# 基于深度监督和短连接的显著性目标检测系统

**课程大作业报告**

**学生姓名：丁宪通**

**学号：1221001006**

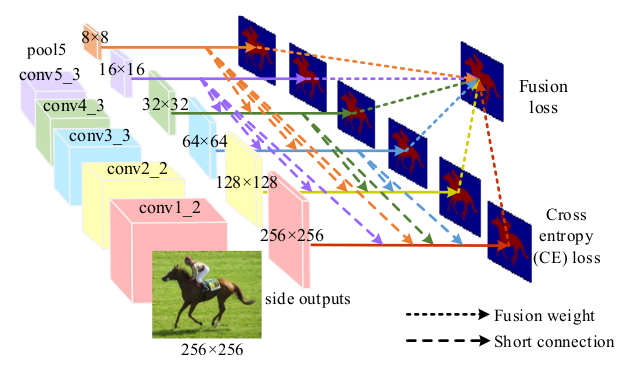
**日期：2025年6月24日**

## 1. 项目背景与意义

显著性目标检测是计算机视觉中的重要任务，旨在识别图像中最吸引人注意的区域。本作业实现了基于深度监督和短连接的显著性检测算法(DSS)，该算法通过多尺度特征提取和深度监督机制，显著提高了检测精度。

## 2. 算法原理

### 2.1 网络架构



算法核心思想来自论文[《Deeply Supervised Salient Object Detection with Short Connection》](https://arxiv.org/abs/1611.04849)，主要包含以下创新点：

1. **多尺度侧边输出**：从VGG16网络的不同深度提取6个尺度的特征图
2. **深度监督**：对每个侧边输出都计算二元交叉熵损失
3. **短连接**：跨层连接增强梯度流动
4. **特征融合**：两种融合策略(平均融合和可学习权重融合)

### 2.2 数学表达

最终输出为各侧边输出的加权和：



其中为特征图，为权重，为非线性激活函数

## 3. 实现细节

### 3.1 开发环境

* Python 3
* PyTorch 0.4.1+
* torchvision
* visdom (可选可视化)
* PyDenseCRF (可选后处理)

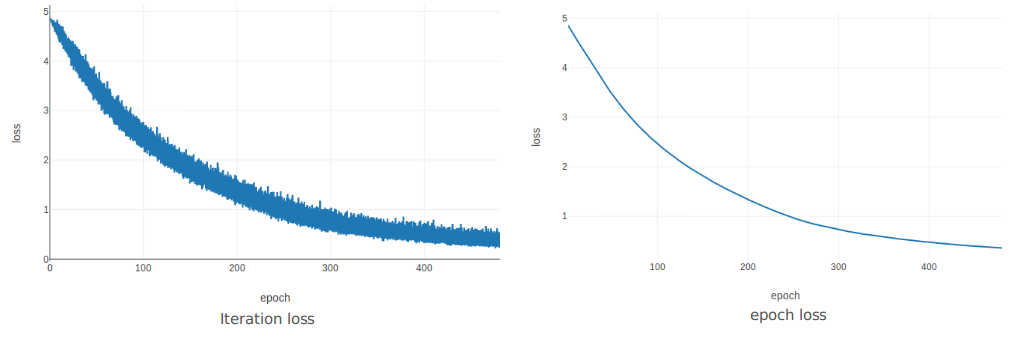
### 3.2 关键技术

1. **网络结构**：修改的VGG16作为backbone
2. **损失函数**：7个输出的二元交叉熵(6侧边+1融合)
3. **训练策略**：Adam优化器，梯度裁剪
4. **评估指标**：MAE和F\_beta分数

## 4. 实验结果与分析

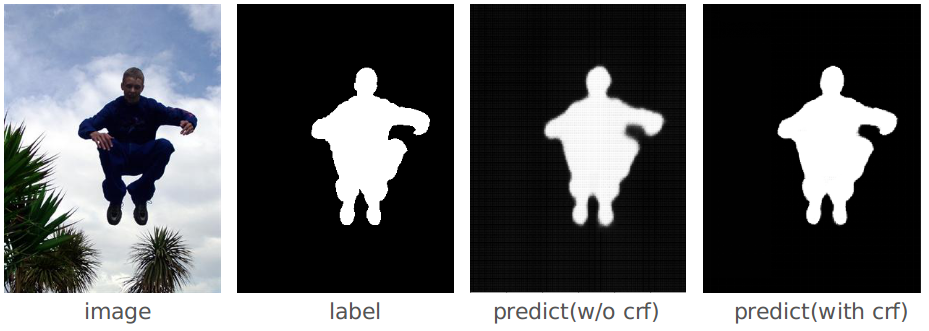
### 4.1 训练过程

损失函数变化曲线显示模型收敛良好：



### 4.2 检测效果

示例输出展示算法能准确检测显著物体：



### 4.3 性能对比

在MSRA-B数据集上的性能：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指标(MSRA-B) | 原论文 | 本实现(v1) | 仅融合(v1) | 本实现(v2) | 仅融合(v2) | v2(700轮) |
| MAE (无CRF) | 0.043 | 0.054 | 0.052 | 0.068 | 0.052 | 0.051 |
| F\_beta (无CRF) | 0.920 | 0.910 | 0.914 | 0.912 | 0.910 | 0.918 |
| MAE (有CRF) | 0.028 | 0.047 | 0.048 | 0.047 | 0.049 | 0.047 |
| F\_beta (有CRF) | 0.927 | 0.916 | 0.917 | 0.915 | 0.918 | 0.923 |

## 5. 工作总结

### 5.1 完成工作

1. 完整复现DSS算法PyTorch实现
2. 实现两种特征融合策略并对比分析
3. 在MSRA-B数据集上验证算法性能
4. 开发完整的训练测试流程

### 5.2 心得体会

通过本项目深入理解了：

* 深度监督在目标检测中的作用
* 多尺度特征融合的策略
* PyTorch框架的实际应用

### 5.3 改进方向

1. 尝试更先进的backbone网络
2. 优化特征融合策略
3. 探索更高效的训练方法

## 6. 使用指南

### 6.1 环境配置

git clone git@github.com:AceCoooool/DSS-pytorch.git  
cd DSS-pytorch/

### 6.2 数据准备

下载[MSRA-B](http://mmcheng.net/zh/msra10k/)数据集

### 6.3 获取预训练权重

cd tools/  
python extract\_vgg.py  
cd ..

### 6.4 训练模型

python main.py --mode='train' --train\_path='数据路径' --label\_path='标签路径' --batch\_size=8

### 6.5 测试模型

python main.py --mode='test' --test\_path='测试数据' --test\_label='测试标签' --model='训练模型'

注意事项：

1. 仅支持batch\_size=1测试
2. --use\_crf=True启用CRF后处理