

# 复现报告

## 文件说明及作用

Inceptionv2.py 是最基础的网络构成，为了适应这个任务我对其进行了一定的魔改。大家也可以选择下载官方训练好的模型作为基础。

Dataset.py 负责将数据集简单的五维 label，转化为最终复杂的 204 维 label。

Train.py，训练用

Pred.py，检验预测结果（这里需要预先训练好的.pth 文件，但是完整模型数据有点大，就没上传到仓库。可以选择自己训练的模型）

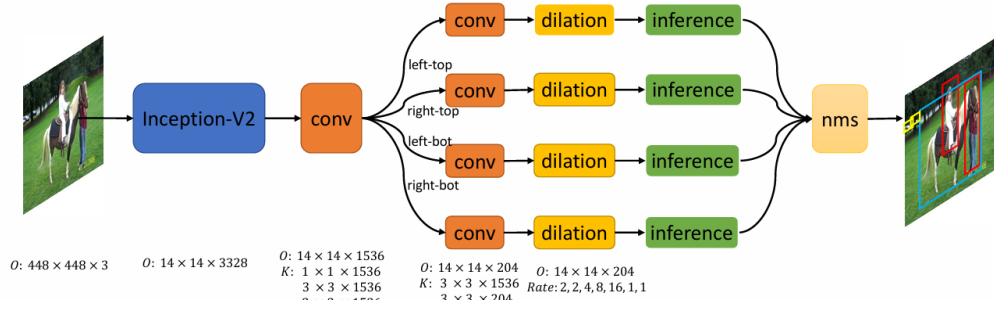
## 论文研读

类似 yolov1，通过回归预测中心点和角点（可以分成四个分支，四对中-角点）来实现预测 Box 的目的，创新点在于可以通过四个分支：中-左上角，中-右上角，中-左下角，中-右下角，再进行 NMS 处理，预测准确性更好，并且对于边界区域更敏感，处理有遮挡的物体效果更好。

## 构建数据集

使用 PASCAL VOC2012 数据集，包含 20 个类别，对每张图的标注框，提取出中心点和角点归一化坐标（我这里省略了从 html 中提取构建 txt 的步骤），进一步处理，构建训练所需的 4\*51 维向量。包括  $P_{ij}$  (1)， $XY_{ij}$  (2)， $Lx_{ij}$  (14)， $Ly_{ij}$  (14)， $Q_{ij}$  (20)。每个框预测四个这样的点（两个角点，两个中心点）

## 网络设计



基础的 inceptionv2 提取特征后，依次经历卷积，扩张卷积，推理以及 NMS 抑制得到输出结果。实际构建过程中主要修改输入输出的维度和尺寸。网络本身只包括 inference 前面，后两个模块是单独的

## 损失函数与优化器

$$Loss_{ij} = \mathbb{1}_{ij}^{pt} Loss_{ij}^{pt} + \mathbb{1}_{ij}^{nopt} Loss_{ij}^{nopt} \quad (1)$$

$$Loss_{ij}^{pt} = (P_{ij} - 1)^2 + w_{class} \sum_{n=1}^N \left( Q(n)_{ij} - Q(\hat{n})_{ij} \right)^2 + w_{coord} \left( (x_{ij} - \hat{x}_{ij})^2 + (y_{ij} - \hat{y}_{ij})^2 \right) + w_{link} \sum_{k=1}^S \left( (L(k)_{ij}^x - L(\hat{k})_{ij}^x)^2 + (L(k)_{ij}^y - L(\hat{k})_{ij}^y)^2 \right),$$

$$Loss_{ij}^{nopt} = P_{ij}^2, \quad (3)$$

which means the existence  $P_{ij}$  should be approached to zero.

In summary, the final loss function of PLN is given as

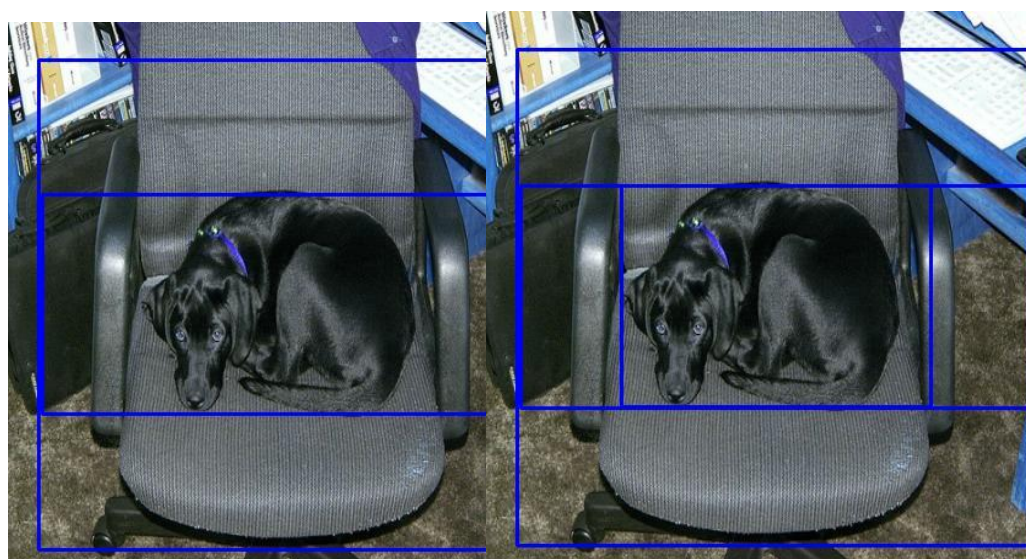
$$Loss = \sum_{i=1}^{S^2} \sum_{j=1}^{2B} L_{ij}. \quad (4)$$

损失函数中计算两类：有点与无点，无点的计算较为简单。有点依次计算剩余 50 维的偏差。原文只是简单的二阶矩，但是实际使用过程中还是给每个部分加了权重。

优化器则是简单的 SGD 优化器。考虑到训练轮数稀少，没有动态规划学习率。

## 训练与预测

采用 PASCAL VOC2012 数据集进行训练，Batchsize=8。跑了 10 轮，总计训练 3 小时左右。受限于设备，训练较少，成功率不算很高。目前可以识别出部分物体，这里放两张效果不错的进行展示。实际使用时记得灵活调整超参（概率阈值与 iou 阈值）来获得更好的效果



## 代码参考

[Shallow-W/PointLN: A simple recurrent of point linking network](#)