****

本科毕业设计（论文）

**基于深度学习的手势检测技术研究**

|  |  |
| --- | --- |
| **学院**  **专业**  **学生姓名**  **学生学号**  **指导教师**  **提交日期** | **自动化科学与工程学院** |
| **自动化** |
| **别远山** |
| **201730643332** |
| **刘伟东** |
| **2021 年 5 月25 日** |

# 摘 要

近年来，随着科学技术的发展，电子设备变得越来越智能，功能也越来越多，与之相关的交互方式也越来越复杂。在众多的交互方法中，基于手势控制的交互方式是比较便捷且易于理解和学习的。目前，手势交互技术在已经在很多场景下得到了应用。在手势检测领域中，大多数检测方法都是对手势类别及其在平面上的坐标位置进行提取，而对手势的深度信息进行提取的相关工作较少。如果能够从RGB手势图像中得到手势目标的深度信息，就可以在花费较低成本的情况下拓宽手势交互的应用范围。本文主要对基于深度学习的手势检测及目标深度估计技术进行了研究。

（1）本次设计中使用YOLOv3目标检测模型在自制的手势数据集上训练，得到基于深度学习的手势检测模型，经测试能够满足实时手势检测的要求。

（2）针对普通的手势检测模型无法获取手势的深度信息的问题，本文尝试使用基于深度学习的单目深度估计算法从单张RGB图像中估计手势目标的深度，包括FastDepth算法和基于YOLOv3的改进模型。在自制手势训练集上训练并测试FastDepth算法的估计效果。之后通过添加输出维度和修改损失函数的方式，给YOLOv3模型增加估计目标深度的功能，并进行训练和测试。最后的实验结果表明，2种方法都能在一定程度上对RGB图像中手势目标的深度信息进行估计。

关键词：手势检测；深度学习；目标检测；深度估计

# Abstract

In recent years, with the development of information technology, electronic devices are becoming more and more intelligent, and have more and more functions and abilities. The related interaction methods are becoming more and more complex. Among all the interaction methods, gesture based interaction is one of the most convenient and easy to learn and use. At present, gesture interaction technology has been used in many scenes. In the field of gesture detection, most of the detection methods extract the gesture’s category and position in 2D plane, but the work of extracting the depth information of gesture is relatively few. If the depth information of gesture object can be obtained from RGB gesture image, the application range of gesture interaction can be broadened. This paper mainly studied gesture detection and target depth estimation based on deep learning.

(1) In this paper, we use the YOLOv3 object detection model to detect gesture in RGB image. We trained the model on the self-made gesture dataset to obtain the gesture detection model based on deep learning. Then by testing the model on the test dataset, we found that the model can meet the requirements of real-time gesture detection while maintaining high accuracy.

1. Aiming at the problem that the common gesture detection model can not get the depth information of gesture, this paper tries to use the monocular depth estimation algorithm based on depth learning to estimate the depth of gesture object from a single RGB image, including FastDepth algorithm and the improved detection model based on YOLOv3. The FastDepth algorithm is trained and tested on the self-made gesture-depth dataset. Then, by adding a depth vector to output dimensions and modifying the loss function, the function of estimating target depth is added to the YOLOv3 model. Then we trained and tested the modified YOLOv3 model on the same gesture-depth dataset. Finally, the experiment results show that both methods can estimate the depth information of gesture object in RGB image to a certain extent.

**Keywords**: Gesture Detection; Deep Learning; Target Detection; Depth Estimation

# 目 录

摘 要 I

Abstract II

目 录 III

第一章 绪论 1

1.1 研究背景与意义 1

1.2手势识别国内外研究现状 1

1.3论文研究内容和结构安排 2

第二章 相关技术基础 4

2.1 卷积神经网络相关技术 4

2.1.1 卷积层 4

2.1.2 池化层 4

2.1.3 全连接层 5

2.1.4 激活函数 5

2.1.5 损失函数 6

2.2 基于深度学习的目标检测 7

2.2.1 YOLO系列算法简介 7

2.3 基于深度学习的单目深度估计 8

2.4 本章小结 9

第三章 基于YOLOv3的手势检测算法 10

3.1 引言 10

3.2 YOLOv3目标检测算法介绍 10

3.2.1 网络结构 10

3.2.2 网络的输出和损失函数 13

3.3 实验及分析 15

3.3.1 数据集简介 15

3.3.2 实验环境和实验设置 15

3.3.3 评价指标 16

3.3.4 实验结果与分析 16

3.4 本章小结 18

第四章 基于深度学习的单目深度估计算法 20

4.1 引言 20

4.2 FastDepth单目深度估计算法 20

4.2.1 网络结构 21

4.2.2 网络的输出和损失函数 22

4.3 基于YOLOv3的单目目标深度估计 22

4.3.1 网络输出修改 22

4.3.2 损失函数 22

4.3.3 网络的预测和训练 23

4.4 实验及分析 23

4.4.1 数据集简介 24

4.4.2 实验设置 26

4.4.3 评价指标 26

4.4.4 实验结果与分析 27

4.5 结合目标检测和深度估计算法的绘图应用 31

4.5.1 手势绘图程序简介 32

4.5.2 手势绘图程序操作及示例 32

4.6 本章小结 34

结论 35

1. 论文工作总结 35

2. 工作展望 35

参考文献 37

致谢 40

# 第一章 绪论

## 1.1 研究背景与意义

近年来，随着科学技术的发展，电子设备变得越来越智能，功能也越来越多，与之相关的交互方式也越来越复杂。在工作、学习、娱乐等多种场景中都会涉及到大量不同的人机交互方式，如键盘、鼠标、触摸屏、语音等等。在众多的交互方法中，基于手势控制的交互方式是比较便捷且易于理解和学习的。目前，手势交互技术在已经在很多场景下得到了应用，比如体感游戏中以手势控制角色的行为，大大提高了玩家的沉浸感；在智能家居的控制中，以手势在远距离控制各种家电的工作，方便了人们的生活；在虚拟现实以手势对仿真环境做出精细的控制等等。上面提及的这些应用场景都与手势识别技术相关，可见基于手势的人机交互方式有很好的发展前景。

深度学习技术是目前解决图像领域的目标识别与检测任务的主流方法，相较于传统的识别与检测算法，使用多层卷积神经网络的深度学习能更好地提取数据中的复杂特征，在复杂场景下的泛化能力和鲁棒性要强于传统算法。基于深度学习的手势检测模型一般是使用手势数据集对所选择的模型进行训练得到，相较于传统的手势检测算法，深度学习方法所需的手势数据较多，计算资源消耗较大，但是能够进行端到端的输出，训练较为方便，而且有较好的扩展性。

目前在手势检测领域中，大多数检测方法都是对手势类别及其在平面上的坐标位置进行提取，而对手势的深度信息进行提取的相关工作较少。如果能够从RGB彩色手势图像中得到手势目标的深度信息，就可以拓宽手势交互的应用范围。

## 1.2手势识别国内外研究现状

手势识别的主要目标是让用户能够通过手势与机器进行自然地交流。手势识别研究最早可以追溯到上世纪八十年代，早期研究者主要是利用装有传感器的硬件设备去采集手势相关信息，例如数据手套。后来，由于数据手套价格昂贵，光学标记法诞生并替代了数据手套。虽然这两种手势识别方法的准确率和稳定性较高，但是这两者均需要额外穿戴设备，增加了用户的使用负担，不符合手势识别交互自然舒适的初衷。在种种原因下基于视觉信息的手势识别被越来越多的研究人员所重视。

一般情况下，基于图像的手势识别技术包括使用传统特征的和使用深度学习的技术。 使用传统特征的手势识别指研究者结合自己的专业知识人为提出的特征[1]，包括SIFT特征等。这种方法有很大的局限性，容易受到光照变化、人体、肤色以及复杂背景的干扰，识别精度通常不高。而深度卷积神经网络不同，它具有强大的自主学习能力，能够处理很多复杂的计算机视觉问题，因此逐渐成为手势识别的研究热门方向。Yingxin[2]使用卷积神经网络来自动提取手势的空间和语义特征，进而实现手势识别。他们在剑桥手势数据集和自己搭建的数据集上做了测试，发现比起传统的SIFT特征与SVM结合的方法，使用卷积神经网络进行特征提取和分类所得的结果要好很多。林汇丰[3]研究了基于YOLOv3的手势检测模型，并使用通道剪枝方法对模型进行了压缩和加速，在检测精度不变的情况下大大减少了模型的推断用时。Kim等人[4]提出了一种结合CNN和加权模型最小最大网络进行动态手势识别的方法。这两个网络分别进行特征提取和特征分析，实验表明该方法能够有效降低特征点时间或空间变化的影响。Molchanov等人[5]提出了三维卷积神经网络的动态手势识别方法，他们首先对视频数据进行时间和空间上的数据增强以防止过拟合，然后直接将3D-CNN直接应用于整段视频序列，最终在VIVA数据集上取得了77.5%的分类准确率。苗启光等人[6]将残差网络(ResNet)[7]和三维卷积网络结合，用于检测视频序列帧中的手势。刘文达[8]研究了基于SSD改进的手势检测模型，设计了一种主网络结构采用MobileNet的轻量手势检测模型Gr-Net，可部署在移动端进行实时检测。倪紫涵等人[9]设计了一种轻量化的基于YOLO的手势检测模型，该模型是在YOLOv2模型[10]基础上进行优化的，使用空间增强模块提高其对图像中较小的手势的检测精度，还使用选择性删除通道修剪方法对算法进行了加速。相比直接使用YOLOv2进行手势检测，在提高了1.26%的精度的基础上，将模型大小降低到了原模型的1.6%。

## 1.3论文研究内容和结构安排

本文主要研究使用深度学习方法对RGB图像进行手势检测，全文分为五个部分，各部分具体内容概括在下面：

第一章简述了手势检测的研究背景和意义，介绍了手势检测技术在国内外的研究现状，最后概括了论文的内容，列出了本论文的结构。

第二章介绍了深度学习技术的基本原理，简单介绍了深度学习中常用的一些网络中的基础结构。之后也对基于深度学习的目标检测和单目深度估计的常用算法及其原理做了介绍。

第三章是基于YOLOv3模型[11]的手势检测算法研究。介绍了所使用的YOLOv3模型的网络结构及其原理，然后对算法的训练和预测过程做了详细说明，比较了使用不同网络和训练方法下检测算法的性能。

第四章是基于深度学习的深度估计算法研究。首先对实时的单目深度估计算法FastDepth[12]进行简单的介绍，包括模型的特征提取主网络结构，深度估计网络结构，输出的处理等原理，然后对算法的训练和预测过程做了详细说明。之后，通过修改YOLOv3模型的输出结构并使用带有目标深度信息的训练数据进行训练，使其能够输出目标的深度信息，并对比了两种方法在提取目标深度信息上的性能，最后通过实验表明基于深度学习的深度估计配合目标检测算法能够检测出手势的类别和其在三维空间中的位置。

第五章是论文的总结和展望，在该部分中对论文的工作做了简短的总结，提出了论文中工作的缺陷，最后也展望了未来在此工作的基础之上还可以进行的工作。

# 第二章 相关技术基础

本章将介绍在本文中所使用到的一些技术，包括深度学习中使用的网络相关技术、深度学习的目标检测和基于深度学习的单目深度估计。

## 2.1 卷积神经网络相关技术

卷积神经网络是深度学习算法中的重要结构，尤其在视觉领域，由于其能够模拟人类视觉感知的过程，它被广泛应用于各类视觉算法当中。对常用的卷积神经网络来说，网络一般分为输入层、卷积层、池化层、全连接层、输出层等。

### 卷积层

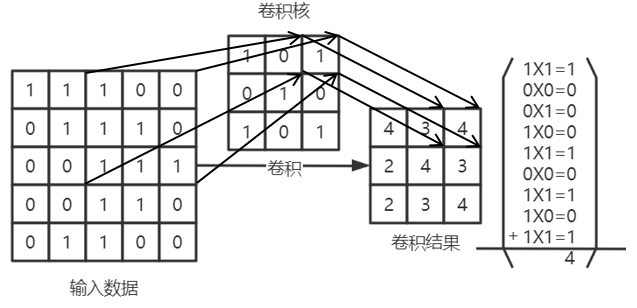
卷积层是卷积神经网络中最重要的结构，用于对输入数据的特征进行压缩和提取。卷积主要是指由卷积核在图像上的所有小区块依次进行如图的卷积操作。在卷积层，每个卷积核在一次计算时只对它所负责的局部图像像素区块进行感知，同一个卷积核使用相同的权值对图像的不同区域进行卷积，最终达成降低输入数据的维数并提取输入数据的重要特征的功能。

图2-1 二维卷积过程

### 池化层

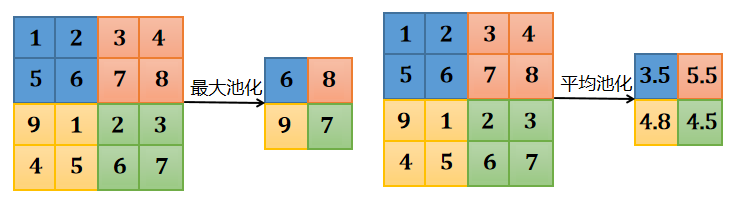
池化层又称降采样层，池化层的所做的池化操作也是一个固定大小的卷积核在图像上的所有小区块依次进行计算，但池化层一般不含有参数，也不计算卷积，而是取计算区域的平均值或最大值进行输出。池化层最大的作用是降低特征维度，保留相对重要的特征，并提取具有尺度不变性的特征。

图2-2 二维池化过程

### 全连接层

全连接层在数学上相当于矩阵乘法，一般位于卷积神经网络的后端。其作用是对前面的特征提取网络所提取到的特征进行处理，输出对特征的处理结果，结果一般会根据任务的不同而进行不同的额外处理。

### 激活函数

激活函数一般位于每一个计算层之后，它主要通过对前方网络的输出做非线性变换，给网络添加拟合非线性函数的能力。目前在深度学习算法中出现最多应用最广泛的激活函数有Sigmoid函数和ReLU函数。

Sigmoid函数，表达式为，它可以把输入的数值变换到[0,1]范围内。Sigmoid函数的优点是上下界稳定且导数较为简单，在前向计算的时候稳定的上下界可以保证输出的稳定，在反向计算时较简单的导数可以提高计算效率。其缺点是由于其在输入大于或小于一定值后就稳定在一个数值上，在反向计算导数时导数就可能为0，影响了训练过程。

相比Sigmoid函数，ReLU目前使用较多的激活函数，它的计算方法很简单：前方的网络输出小于0时，ReLU输出0；前方网络输出大于0时，ReLU函数将输入值直接输出，对应的函数图像如图2-3。ReLU激活层在反向传播时其导数恒为1，计算效率很高，而且梯度较为稳定，因此得到广泛使用。由于ReLU对负值输入不敏感，会在网络的训练中产生一些问题，之后又出现了LeakyReLU，其图像如图2-4，为一个分段函数，在输入小于0时也有不为零的输出。

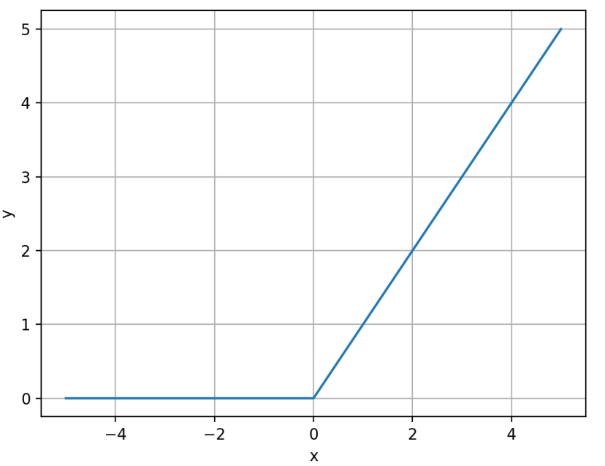


图2-3 ReLU函数

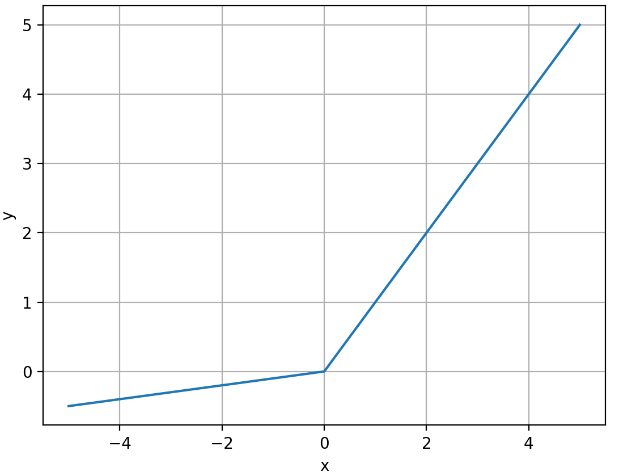


图2-4 LeakyReLU函数

### 损失函数

损失函数主要用来评价算法预测值和真实值之间的误差，然后根据误差来对算法进行优化。在算法的训练过程中，损失函数的输出越小，就证明算法在训练集上越准确。最常用的损失函数包括在回归问题中广泛应用的L1损失函数，均方误差(MSE)损失函数，在分类任务中大量使用的交叉熵损失函数等。

L1损失函数又叫最小绝对值误差，其目的是将真实值与估计值的绝对差值的总和最小化，公式如下：

（2-1）

均方误差（MSE）是回归损失函数中使用比较广泛的误差，它计算真实值和预测值差值的平方和，再对其求均值，公式如下：

（2-2）

交叉熵损失函数，在分类的任务中使用比较广泛的损失函数是交叉熵损失函数，公式如下：

（2-3）

上式中，为类别数目，为类别标签值，为算法所预测的属于第*i*类的概率。

## 2.2 基于深度学习的目标检测

基于深度学习的目标检测主要有以YOLO[13]为代表的One-Stage模型和RCNN[14]为代表的Two-Stage模型[15]。RCNN和其他双阶段模型一般都是先通过一些方法生成很多候选框，再通过分类算法或者分类网络对这些候选框的特征进行提取并分类。YOLO和其他单阶段模型，比如SSD[16]等，则将目标边框定位的问题视为回归问题进行处理，将图像输入到网络中之后经过网络的计算直接输出目标的检测结果。由于单阶段方法不用产生和过滤众多的候选框，其检测的速度比双阶段的要快得多。但由于只能在图像中限制的位置产生较少的检测框，单阶段方法的检测精度比双阶段的低一些。考虑到对算法运行速度的要求，下面重点介绍YOLO系列算法的研究现状。

### 2.2.1 YOLO系列算法简介

2015年Joseph Redmon等提出的YOLOv1模型是目标检测的One-Stage方法，在提出后因为其推断速度相较于RCNN大幅增加，且检测精度也很高，迅速成为目标检测算法的领先者。YOLOv1模型通过对整张图片进行处理来进行目标检测，主要步骤是先将缩放输入图像到固定的尺寸以便模型进行处理，之后通过特征提取网络对图像中的目标的特征进行提取和压缩，最后用全连接层输出目标检测的结果。由于存在预先的网格划分，该模型难以检测尺寸较大或尺寸较小的物体，这很大程度上影响了它的精确度。

YOLO模型具体的检测原理如下：将整张尺寸匹配的图像直接输入到网络中，将图像分成网格，并预测每个网格的类概率和边界框。以一个的RGB彩色图像为例。首先对它进行网格划分，平均分成的网格，每个网格只负责预测一个目标，且该目标的中心必须位于这个网格中。之后让每个网格都预测两个边界框的坐标和置信值，和该网格所负责预测的目标属于每个类别的概率。每个边界框预测信息由五个值组成：边界框的中心点在网格中的相对坐标，边界框的高度和宽度值，边界框中存在对象的概率。一个网格只会负责一个目标，每个网格中将有多个边界框。在训练过程中，对于单个网格所预测出的多个边界框，分配与标签框重叠度最高的预测框来负责检测该目标。

2017年YOLOv1的改进版本YOLOv2被提出。YOLOv2更换了所使用的主网络结构，采用作者借鉴GoogLeNet结构[17]所创建的Darknet-19网络，在网络结构上组合使用和的卷积层，在卷积层后增加了批归一化层，在ImageNet数据集上以较低分辨率进行预训练，之后切换为较高分辨率使其更好地进行检测任务。在网络基础结构方面，YOLOv2中的Darknet-19还将浅层特征连接到了网络后部，使检测结构能接收到更多图像中的细粒度特征，能更好地检测图片中远处的目标和小目标。在检测方面，YOLOv1是网络的输出经过简单的变换后直接预测目标框的位置和大小，YOLOv2则使用了了先验框机制，通过对数据集中的各类目标的目标框进行K均值聚类，得到训练集中目标大小的先验知识，并作为先验框数据来辅助检测网络的训练，给算法提供了目标框常见位置和大小的先验知识，这种对网络输出往真实值方向的约束提高了模型的精确度，也使得模型更容易训练，收敛更快。

2018年Joseph Redmon等发布了新的检测模型YOLOv3，YOLOv3再次更换了其主网络结构，采用作者借鉴ResNet的残差结构所创建的Darknet-53网络，使用由卷积层、激活层、批归一化层和残差连接所构成的残差块结构来作为主要的网络基础结构。在YOLOv2的Darknet-19网络中将浅层特征连接到了网络后部使检测网络获得更多图像中的细节，在YOLOv3的Darknet-53网络中则没有使用这种网络层之间的跳跃连接，而是直接将浅层特征输出到检测结构中进行检测任务，其所输出的三种不同大小的特征图分别用来检测图像中不同大小的物体。

目前最新的YOLOv4[18]则使用了各种技巧微调了算法，在模型大小和检测性能上达到了很高的水平，而且有各种侧重点不同的版本可选。

## 2.3 基于深度学习的单目深度估计

深度估计算法的功能是估计图像中像素、物体或场景的绝对或相对深度，输出所估计的深度信息。单目深度估计是指使用机器学习模型估计单个RGB摄像头所拍摄图像的深度信息，分为无监督、半监督和有监督方法[19,20]。无监督和半监督方法的训练过程依赖于单目视频或立体图像对，利用相邻视频帧，产生近似的多视角图像，对相机位姿和场景深度进行估计，训练后的深度网络在测试过程中可从单目图像预测深度图[21]。有监督的方法一般是按照回归问题进行处理，在训练时根据标签深度图，通过深度学习模型对单目RGB图像进行深度估计，利用所输出的预测深度图和实际深度图之间的误差对算法进行监督。

2014年David Eigen等人[22]首先用深层卷积网络的方式尝试单目深度估计，该模型的主网络分为两个模块，一个是大尺度网络，大概预测整个图片中场景的深度；另一个是精细网络，在场景粗略深度的基础上，对图片中的细节部分做更进一步的深度预测和调整，最后使用均方误差作为算法预测与真实值的误差。此后对该算法也有较多的改进，包括联合训练、网络间跳跃连接[23]等。在实时深度估计方面，2019年，Diana等人提出了一种实时的端到端的深度估计算法FastDepth，网络为编码-解码器结构，实现了端到端的深度估计，在保证准确率的情况下，大大提高了模型的计算速度，可在嵌入式设备上实时运行。

## 2.4 本章小结

本章首先介绍了在深度学习中广泛使用的卷积神经网络相关结构的原理，介绍了网络中各层的原理和构成，然后对基于深度学习的目标检测算法进行了介绍，其中重点介绍了YOLO系列单阶段目标检测模型，最后简单介绍了基于深度学习的单目深度估计算法。

# 第三章 基于YOLOv3的手势检测算法

## 3.1 引言

YOLOv3是YOLO目标检测网络系列的第三代版本，经过一系列的迭代改进之后该版本较好的兼顾了目标检测的速度与精度。

## 3.2 YOLOv3目标检测算法介绍

### 3.2.1 网络结构

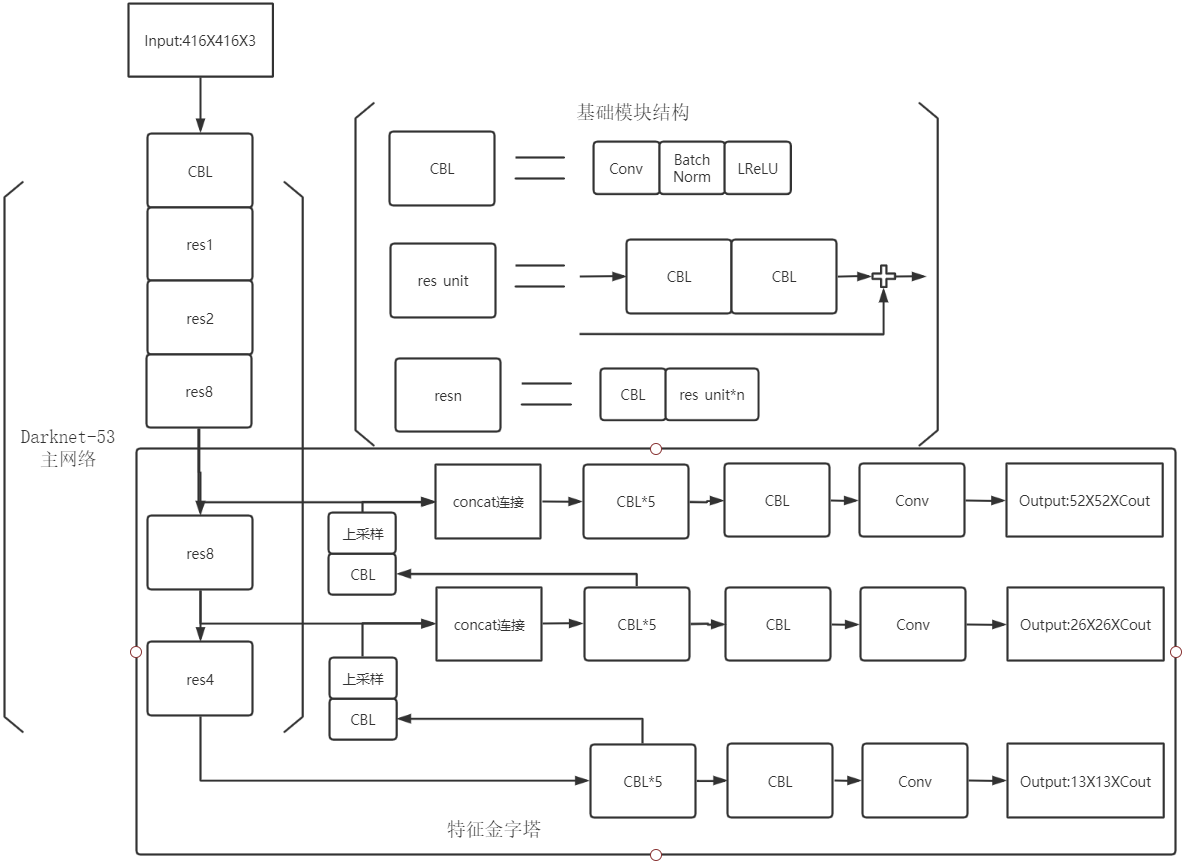
YOLO系列网络进行目标检测的思路都是基本相同的，首先将输入图片均匀划分成正方形网格(grid)，然后由每个网格来判断其所在的区域中是否有目标，若区域中有目标，就预测目标的具体位置和形状大小，最后综合所有网格得到的预测结果得到最终的预测结果。下面将详细地介绍YOLOv3的模型结构。

图3-1 YOLOv3网络结构

在特征提取方面，YOLOv3使用了全新的主网络Darknet-53。其主网络结构借鉴了ResNet，使用了大量的小卷积核进行卷积，并在网络层之间加入了跳跃连接，也就是残差结构。Darknet-53的结构图如3-1中左边的特征提取部分所示，大部分特征提取任务都通过CBL模块来完成，其主要组成为卷积层，负责提取特征；批归一化层，负责将前方网络的输出进行归一化处理；LeakyReLU激活函数，负责对前方网络的输出进行非线性变换。图中的res\_unit 表示残差单元，由两个CBL模块和残差连接组成；resn表示由一个CBL模块和n个res\_unit模块组成的模块。

在目标检测方面，YOLOv3为提高对不同大小目标的检测精度，使用了特征金字塔结构，输出包括了多个不同尺寸大小的特征图，在这些特征图上都进行检测任务。一般情况下，在特征提取网络中，浅层特征的细节较多，能更好地检测小物体。由于未经过过多的下采样，浅层特征与原像素之间的位置对应关系较准确，利于对物体进行定位。但是浅层特征所含的无用信息也很多，无法直接使用。深层特征图的特征语义层级较高，所含有的大部分为目标信息，可直接使用，但缺点是分辨率较低。特征金字塔结构将深层特征通过上采样和卷积操作与浅层特征图进行融合，得到各层语义性都较强的融合特征，可以提升模型对小目标的检测能力以及定位目标的准确度。图3-1右下是YOLOv3的特征金字塔结构，共有3层，分别以Darknet-53中第一个res8块，第二个res8 块和最后一个res4块的输出向量为基础构建而成。由于特征图分辨率的差异，低分辨率的特征图在与高分辨率特征图进行融合前会先进行上采样。使用了特征金字塔结构的YOLOv3模型对于小目标的检测性能有了较大的改善。

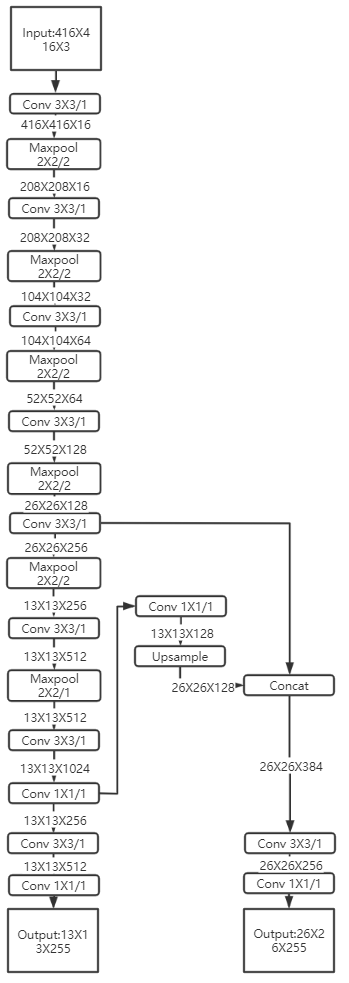
除了检测精度较高的使用Darknet-53作为主网络的YOLOv3之外，YOLOv3的论文作者还在其个人网站上介绍了检测速度较快的使用轻量化主网络的TinyYOLOv3，其主网络结构如图3-2。

图3-2 TinyYOLOv3网络结构

TinyYOLOv3的网络结构较YOLOv3简单了不少，去掉了一些特征提取层，去除了残差连接，最后的检测结构也只保留了两个较小的预测分支。在COCO数据集[24]上，TinyYOLOv3的检测精度比YOLOv3低了23.3%mAP，但运行速度要快很多，适合在不具有高性能GPU的设备上进行较为简单的目标检测任务。

### 3.2.2 网络的输出和损失函数

图3-3 YOLOv3每个单元格的检测输出

对于使用Darknet53结构的YOLOv3，在输入一张的RGB图像后，最终的输出为三个不同大小的特征图：，，。对于的输出，可以看作是将原图像划分成相等大小的个单元格，预测特征图中的每个单元格与原图上对应位置的单元格是相关的，每个单元格负责检测对应位置附近的一片区域中是否有物体。在YOLOv3中，预测特征图中的每个单元格中能够预测3个对应3种不同尺度先验框的边界框，分为大中小3种。先验框是边界框的默认宽高尺寸，通过*K*均值聚类的方式从训练集中得到，网络只需在该默认值的基础上预测修正量，而不需直接预测出边界框的长和宽，降低了模型学习的难度。单元格的输出按如下含义解释：和是对应格子左上角的边界框的中点坐标偏移，和表示相对于先验框的边界框大小的修正值，*p*表示目标中心处于该格子中的置信度。到表示若该格子负责该目标时该目标属于各个类别的概率。因此，预测特征图的通道数为，其中*B*表示每个单元格能预测的边界框的个数，*C*表示类别的数目。网络会输出个预测边界框，之后使用置信度阈值过滤掉置信度过低的预测框。然后将保留的预测框的位置和大小数据转换成边界框的实际中心坐标和长宽尺寸，转换公式如下：

（3-1）

（3-2）

（3-3）

（3-4）

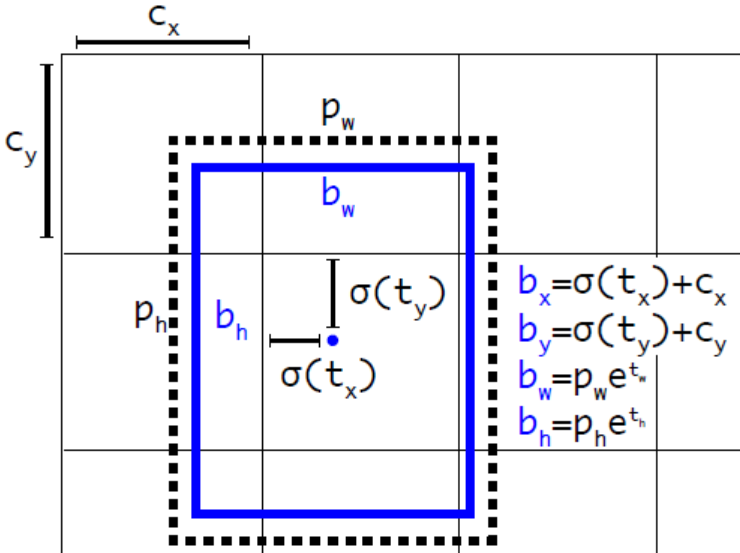
其中、、和分别表示目标中心点的横纵坐标、预测框的宽和高，和表示单元格左上角的坐标，和分别表示先验框的宽和高，函数为常用的Sigmoid函数，用于将预测的中心坐标限制在0~1之间。由于可能存在多个预测框同时检测同一个物体的情况，最后使用非极大值抑制(NMS)的方法去除重复的预测框，保留下来的少量的预测框将作为最后的预测结果。

图3-4 模型预测值与实际值的转换关系图[11]

在将原始标签数据转换成可供模型使用的训练数据时，首先计算出标签框的中心落在预测特征图上哪个单元格中，然后计算出标签框与该单元格上的*B*个先验框的交并比，交并比最大的先验框负责对该真实边界框进行学习。经过上面的分配后，每个标签框都与一个先验框对应，然后将矩形框的实际中心坐标和长宽数值逆向转换成相对于先验框的修正量，并设置目标置信度和类别信息。对于未匹配到目标物体的先验框，则将其置信度设置为0。

YOLOv3的损失函数包括3个部分：关于边界框位置和大小的损失，关于目标置信度，关于分类的损失。总损失由3个部分的损失加权得到，其计算公式如下：

（3-5）

其中的，，分别表示对应损失部分的加权系数。在初始的YOLOv3实现中一般是分为和两部分进行计算，现在流行的实现中则大多数使用了GIoU损失[25]。对于和则对每个类别单独使用二元交叉熵损失。

## 3.3 实验及分析

### 3.3.1 数据集简介

本章的手势数据检测实验是在自己制作的手势数据集上进行的。该数据集是一个规模较小的数据集，共包含4种手势的1200张RGB彩色图片。

原始数据集被划分成了训练集、验证集、测试集，其中训练集中共有960张图片，验证集中一共有120张图片，测试集中有120张图片，每张图片中只有1个目标。由于数据量较少，我们将训练集和验证集进行了合并得到一个更大的训练集，测试集则使用原始的测试集，数据集中的样本示例如图3-5。



图3-5 自制手势数据集中的样本示例

### 3.3.2 实验环境和实验设置

本实验在装有Windows 10 64位家庭版操作系统的电脑上进行，CPU型号为Intel(R) Core(TM) I7-10400F，主频为2.9GHz，内存大小为16GB，硬盘为固态硬盘，GPU型号为GeForce RTX 3070显卡，显存频率14000MHz，显存容量8GB。

本实验所使用的YOLOv3模型是使用Python语言，基于Pytorch库搭建的，可在CPU或者GPU上进行训练或测试。

训练时，由于显存的限制，将训练数据批大小设置为8，学习算法则使用了常规的带动量的随机梯度下降算法(SGD)，初始学习率设置为0.002324，动量项参数设置为0.97，最大迭代次数为150，学习率曲线如下图。

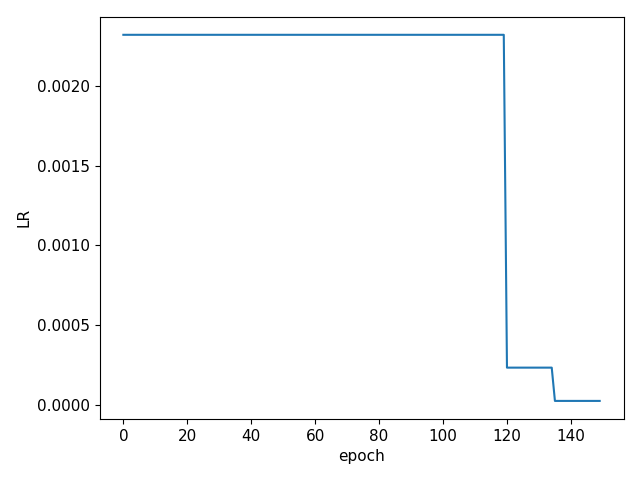


图3-6 训练YOLOv3检测模型时的学习率曲线

### 3.3.3 评价指标

本章给出的YOLOv3模型进行的是常规的多类别的目标检测任务，所以可以使用检测任务常用的mAP来评价模型的精确度。

mAP是数据中的每个类别的检测精确度(AP)的平均值，对于目标检测任务来说，mAP越高，则模型在整个数据集上的检测精确度越高。具体某一类别目标的AP的计算方式如下：对于某个目标类，以模型的召回率为x轴，精确率为y轴画出曲线，其下方所包围的面积就是AP值。对所有类别的AP值求平均，所得的就是模型在该数据集上的mAP。

### 3.3.4 实验结果与分析

本文使用2种不同的网络结构，第1个为标准的YOLOv3，以Darknet-53为主网络；第2个模型为计算速度较快的TinyYOLOv3模型，以YOLOv3-tiny网络为主网络。除了主干网络不同之外，模型的其他训练设置均一致，两个模型经过训练后的实验结果如表3-1。

表3-1 不同主网络结构的YOLOv3模型在自制数据集上的实验结果

|  |  |  |
| --- | --- | --- |
| 训练设置 | 平均精确率mAP(%) | 推断速度(FPS) |
| YOLOv3 | 99.5 | 47 |
| YOLOv3-tiny | 97.2 | 144 |

可以看出，两种网络结构的模型均能实现对手势的检测，YOLOv3-tiny由于使用了较轻量化的网络，相比于使用Darknet-53的模型，其mAP低了2.3%，但由于参数量的减少，推断速度提高了将近100FPS，适合在不具有高性能GPU的设备上执行对简单目标的检测任务。

在训练方面，两种模型对应的训练损失函数曲线如图3-7和图3-8。

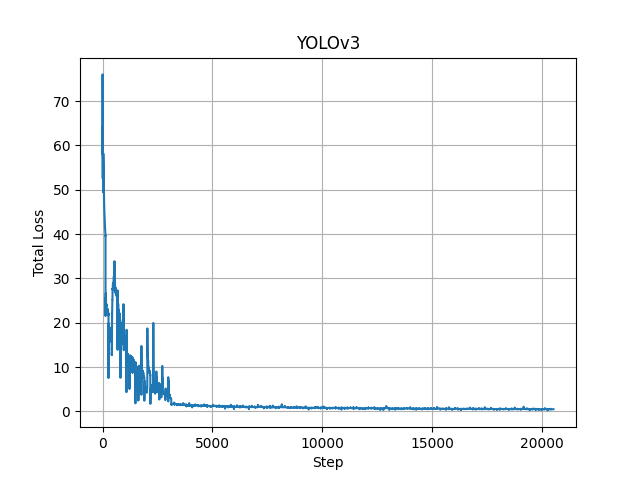


图3-7 YOLOv3的训练损失曲线

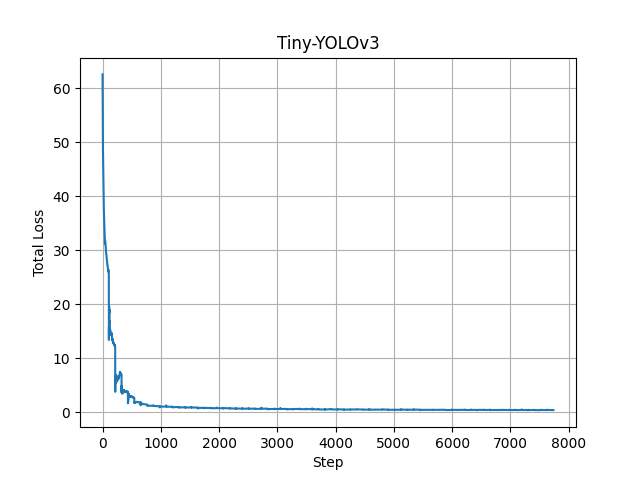


图3-8 TinyYOLOv3的训练损失曲线

由图3-7和图3-8可以看出YOLOv3模型和TinyYOLOv3模型在手势数据集上进行训练时都能够收敛，TinyYOLOv3由于主网络较小，其收敛速度较快，在第1000个step时就已经接近收敛。

使用该模型检测手势的效果如图3-9。

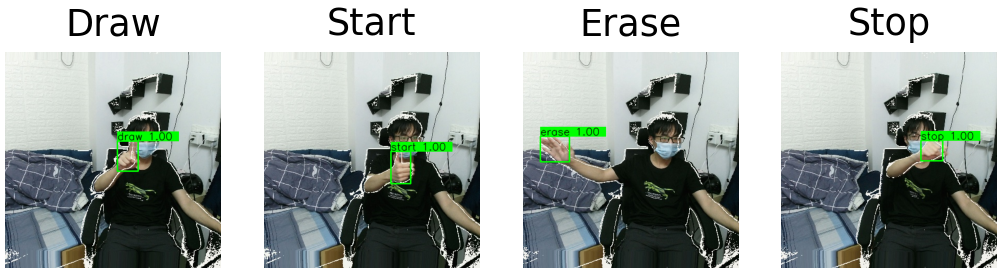


图3-9 YOLOv3检测4种手势的效果

图3-9是使用YOLOv3模型在自制手势数据集的测试集上进行检测的结果可视化，模型的预测结果以绿色的矩形框标出。从图中可以看到，该模型对数据集中的4种手势目标都能进行较为精确的定位，并准确判断手势类别。

## 3.4 本章小结

YOLOv3目标检测网络是一种兼顾速度和精度的目标检测框架，本章中首先对其网络结构、训练和测试方式等做了较详细的介绍，之后使用自制手势数据集训练YOLOv3及其变种TinyYOLOv3得到了2个手势检测模型，模型在相应的测试集上都取得了较好的检测效果，最后对比了YOLOv3和TinyYOLOv3的检测性能。使用YOLOv3模型能够很好地进行手势类别和二维位置的检测，但无法得到手势的深度信息，下一章节将介绍基于深度学习的单目深度估计方法，并尝试对本章节所得的检测模型进行修改以支持估计目标深度。

# 第四章 基于深度学习的单目深度估计算法

## 4.1 引言

在第三章中介绍了基于YOLOv3的手势检测算法，该算法能够实现对手势类别和平面位置的检测，对于一般的手势控制的应用已足够使用，但如果能够从RGB图像中得到手势位置的深度信息，则能在使用低成本的硬件的情况下提高手势控制的交互能力。从数学上讲，在没有相机信息以及第二帧图像的情况下是无法对图像中物体的深度进行估计的，因为无法通过三角化方法对目标或者像素的深度进行求解。然而，人类在生活中通过学习场景中各个物体之间的关系获得了单眼估计深度的能力。因此，仿照人类的学习能力，利用深度学习技术是有可能从单张RGB图中估计得到深度信息的。随着技术的发展，基于深度学习的深度估计技术有了一些不错的成果。本章将首先介绍实时的单目深度估计算法FastDepth，其网络输出为像素级别的深度图，然后介绍基于YOLOv3的目标深度估计算法，其除了输出目标的类别和二维平面上的坐标外，还可对目标中心点的深度进行估计，最后对两种深度估计方法的优缺点进行了比较。

## 4.2 FastDepth单目深度估计算法

FastDepth是一种基于深度学习的单目深度估计算法，其使用了编码-解码器结构对图像进行处理，由于采用了端到端的计算方式，和轻量的Mobilenet网络[26]作为特征提取网络，算法的运行速度很快，能够在嵌入式设备(NVIDIA Jetson)上以较高的FPS运行。模型在NYU Depth V2 dataset上进行训练，基本实验结果如图4-1所示，可以看出相对于其他算法，FastDepth的估计精度稍低，但是运算速度有着大幅提升。

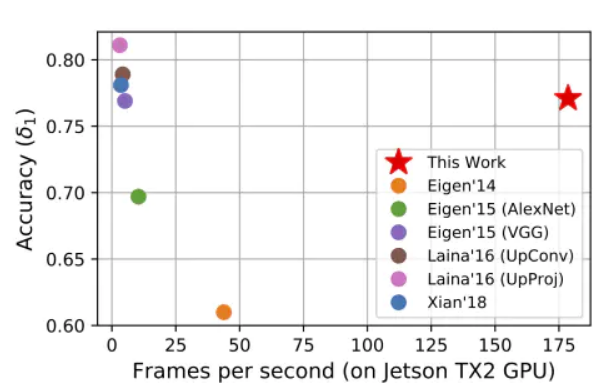


图4-1 FastDepth与一些其他深度估计算法的性能比较[12]

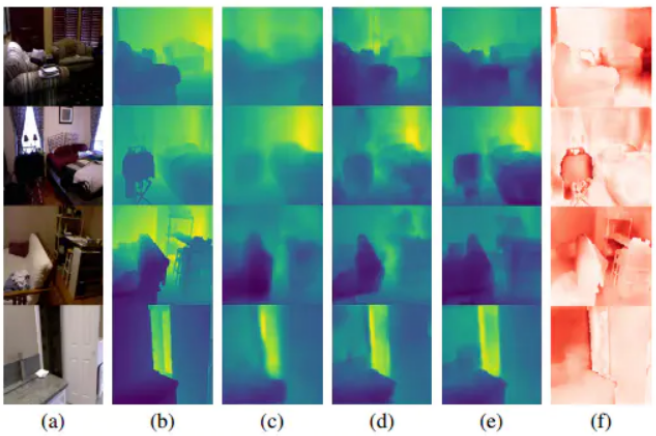


图4-2 FastDepth在NYU数据集上进行深度估计的可视化效果[12]

FastDepth在NYU数据集上进行深度估计的可视化效果如图4-2所示，(a)为输入的RGB图像，(b)为对应RGB图像的深度图真值，(c)(d)(e)分别为未进行模型剪枝、带有跳跃连接且未剪枝和带有跳跃连接并进行模型剪枝的FastDepth模型的深度估计结果，(f)为深度估计值与深度真值间的误差图。

### 4.2.1 网络结构

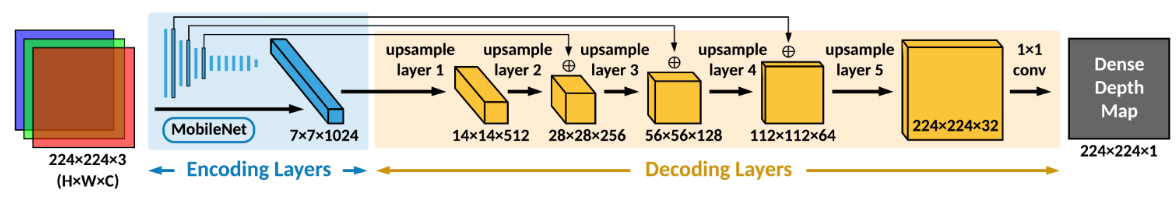


图4-3 FastDepth的网络结构[12]

FastDepth模型的主网络主要包括三个部分：编码器网络，解码器网络，网络间的跳跃相连。主网络中的特征提取部分，也就是编码器部分，采用了Mobilenet特征提取网络，将输入的的RGB图片输出为的特征；解码器部分由5个级联的上采样层和一个单独的点卷积层组成，称为NNConv5。每个上采样层使用卷积，输出大小是输入大小的一半，卷积之后通过最近邻插值的方法将输出特征向量的分辨率加倍。中间三次上采样计算的结果通过跳跃连接的方式分别与编码器部分的特征进行了特征融合，有助于解码器重建更细节的密集输出，这是在UNet语义分割模型[27]中被使用并且被证明有效的技术。最后使用的卷积得到深度图。

### 4.2.2 网络的输出和损失函数

FastDepth网络最终的输出为和原图像按像素对应的深度图，再乘以放大系数后可直接作为估计深度使用，无需复杂后处理过程。模型的损失函数为像素级深度估计值和标签值的L1损失，即最小化估计深度和标签深度间的距离。

## 4.3 基于YOLOv3的单目目标深度估计

在4.2所介绍的FastDepth模型中，最终要获得的是所有像素的深度值，所以在使用编码器网络提取特征后，必须使用解码器网络才能输出像素级的深度图。而在YOLOv3目标检测算法中，所关注的只是目标中心的深度值，所以使用特征提取网络提取特征后，只需调整检测部分网络的结构，就能够输出目标中心深度的估计值。

### 4.3.1 网络输出修改

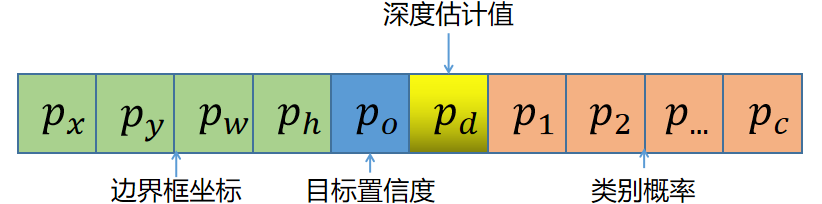
所使用的模型结构为第三章中的YOLOv3和TinyYOLOv3模型，为了使模型支持估计目标的深度，需要修改网络最后的检测输出。在每个边界框所预测的向量中增加一个深度维度*d*，修改后每个边界框的输出向量如图4-4。

图4-4 修改后的YOLOv3的每个边界框的输出

### 4.3.2 损失函数

对于其他的输出，包括边界框坐标，目标置信度和类别概率，其损失函数不变；对于深度信息的输出，为使其与标签值匹配，按如下公式变换后再进行损失值计算。

（4-1）

其中表示用于参与深度损失计算的预测深度，表示网络输出的未经处理的预测深度，函数为常用的Sigmoid函数，用于将预测的中心坐标限制在0~1之间。

网络的深度估计值输出经变换后其输出范围在-1~1之间，如图4-5。

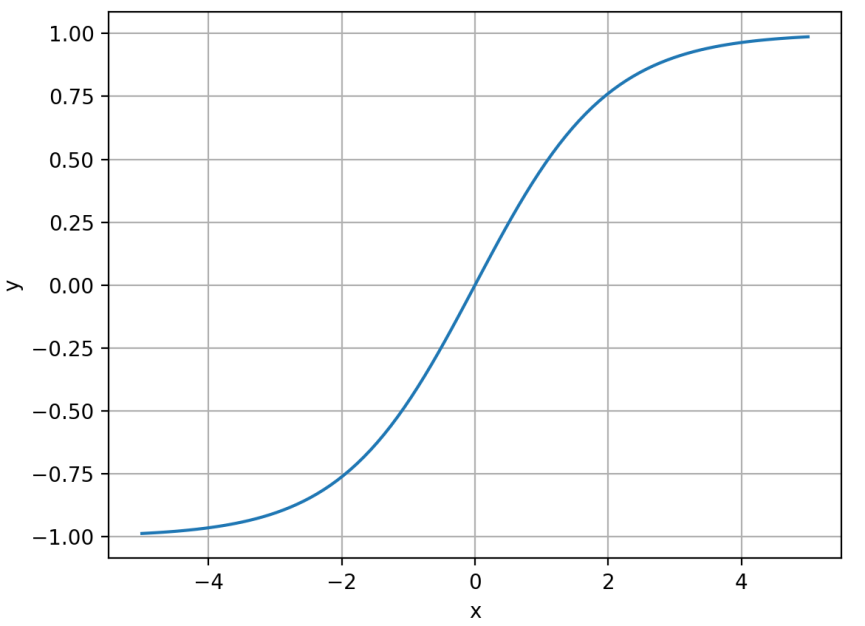


图4-5 原始深度估计值的变换函数图像

深度部分的损失函数使用L1距离损失函数进行度量，即：

（4-2）

整个模型的损失函数也在YOLOv3检测模型的损失函数上添加深度损失得到，损失函数如下：

（4-3）

其中为深度损失的权重，在训练中设为10。

### 4.3.3 网络的预测和训练

网络的训练和预测与第三章中的YOLOv3检测模型相同。

## 4.4 实验及分析

实验中的PC配置与训练YOLOv3手势检测网络时所使用的PC相同，实验中训练所用的数据集都使用自制的手势深度数据集。

### 4.4.1 数据集简介

实验所用的手势深度数据集为第三章中自制RGB手势数据集所对应的深度数据，其中包含1200张共4类的手势图像以及对应的深度图像，深度图像使用Kinectv2深度相机采集得到。

图4-6 手势数据集中的RGB图像及对应的可视化深度图

图4-6下方的4张图为经过可视化处理的深度图，原始深度数据为的浮点数矩阵，深度数值的范围在0至8000之间（单位为mm）。

对于FastDepth模型，在训练时将标签深度值除以8000得到归一化的处在0~1之间的可用深度标签，用于深度损失的计算。

对于YOLOv3+depth模型，由于是对目标中心点深度进行估计，所以对一张RGB图像只需要目标中心点的深度数值来作为标签。我们提取了数据集中所有手势目标的中心点深度，其分布如图4-7。

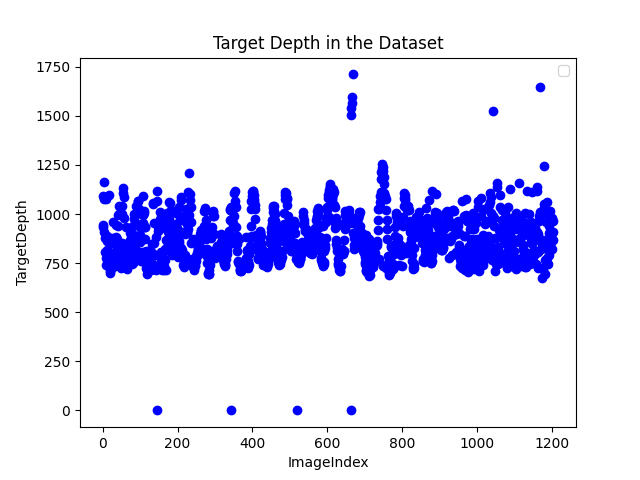


图4-7 数据集中手势目标中心点的深度值分布

其中有一些离群点，去除后的分布如图4-8。

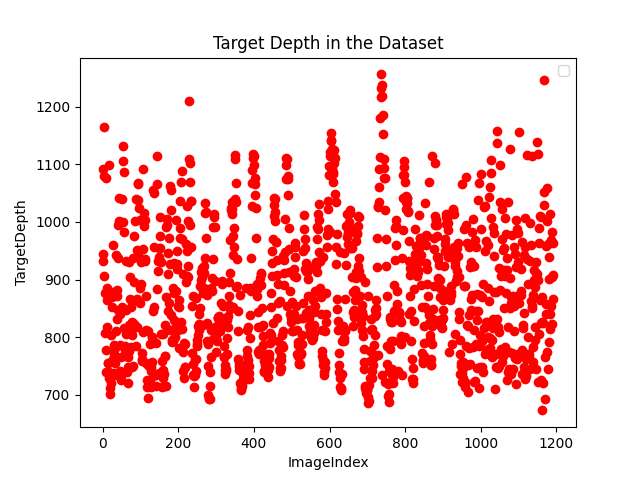


图4-8 去除离群点后的数据集中手势目标中心点的深度值分布

由图4-8可知，手势目标中心的深度范围位于650~1300之间(单位为mm)。为了让网络能够更容易地学习对深度的估计，我们将目标中心深度数值按公式4-4进行处理，得到可用的目标中心深度标签。

（4-4）

其中表示用于参与深度损失计算的真实深度值，表示采集所得的未经处理的真实深度值，*n*为数据集中的样本数目，处理后的数据分布如图4-9。

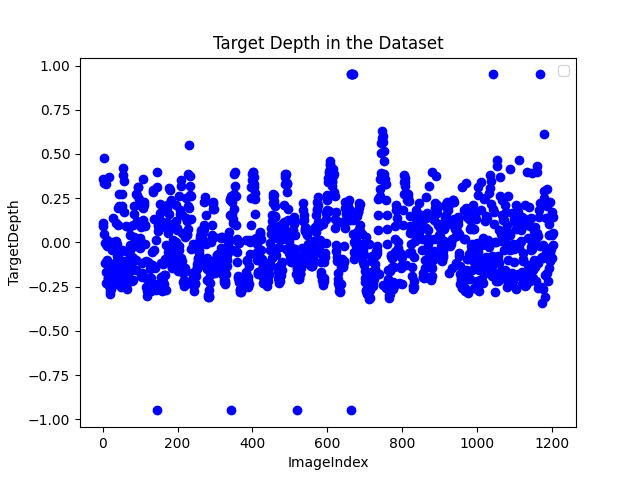


图4-9 用于训练的手势目标中心点的深度值标签的分布

### 4.4.2 实验设置

对于FastDepth模型的训练，将训练数据批大小设置为16，学习算法则使用了带动量的随机梯度下降算法(SGD)，初始学习率设置为0.01，学习率下降权重设为0.0001，动量项参数设置为0.9，最大迭代次数为20。

对于YOLOv3+depth模型的训练，与第三章的YOLOv3检测模型的训练参数设置完全相同。

### 4.4.3 评价指标

在深度估计领域常用的评价指标为Delta1精确度，其公式如下：

（4-5）

其中*T*在Delta1下为1.25，为估计值，为真实值，Delta1精确度的含义为在整张估计深度图上估计值与深度值间的差距小于25%的像素点数目占所有像素的比例。

对于FastDepth，可直接使用Delta1精确度来评价其精度，但基于YOLOv3的目标深度估计算法并不输出稠密深度图，只输出对目标中心点深度的估计，所以使用Delta1进行评估没有太大意义。由于模型所估计的是目标中心的深度，可以使用仪表测量误差中的引用误差来进行度量，公式如下：

（4-6）

（4-7）

其中为深度的估计值，为深度的标签值，为可估计的最大深度，设为1200，为可估计的最小深度，设为700，单位均为mm。

### 4.4.4 实验结果与分析

一、FastDepth模型实验结果与分析

FastDepth使用NYU深度数据集预训练权重在手势深度数据集上进行训练的训练损失如图4-10。

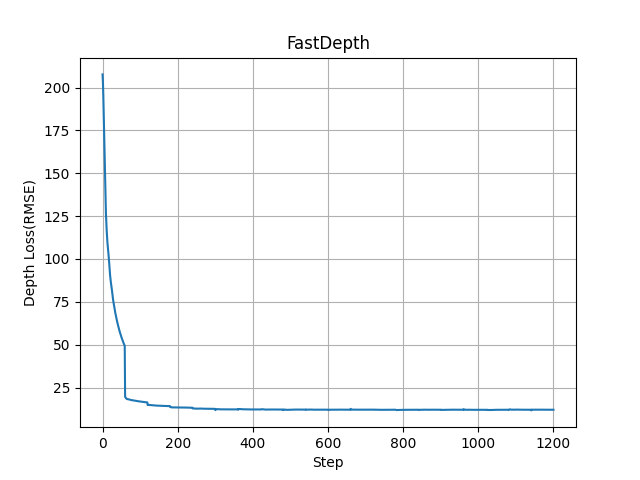


图4-10 FastDepth的训练损失图

其对应的Delta1精确度如图4-11。

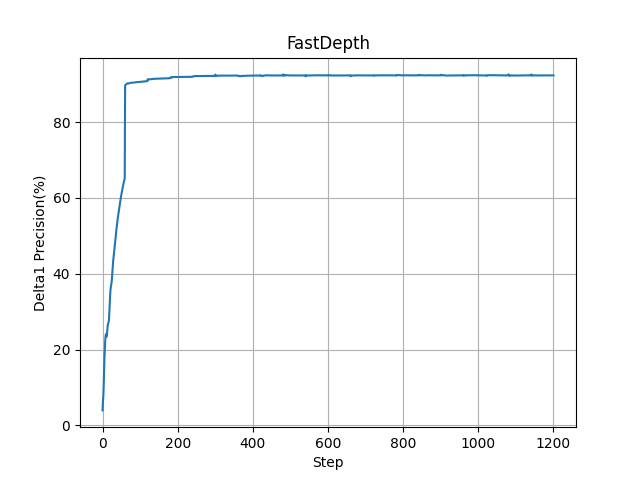


图4-11 FastDepth的Delta1精度图

由图4-10可以看出FastDepth模型在手势数据集上进行训练时收敛速度较快，在150个step时就已经接近收敛。对应的图4-11中的Delta1精确度也已接近90%，说明模型对整张图像所预测的像素级深度图与真实深度图之间的误差较小，只有10%左右的像素估计深度与真值间的误差大于25%。

FastDepth使用不同的预训练权重进行训练的结果如表4-1。

表4-1 FastDepth使用不同的预训练权重进行训练的实验结果

|  |  |  |
| --- | --- | --- |
| 训练设置 | 精确率Delta1(%) | 推断速度(FPS) |
| 使用MobileNet特征提取预训练权重在手势深度数据集上进行训练 | 85.2 | 85 |
| 使用NYU深度数据集预训练权重在手势深度数据集上进行训练 | 92.3 | 85 |

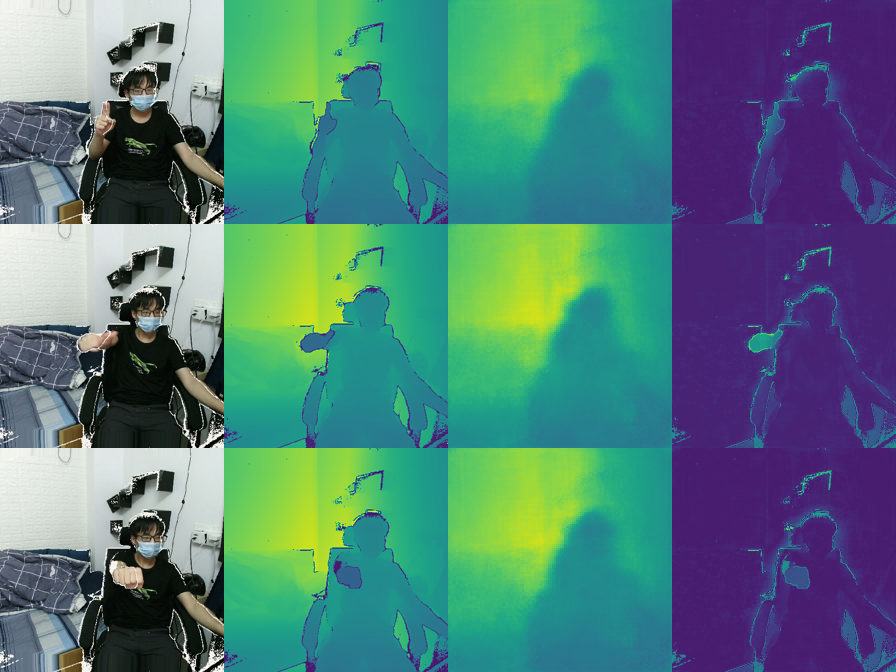


图4-12 FastDepth对手势图像进行深度估计的结果

图4-12为使用NYU深度数据集预训练权重在手势深度数据集上进行训练的FastDepth对手势的深度估计结果，左一为原始RGB图像，左二为Kinectv2采集得到的深度真值图，右二为FastDepth模型输出的深度估计图，右一为深度估计与深度真值间的误差图。

由FastDepth模型输出的深度估计图可以知道，由手势数据集所训练的FastDepth可以估计出场景内较大物体或者背景的深度信息，但对于前景目标的估计能力很差。这可能有如下两点原因：一是模型本身的原因，由于FastDepth模型本身对于小目标并不特别关注，只对宏观场景有一个认识，所以对小目标的感知能力不强。二是手势数据集的原因。首先自制的数据集数据量不够大，只有一千张左右的图片用于训练，不够算法进行学习。其次是数据本身的问题，由于数据集中的图片都只有手势的位置和形态在进行微小的变动，背景并没有变化，而模型又是将整张图片中所有的像素同等对待，模型在进行学习时很容易只学习到背景的深度知识而忽略了变动的手势。

二、YOLOv3+depth模型实验结果与分析

YOLOv3+depth训练时损失值图如图4-13和图4-14。

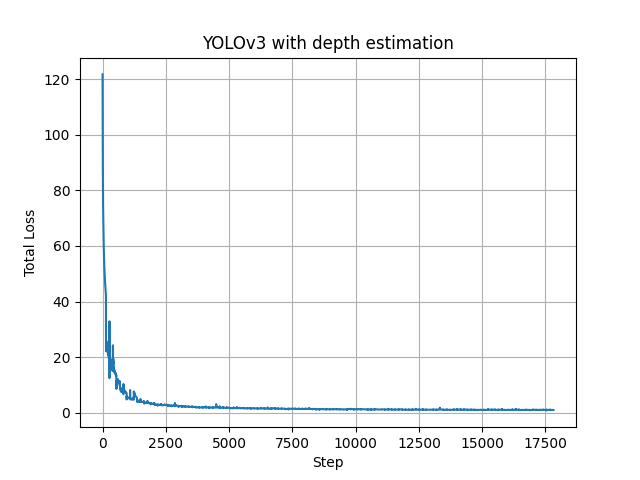


图4-13 YOLOv3+depth的训练损失图

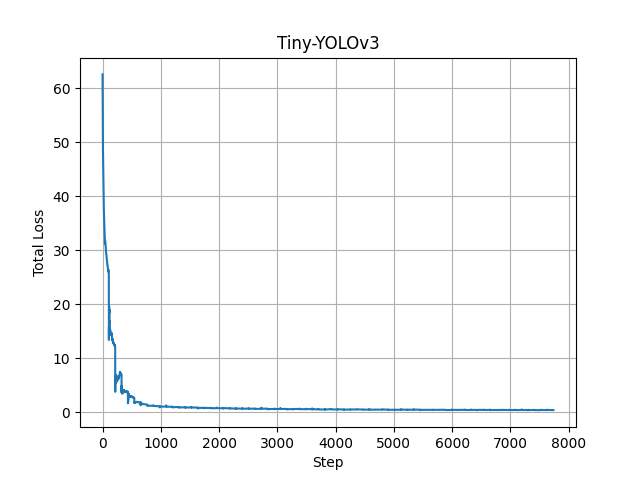


图4-14 TinyYOLOv3+depth的训练损失图

由图4-13和图4-14可以看出YOLOv3+depth模型和TinyYOLOv3+depth模型在手势数据集上进行训练时都能够收敛，TinyYOLOv3+depth由于主网络较小，其收敛速度较快，在第1000个step时就已经接近收敛。

YOLOv3+depth和TinyYOLOv3+depth在进行检测和深度估计时的准确率以及计算速度对比如表4-2。

表4-2两种网络结构的YOLOv3+depth模型的检测和估计效果

|  |  |  |  |
| --- | --- | --- | --- |
| 训练设置 | 目标检测平均精确率mAP(%) | 目标深度估计平均精确度(%) | 推断速度(FPS) |
| YOLOv3+depth在手势深度数据集上进行训练 | 99.5 | 93.4 | 43 |
| TinyYOLOv3+depth在手势深度数据集上进行训练 | 97.3 | 93.9 | 144 |

可以看出，两种网络结构的模型均能实现对手势的检测和对目标深度的估计，TinyYOLOv3+depth由于使用了较轻量化的网络，相比于使用Darknet-53的模型，其mAP低了2.2%，但由于其参数量较少，能更快地学习到目标的深度信息。

对于YOLOv3+depth和TinyYOLOv3+depth在目标检测和深度估计的精度上差距不大的原因，我们认为有两点：一是数据量较少，只有1000张左右的训练图片；二是任务较为简单，要检测的目标只有手势，训练所用的图片中的背景基本不变，手势的颜色也和背景有较大差异，较容易进行分辨。所以两者都能够较好地学习到图片中的知识，而且TinyYOLOv3由于结构更简单参数量更少，能更快地对训练数据进行拟合，而且相对于较大的YOLOv3，有更好的正则能力，可以防止过拟合。

YOLOv3+depth的检测效果如图4-15。



图4-15 YOLOv3+depth检测4类手势的效果

图4-15是使用YOLOv3+depth模型在自制手势数据集的测试集上进行检测的结果可视化，模型的预测结果以绿色的矩形框标出。从图中可以看到，该模型能够较好的进行手势位置检测、类别判断和手势中心点深度的估计。

## 4.5 结合目标检测和深度估计算法的绘图应用

YOLOv3+depth模型能够对手势的位置类别深度进行估计，相当于从单目RGB图像中得到了手势的三维坐标，下面使用PyOpenGL结合于YOLOv3的目标深度估计算法，编写了一个简单的使用RGB摄像机获取手势进行三维图形绘制的程序，以展示模型可能的用途。

### 4.5.1 手势绘图程序简介

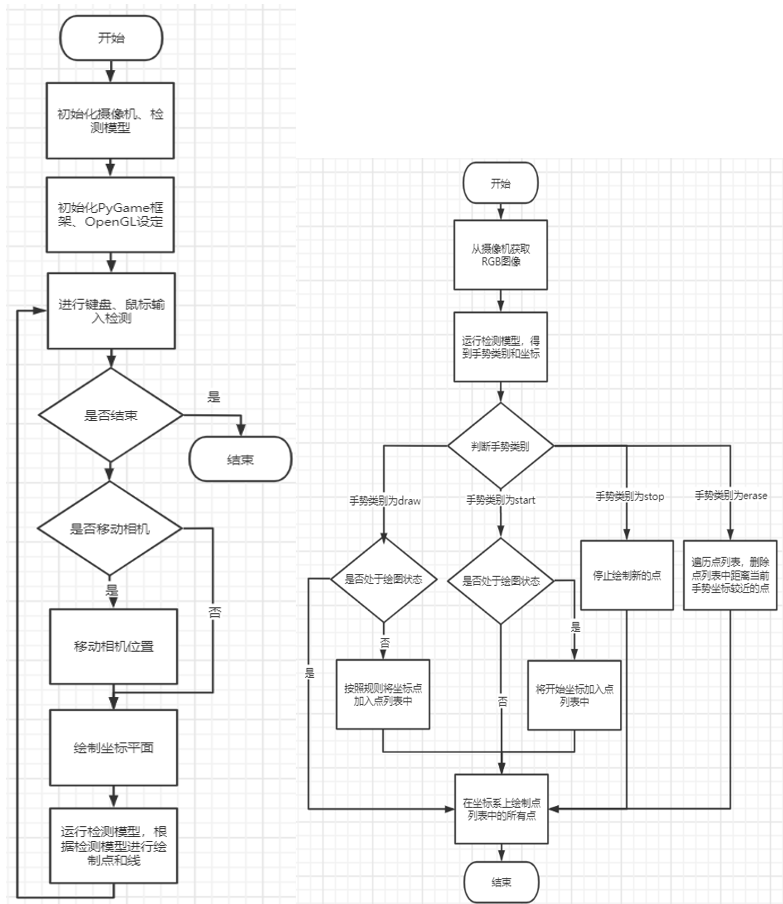
该手势绘图程序能够根据RGB图像中手势的类别得到控制指令的类型，根据手势的坐标进行空间点的绘制，其运行流程图如图4-16。

图4-16 手势绘图程序流程图

程序上使用pygame进行程序生命周期控制，使用PyOpenGL进行图形绘制，使用TinyYOLOv3+depth进行手势信息的获取，使用KinectV2的RGB相机进行彩色图像的获取。

### 4.5.2 手势绘图程序操作及示例

该程序的操作比较简单，使用WASD可以控制视角相机的位置，移动鼠标可以控制相机的朝向，检测模型检测到start手势就会进入图形绘制状态，获取每个所检测到的draw手势的坐标，并将其绘制到坐标系上，检测到erase手势就会擦除掉当前坐标附近一定范围的已绘制点，当检测到stop手势时绘制结束，进入观察模式。

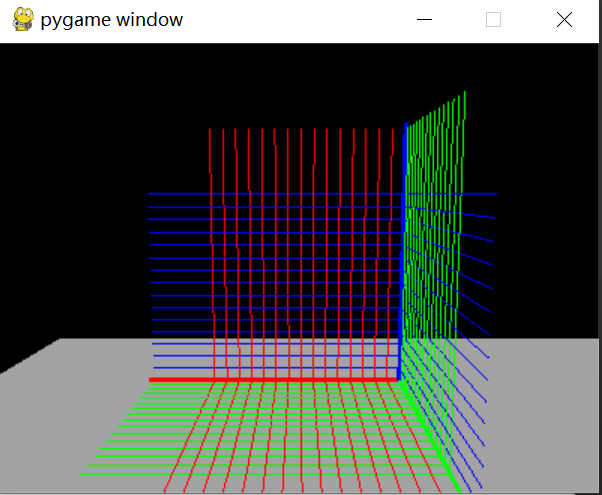


图4-17 手势绘制程序中的坐标系

将检测到的深度坐标设为0，可使用手势进行平面图形的绘制，绘制结果如图4-18。

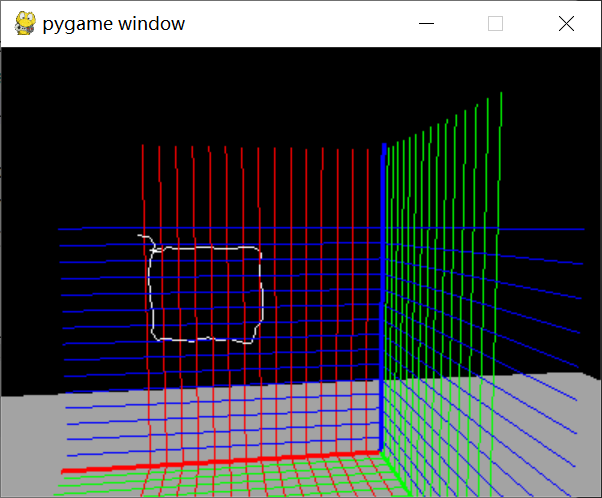


图4-18 利用手势进行平面图形的绘制

利用估计得到的手势深度信息，可使用手势进行三维图形的绘制，绘制结果如图4-19。

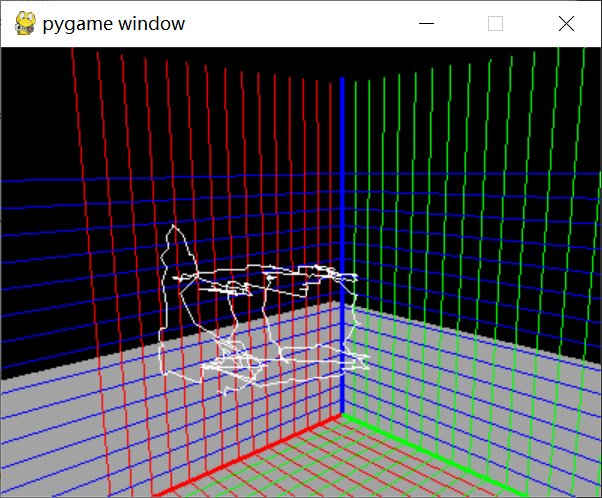


图4-19 利用手势进行三维图形的绘制

## 4.6 本章小结

本章中首先介绍了实时的单目深度估计算法FastDepth，对其网络结构、训练和测试方式等做了较详细的说明。然后介绍基于YOLOv3的目标深度估计算法，其除了输出目标的类别和XY坐标外，还可对目标中心点的深度进行估计。之后使用手势深度数据集对两种模型进行了训练和测试。FastDepth模型擅长对整个场景的深度进行估计，但对前景小目标的深度估计能力很差；YOLOv3+depth和TinyYOLOv3+depth无法估计场景深度，但能对目标中心的深度能做出较准确的估计。TinyYOLOv3+depth更是能在保持较高的检测和深度估计精度的情况下达到实时推断，在RTX3070上的运行速度达到了144FPS。最后，使用PyOpenGL结合TinyYOLOv3+depth的目标深度估计算法，编写了一个简单的使用RGB摄像机获取手势进行三维图形绘制的程序，以展示模型可能的用途。

# 结论

## 论文工作总结

本文主要研究的内容为基于深度学习的手势检测。基于手势的人机交互方式为人们控制计算机或机器人等设备提供了更自然、直观、便捷的方式，其在各种场景下均有广泛的应用，与之密切相关的手势检测一直以来都是视觉算法中的研究热点。尽管手势检测算法已经得到了广泛的研究，但是它们在实际应用场景中所必然会面临的复杂背景、遮挡、光照变化、运动模糊等问题仍未得到有效的解决。深度学习的算法由于使用卷积神经网络来进行拟合和学习，相较于传统的机器学习模型，虽然有较大的性能消耗，但是在泛化能力、鲁棒性等方面有巨大的优势。本文所研究的基于深度学习的手势检测算法，主要是从两方面进行了研究，一是基于通用的目标检测模型训练得到适合手势检测任务的检测模型，二是提出结合深度估计的目标检测模型来检测手势类别和其在空间中的位置。本文主要完成了以下几方面的工作：

（1）使用了单阶段目标检测网络的代表YOLOv3模型用于手势检测。在数据准备上，使用了K均值聚类计算最优手势数据的最优先验框，使用在ImageNet上预训练的特征提取权重为模型提供更优的参数初始化；在训练时使用了适当的数据增强方法，提高了模型的人手检测效果。与传统的机器学习模型相比，基于深度学习的检测模型可以在复杂背景下完成手势检测的任务，训练所得的YOLOv3手势检测模型的mAP达到99.5%，TinyYOLOv3手势检测模型的mAP达到97.2%，推断速度在RTX3070上达到了144FPS，可以实时运行。

（2）使用基于深度学习的深度估计模型FastDepth来估计手势在空间中的深度信息。修改了YOLOv3目标检测算法的结构，得到能够估计目标中心点深度信息的YOLOv3+depth模型，使其能够对目标的深度进行估计。分析了两种对手势深度信息进行估计的模型的优劣。同时，本文实现了一个简单的基于手势检测的三维绘图程序。

## 工作展望

本文研究了基于深度学习的手势检测模型，并且用实验展示了相关算法的效果。结合目前相关领域的研究现状，可以从以下方面对本文的工作做进一步的完善：

（1）在第3章中我们在手势检测时使用了YOLOv3模型，模型中以Darknet-53网络作为主网络结构。可以考虑使用MobileNet或ShuffleNet[28]等轻量化模型对Darknet-53结构进行替换，实现更加高效的特征提取，提升整个目标检测模型的速度。

（2）模型压缩，可以考虑使用模型剪枝和量化等方式来减少主网络部分的计算消耗，增加主网络的计算速度，让深度估计模型和手势检测模型在计算性能较低的机器上也能够以较高速度进行推断计算。

（3）调整深度估计模型，使其能够更关注场景中的目标，这可通过调整损失函数做到。本文中的FastDepth单目深度估计模型在计算损失值时将图像中所有像素等同对待，如果针对含有目标的场景设置一个目标权重遮罩，将其与未求和的损失矩阵相乘，就可以提高目标部分深度像素的重要性，使模型更关注对目标的深度估计。

（4）本文所提出的YOLOv3+depth模型只能估计目标中心点的深度值，FastDepth模型则能够估计整个场景所有像素的深度值，输出像素级的深度图。如果将两者结合，并调整目标检测和深度估计模型的网络结构，使其共享特征提取网络所提取的特征，就能在稍稍损失一些计算性能的情况下得到目标的类别、坐标、中心点的较精确深度和整张图像的像素级深度图。

# 参考文献

1. Zabulis X, Baltzakis H, Argyros A. Vision-Based Hand Gesture Recognition for Human-Computer Interaction[J]. universal access handbook, 2009, 34:30
2. Xing Y, Li J, Wang L, et al. A Robust Hand Gesture Recognition Method via Convolutional Neural Network[C]// International Conference on Digital Home. IEEE Computer Society, 2016.
3. 林汇丰. 基于深度学习的手势识别、人手检测及模型压缩的研究[D]. 华南理工大学, 2020.
4. Kim H J, Lee J S, Park J H. Dynamic hand gesture recognition using a CNN model with 3D receptive fields[C]// International Conference on Neural Networks & Signal Processing. 2008.
5. Molchanov P, Gupta S, Kim K, et al. Hand gesture recognition with 3D convolutional neural networks[C]// Computer Vision & Pattern Recognition Workshops. IEEE, 2015.
6. Miao Q, Li Y, Ouyang W, et al. Multimodal Gesture Recognition Based on the ResC3D Network[C]// 2017 IEEE International Conference on Computer Vision Workshop (ICCVW). IEEE, 2017.
7. Deep Residual Learning for Image Recognition[C]// IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2016.
8. 刘文达. 基于单目摄像头的静态手势识别方法研究[D]. 湖北工业大学, 2020.
9. Ni Z, Jia C, Nong S, et al. Light YOLO for High-Speed Gesture Recognition[C]// 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018.
10. Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. 2017:6517-6525.
11. Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv e-prints, 2018.
12. Wofk D, Ma F, Yang T J, et al. FastDepth: Fast Monocular Depth Estimation on Embedded Systems[J]. 2019.
13. Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. IEEE, 2016.
14. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
15. 赵永强, 饶元, 董世鹏, 张君毅. 深度学习目标检测方法综述[J].中国图象图形学报,2020,25(04):629-654.
16. Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. European Conference on Computer Vision, 2016.
17. Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[J]. IEEE Computer Society, 2014.
18. Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. 2020.
19. 郭继峰, 白成超, 郭爽. 基于深度学习的单目视觉深度估计研究综述[J].无人系统技术,2019,2(02):12-21.
20. Zhao C, Sun Q, Zhang C, et al. Monocular Depth Estimation Based On Deep Learning: An Overview[J]. 中国科学:技术科学英文版, 2020(9):1612-1627.
21. Godard, Clément, Mac Aodha O, Brostow G J. Unsupervised Monocular Depth Estimation with Left-Right Consistency[J]. 2016.
22. Eigen D, Puhrsch C, Fergus R. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network[J]. 2014.
23. Eigen D, Fergus R. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture[C]// 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2014.
24. Chen X, Fang H, Lin T Y, et al. Microsoft COCO Captions: Data Collection and Evaluation Server[J]. Computer Science, 2015.
25. H Rezatofighi, Tsoi N, JY Gwak, et al. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019.
26. Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.
27. U-Net: Convolutional Networks for Biomedical Image Segmentation[C]// International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2015.
28. Zhang X, Zhou X, Lin M, et al. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices[J]. 2017.

# 致谢

大学生活即将结束，四年的学习生活使我受益匪浅。经历大半年时间的反复琢磨，毕业论文最后完稿，回顾这期间来收集资料、阅读论文、构建模型、不断迭代修改的过程，我得到了许多人的帮助，此刻要向帮助我的人们表达我最诚挚的谢意。

首先，我要感谢我的导师刘伟东老师。在论文的选题、搜集资料和写作阶段，刘老师都投入了极大的关怀和鼓励。在论文的写作过程中，每当我有所疑问，刘老师总会放下繁忙的工作，耐心地指点我；在我初稿完成之后，刘老师在百忙之中抽出空来对我的论文认真批改，提出许多指导意见。他严谨的作风和对学术的追求大大激励了我，他的关心和教诲我更将永远铭记。

我还要感谢我的班主任许玉格许老师这几年来对同学们的关心与支持，同时也感谢这三年来与我互帮互助的诸位同学，在各位同学的共同努力之下，我们维持了良好的生活环境和积极向上的学习氛围，能在这样一个团队中度过大学生活，是我的荣幸。

同时也感谢我的舍友们，他们以极大的热情，在我的数据收集和处理上帮了很大的忙，感谢他们对本文工作所带来的帮助。

最后，我要感谢参与我论文评审和答辩的各位老师，他们给了我一个检验几年来学习成果的机会，让我能够明确今后的发展方向。我将在今后的工作和学习中更加努力，取得更多成果回报他们、回报社会。再次感谢他们，祝他们工作顺利、家庭幸福！