像内其他颜色的效果，可以对图像的饱和度进行处理。因此，色彩校正模块异常时图像数据的具体扩增方法如下：

1、对图像进行归一化处理，并将其像素值且转换为浮点型。

2、由于要对图像的饱和度进行操作，我们对图像的色彩空间进行转换，将原图像的 RGB 色彩空间转为 HLS 色彩空间。

3、随机生成图像饱和度比例，将图像饱和度 (S) 色彩空间对应的原饱和度数值与该比例乘积，对于 S 色彩空间大于一的部分全部截取。

4、将 HLS 色彩空间图像转化为 RGB 图像并进行逆归一化处理，得到颜色异常的错误图像。

g、后处理模块异常

后处理模块的功能主要包括边缘增强和去除噪声两种功能。为了实现这两种效果，可以使用图像锐化和图像平滑处理技术。由上文分析可知，为了模拟出该模块失效时图像的异常情况，我们可以使用图像平滑处理技术来模糊图像的边缘细节，同时在图像上随机添加噪声达到预期效果。根据该模块负责功能，本文总结出后处理模块失效后异常图像数据的扩增方法，具体的扩增方法如下：

1、对原图像进行模糊处理，该过程用来抵消后处理模块中图像锐化的作用。

2、对模糊后的图像添加随机噪声，本文采用添加椒盐噪声或高斯噪声对图像进行修改，该过程可以抵消后处理模块中的去除噪声两种功能。

3.3.3 板级模块异常

根据上文所述，板级模块类型异常主要是由于 PCB 电路板上焊接器件、连接线束及其所负责功能模块失效所引起的。在这种异常类型中本文主要考虑主控芯片时序异常、图片缓存异常、线束传输异常给图像采集带来的影响。在该部分，本文主要讨论板级模块中各个功能器件以及相应功能点发生异常时，异常图像数据的扩增方法。

3.3.3.1 主控芯片时序异常

根据上文所述，PCLK，VSYNC 和 HERF 这三个信号可以完成对摄像机的时序逻辑的控制，使相机内部各个组件有条不紊地协同进行工作。其中 VSYNC 信号可以控制图像数据帧的同步，在同一帧画面内 HERF 信号可以控制每一行像素数据的同步。在该异常类型中本文主要考虑由 HERF 信号出错，导致像素数据行缺失相位不同步、数据采集有错位引发的问题。主控芯片时序发生错误时异常图像数据的扩增方法如下：

- 1、生成一张黑色图像，图像内黑色部分是由于时序错乱而导致图像数据的未填充部分。图像大小和类型与原始图像一致。
- 2、随机生成 HERF 时序异常发生的次数，异常发生时所处行数以及每次异常所持续的时间（行数）。
- 3、根据异常发生时所处的行数和异常发生次数，使用原图像的像素点对异常图像进行填充，得到 HERF 时序异常发生时的异常图像数据。

3.3.3.2 图片缓存异常

根据上文可知，摄像头在采集到一帧一帧的图像后，若要将图像组合成为视频流数据就需要相机将拍摄过程中的图像信息存储到板级缓存区中。当缓存区发生故障时将会产生异常数据，根据数据在内存中的组织形式，本文将缓存的损坏形式分为两种：缓存区的随机损坏和缓存区的块级损坏。本文总结了图片缓存异常发生时，异常数据的扩增方法：

a、缓存区随机损坏：

- 1、根据图像在内存中的组织形式，本文首先对 RGB 图片进行通道分离，将原图像分解为 R 通道灰度图、G 通道灰度图、B 通道灰度图。
- 2、随机生成 R、G、B 三个内存损坏矩阵，每个矩阵大小与原图像尺寸一致，矩阵中元素为零表示该位置内存损坏。

3、对内存损坏矩阵进行二值化阈值处理，对于大于阈值的位置我们将其值设置为 255。

4、将原图像的 R 通道灰度图、G 通道灰度图、B 通道灰度图分别与经过处理的内存损坏矩阵按位与运算，将内存损坏的点设置为零。

5、将三个经过与运算的灰度图像进行融合，生成缓存故障发生后的异常图像。

b、缓存区块级损坏：

1、根据图像在内存中的组织形式，我们首先对 RGB 图片进行通道分离，将原图像分解为 R 通道灰度图、G 通道灰度图、B 通道灰度图。

2、随机选取内存坏块起始点以及坏块的长度。根据这些参数分别对原图像的 R 通道灰度图、G 通道灰度图、B 通道灰度图进行处理，将损坏位置的像素值设为零。

3、将三个经过与运算的灰度图像融合，生成缓存故障发生后的异常图像。

3.3.3.3 线束传输异常

根据上文所诉，图像数据对外输出需要通过 DATA 接口的 8 个引脚。根据 OV3640 的使用手册我们了解到，相机系统以 YUV 格式的图像数据对外进行输出。因此，在该异常类型中本文主要考虑由于线束传输中断路、虚焊、连接不稳定这些情况引起 DATA 接口引脚故障导致的图像数据异常。线束传输异常图像数据的扩增规则如下所示：

1、根据图像在数据传输中所遵守的数据格式，首先将图像转化为 YUV 格式图像。

2、对 YUV 格式图像进行 Y, Cr, Cb 通道分离。

3、创建数据传输通道损坏矩阵，同时随机生成一个 255 以内的数字，将其转换为八位二进制后，零位置表示该引脚损坏，一位置表示该引脚传输正常。

4、根据随机生成的数字对通道损坏矩阵的每个点进行赋值。

5、将单通道图像与通道损坏矩阵做与运算，得出对应通道的传输后的数据。

6、将经过与运算的三通道进行融合，生成线束传输异常发生后的异常图像。

3.4 数据扩增的实现

通过正常情况下采集的数据生成异常状态时采集的图像数据是本文的重点部分。异常数据的生成关系到目标识别网络的识别结果和自动驾驶系统功能安全性检验的结果，这是本文工作中的核心。本节主要介绍异常数据的生成过

程，考虑到算法过程的复杂性和数据生成过程的可理解性，我们采用算法和伪代码的形式对数据扩增的实现过程进行介绍。

3.4.1 光线场景异常数据扩增

本部分主要介绍光线场景异常数据扩增的实现过程。该过程主要包含两种非正常的天气状况：1、雾天场景异常数据扩增实现过程。2、雨天场景异常数据扩增实现过程。下面我们分别对这两种情况进行介绍。

Algorithm 1: 生成雾场景图像

Data: Image to be processed, Ambient brightness: A ,
scattering coefficient: β

Result: Fog scene image

- 1 Image normalization processing;
- 2 $size = \text{math.sqrt}(\max(\text{image.shape}[0], \text{image.shape}[1]));$
- 3 Select the $1/4$ position above the image center as the atomization center;
- 4 **foreach** row in $\text{image.shape}[0]$ **do**
- 5 **foreach** col in $\text{image.shape}[1]$ **do**
- 6 $d = -0.04 * \text{math.sqrt}((j - \text{center}[0]) ** 2 + (k - \text{center}[1]) ** 2) + size;$
- 7 $td = \text{math.exp}(-\beta * d);$
- 8 $\text{img_f}[j][k][:] = \text{img_f}[j][k][:] * td + A * (1 - td);$
- 9 Image inverse normalization processing;
- 10 return $\text{img_f};$

1、雾天场景异常数据扩增实现过程，该过程主要实现对原图像加雾的效果。这种异常数据扩增实现的核心是根据大气散射模型，计算出光线到达各个像素点的强度。由大气散射模型可知，光线的强度与环境光强度、传播的距离和散射系数有关。其中，环境光强度为算法的输入值，通过改变该数值可以调整图像整体的亮暗。同样，散射系数也是算法的输入值，在数据扩增的过程中我们可以根据该系数调整雾的浓度。在这些控制系数中光线传播的距离不能直接获得。因此，结合自动驾驶汽车在实际行驶过程中的规律，本文采用了一种巧妙的方法来大致计算光线传播的距离。由于自动驾驶汽车的摄像头一般安装在挡风玻璃后的后视镜前方，所以通常情况下由该位置相机采集的图像中，图像的中心位置上 $\frac{1}{4}$ 处像素点的视野距离最远。可以认为该点处光线的传播距离最远，因此该点的雾化程度最大。所以在计算过程中，我们将该点作为雾化中心。在现

实世界中，光线到达相机的距离也由该点向周围延伸越来越远。因此，在数字图像矩阵中，可以通过计算像素点与雾化中心的距离近似代替光线的传播距离，这样我们就使用大气散射的数学模型来进行数据扩增的实现。该部分异常数据扩增的具体实现方法如算法1所示。

2、雨天场景异常数据扩增实现过程，该过程主要通过 `get_noise`、`rain_blur`、`add_rain` 三段代码进行实现。其中 `get_noise` 的作用为生成雨线模糊核心。而 `rain_blur` 方法的作用为生成对雨线模糊核心进行运动模糊，模仿雨滴下落的姿态，最后 `add_rain` 方法负责将运动模糊后的雨线图像与原图像进行融合。该部分异常数据扩增的具体实现如算法2所示。

Algorithm 2: 生成雨场景图像

```

1 noise = np.random.uniform(0, 256, img.shape[0:2]);
2 noise[np.where(noise < (256 - value))] = 0;
3 k = np.array([[0, 0.1, 0], [0.1, 8, 0.1], [0, 0.1, 0]]);
4 noise = cv2.filter2D(noise, -1, k);
5 Generate a motion blur kernel;
6 rainline = cv2.filter2D(noise, -1, kernel);
7 result = cv2.addWeighted(image, alpha, rainline, 1 - alpha, 1);
8 return result;

```

3.4.2 摄像头异常数据扩增

本部分主要介绍车载摄像头部分发生异常时异常数据扩增的实现过程。车载摄像头部分主要包含三个子部分：1、镜头组部分。2、感光芯片部分。3、图像信号处理器部分。其中，镜头组部分异常我们又将其分为两种情况 a、镜头异常（挡风玻璃水滴）。b、对焦马达失效异常。感光芯片部分的异常情况主要为激光照射导致传感器失效异常，而图像信号处理器部分的异常，我们参考 ISP 的工作流程的先后情况，将其分为以下七种情况：a、黑电平校正异常。b、镜头 j 校正异常。c、坏点校正异常。d、CFA 插值异常。e、自动白平衡校正异常。f、色彩校正异常。g、后处理模块异常。下面我们分别对这三部分组件发生异常时异常数据的生成过程进行介绍：

1、镜头组部分异常数据扩增过程。

a、镜头异常（挡风玻璃雨滴）异常数据扩增实现过程，该过程主要模拟雨滴在镜头上的效果。本文参考了凸透镜成像基本原理以及自动驾驶汽车中摄像

Algorithm 3: 生成挡风玻璃雨滴图像

```

1 Reduce image brightness;
2 foreach num in number of raindrops do
3   Generate raindrop coordinates randomly;
4   roi = the square area around the raindrop coordinates;
5   roi rain = cv2.resize(roi, (scale*roi.hight, scale*weight));
6   cv2.flip(roi rain, 0, roi rain);
7   Intercept the processed square area as a raindrop image;
8   foreach The column of pixels in range(raindrop_image.height) do
9     foreach The row of pixels in range(raindrop_image.weight) do
10      Replace the pixel value of the corresponding position of the
          original image to the raindrop image;
11 return result;

```

头、挡风玻璃、雨滴和现实场景的实际相对位置进行数据扩增。该部分异常数据扩增的具体实现如算法3所示。

Algorithm 4: 生成对焦马达故障图像

```

1 if Fuzzy type == Homogeneous then
2   mean_blur = cv2.blur(img, (Fuzzy_level * 2 - 1, level * 2 - 1));
3   return mean_blur;
4 if Fuzzy type == Gauss then
5   gauss_blur = cv2.GaussianBlur(img, (Fuzzy_level * 2 - 1, level * 2 - 1),
        0, 0);
6   return gauss_blur;

```

b、对焦马达失效异常数据扩增实现过程，该过程主要模拟镜头组对焦马达失效时图像模糊的结果。该部分异常数据扩增的具体的实现如算法4所示。

2、感光芯片部分异常数据扩增过程，该部分异常主要考虑受激光照射后感光芯片损坏后图像采集的效果。结合图像传感器相关知识以及激光对图像传感器的损伤程度，本文将异常数据分为三种类型并得出了三种类型下异常图像的表现形式：

a、图像传感器轻度损伤，在激光的照射下发生感光器件损伤，在损伤点附

近产生亮色十字干扰线。

b、图像传感器中度损伤，由于激光的照射使图像传感器损坏点附近的感光器件出现暂时性损伤，发生色彩偏移。

c、图像传感器中严重损伤，在激光的照射下感光芯片发生融化，感光元件出现大面积失效导致图像出现无效区域。该部分异常数据扩增的具体实现如算法5所示。

Algorithm 5: 生成感光芯片故障图像

```

1 Generate X and Y coordinates of laser irradiation points randomly;
2 Generate a cross bright line according to this position;
3 if Spots == True then
4   foreach i in number of spots do
5     cv2.circle(result, center, spot radius, spot color, -1);
6 if Deviation == True then
7   img_deviation = vertical color deviation band;
8   result = cv2.addWeighted(result, 1, img_deviation, 0.6, 0)
9 if SevereDamage == True then
10  foreach i in weight of damage do
11    result[damage area, :] = 0;
12 return result;
```

3、图像信号处理（ISP）部分异常数据扩增过程。

a、黑电平校正异常，该过程主要模拟 ISP 流水线上黑电平校正功能模块失效后的效果。该部分异常数据的扩增过程算法主要实现了黑电平校正的逆过程，即对图像中每个像素点进行阈值操作和数值运算。结合相关的参考资料，本文对图像扩增后整体的色彩效果进行了改变，使得扩增后的图像更接近真实情况。该部分异常数据扩增的具体实现如算法6所示。

b、镜头校正异常，该过程主要模拟 ISP 流水线上镜头校正功能模块失效后采集图像的效果。该算法中的核心部分为生成阴影颜色深度比例矩阵。该矩阵根据像素点距离图像中心距离的远近控制矩阵中元素的比例系数，从而模拟镜头校正失效时图像的状态。阴影颜色深度比例矩阵分为两部分，以图像中心为圆心的特定距离为半径的矩阵元素的比例系数为一，图像中这部分像素不会发生改变。其余部分为渐晕现象影响部分，这部分像素点距离图像中心位置越远，

Algorithm 6: 生成黑电平校正异常图像

```

1 Convert the Image to Raw-image;
2 t, rst = cv2.threshold(image, 240, 255, cv2.THRESH_TRUNC);
3 tmp = rst[:, :] + 15;
4 result = cv2.cvtColor(tmp, cv2.COLOR_BAYER_BG2BGR);
5 mask = np.zeros(result.shape, result.dtype);
6 if Deviation == True then
7     | The image was processed with a mask to simulate the effect of color
8     | offset;
9 return result;

```

矩阵元素的值越小。镜头校正异常图像的具体实现如算法7所示。

Algorithm 7: 生成镜头校正异常图像

```

1 Convert the Image to Raw-image;
2 According to the position of the pixel in the image, generate the shadow color
  depth ratio matrix;
3 foreach row in image.shape[0] do
4     | foreach col in image.shape[1] do
5         | if The distance is less than the vignetting radius then
6         |     | continue;
7         | if The distance is greater than the vignetting radius then
8         |     | The pixel point at this position is multiplied with the
9         |     | corresponding position element of the proportionality matrix;
10 result = cv2.cvtColor(image, cv2.COLOR_BAYER_BG2BGR);
11 return result;

```

c、坏点校正异常，该过程主要模拟 ISP 流水线上坏点校正功能模块失效后的效果。结合图像传感器在现实中的生产工艺特点，本文首先在图像中选取一块坏点存在区域从而模仿制造过程中图像传感器上坏点集中分布的特点。该部分异常数据扩增的具体实现如算法8所示。

d、CFA 插值异常。该过程主要模拟 ISP 流水线上 CFA 插值功能模块失效后采集图像的效果，本文中假设图片为拜耳格式 (RGGB)。此时只需要根据像素

Algorithm 8: 生成坏点校正异常图像

```

1 Convert the Image to Raw-image;
2 Select areas of bad spots in the image randomly;
3 while  $i \leq \text{Number of bad pixel}$  do
4     Generate bad point types randomly;
5     if  $\text{type} == \text{black}$  then
6         Select a pixel in the bad spot area randomly and change its value to 0;
7     if  $\text{type} == \text{white}$  then
8         Select a pixel in the bad spot area randomly and change its value to
           255;
9 result = cv2.cvtColor(image, cv2.COLOR_BAYER_BG2BGR);
10 return result;

```

点在图像中的位置，改变其他两个通道的灰度值为零。最后将图像转换为单通道图像，就可以得到 CFA 插值模块失效后图像的状态。该部分异常数据扩增的具体实现如算法9所示。

Algorithm 9: 生成 CAF 插值异常图像

```

1 Change the length and width of the image to a multiple of 4;
2 image[0:row:2, 0:col:2, 0:2] = 0, 0;
3 image[0:row:2, 1:col:2, [0, 2]] = 0, 0;
4 image[1:row:2, 0:col:2, [0, 2]] = 0, 0;
5 image[1:row:2, 1:col:2, 1:3] = 0, 0;
6 Convert the image to a single channel image;
7 return result;

```

e、自动白平衡校正异常，该过程主要模拟 ISP 流水线上自动白平衡校正功能模块失效后的采集图像效果。结合自动白平衡校正的原理，本文在该部分异常数据的扩增过程中主对图像中的三基色混合比例进行调整，从而模拟图像在不同光源场景下采集后未进行白平衡校正的状态。该部分异常数据扩增的具体实现如算法10所示。

f、色彩校正异常，该过程主要模拟 ISP 流水线上色彩校正功能模块失效后的效果。由于色彩校正必须在不改变图像中黑色与白色的像素点的情况下调整

Algorithm 10: 生成自动白平衡校正异常图像

```

1 Image normalization processing;
2 Separate the RGB channel of the image;
3 b[:, :] = b[:, :] * random.uniform(0, 1);
4 g[:, :] = g[:, :] * random.uniform(0, 1);
5 r[:, :] = r[:, :] * random.uniform(0, 1);
6 Merge RGB channel images to generate a new image;
7 Image inverse normalization processing;
8 return result;

```

其他颜色的显示效果，因此，本文采用了一种巧妙地方法对图像中除黑色或白色的像素点数值进行调整。在色彩概念中饱和度是指彩色的纯度，纯度越大，饱和度就越高。黑白色以及它们之间的过度颜色都不属于彩色范围，所以调整图像的饱和度同样不会对黑色和白色产生影响。因此，在进行色彩校正图像数据扩增的过程中，本文选择对颜色的饱和度进行调整。该部分异常数据扩增的具体实现如算法11所示。

Algorithm 11: 生成色彩校正异常图像

```

1 Image normalization processing;
2 hlsImg = cv2.cvtColor(Image, cv2.COLOR_BGR2HLS);
3 Separate the HLS channel of the image;
4 Adjust the image saturation channel S;
5 rgbImg = cv2.cvtColor(hlsImg, cv2.COLOR_HLS2BGR);
6 Image inverse normalization processing;
7 return result;

```

g、后处理模块异常，该过程主要模拟 ISP 流水线上后处理功能模块失效后的效果。本文对后处理模块所负责的功能（锐化处理，平滑处理）进行逆处理。在后处理模块异常数据的扩增过程中必须注意逆处理操作的顺序性。如果将图像锐化处理的逆过程（平滑处理）放在图像平滑处理的逆过程（向图像中添加噪声）之后，那么图像平滑处理的逆过程对图像质量的影响效果将显著降低。因此，本文在实现该部分异常数据的扩增的过程中首先对待处理图像进行平滑处理，随后在向图像中添加噪声。该部分异常数据扩增的具体实现如算法12所示。

Algorithm 12: 生成后处理模块异常图像

```

1 Image processing by means of blur;
2 foreach row in range(image.shape[0]) do
3   foreach col in range(image.shape[1]) do
4     Add the random noise to the image;
5 return result;

```

3.4.3 板级模块异常数据扩增

本部分主要介绍图像采集设备中由于板级模块的异常问题导致的异常数据扩增的实现过程。图像采集设备中板级模块主要包含三个部分：1、主控芯片部分。2、图像缓存部分。3、数据传输部分。根据不同子部分所负责的功能以及常见的异常，本文将扩增的异常数据分为三种类型：1、主控芯片时序异常。2、图像缓存器损坏异常。3、数据传输线束断开异常。下面我们分别对这三种情况进行介绍：

Algorithm 13: 生成主控芯片时序异常图像

```

1 foreach num in range(errorNum) do
2   error.append(int(random.uniform(0, row)));
3 foreach num in range(errorNum) do
4   Generate error duration randomly;
5   if error[num] + error duration > image.row then
6     break;
7   Fill the pixels from the original image into the result image;
8 Fill the remaining rows of the original image into the result image;
9 return result;

```

1、主控芯片时序异常数据扩增过程，该过程主要模拟主控芯片时序信号出错时车载摄像头所采集图像效果。本文主要考虑信号量 HERF 和 PCLK 出现异常时图像的状态。在实际的数据采集过程中，会出现相邻线路之间存在耦合导致 HERF 不能迅速拉升至 90% 的区域。在发生该种情况时，HERF 信号的缺失会导致同一帧数据内缺少对应的数据行，从而导致一帧画面内的相位不同步，最终像素点出现错位，导致采集的图像数据异常。为了模拟由 HERF 和 PCLK 信

号量导致像素点出现错位的情况，本文随机选取图像中的像素行和像素点进行删除，并将该行后的像素依次向前补齐。图像中空出的位置用黑色代替。该部分异常数据扩增的具体实现如算法13所示。

2、图像缓存区损坏时异常数据扩增过程。该过程主要模拟图像缓存区损坏时车载摄像头所采集图像效果。根据缓存区损坏的特点，我们将其分为 a、图像缓存区随机损坏；b、图像缓存区块级损坏两种情况。图像缓存区损坏的异常数据扩增过程中，缓存区损坏矩阵的生成是该算法的关键。在图像缓存区随机损坏时，本文首先生成和图像大小一致的缓存区矩阵，并在其中随机选取坏点将其值设置为零。而在图像缓存区发生块级损坏时，本文同样生成和图像大小一致的缓存区矩阵，然后在其中选取连续随机大小的长度元素将其值设置为零，这些点代表缓存区损坏区域。最后将图像损坏矩阵与原图像各通道灰度图做与运算即可。该部分异常数据扩增的具体实现如算法14所示。

Algorithm 14: 生成图像缓存区损坏图像

```

1 Separate the RGB channel of the image;
2 if type == random then
3     Generate three memory corruption matrices randomly ;
4     cv2.threshold(memory corruption matrices, 32, 255,
5         cv2.THRESH_BINARY);
6     The image RGB matrix and three memory damage matrices were
7         respectively bitwise and operation;
8 if type == continuous then
9     foreach ch in image.shape[3] do
10         Select the starting point and length of the bad block randomly;
11         Set the matrix element value for the memory corruption region to 0;
12 Merge RGB channel images to generate a new image;
13 return result;
```

3、数据传输线束损坏时数据扩增过程，该过程主要模拟线路板上数据传输线束或引脚损坏时车载摄像头所采集图像效果。数据传输线束损坏异常数据扩增算法的关键是依据数据传输的基本原理生成数据传输损坏矩阵。由于图像在传输过程中每个像素点的数据通过八个引脚并行传输，同时每个像素点的数据格式也为八位二进制。因此，每个引脚负责像素点中八位二进制的每一位数据传输。所以如果某个引脚发生损坏或某条线路发生断路，均会造成该位置的传

输数据为零。所以传输损坏矩阵内元素的值应该一致且范围在 0~255 之间，当用该矩阵与图像矩阵做按位与运算时就可以模拟出数据传输线束损坏时图像的结果。该部分异常数据扩增的具体实现如算法15所示。

Algorithm 15: 生成传输线束损坏图像

```
1 YUVImg = cv2.cvtColor(image, cv2.COLOR_BGR2YUV);
2 Separate the YUV channel of the image;
3 Generate an 8-bit binary number randomly ;
4 Use this number to initialize the transmission harness damage matrix;;
5 foreach channel in image.shape[3] do
6   | Perform bitwise and operation between image and transmission wire
   | harness damage matrix;
7 Merge YUV channel images to generate a new image;
8 return result;
```

3.5 本章小结

本章为论文的核心部分，首先我们分析了自动驾驶汽车图像数据采集通路的硬件架构，对图像输入流程进行了切片分析，明确了各个硬件模块和功能点的组成、作用和原理，并根据各部分流程对过程中的异常进行了分类。然后我们采用失效模式与影响分析 (FMEA) 的方法深入探讨了每一种异常问题的形成原因、失效形式 (规律) 和失效后果，对各个模块的失效模式和失效原因进行分析，总结得到各故障所导致的后果表。扩增规则作为数据扩增的核心是本文研究的重点工作，我们根据不同因素导致异常数据的形成原因及异常情况下采集的图像结果，对各种异常场景的数据扩增方法进行了归纳总结，得出了不同异常问题情况下的数据扩增方法。最后，本文结合总结的以上相关扩增规则介绍了异常数据的生成过程，考虑到算法过程的复杂性和数据生成过程的可理解性，本文采用算法和伪代码的形式对数据扩增的实现过程进行了简要介绍。

第四章 扩增数据的准备与检验

4.1 数据准备与处理

考虑到本文研究的对象为图像数据，研究的具体内容为道路图像中的行人与车辆的检测。所以，我们选取的数据集对象格式必须为图像数据，并且这些数据中要尽可能包含这些物体和场景。同时，为了保证目标检测网络学习的充分性，我们还需要对数据量和数据标签等进行要求。因此，本文选择公开数据集 KITTI 数据集进行数据扩增实验以及网络结构的训练。

4.1.1 KITTI 数据集简介

KITTI 数据集是由德国卡尔斯鲁厄理工学院和丰田美国技术研究院合作完成的。作为国际上最为盛行的自动驾驶场景计算机视觉算法的评测数据集，KITTI 数据集被广泛的用于车辆及行人的目标检测，它包含了不同场景和多种车辆。KITTI 数据集是经过安装在汽车上的相机在真实的道路上采集得到的，其中包含了高速公路、乡村、市区等丰富多样的驾驶场景图像数据 [39]。

KITTI 数据集主要由两部分构成：1、数据文件夹 data object image，该文件夹内主要包括两部分数据。一部分是用于训练深度学习模型的 training 数据集，另一部分为用于测试模型的 testing 数据集。这两部分数据集中的图像数据均为相机采集得到的 2D 图像数据。2、标签文件夹 label2，该文件夹内所包含的文件中记录了 training 数据集中对应图片的目标标注信息。

```
Truck| 0.00 0 1.86 409.55 177.73 520.82 250.74 1.76 1.77 4.41 -3.82 1.90 19.65 1.67
Car 0.00 1 1.82 502.22 185.85 574.38 229.91 1.48 1.65 4.40 -2.63 1.99 27.12 1.72
Car 0.00 0 1.71 602.85 184.10 638.97 209.35 1.42 1.66 3.64 0.67 2.11 43.31 1.73
Car 0.00 1 1.81 677.29 181.51 704.82 198.00 1.44 1.64 3.78 7.44 2.25 66.05 1.92
Cyclist 0.00 0 -1.64 692.58 175.37 717.50 229.83 1.78 0.69 1.81 3.18 1.87 24.54 -1.51
DontCare -1 -1 -10 719.13 178.07 794.77 189.69 -1 -1 -1 -1000 -1000 -1000 -10
```

图 4.1: KITTI 数据集 Label 文件格式示意图

KITTI 数据集总共包含图片 7481 张，图像大小一般为 1242*375 像素。按 7: 3 的比例将数据集分为训练集和测试集。其中训练集 5236 张，测试集 2245 张。KITTI 数据标注共有九个类别，分别为：‘Car’，‘Van’，‘Truck’，‘Pedestrian’，‘Person’，‘Cyclist’，‘Tram’，‘Misc’，‘Don’ t care’。KITTI 数据集使用这些标签分别对行人与车辆类别进行了详细的分类。其中 ‘Don’ t care’ 标

签定义为不关心区域的测试集，它不作为最后对训练集类别，该类别在目标检测算法的实际训练过程中，不会产生任何影响。KITTI 数据集中 label 文件的内容和格式如图4.1所示。在标签文件中，每个目标由 15 个字符串字段构成，每个字符串表示的意义如表4.1所示。

表 4.1: 标注文件含义示意表

	字符串编号	解释
类别	0	对象的分类是八类目标之一或 DontCare
截断	1	对象被截断的程度，范围从 0（没有阶段）~1（完全截断）
遮挡	2	对象被遮挡的程度，从 0 级到 1 级被遮挡程度逐渐加深
观察角	3	物体的观察角度，范围为 $[-\pi, \pi]$
2D box	4-7	2D box 在图像上的 Xmin, Ymin, Xmax, Ymax
3D box 尺寸	8-10	3D box 的长宽高
3D box 位置	11-13	目标中心在相机坐标系中的三维坐标
旋转角	14	目标在相机坐标系绕 y 轴的旋转角，范围为 $[-\pi, \pi]$

4.1.2 数据集标签类别转换

结合本文的实际情况，为了简化目标分类类别，提高模型计算和训练的效率，本文对 KITTI 数据集做了进一步的处理。本文只保留数据集中 Car、Pedestrian、Cyclist 三个类别，把原来的 Car、Van、Truck、Tram 合并为 Car 类，把原来的 Person(sit-ting) 合并为现在的 Pedestrian，原来的 Cyclist 这一类保持不变。本文编写 modify annotations txt.py 脚本，遍历数据集中所有的 label 文件，对其中的 Person (sitting)、Tram、Van、Truck 项进行合并转换，得到新的 label 文件。其中标签转换脚本如算法16所示。

Algorithm 16: 对 KITTI 数据集进行标签合并

```

1 foreach line in this document do
2   if labeldata[0] in ['Truck', 'Van', 'Tram'] then
3     labeldata[0] = labeldata[0].replace(labeldata[0], 'Car');
4   if labeldata[0] == 'Person_sitting', then
5     labeldata[0] = labeldata[0].replace(labeldata[0], 'Pedestrian');
6   if labeldata[0] == 'DontCare' or labeldata[0] == 'Misc' then
7     continue;

```

4.1.3 数据集 VOC 格式及转换

由于本文使用 YOLO V3 目标检测算法进行训练和检测，因此，我们需要对图像标签数据进行 VOC 格式的转化，才可以对 YOLO V3 网络进行训练 [40]。原数据文件经过 VOC 格式的转化后，可以直接通过寻找对应图像路径进行目标识别网络训练及后续检测。本文研究将 KITTI 数据集转化为 VOC 2007 类型数据集，VOC 格式数据集主要包含三个文件夹：Annotation、JPEGImages、ImageSets。Annotation 文件夹保存训练所需的标签文件，其文件名与图像名一一对应。JPEGImages 文件夹存放用于训练的图像，图像为.JPG 格式。ImageSets 中包含三个子文件夹：Main、Layout、Segmentation。在本文中我们主要使用 Main 文件夹中的数据，在 Main 文件夹中存放了 train.txt、val.txt、trainval.txt、test.txt 四个文件。其中 train.txt 记录了训练集所有的图像名称，val.txt 记录了训练过程中用于验证的图像名称，trainval.txt 记录了 train 与 val 的合集，test.txt 记录了用于测试的图像名称。下面本文将对每个文件夹所负责功能以及包含的文件做详细介绍。

```
<?xml version="1.0" ?>
<annotation>
  <folder>KITTI</folder>           // 文件夹
  <filename>000002.jpg</filename>   // xml对应文件名称
  <source>
    <database>The KITTI Database</database>
    <annotation>KITTI</annotation>
  </source>
  <size>                             // 图片大小信息
    <width>1242</width>
    <height>375</height>
    <depth>3</depth>
  </size>
  <object>                             // 图片中标注的物体
    <name>Car</name>                 // 物体类别
    <bndbox>                          // 标注物体的 bounding box
      <xmin>657</xmin>
      <ymin>190</ymin>
      <xmax>700</xmax>
      <ymin>223</ymin>
    </bndbox>
  </object>
</annotation>
```

Annotation 是用来存放 xml 文件的，文件夹中储存的 XML 文件中包含了相对应的 bounding box 的位置及种类等信息。上文展示的代码段为数据集中某图片样本的标注信息，folder 标签记录文件所处位置，filename 标签记录标注图片

对应名；size 标签记录图片的尺寸信息和图片通道数，文件中 depth=3 表示该图像为三通道的彩色图像，object 标签记录目标属性的类别以及标注框的左上角和右下角的坐标信息。

根据上文可知，由于 KITTI 数据集和 VOC 数据集的格式类型存在差异，同时，训练 YOLOv3 网络需要直接获取对应文件的名称和标注框左上角和右下角坐标和尺寸信息，因此我们需要对 KITTI 格式的标注信息进行转换将其转换为 VOC 2007 格式标注信息，其具体过程如算法17所示：

Algorithm 17: KITTI 数据集标签转换为 VOC 2007 格式标签

Data: KITTI data label

Result: VOC 2007 data label

```

1 Create DOM document object;
2 Use the DOM object to create the tags needed in the XML file;
3 foreach line in KITTI label do
4     line = split_line.strip().split();
5     if line[0] in ['Pedestrian', 'Car', 'Cyclist'] then
6         Obtain various information about the image and the target;
7         Export the information to XML;
8 f = open('Annotations/' + name + '.xml', 'w');
9 Writes the DOM object to the file(f);
10 f.close();

```

在 JPEGImages 文件夹中存放的文件格式为.jpg 的图片文件。因此，我们需要对 KITTI 数据集的.png 图像文件进行批量转换。本文中直接使用命令行：`ren *.png *.gif` 进行转换，并将转换后的文件放入 JPEGImages 文件夹中。

在 ImageSets 文件夹的子文件夹 Main 中，我们通过脚本程序 `voc2yolo3.py`，读取 Annotations 文件夹中的 xml 文件的标签数据，生成对应的 `train.txt`、`val.txt`、`trainval.txt` 和 `test.txt` 文件用于进行深度神经网络的训练。

本文首先将需要训练的图片文件转换成.jpg 格式，然后将其放置在 VOC2007 文件夹下的 JPEGImages 中。同时将每张图片所对应的标签文件放置在 VOC2007 文件夹下的 lable 文件夹中。然后，我们运行 `modify annotations txt.py` 程序，遍历文件夹中所有的 label 文件，将原来的八类物体类别转换为所需的三类物体类别，得到新的 label 文件。然后，我们运行 `kitti_txt_to_xml.py` 程序，将 KITTI 数据集.txt 标签文件转化为 VOC 数据集.xml 标签文件，并将其保存在 Annotation

下。最后，我们运行通过程序 `voc2yolo3.py`，读取 `Annotations` 文件夹中的标签数据，生成对应的 `train.txt`、`val.txt`、`trainval.txt` 和 `test.txt` 文件。至此，本文深度神经网络训练所需的数据集制作完成。

4.2 扩增数据有效性检验

4.2.1 YOLO V3 环境搭建及配置文件调试

表 4.2: 系统软硬件环境表

	环境名称	版本号
CPU	Intel Core	Intel Core i5 4210M
GPU	NVIDIA	NVIDIA GTX960M 4G
内存	Samsung	DDR3L 8G
操作系统	Windows	windows 7
环境管理工具	Anaconda	Anaconda 3.7
NVIDIA 运算平台	CUDA	CUDA 9.1.48
编程语言	Python	Python 3.7.4
机器学习库	Pytorch	Pytorch 1.1
计算机视觉库	Opencv	Opencv-Python 4.5.1.48
包安装与管理工具	pip	pip 19.2.3

本文中，我们所使用的环境为：`windows 7 + Anaconda 3.7+ CUDA 9.1.48 + Python 3.7.4 + Pytorch1.1 + Opencv-python4.5.1.48`。具体软硬件环境如表4.2所示。接下来，我们运行根目录下的 `voc_annotation.py` 程序，该程序的作用是在根目录下生成 `2007_train.txt` 文件，在该文件中每一行对应训练图片的绝对位置及其真实框的位置，运行前我们需要对文件中的 `classes` 进行改动，将其换为我们生成的三种目标识别类型。同时，我们还需要修改 `utils/config.py` 文件和 `model_data` 里面的 `voc_classes.txt` 文件，修改内部的 `Num_Classes` 和 `classes`，将其变成所分的种类的数量和我们需要检验的三中种类型。

4.2.2 扩增数据生成及有效性检验

本文使用 KITTI 数据集中的测试集作为数据扩增的作用对象进行图像数据的扩增和实验，将上文归纳的 15 种扩增方法作用于待处理图片最终得到不同情况下的异常图像数据。随机选取一张图像进行数据扩增测试，图4.2中各子图像为使用不同扩增方法生成的图像数据。



图 4.2: 各种扩增方法生成的图像数据

4.2.2.1 研究问题

为了探究图像数据采集通路中由于某些异常或设备故障产生的异常图像对自动驾驶汽车的影响、讨论深度学习系统对扩增前后数据判别的差异性、研究扩增后的数据对目标识别系统产生何种影响，本文安排并设计了扩增数据生成及有效性检验实验。该实验选取 KITTI 数据集中测试集作为待扩增数据，将上文所总结的扩增算法作用于这些图像数据上就可以得出不同异常类型下的图像数据。将这些异常图像数据输入给训练好的目标识别模型中，可以研究数据扩增前后对目标识别模型产生的相关影响，探究在这些异常情况下目标识别模型能否正常工作。同时，实验还可以进一步检测出在异常情况发生时深度学习系统中的漏洞与不足，反映出系统安全性和鲁棒性问题。

4.2.2.2 实验对象

本实验在 KITTI 2D 图像数据集的测试集中随机选取 50 张图像数据作为待扩增数据对象。选取运行速度较快的 YOLO V3 作为目标检测模型对扩增前后的数据进行检测，同时选择 KITTI 2D 图像数据集的训练集中前 1500 张图像数据作为训练数据对 YOLO V3 网络进行训练。

4.2.2.3 实验设计

本文首先对 YOLO V3 进行训练,我们选取 KITTI 数据集中训练集的前 1500 图像来训练 YOLO V3 模型,通过上文可知,我们对 KITTI 数据集的标签和图像文件进行了处理,将其重新分为三类 (Car、Cyclist、Pedestrian),然后我们调整训练参数,将其按照 9:1 的比例划分训练集和验证集。我们设置第一阶段训练参数学习率 lr 为 1×10^{-3} , 每批输入的图像大小 Batch_size 为 8, 起始世代 Init_Epoch 为 0, 冻结训练的世代 Freeze_Epoch 为 50, 设置第二阶段训练参数学习率 lr 为 1×10^{-4} , 每批输入的图像大小 Batch_size 为 4, 起始世代 Init_Epoch 为 50, 冻结训练的世代 Freeze_Epoch 为 100。然后对其进行训练,训练后模型的识别效果如图4.3所示。

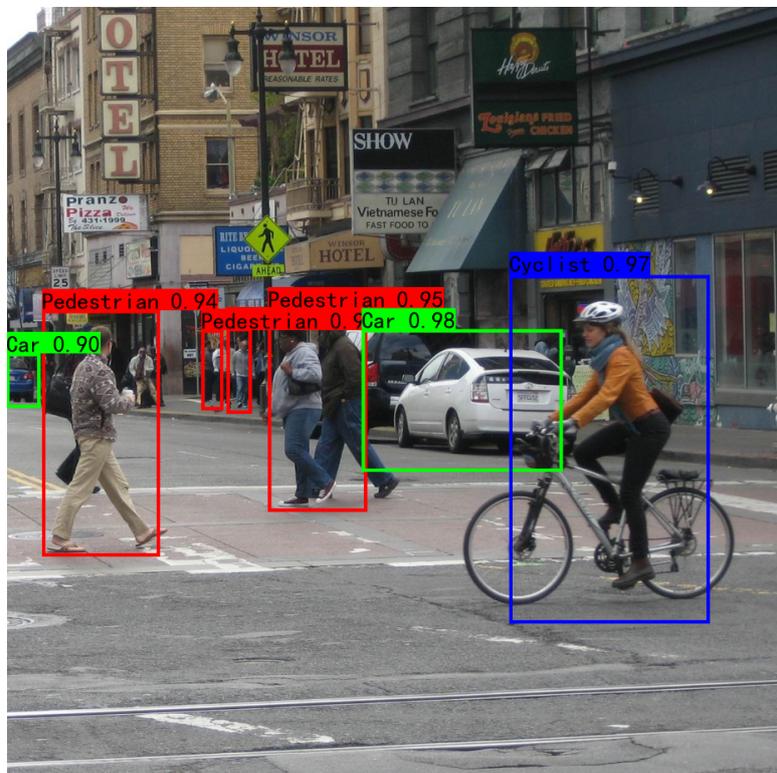


图 4.3: YOLO V3 识别效果图

为了更为直观的显示出图像经各模块扩增算法处理后对深度学习的不同影响程度,本文对同一张图像施加了所有的数据扩增算法。这些算法作用于原始图像后会生成 15 张不同异常类型的图像数据。对扩增前后的 16 张图像数据进行目标检测,其检测结果如图4.4所示。

在实验过程中,我们首先对 KITTI 2D 图像数据集的测试集中随机选取的 50 张图像数据进行数据扩增处理,将其扩增为 750 张不同类型异常的图像数据。

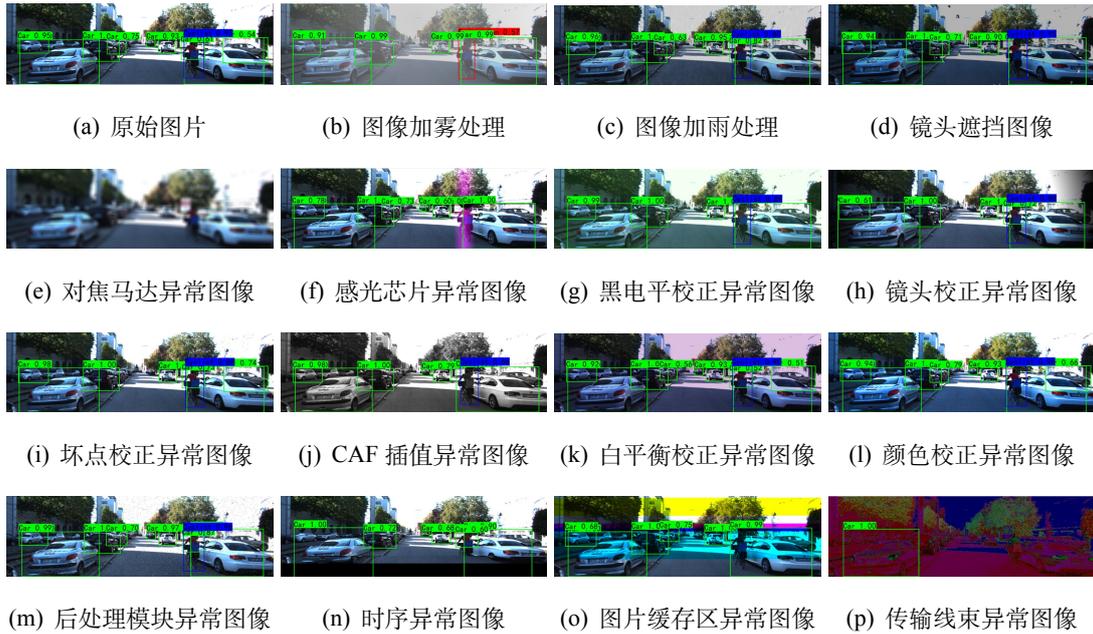


图 4.4: 扩增数据识别结果示意图

然后，我们将原始图像数据与扩增后的图像数据一并输入给训练好的 YOLO V3 目标识别模型中，观察并记录目标识别模型对这些图像数据的测试结果。最后，我们对所有的测试结果进行汇总和统计，依据统计目标的不同绘制出相关表格和实验统计图，同时对实验结果进行分析和总结。

4.2.2.4 评价指标

通过本文提出的扩增算法，可以将一张图像扩增成 15 张不同故障或异常场景下的图像数据。因此，本文提出的基于功能安全的自动驾驶图像数据扩增技术可以在短时间内生成大量异常情况下的图像数据。用这些图像数据对自动驾驶系统进行测试可以解决自动驾驶汽车异常图像数据集难以获取的问题。通过相关测试，研究人员可以探究在图像数据采集通路中由于某些异常或设备故障产生的异常图像对自动驾驶系统的影响。本文用深度学习模型 YOLO V3 代替自动驾驶汽车中的目标识别系统，将处理前后的图像作为模型的输入数据进行测试。在测试的过程中，本文发现扩增后的数据可以对深度学习模型产生不同程度的影响。根据影响程度的不同，本文将其分为三类：1、扩增后的图像数据造成深度学习模型对目标判断的置信度发生变化。2、扩增后的图像数据造成深度学习模型对目标的错误判断，造成目标分类错误。3、扩增后的图像数据造成深度学习模型对目标的漏判。这种影响最为严重，如果发生在自动驾驶汽车上会很大概率发生交通事故。

根据上文总结出的扩增后的数据对深度学习模型产生的 3 种不同程度的影响，我们提出了以下几种评价指标：

1、图像数据扩增后目标检测结果漏判次数：该评价指标记录了实验过程中每种异常扩增数据相比于原始数据目标检测漏检测次数的总和。该指标可以反映出目标检测模型对于异常数据判断的精确能力。漏检测次数越多说明扩增后的异常图像数据对目标检测模型判断的精确性影响越大，目标检测模型的安全性和鲁棒性越差。

2、图像数据扩增后目标检测结果误判次数：该评价指标记录了实验过程中每种异常扩增数据相比于原始数据目标检测误检测次数的总和。该指标可以反映出目标检测模型对于异常数据判断的准确能力。误检测次数越多说明扩增后的异常图像数据对目标检测模型判断的准确性影响越大，目标检测模型的安全性和鲁棒性越差。

3、图像数据扩增后目标检测结果漏判率：目标检测结果漏判率 = 扩增后异常图像数据漏判次数总和 / 原图像数据目标检测数目总和。该评价指标反映了扩增后异常图像对于目标识别模型精确度的影响。

4、图像数据扩增后目标检测结果误判率：目标检测结果误判率 = 扩增后异常图像数据误判次数总和 / 原图像数据目标检测数目总和。该评价指标反映了扩增后异常图像对于目标识别模型准确度的影响。

5、图像数据扩增识别目标数之比：图像数据扩增识别目标数之比 = 扩增后异常图像数据目标检测数目总和 / 原图像数据目标检测数目总和。该评价指标可以反映深度学习系统对扩增前后图像数据判别的差异性。

6、图像数据扩增前后目标识别模型置信度总和之比：目标识别模型置信度总和之比 = 扩增后异常图像数据目标检测置信度总和 / 原图像数据目标检测置信度总和。该评价指标可以反映扩增后数据对于深度学习模型置信度影响的程度。

4.3 实验结果与分析

本文随机在测试集中选取 50 张图片进行数据扩增处理。我们将以上归纳的 15 种扩增方法作用于待扩增图片，最终生成了 750 张不同异常下的图像数据。随后，本文对这些图像数据进行目标检测后进行统计总结，最终得到的结果如表4.3所示。

从表中数据可以看出，扩增后的数据相比于原图像数据对深度学习系统产生了影响，导致了系统对目标检测的误判和漏判以及目标检测置信度的下降，其

表 4.3: 扩增前后图像目标检测结果表

	漏判次数 (相对于原图像)	误判次数 (相对于原图像)	识别目标数之比 扩增后: 原图像	置信度总和之比 扩增后: 原图像
图像加雾处理	137	15	93:230	86.57:192.55
图像加雨处理	35	5	195:230	165.42:192.55
镜头水滴遮挡	49	4	181:230	161:192.55
对焦马达异常	187	0	43:230	44.4:192.55
感光芯片异常	30	11	200:230	167.75:192.55
黑电平校正异常	84	0	146:230	127.49:192.55
坏点校正异常	61	3	169:230	148.53:192.55
镜头校正异常	73	7	157:230	133.23:192.55
CAF 插值异常	90	0	140:230	122.47:192.55
白平衡校正异常	17	0	213:230	176.58:192.55
颜色校正异常	13	0	217:230	181.87:192.55
后处理模块异常	50	3	180:230	161.05:192.55
主控芯片时序异常	87	21	143:230	128.81:192.55
图片缓存区异常	137	6	93:230	92.13:192.55
传输线束异常	223	0	7:230	7.83:192.55
总计	1273	80	2177:3450	1823.64:2888.25

中一共发生 80 次误判, 1273 次漏判。因异常发生产生的误判概率为 2.32%, 因异常发生产生的漏判概率为 36.89%, 使得目标检测的识别率总体下降了 39.21%, 总体置信度下降了 36.86%。

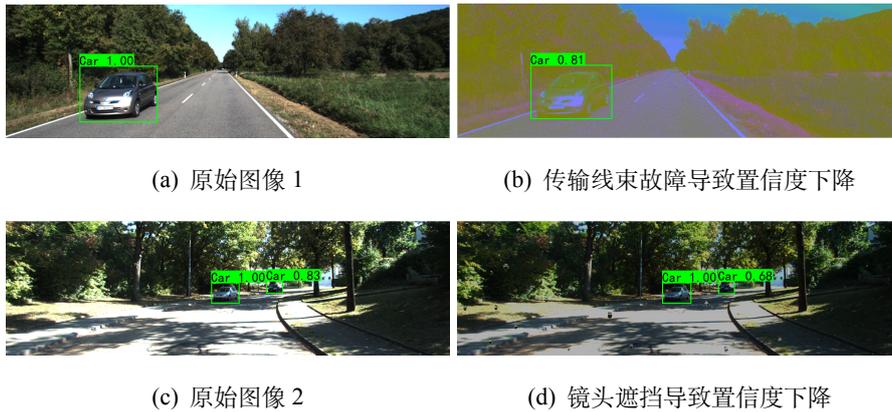


图 4.5: 扩增前后目标识别置信度变化示意图

在实验中, 我们发现图像数据扩增后, 输入给深度学习模型进行目标识别检测, 会导致模型对目标判断的置信度发生下降。这种情况的目标检测结果如图4.5所示。在图4.5的 (a) (b) 图像中对小车识别的置信度发生了变化。这种变

化是由于经过数据传输线束损坏扩增过程后图像的质量发生变化而造成的。而 (c) (d) 图像中对小车识别的置信度降低的原因是图像经过镜头遮挡算法后图像质量下降，导致图像内容细节的丢失。

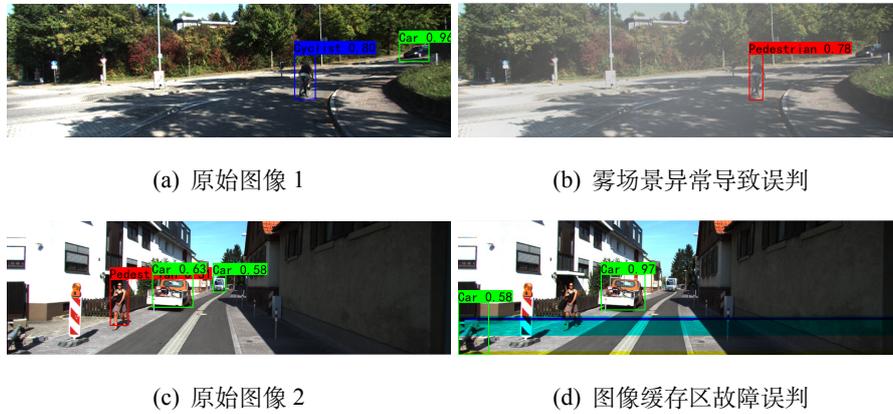


图 4.6: 扩增前后目标识别判断错误示意图

我们同样发现图像数据经过扩增处理后会导导致模型对目标判断发生错误，造成目标分类错误或将原来图像中不属于任何类型的内容错误的划分为某一类型。这种情况的目标检测结果如图4.6所示。可以发现，在图4.6的 (a) (b) 图像中由于对原图像进行了加雾处理导致目标识别系统对 Cyclist 的类型划分产生错误。而 (c) (d) 图像则是由于图像经过缓存区域故障算法处理后将原来图像中不属于任何类型的内容错误的识别为 Car 类型。原图像经过扩增算法的处理后造成图像内容细节模，从而导致目标识别系统产生了这些错误判断。

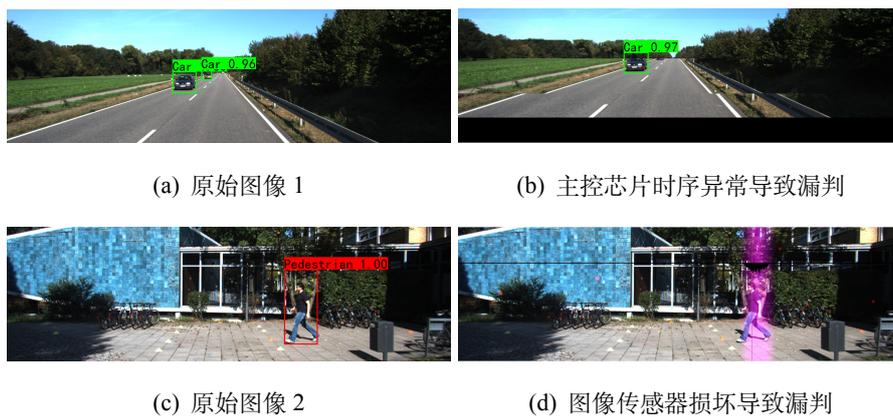


图 4.7: 扩增前后目标识别漏判断示意图

第三类错误为扩增后的图像数据造成深度学习模型对目标的漏判。这种影响最为严重，如果发生在自动驾驶汽车上会很大概率引发交通事故。在实验过

程中，我们在对扩增图像的测试中也发现了这种情况。如图4.7所示，(a) 图像由于经过时序异常算法的处理，图像中内容和质量大大降低导致了对图像 (b) 的漏判。同样 (c) 图像经过感光芯片异常扩增后生成的图像数据 (d)，由于图像内容发生了明显的改变造成了深度学习模型的漏判。

为了更为直观的观察每种数据扩增方法对目标识别系统的影响，本文对表4.3中的内容进行了汇总统计，并根据统计结果绘制了相应的数据统计图。扩增算法导致系统发生漏判次数统计图，如图4.8所示。

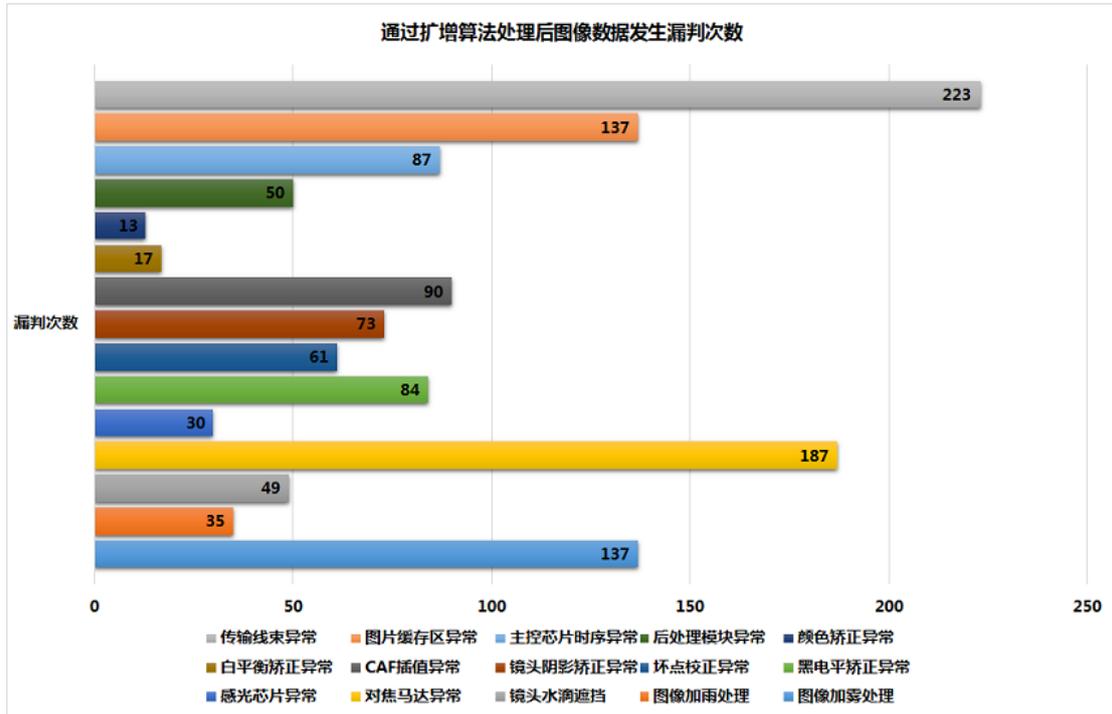


图 4.8: 扩增算法导致系统漏判统计图

通过观察图4.8，我们发现由线束传输故障、对焦马达故障、图像缓存区故障和雾场景扩增算法生成的异常图像数据对深度学习模型的漏判影响程度最大。而在实验中，由颜色校正异常和白平衡校正异常扩增算法生成的图像数据对深度学习模型的漏判影响程度最小。

图4.9为各扩增算法处理后的图像导致系统发生误判次数的统计图。可以发现由主控芯片时序故障和雾场景扩增算法生成的异常图像数据导致深度学习模型发生错误判断的次数最多。而在本文的实验中，由对焦马达异常、黑电平校正异常、CAF插值异常、白平衡校正异常、颜色校正异常、传输线束异常扩增算法生成的图像数据不会造成深度学习模型的误判。

图4.10展示了各扩增算法处理前后的图像中目标被系统识别的数量之比。可

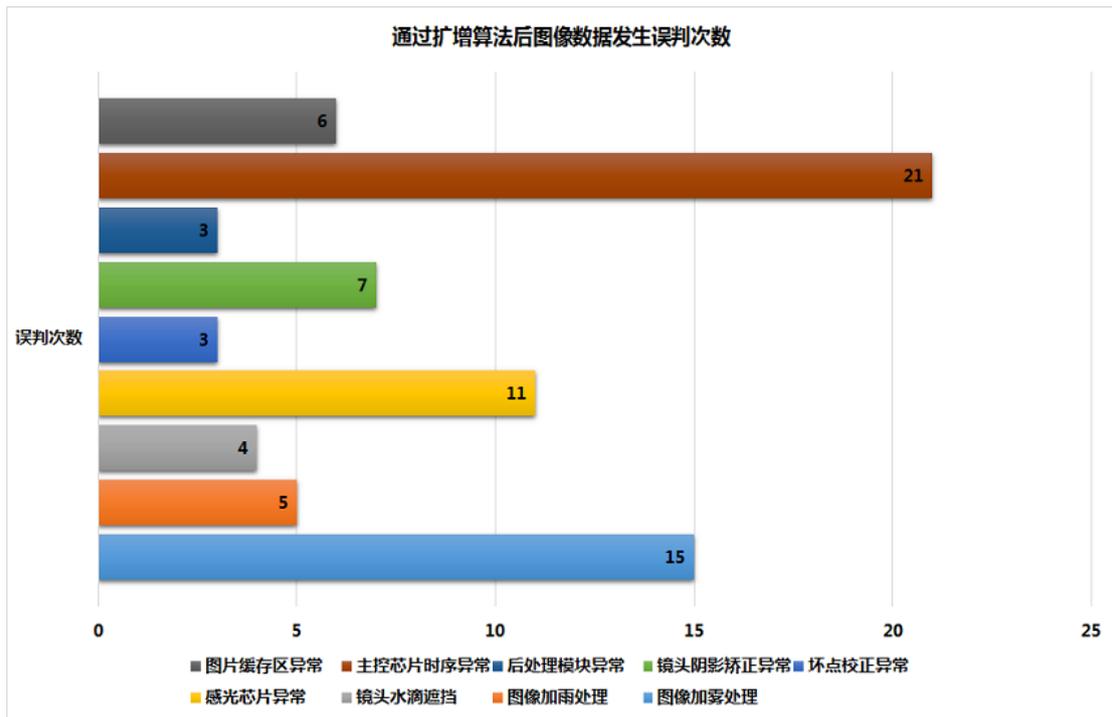


图 4.9: 扩增算法导致系统误判统计图

可以发现由传输线束故障和对焦马达故障扩增算法生成的异常图像数据导致深度学习模型识别目标的数量相较于原图像中目标识别的数量严重下降。而实验中由白平衡校正异常、颜色校正异常扩增算法生成的图像数据与原图像被深度学习模型识别的数量相差不大。

图4.11为各扩增算法处理前后的图像内识别目标的置信度总和比统计图。可以发现由传输线束故障和对焦马达故障扩增算法生成的异常图像数据导致深度学习模型识别目标的置信度相较于原图像严重下降。而由白平衡校正异常、颜色校正异常扩增算法生成的图像数据对深度学习模型识别目标的置信度影响最小，这两种故障算法生成的图像数据导致深度学习模型的置信度下降在 10% 以内。

参考实验结果，我们可以进一步得出以下结论：

1、本文基于功能安全的数据扩增方法可以模拟出在不同异常情况下自动驾驶汽车采集的异常图像数据。

2、这些异常的图像数据会对深度学习系统产生不同作用的影响，导致系统对目标检测的误判和漏判以及目标检测置信度的下降。

3、在本文总结的自动驾驶汽车图像数据采集通路的 15 类异常中，传输线束异常发生时对深度学习系统的影响最大，造成了 96.95% 的漏判概率。白平衡校

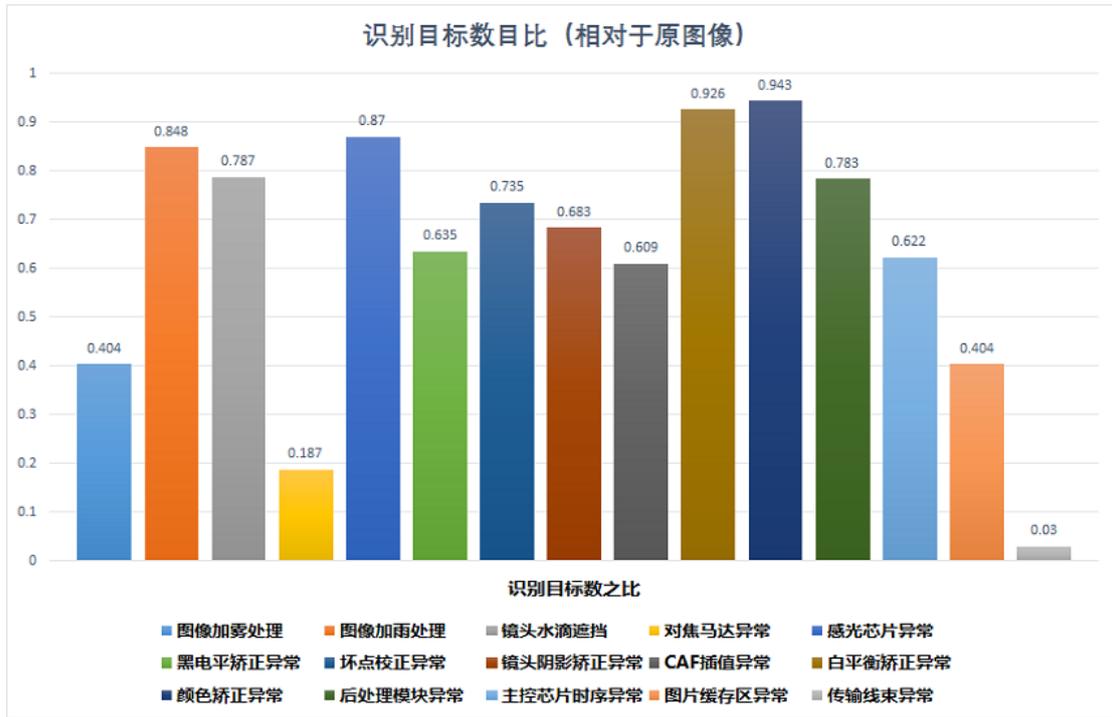


图 4.10: 扩增数据前后目标识别数目比统计图

正模块和颜色校正模块发生异常对深度学习系统的影响最小，分别造成了 7.39% 和 5.65% 的漏判概率，且均未造成深度学习系统的误判。

4、在本文总结的自动驾驶汽车图像数据采集通路的 15 类异常中，图像加雾处理、图像加雨处理、镜头水滴遮挡、感光芯片异常、坏点校正异常、镜头校正异常、后处理模块异常、主控芯片时序异常以及图片缓存区异常都会导致深度学习系统对目标分类的误判，其中主控芯片时序异常最容易造成深度学习系统对目标分类的误判，当该部分发生异常时造成模型 9.13% 概率的误判。

5、传输线束异常、对焦马达异常和图像加雾处理对深度学习系统对目标分类置信度的影响最大，这三类异常发生时会导致目标检测置信度的急剧下降。

6、对焦马达异常发生异常时图像的模糊等级对深度学习系统的影响成正相关，图像越模糊深度学习系统发生错误的概率越大。

7、通过本文所概括的方法生成的数据可以检测到在异常情况发生时深度学习系统中的不足与漏洞，反映出系统安全性和鲁棒性问题。

4.4 本章小结

本章主要介绍了自动驾驶图像数据的来源、准备、处理和具体的扩增算法。同时，我们还探究了扩增数据对于深度学习系统的具体影响。首先，本章对训练

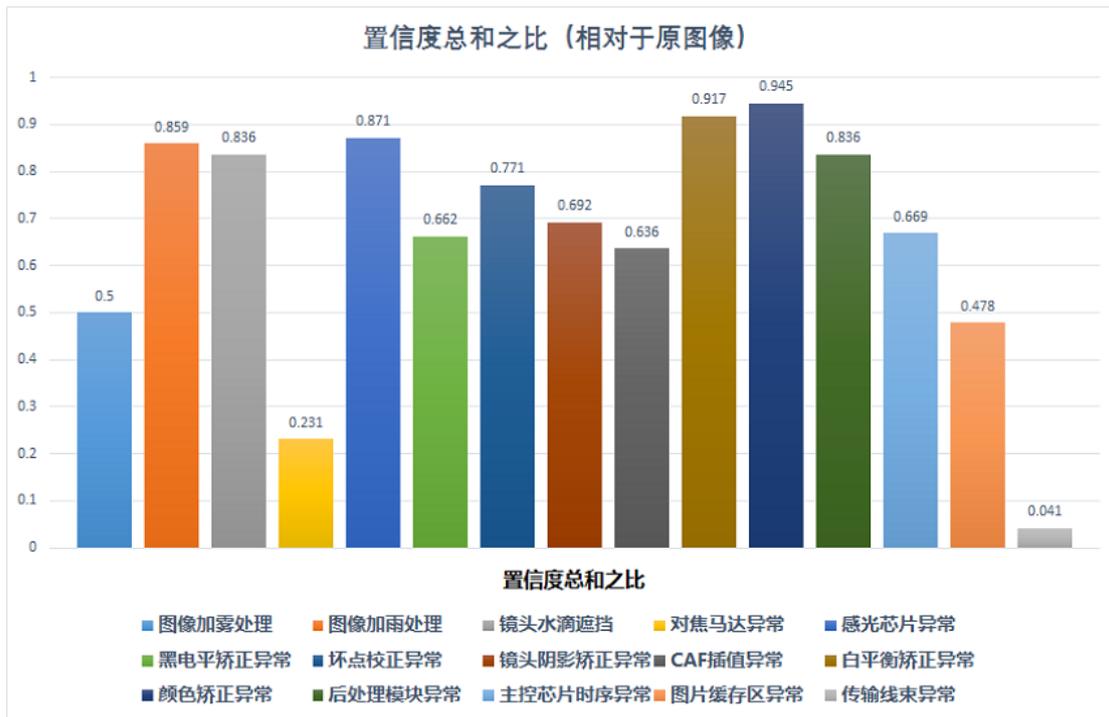


图 4.11: 扩增数据前后目标识别置信度总和比统计图

和待扩增图像数据的来源、特点及其相关格式进行了介绍，由于 YOLO V3 训练数据时所采用的格式与 KITTI 数据集格式不同，我们对其格式转换也进行了简单介绍。然后，我们对整体系统的软硬件环境进行简要介绍。其次，我们将扩增方法作用于相关数据进行展示并总结统计，发现在异常情况发生时深度学习系统中的不足与漏洞，反映出系统安全性和鲁棒性问题。最终，我们参考实验结果总结得出了七个具有实际价值的结论。

第五章 总结与展望

5.1 本文研究内容总结

本文基于汽车电子电器产品的研发和管理流程中的功能安全分析方法对自动驾驶汽车摄像头采集的图像数据进行扩增研究。本文采用 FMEA 分析方法,对自动驾驶汽车图像数据流通路进行分析切片,总结各种异常发生时数据的规律,得出了不同的扩增规则,并基于 Python 语言,实现了异常数据的扩增。该技术可以模拟自动驾驶汽车图像数据采集通路中各模块节点发生故障异常时的图像数据。通过深度学习模型和这些扩增数据,本文发现了自动驾驶汽车图像数据采集通路异常发生时深度学习模型中的不足与漏洞,反映出系统安全性和鲁棒性问题。本文主要的研究工作与成果如下:

1、对自动驾驶汽车的图像数据采集系统进行功能安全分析。本文采用 ISO26262 中推荐的失效模式与影响分析方法 (FMEA) 对自动驾驶汽车的图像数据的整体输入流程进行切片分析,并对图像采集过程中各个部分中可能产生的故障和异常失效形式进行分类,总结出了图像数据流通路中最容易发生事故的 15 种异常类型。

2、本文对图像数据流通路中最容易发生事故的 15 种异常类型进行分析,分析产生异常图像的原因、硬件或摄像机中的模块发生错误的机理以及故障发生时图像数据因这些异常发生的变化。根据这些原因和机理,总结归纳出这些异常发生时图像数据扩增的规律,并通过 Python 代码和 OpenCV 库进行扩增数据的实现。

3、本文对 KITTI 数据集进行收集和修改,使用 Anaconda 对系统整体环境进行配置,利用修改后的 KITTI 数据集中的训练集对 YOLO V3 深度学习网络进行训练,得出的深度学习模型可以对图像的内容进行“Car”、“Pedestrian”、“Cyclist”三种类型的分类和识别。

4、本文根据文中所总结的扩增规则对 KITTI 数据集中的测试集数据进行扩增,得出了在不同故障发生时异常的图像数据,并对扩增数据的结果进行保存。

5、本文在 KITTI 数据集中的测试集数据中随机选取 50 张图像进行数据扩增,共得出 750 张扩增后的图像,其中每种异常类型各 50 张。利用训练好的深度学习模型对包括原图像的 800 张图像进行检测和统计,最终得出结论:扩增后异常的图像数据会对深度学习系统产生不同程度的影响,导致系统对目标检

测的误判和漏判以及目标检测置信度的下降。通过本文所概括的方法生成的数据可以检测到在异常情况发生时深度学习系统中的不足与漏洞，反映出系统安全性和鲁棒性问题。

6、本文将汽车工业开发测试领域的功能安全分析技术与计算机视觉领域的图像处理技术相结合，提出了一种全新的自动驾驶汽车数据扩增方法。相较于 DeepTest 和 DeepRoad 中提出的驾驶场景变换图像数据扩增方法，本文所总结的扩增方法涵盖了自动驾驶汽车相机采集通路的各个功能点（不局限于驾驶场景）。因此，本文提出的扩增方法在自动驾驶汽车的实际生产开发和测试过程中，更加具有针对性和专业性。

5.2 本文工作内容局限与展望

尽管本文提出的基于汽车电子电器产品研发和管理流程中的功能安全分析方法对自动驾驶汽车采集的图像数据进行扩增研究具有一定价值，但是在实际的过程中还存在一些问题有待进一步优化：

1、除了本文总结归纳出的 15 种异常情况外，自动驾驶汽车图像采集通路中还存在一些潜在类型的故障，因此如果将自动驾驶汽车图像采集通路中所有的故障一一列举，并针对不同故障数据进行扩增可以检测到更多异常情况发生时深度学习系统中的不足与漏洞。

2、图像数据相较于视频数据为间断性数据，不具有连续性。因此使用图像数据在分析自动驾驶汽车安全性、目标检测系统鲁棒性的过程中仍然存在局限性。

3、本文仅针对单一图像数据实现了扩增，但在实际的过程中自动驾驶系统往往基于不同传感器采集的融合数据进行目标的识别和判断。因此可以采用相同场景下不同传感器采集的数据进行统一扩增，这样的结果会更加准确。

4、本文采用 YOLO V3 对扩增数据进行检测和判断，单一的深度学习网络本身就具有一定的局限性。因此下阶段可以采用不同的深度学习网络对扩增后的数据进行检测判断，这样得出的结果会更加客观真实。

参考文献

- [1] 马桂强, 汽车市场需求调查与预测, 营销界 (44) (2020) 201–202.
- [2] 章军辉, 陈大鹏, 李庆, 自动驾驶技术研究现状及发展趋势, 科学技术与工程 v.20;No.514 (09) (2020) 37–46.
- [3] 赵炯, 王伟, 基于传感器融合技术的电动汽车自动驾驶系统的开发, 制造业自动化 35 (009) (2013) 43–46.
- [4] 李时雨, 基于线控转向系统的无人驾驶技术发展, 汽车工程师 No.251 (03) (2018) 16–19.
- [5] G. Venkat, 自动驾驶辅助系统聚焦于图像传感的功能性安全, 电子产品世界 v.25; No.355 (08) (2018) 26–28.
- [6] 《汽车与安全》编辑部, 自动驾驶汽车的概念和历史, 汽车与安全.
- [7] 袁建华, 陆文杰, 宋群, 《智能网联汽车道路测试管理规范 (试行)》解读, 道路交通科学技术 000 (003) (2018) 3–4.
- [8] 王艺帆, 自动驾驶汽车感知系统关键技术综述, 汽车电器 (12) (2016) 12–16.
- [9] 秦硕, 周礼, 许淞斐, 基于深度学习的图像数据扩增方法 (2018).
- [10] 田凯茜, Gan 在自动驾驶数据集生成方面的应用, 技术与市场 026 (011) (2019) 30–31.
- [11] Y. Tian, K. Pei, S. Jana, B. Ray, Deeptest: Automated testing of deep-neural-network-driven autonomous cars.
- [12] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.
- [13] L. Taylor, G. Nitschke, Improving deep learning using generic data augmentation.
- [14] M. Zhang, Y. Zhang, L. Zhang, C. Liu, S. Khurshid, Deeproad: Gan-based metamorphic autonomous driving system testing.

- [15] 闫施违, 自动驾驶系统测试用例生成技术的实证研究, Ph.D. thesis, 哈尔滨工业大学.
- [16] H. Inoue, Data augmentation by pairing samples for images classification.
- [17] F. Perez, C. Vasconcelos, S. Avila, E. Valle, Data augmentation for skin lesion analysis.
- [18] 李佳玉, 员春欣, Iec61508 功能安全国际标准及安全性分析, 中国铁路 (1) (2001) 44–45.
- [19] T. Dittel, H. J. Aryus, How to "survive" a safety case according to iso26262, in: Computer Safety, Reliability, Security, International Conference, Safecomp, Vienna, Austria, September, 2010.
- [20] 阮秋琦, 数字图像处理学, 数字图像处理学, 2007.
- [21] 徐黎明, 刘航江, 数字图像处理技术研究综述, 软件导刊 15 (003) (2016) 181–182.
- [22] C. Solomon 等, 吴永礼, 数字图像处理基础, 国外科技新书评介 000 (010) (2011) 21–22.
- [23] 李迎春, 曾峦, 丁向丽, 图像的几何运算方法, 装备指挥技术学院学报 (01) (2000) 46–49.
- [24] 王鹏飞, 金杨, 刘真, 图像颜色空间与其分量自相关研究, 中国印刷与包装研究 (05) (2011) 40–45+51.
- [25] 王庆有, 图像传感器应用技术, 图像传感器应用技术, 2013.
- [26] 李彦冬, 郝宗波, 雷航, 卷积神经网络研究综述, 计算机应用 36 (009) (2016) 2508–2515.
- [27] 李建伟, 各类摄像头在 L3 级自动驾驶视觉感知中的应用, 汽车电器 (8).
- [28] R. Hide, Optics of the atmosphere: Scattering by molecules and particles, Physics Bulletin.
- [29] 林岚, 基于大气散射模型的去雾算法研究, Ph.D. thesis, 西安电子科技大学 (2014).

- [30] 王勇, 单幅雾天降质图像复原方法研究, Ph.D. thesis, 合肥工业大学 (2010).
- [31] 石晓晴, 单幅图像中雨滴检测与去除方法的研究, Ph.D. thesis, 北京交通大学 (2016).
- [32] J. C. Halimeh, M. Roser, Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation, IEEE.
- [33] 林均仰, 舒嵘, 黄庚华, 方抗美, 闫志新, 激光对 ccd 及 cmos 图像传感器的损伤阈值研究, 红外与毫米波学报 (06) (2008) 475–478.
- [34] 倪晓武, 沈中华, 强激光对光电器件及半导体材料的破坏研究, 光电子·激光 (6 期) 487–490.
- [35] 盛良, 张震, 张检民, 徐作冬, 林新伟, 李云鹏, 激光导致 cmos 输出光斑点阵的现象及机理, in: 2015 光电防御技术学术研讨会摘要集, 2015.
- [36] 王森, 图像信号处理器中关键技术研究, Ph.D. thesis, 上海交通大学.
- [37] 李方宁, 高清高速 cmos 相机系统设计, Ph.D. thesis, 中国科学院研究生院 (长春光学精密机械与物理研究所) (2014).
- [38] 许鹏, 梁发云, 王兴鹏, 宋凡, 刘敏, 图像传感器数据采集与显示系统设计研究, 南昌大学学报 (工科版) (02) (2015) 176–179.
- [39] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, International Journal of Robotics Research 32 (11) (2013) 1231–1237.
- [40] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv e-prints.

简历与科研成果

基本情况 徐彬桐，男，汉族，1996年9月出生，河南省安阳市人。

教育背景

2019.9 ~ 2021.6 南京大学软件学院 硕士

2015.9 ~ 2019.6 吉林大学汽车工程学院 本科

这里是读研期间的成果（实例为受理的专利）

1. 陈振宇，**徐彬桐**，倪焯，“一种面向自动驾驶图像数据的测试用例生成方法”，申请号：CN202010712788.4，已受理。
2. 王晓冰，**徐彬桐**，倪焯，“一种面向自动驾驶系统的蜕变测试方法”，申请号：CN202010714664.X，已受理。

致 谢

光阴似箭，转眼间硕士两年的生活就要结束了，蓦然回首，过往岁月，历历在目，留恋之情，油然而生，感恩之情，铭记在心。能够顺利完成学业，开启人生的新征程，离不开身边的人给予我的鼓励和帮助，或振聋发聩，或醍醐灌顶，或悄然无声，都激励着我做更好的自己。在本论文的完成之际，我的内心感触颇多，很高兴能够借此机会，向他们表达我的感激之情，并奉上最诚挚的祝福，在此我要感谢硕士研究生期间给予我帮助和关心的所有人。

我的导师，陈振宇老师，他那严肃的科学态度，一丝不苟的学术精神，求同存异的工作作风时时刻刻都在激励着我，他不仅仅只是在学业上给我以悉心指导，更是在思想、生活上给予了我无限的鼓励，在此，我谨向陈老师致以十二分诚挚的谢意。

同样要感谢冯洋老师、刘嘉老师在硕士期间对我的指导和帮助，学习和生活中遇到问题时，老师们总是一如既往的鼓励我，启发我解决问题。

除此之外，我还要感谢软件学院和实验室的其他老师、室友、同学。我们彼此交流心得，相互帮助，期间我收获了许多，也从中取得了许多进步。软件学院如同一个大家庭，在温馨美好的生活环境中，让我保持良好的心态进行学习科研与论文的撰写。

最后我想特别感谢我父母，女友和家人，是他们一直都在默默的为我付出，不管在经济上还是精神上都给予了我极大地支持，再一次向我最亲爱的人表示由衷的感谢。

感谢评阅本论文的老师，感谢课题研究相关的学术前辈们。