# 演讲稿

## 1.开题报告

### 前两页PPT-首页+目录

* **第一页**：各位老师、同学们，大家早上好。我是计科专业的薄劲阳。今天呢，我的答辩题目是轻量化目标检测算法的设计。
* **第二页**：今天我主要从以下四个方面开展答辩。

### 第三页PPT-选题背景

* 首先说一下这个选题背景，目前目标检测是计算机视觉中一个相当重要的方向。所谓目标检测就是目标定位再加一个目标分类。传统的分类问题只需要判断类别就好了，而不需要定位。
* 目前目标检测主要的方法就是深度学习。基于深度学习的目标检测算法虽然在准确性方面表现出色，但它们通常具有较大的模型大小和高计算复杂度，导致在资源受限的环境下，比如说单片机这种设备，因为资源较少，难以应用高精度模型。因此通过牺牲一小部分检测精度而实现加速的轻量化网络获得了广大关注。轻量化的轻就是轻在模型大小和参数规模。
* 然后目标检测算法当前的特点就是模型参数多、计算复杂度高、难以部署到嵌入式边缘设备。

### 第四页PPT-解决问题与意义

* 我的这个毕设呢，就是想办法解决这个问题。我计划是复现现有的Yolov5、Yolov5-Lite目标检测模型。Yolov5-Lite是一个轻量化的Yolov5。我计划比较二者的不同，找出其如何实现轻量化，并在这个基础上，尝试修改一下这个网络架构。
* Yolo模型训练是在我的笔记本上，操作系统用的是Ubuntu，处理器2.5Ghz双核，显卡是这个NVIDIA RTX 4060的显卡。训练好的模型最终跑在树莓派上，树莓派的处理器是1.5Ghz四核，内存是4GB。大家可以看到相对这个笔记本来说，树莓派的资源算是比较差了。

### 第五-第九页PPT-技术路线

* **第五页**：接下来讲一下这个技术咋实现。我主要是分成5个流程。首先构建数据集，然后用构建好的数据集分别训练一个Yolov5和Yolov5-Lite，然后部署到树莓派上并运行。到时候用PyQT给程序写个界面，然后可以展示一些性能一些的参数就好了。具体实现看下面的讲解。
* **第六页**：数据集制作这个，主要是先使用树莓派的摄像头，写一个拍照程序，对要检测的物体进行拍照。然后再使用这个Labelme软件，对拍好的图片描框、标注。就跟左边这张图片差不多。因为Labelme标注后的文件是.JSON格式，而Yolo训练需要txt格式，所以到时候还得写个程序转换一下两个的格式。
* **第七页**：模型训练就很简单了，从GitHub上下载源码之后，将制作好的数据集放到对应目录下，配置一下相关参数，训练一下，得到模型的相关参数，就可以部署到树莓派了。
* **第八页**：模型部署呢我主要使用的是ONNX，ONNX是一种模型格式，他可以实现跨平台。就跟左边这个图展示的原理差不多。像是Pytorch、TensorFlow这些主流模型框架，转换成ONNX模型格式，然后再部署到设备上。运行ONNX模型需要使用软件onnxruntim，所以到时候还需要再树莓派上下载这个软件。
* **第九页**：最终就是在树莓派上运行这个目标检测程序，调用训练好的模型，对目标进行检测。计划到时候写一个前端界面吧，可以展示这个摄像头画面和一些基本的性能指标。

### 第十页PPT-工作安排

* 目前我的工作是这样安排的。前期的话就是理论学习。后期可能就是复现代码和完成论文了。
  + 我目前工作的重心也就是这个分析Yolov5、Yolov5-Lite模型，难点也是在这里。前段时间我尝试读了下论文，发现这个深度学习底子不太够，所以就去补深度学习的知识了。
  + 目前我的计划是这样的，就是最低目标就是能够搞懂这个Yolov5、Yolov5-Lite检测模型的所有知识，分析一下Yolov5-Lite相比Yolov5，是怎么实现轻量化的。然后还有一个理想目标，就是能根据Lite实现轻量化的原理，改一下它的网络结构，提出一个新的网络结构。不过这个的话，我自我评价，感觉可行性有点低，难度有点大。
* 后期的话主要就是编程、写论文了。
* 计划打算年前继续学习深度学习理论，然后顺带完成一下这个数据集制作这块。年后可以尝试看看Yolo的论文这些了。计划四月中旬初步拿出个东西，后面5月份的时候在完善修改。

## 答辩演讲

### 前两页PPT-首页+目录

* **第一页**：各位老师，大家早上好。我是计科专业的薄劲阳。今天我的答辩题目是轻量化目标检测系统的设计与实现。
* **第二页**：今天我主要从以下五个方面开展答辩。

### 第三页PPT-研究背景及意义

* 首先说一下研究背景及意义，目前目标检测是计算机视觉中一个相当重要的方向。传统的目标检测算法通常要求较大的模型大小和高计算复杂度，这会对计算和存储资源造成较大的负担。轻量化目标检测算法通过优化网络结构、减少参数数量以及采用高效的计算方式，可以大幅降低算法对资源的需求。

### 第四页-第七页PPT-YOLO基本原理及网络架构

* **第四页**：YOLO的基本原理如右图所示，主要可以分为三个阶段。
  + 第一个阶段是将图片进行处理，输入到网络模型中。
  + 第二阶段是网络模型进行推理，得到一个tensor张量。这个张量包含了物体的可能位置以及各个类别的概率。
  + 第三阶段是对第二阶段的tensor进行非极大值NMS处理，去除多余的候选框，最终确定图像中物体的具体位置和类型。
* 我的毕设实验，主要是针对YOLOv5、YOLOv5-lite、YOLOv7三个模型进行实验的。 YOLO目标检测模型的网络结构，主要分为Backbone、Neck、Head结构。Backbone结构是网络的核心部分，主要负责从输入图像中提取关键特征。Neck结构主要负责输入特征的融合以及加强。Head结构负责去除多余的预测框、物体类别预测。我们主要的研究对象是网络的Backbone结构。
* **第五页**：YOLOv5的Backbone结构，由Focus、CBL以及CSP组成。Focus模块负责对图像进行Slice切片操作，将高维图像转换为低维图像（会导致增加通道数）。CBL模块是由Conv卷积模块、BatchNorm归一化处理模块、Leaky ReLU激活函数模块组成。CSP由CBL、Res Unit、Conv构成，主要负责XXX。
* **第六页**：YOLOv5-lite的Backbone结构由CBL、ShuffleNet结构构成。相比于YOLOv5，去除了Focus 层。另外在提取特征模块，使用 ShuffleNet替代CSP模块。相比于CSP 模块，ShuffleNet减少了缓存的使用。
* **第七页**：YOLOv7的Backbone结构由CBS、ELAN、MP1模块组成。CBS模 块由Conv卷积层、BatchNorm归一化层、 SiLU激活函数。ELAN则是由5个连续的CBS模块组成。MP1是负责最大池化的 模块。模块中存在大量并行的网络结构。

### 第八-第十三页PPT-模型对比实验及结论分析

* **第八页**：我主要将训练好的模型分别在PC端和树莓派端运行，比较不同模型的轻量化效果。PC和树莓派的配置环境如下表所示。然后训练模型采用的是Fruits detection数据集，他总共有五种类别。测试数据选择了一段时长为58s的视频，用来进行模型的比对实验。
* **第九页**：模型轻量化的效果，我主要使用了FPS、参数量以及模型占用内存三个指标反应。FPS代表了模型一秒内能够推理完成的图像张数。其从图片处理处开始计时，到推理结束停止。具体计算公式如下所示。参数量反应了网络结构的复杂程度。参数量越多，代表模型越复杂。模型占用内存则反应了模型所占用的存储空间大小。
* **第十页**：具体的实验结果如下表所示。FPS层面，YOLOv5、YOLOv5-LIte、YOLOv7在PC端的运行效果要远高于在树莓派的运行效果。在PC端，YOLOv5-lite的运行效果最好，YOLOv5次之，YOLOv7最差。在树莓派端，YOLOv5-lite的效果最好，YOLOv5与YOLOv7相近，都很差。
* **第十一页**：从参数量和模型占用内存大小来看，YOLOv7的参数量和模型占用内存最大，这也说明了YOLOv7的网络最复杂。而YOLOv5-lite则在这两方面都是最小的，也正确反映了为什么其在树莓派上的效果好的原因。
* **第十二页**：针对实验结论，我阅读了Shufflenet v2论文。这篇论文指出了轻量化的四个原则。其从内存访问代价和GPU并行性方向分析了网络应该如何设计才能减少存储器开销。概括下来就是要减少存储器的使用，增加网络的并行度。具体四个准则如下所示。
* 第十三页：读PPT。

### 第十四页-第十五页-成果展示

* 第十四页：另外我对模型搭建了一个简单的QT平台。介绍PPT
* 第十五页：读PPT
* 第十六页：另外，我目前还存在一些小问题。读PPT