嵌入式系统设计综合项目

开发文档

基于GFNet的交通标志识别系统

第三组

2052538 陈 诺 1851903 边 策

2051840 梁 厚 2053171 白 钰

1953729 吴浩泽

2051374 吴雨阳

2051849 王崧宇

2052225 张勤杭

2052134 刘治华

目录

[项目背景 3](#_Toc138686138)

[相关工作 3](#_Toc138686139)

[Glance-and-Focus（通用、高效的神经网络自适应推理框架） 3](#_Toc138686140)

[开发平台 4](#_Toc138686141)

[硬件平台 4](#_Toc138686142)

[软件依赖 5](#_Toc138686143)

[开发说明 6](#_Toc138686144)

[设计 6](#_Toc138686145)

[架构设计 6](#_Toc138686146)

[用例模型 7](#_Toc138686147)

[流程图 8](#_Toc138686148)

[实现 9](#_Toc138686149)

[源码 9](#_Toc138686150)

[效果展示 9](#_Toc138686151)

[测试与评估 10](#_Toc138686152)

[数据集说明 10](#_Toc138686153)

[测试结果 10](#_Toc138686154)

[不足与改进思路 10](#_Toc138686155)

[参考资料 10](#_Toc138686156)

# 项目背景

交通标志检测是道路信息提取的重要分支之一。近年来，交通标志的识别越来越受到人们的重视，甚至被认为是智能车辆的一个非常重要的特征。交通标志包含大量有用的信息，这些信息可能会因驾驶疲劳或寻找地址而被驾驶员忽略。这些司机在危险天气驾驶时也可能不太注意交通标志。因此，采取提高行车安全性、完善自动检测和路标识别系统等措施，对降低道路死亡人数具有重要意义。

随着人工智能技术的发展，近几年在图像处理领域越来越多地采用深度学习的方式进行图像中物体的识别。使用深度学习的方法识别图像，不仅性能更为鲁棒，而且相比于设定规则的方式，识别率更高。

从数据的角度来说，交通标志检测属于图像检测与识别的范畴；从方法的角度来说，交通标志检测当前主要依托于深度学习算法。

# 相关工作

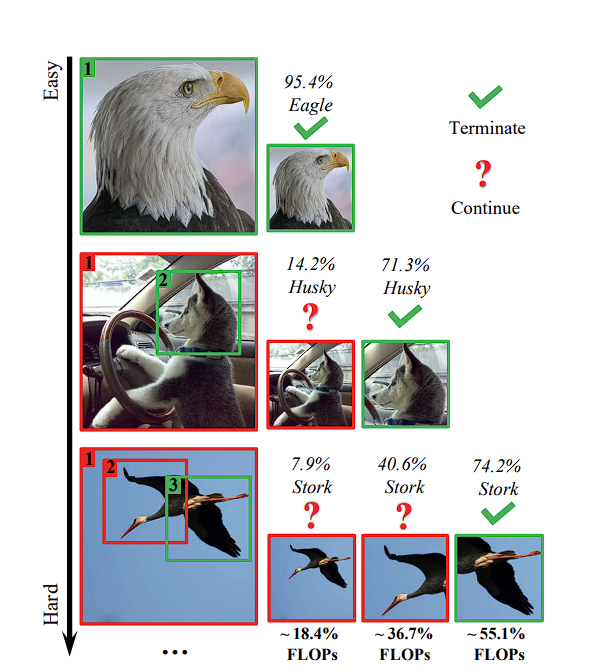
## Glance-and-Focus（通用、高效的神经网络自适应推理框架）

我们根据NeurIPS 2020会议录用的一篇文章[Glance and Focus: a Dynamic Approach to Reducing Spatial Redundancy in Image Classiﬁcation](https://arxiv.org/abs/2010.05300)，其中代码和预训练模型已经在Github上面放出（<https://github.com/blackfeather-wang/GFNet-Pytorch>）

这项工作提出了一个通用于绝大多数CNN的**自适应推理框架**，其效果比较明显，在同等精度的条件下，将MobileNetV3的平均推理速度加快了30%，将ResNet/DenseNet加速了3倍以上，且在iPhone XS Max上的实际测速和理论结果高度吻合。此外，它的计算开销可以简单地动态在线调整，无需额外训练。

简单地来说，下面一张图可以概括我们做的事情：

将图像识别建模为序列决策过程，先将缩略图输入神经网络（Glance），再不断选择最关键的图像区域进行处理（Focus，利用强化学习实现），直至网络产生一个足够可信的预测结果时停止；对于简单和困难的样本分配不同的计算资源，以提升整体效率。



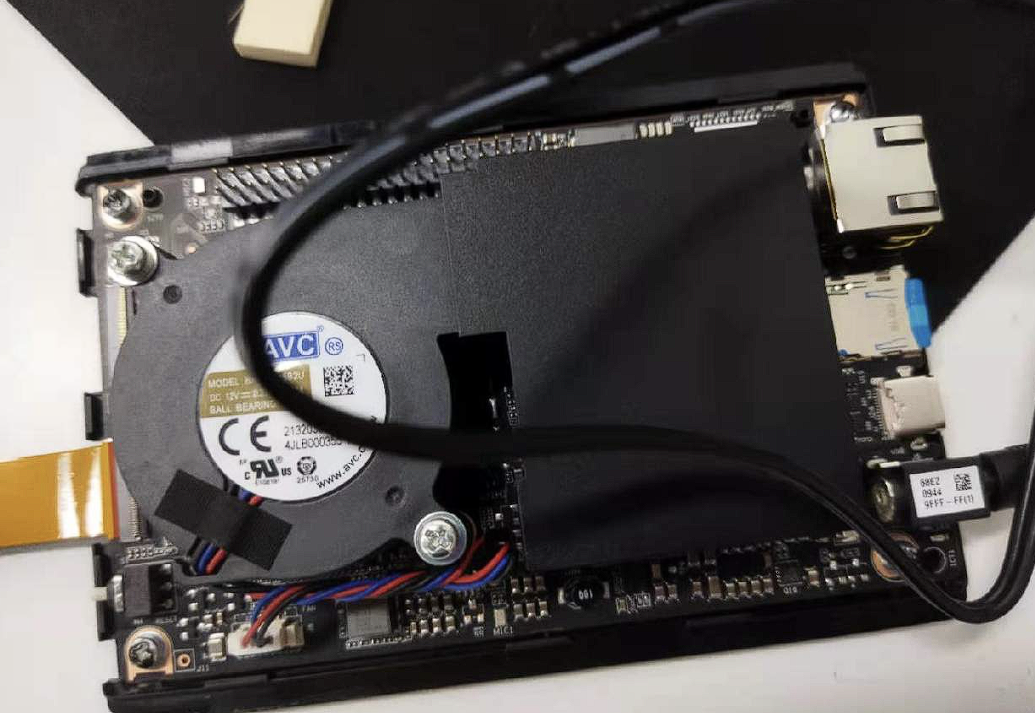
对该论文的详细分析已经在中期报告中展示，这里不再赘述。

# 开发平台

## 硬件平台

**Atlas 200 DK**

Atlas 200 DK开发者套件（型号 3000）是以Atlas 200 AI加速模块为核心的开发者板形态的终端类产品。主要功能是将Atlas 200 AI加速模块的接口对外开放，方便用户快速简捷的使用Atlas 200 AI加速模块，可以运用于平安城市、无人机、机器人、视频服务器等众多领域的预研开发。



**Atlas 200 AI加速模块**

Atlas 200 AI加速模块是一款高性能的AI智能计算模块，集成了昇腾310 AI处理器（Ascend 310 AI处理器），可以实现图像、视频等多种数据分析与推理计算，可广泛用于智能监控、机器人、无人机、视频服务器等场景。

**昇腾310芯片**

昇腾310是一款华为专门为图像识别、视频处理、推理计算及机器学习等领域设计的高性能、低功耗AI芯片。芯片内置2个AI core，可支持128位宽的LPDDR4X。

昇腾310芯片的最大功耗仅为8W。主打极致高效计算低功耗AI芯片。半精度（FP16）运算能力8TFLOPS，整数精度（INT8）16TOPS，支持16通道全高清视频解码（H.264/265）。

## 软件依赖

本项目参考了[华为Atlas 200 DK开发者套件的CANN样例仓](https://gitee.com/ascend/samples)进行开发，其中使用了CANN AscendCL接口。

**CANN AscendCL（Ascend Computing Language）**

CANN AscendCL（Ascend Computing Language）提供Device管理、Context管理、Stream管理、内存管理、模型加载与执行、算子加载与执行、媒体数据处理等C语言API库供用户开发深度神经网络应用，用于实现目标识别、图像分类等功能。用户可以通过第三方框架调用AscendCL接口，以便使用昇腾AI处理器的计算能力；用户还可以使用AscendCL封装实现第三方lib库，以便提供昇腾AI处理器的运行管理、资源管理能力。

# 开发说明

本系统在以下环境中开发：

**CANN版本**

>=6.0.RC1.alpha005

**Python 版本**

>=3.6（小于3.6的未测试）

**包管理器**

Anaconda

**依赖**

见env.yml

**SD卡**

>=64G

**开发环境**

Visual Studio Code + SSH（远程连接）

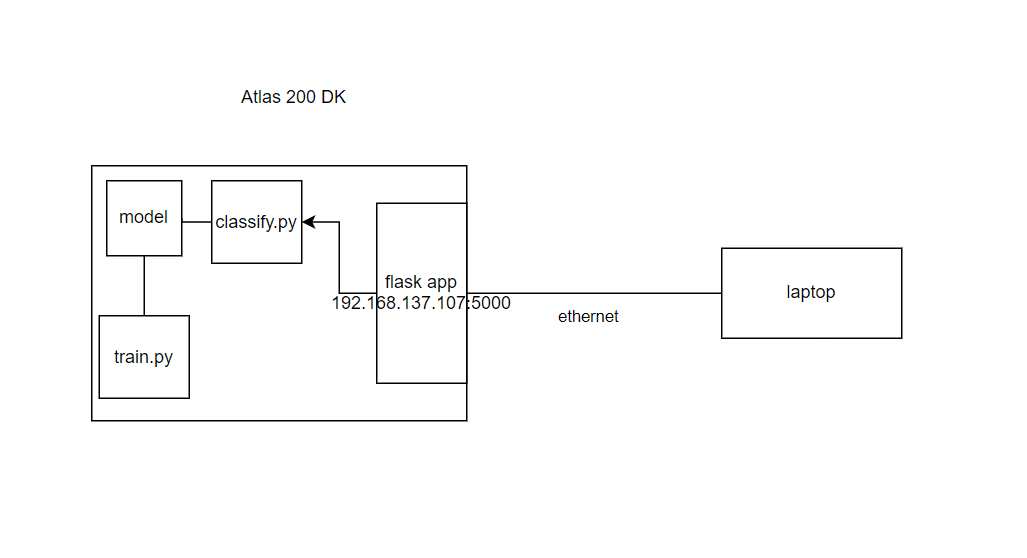
# 设计

## 架构设计

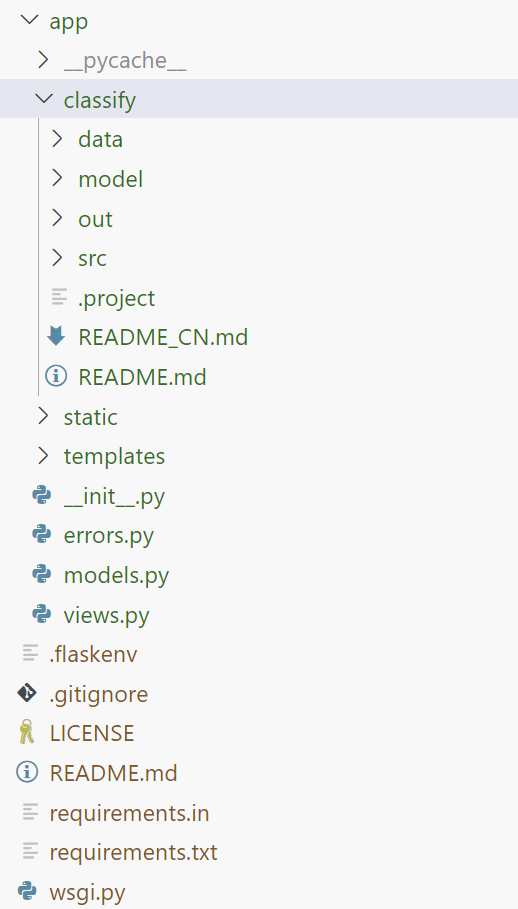
**说明**

* PC通过以太网线和开发板连接，开发板通过PC代理上网，进行相关资源的下载。
* 提供前端进行效果展示，使用WSGI+Flask框架提供Web服务，以及花生壳DDNS进行内网穿透，方便远程调试。
* Web服务和推理计算分离，使得各模块或组件之间的依赖尽可能小，便于维护。

**架构图**



**目录**

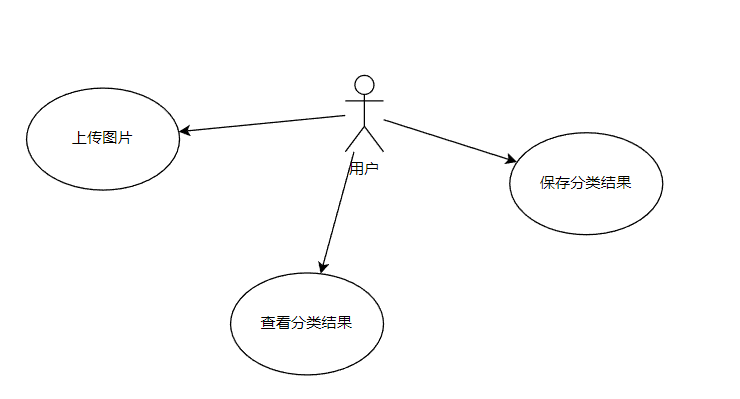
****

Web服务部分

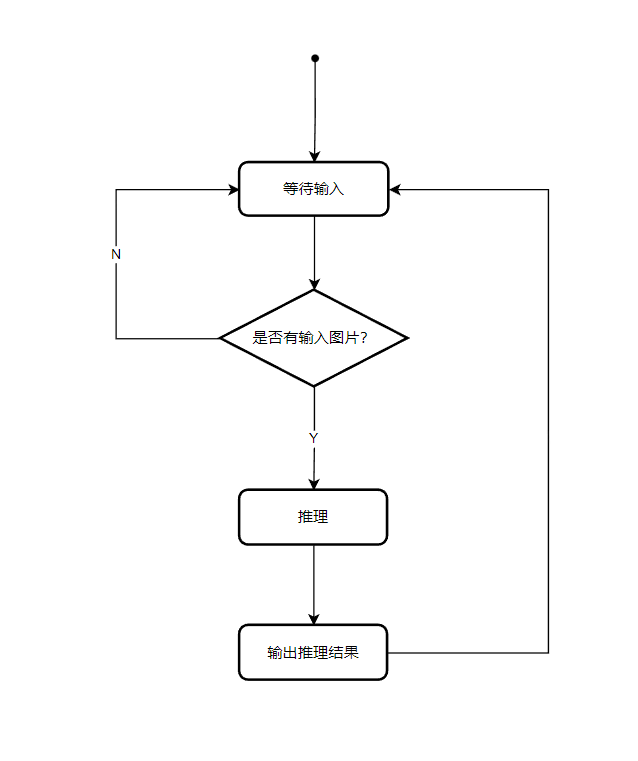
推理部分

## 用例模型

用户点击上传图片，选择图片，查看分类结果



## 流程图



# 实现

## 源码

本地目录 /home/HwHiAiUser/traffic-sign-recog



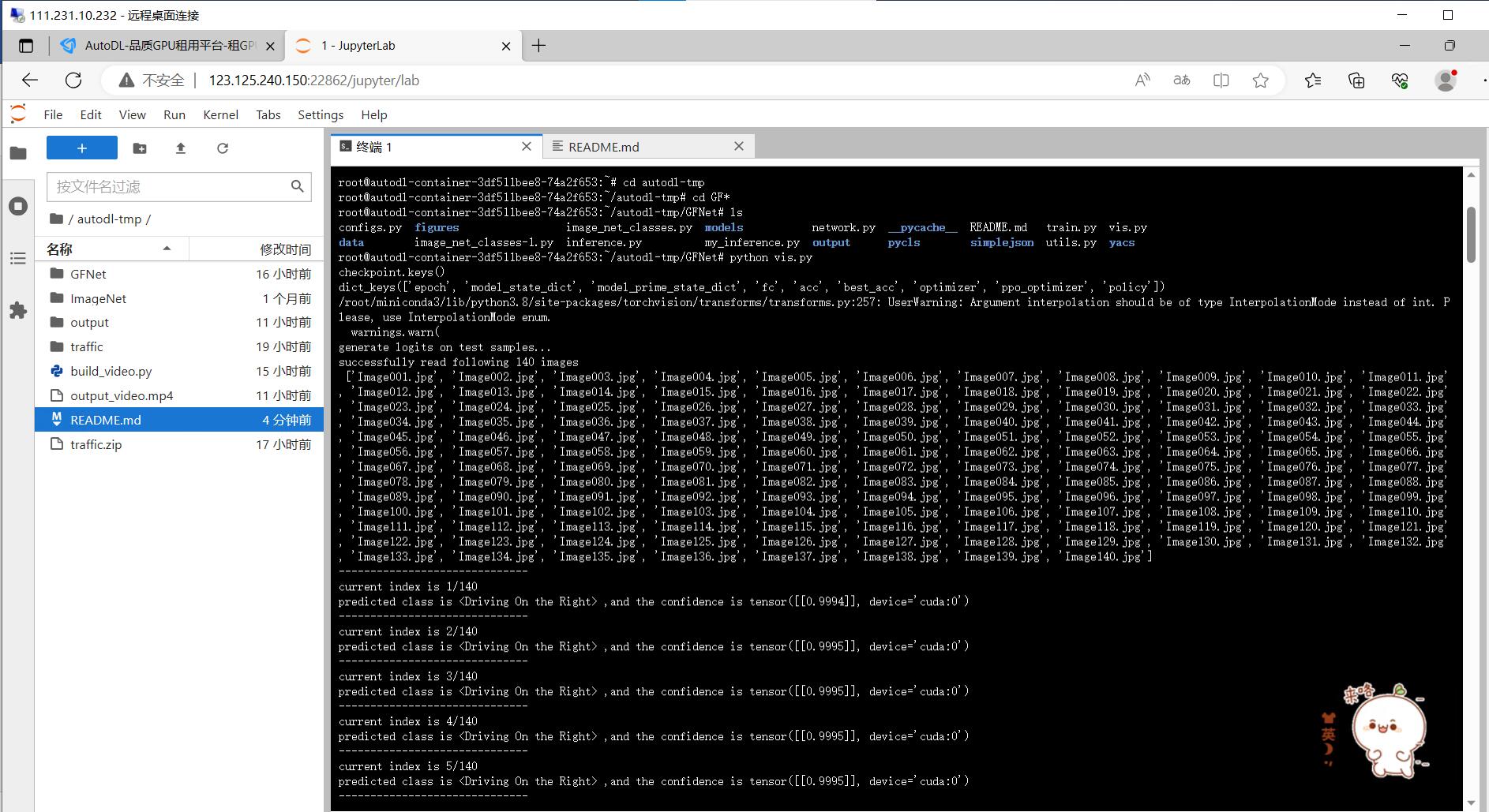
也可见附件

## 过程

**准备环境**

训练平台：autodl





下载数据集并解压缩。将训练集（39209图像）分成训练和验证集，并将文件移动到正确的文件夹。我用80％的样本进行训练，20％的样本用于验证。

**训练**

加载在ImageNet数据集上预训练的ResNet34模型。删除最后一层并在顶部添加一个新的softmax层。

arch = resnet34

learn = ConvLearner.pretrained(arch, data, precompute=False)

挑选了一些尺寸较小的图片作为输入,从32x32图像大小开始,并缩小训练轮次（总共7个epochs）。

优化batch size：尝试使batch size与内存允许的一样大。较大的批量可以缩短训练时间。实验中，发现过大的batch（如1024个样本）会导致较低的验证准确度。推测这个模型很早就开始过度配合。

最终设置batch size为256。

在找到一组合适的超参数后，切换到在更大的图像上进行更长时间的细粒度训练。

最终使用96x96图像和19个epoch的训练，如下所示。

**图像增强**

sz = 96

def get\_augs():

x,\_ = next(iter(data.aug\_dl))

return data.trn\_ds.denorm(x)[1]

aug\_tfms = [RandomRotate(20), RandomLighting(0.8, 0.8)]

tfms = tfms\_from\_model(arch, sz, aug\_tfms=aug\_tfms, max\_zoom=1.2)

data = ImageClassifierData.from\_paths(path, tfms=tfms, test\_name='test')

ims = np.stack([get\_augs() for i in range(6)])

plots(ims, rows=2)

**微调最后一层**

wd = 5e-4

learn.fit(0.01, 1, wds=wd)

**微调整个模型**

learn.unfreeze()

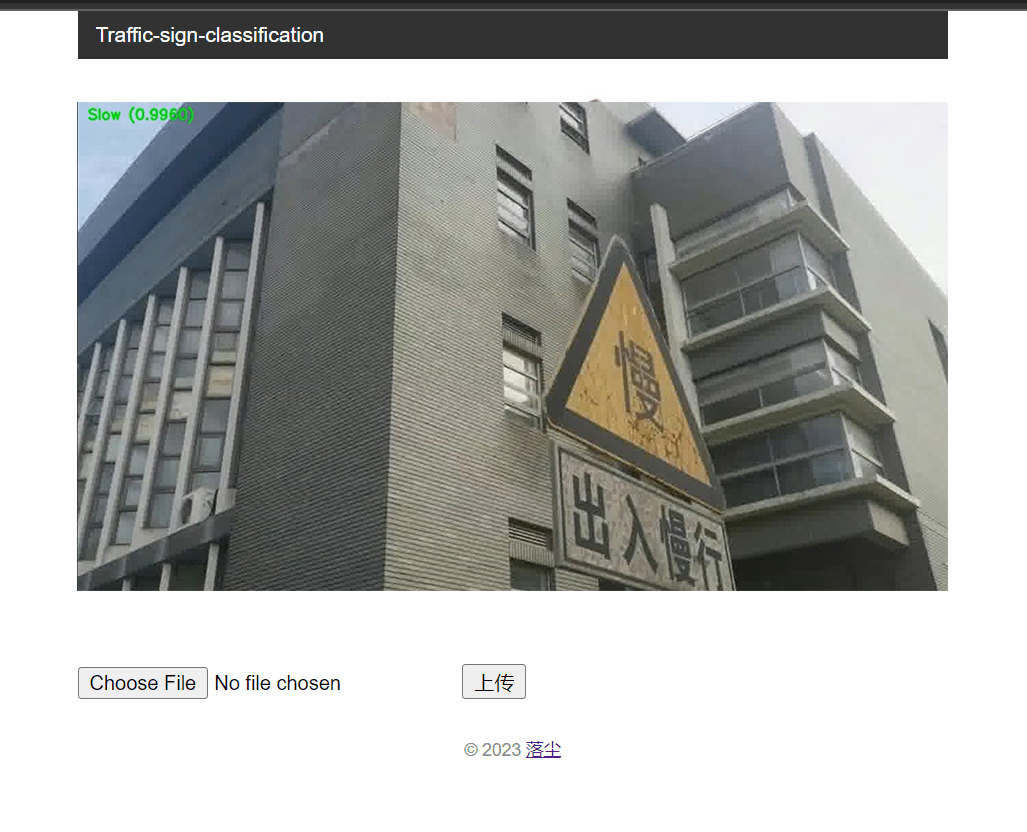
learn.fit(0.01, 3, wds=wd)

learn.fit(lr, 4, cycle\_len=1, cycle\_mult=2, wds=wd)

**集成到项目中**

作为模块classify，集成到Web项目中

## 效果展示



# 测试与评估

## 数据集说明

**中国交通标志检测数据集(CCTSDB)**

来源于论文 A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2一文提出的训练数据集。

论文地址：<https://doi.org/10.3390/a10040127>

CSUST Chinese Traffic Sign Detection Benchmark 中国交通数据集由长沙理工大学综合交通运输大数据智能处理湖南省重点实验室张建明老师团队制作完成。到目前为止，已经包含可用图像15734张。结合本项目实际类别需求，筛选后可用图片为9006张。

此外，我们在校园中拍摄了323张交通标志图片，将其中的80%加入训练集，20%作为测试集。本项目中，区分的交通标志有三类：靠右行驶、慢行、限速。

## 测试结果

见测试文档

## 不足与改进思路

① 目前的数据集只包含了正常环境条件下的交通标志图片，缺乏雨天、夜晚或者阴天等视野环境恶劣情况中的交通标志图片，将来应当进一步扩充数据集，检验模型在恶劣视野条件中的检测能力，提升算法鲁棒性；

② 目前的检测速度还未能达到实时水平（一般认为实时性需要达到每秒处理的图片数量不低于25帧），缺乏在指令集、算法等更底层层面的优化实践，这是未来需要努力的方向。

# 参考资料

<https://pdf.hanspub.org/SEA20210300000_65979652.pdf>

<https://zhuanlan.zhihu.com/p/600136937>

<https://www.researchgate.net/publication/357733836_Glance_and_Focus_Networks_for_Dynamic_Visual_Recognition>

<https://blog.csdn.net/weixin_39540834/article/details/110153595>

<https://www.zkxjob.com/37083>

<https://www.zhihu.com/tardis/zm/art/57160727?source_id=1005>