## 喻志勇日报 2019.10.28状态列表: <font color=red>[已完成]</font><font color=green>[进行中]</font><font color=yellow>[有问题]</font><font color=blue>[待办项]</font><font color=orange>[延误中]</font><font color=pink>[暂停中]</font>#### \*\*Highlight\*\*

<font color=red>[已完成]</font>经过上一周的学习，已经掌握了基于Pytorch的深度学习重要模块组成及其原理，并根据自己的理解拼接了其中重要的模块，并实现了对物品的检测。同时尝试了修改网络层及transforms模块让仅有2层卷积网络的分类器检测精度达到了84%，以上传至learn3.py。

<font color=green>[进行中]</font> 接下来主要目标熟练灵活的应用、调整网络，同时深度的理解torch各函数。

#### \*\*工作总结\*\*

上周针对深度学习常用的简单CNN网络、AlexNet、VGG、ResNet网络进行了学习，同时理解了基于Pytorch实现的步骤。特别是ResNet其残差模块是实现网络更深层而不会出现梯度弥撒的关键，在基于Pytorch代码实现中ResidualBlock是构建上百层网络的基础。跳连接中，以输入与输出对应的channels为依据将代码分为是否需要在跳连接中加一个nn.Con2d。简单的四层self.make\_layer(BesidualBlock,channels,num\_block,stride=n)，构建几千层的网络，以ResNet18、34、101、152为例只要将4层对应的num\_block1、2、3、4分别设为[2,2,2,2]、[3,4,6,3]、[3,4,23,3]、[3,8,36,3]即可。

对基于Pytorch实现深度学习框架，采用的是基于模块化的思想。学习中将模块主要分为数据加载模块、前处理模块（好的调参实现高的精度与召回率）、网络构建模块（重要的实现迁移学习、好的网络使特征的提取更易辨识）、训练模块（训练的批次大小调参提高精度）、测试模块、验证模块、可视化及数据筛选模块。数据加载模块能将需要的数据和标签转为inputs，labels=data形式，data通过迭代返回tensor。

接下来在目标熟练灵活的应用调整网络，同时深度的理解torch各函数，针对具体的项目与问题提高对整体网络的把握能力。

#### \*\*工程项目\*\*

#### 基础研究

<font color=red>[已完成]</font>

<font color=red>[已完成]</font>

<font color=green>[进行中]</font>

<font color=green>[进行中]</font>

<font color=green>[进行中]</font>