



山东大学
SHANDONG UNIVERSITY

信息基础2 机器学习与深度学习

实验四

陈雷

山东大学信息科学与工程学院

实验内容

实验一：常规神经网络函数逼近实验

实验二：基于LeNet-5的MNIST字符识别

实验三：ResNet-18

实验四：Selective search

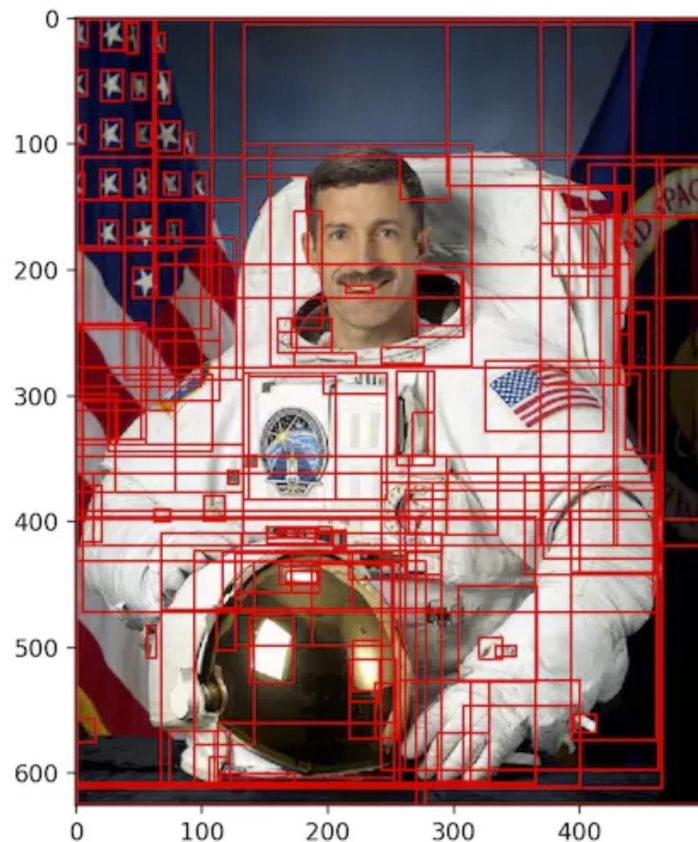
实验五：Yolo

期末项目：命题项目和自选项目



实验四： Selective search

选择性搜索（**Selective Search**），简单来说，就是从图片中找出物体可能存在的区域。下图红色的框就是 Selective Search 找出来的可能存在于物体的区域。

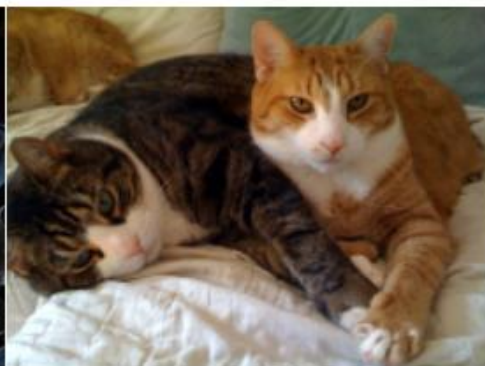


实验四： Selective search

如何判别哪些区域属于一个物体？



(a)



(b)

单一的策略无法应对多种类别的图像。



(c)



(d)



实验四： Selective search

物体检测的核心是物体识别。为了定位物体，我们需要选择一些子区域并在子区域上运行物体识别算法。

产生候选子区域最直接的方法就是滑窗法，但是这种方法效率比较低。

候选区域（Region Proposal）算法用分割不同区域的办法来识别潜在的物体。在分割的时候，合并那些在某些方面（如颜色、纹理）类似的小区域，通过这种方法保证拥有物体的区域在候选区域列表里。



实验四： Selective search

选择性搜索是区域规划的方法，常用于目标检测。

原理：基于颜色、纹理、大小和形状等，将图像分层划分为诸多相似区域。

方法：利用基于图的图像分割方法先得到小尺度的区域，然后一次次合并得到大的尺寸。

特点：选择性搜索用于为物体检测算法提供候选区域，它速度快、效率较高。

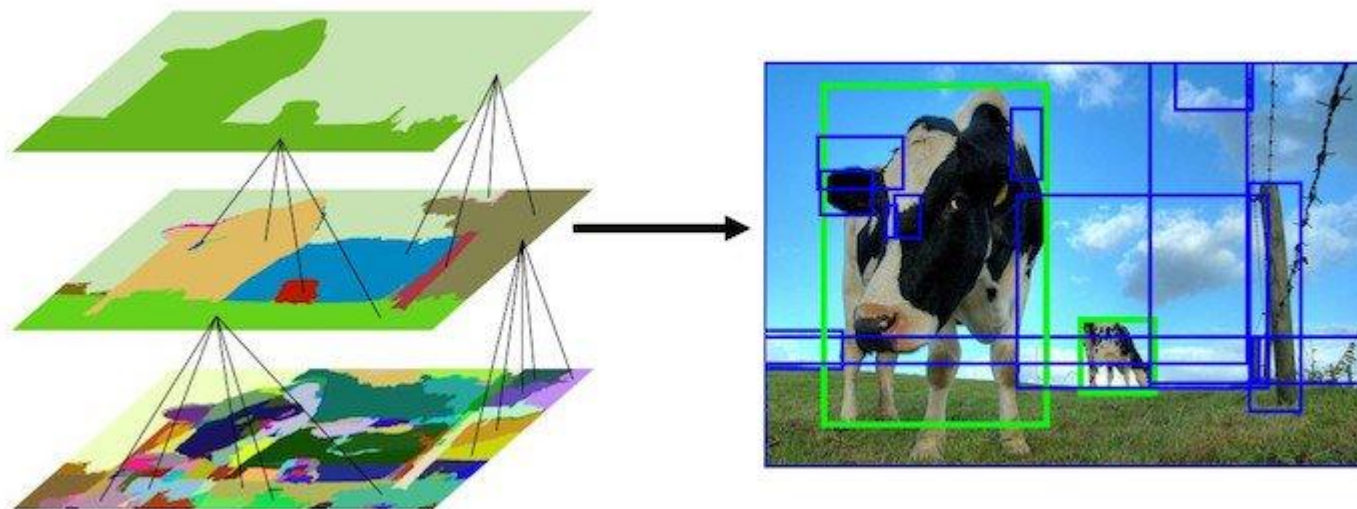


实验四： Selective search

产生初始的分割区域作为输入，通过下面的步骤进行合并：

1. 首先将所有分割区域的外框加到候选区域列表中；
2. 基于相似度合并一些区域；
3. 将合并后的分割区域作为一个整体，跳到步骤1；

通过不停的迭代，候选区域列表中的区域越来越大。通过自底向上的方法创建了越来越大的候选区域。



实验四： Selective search

Algorithm 1: Hierarchical Grouping Algorithm

Input: (colour) image

Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \dots, r_n\}$ using Felzenszwalb and Huttenlocher (2004) Initialise similarity set $S = \emptyset$;

foreach Neighbouring region pair (r_i, r_j) **do**

 Calculate similarity $s(r_i, r_j)$;

$S = S \cup s(r_i, r_j)$;

while $S \neq \emptyset$ **do**

 Get highest similarity $s(r_i, r_j) = \max(S)$;

 Merge corresponding regions $r_t = r_i \cup r_j$;

 Remove similarities regarding r_i : $S = S \setminus s(r_i, r_*)$;

 Remove similarities regarding r_j : $S = S \setminus s(r_*, r_j)$;

 Calculate similarity set S_t between r_t and its neighbours;

$S = S \cup S_t$;

$R = R \cup r_t$;

Extract object location boxes L from all regions in R ;

算法 1: 层次分组算法

Input: (彩色) 图像

Output: 目标定位假设 L 的集合 (区域集合)

使用 Fel&Hut (2004) 得到初始区域 $R = \{r_1, \dots, r_n\}$

初始化相似度集 $S = \Phi$

For each 相邻的区域对 (r_i, r_j) **do**

 计算 (r_i, r_j) 的相似度 $s(r_i, r_j)$

$S = S \cup s(r_i, r_j)$

End

While $S \neq \Phi$ **do**

 得到最高的相似度值: $s(r_i, r_j) = \max(S)$

 对相应区域进行合并: $r_t = r_i \cup r_j$

 从 S 里面移除所有关于区域 r_i 的相似度: $S = S \setminus s(r_i, r_*)$

 从 S 里面移除所有关于区域 r_j 的相似度: $S = S \setminus s(r_j, r_*)$

 计算 r_t 与它相邻区域的相似度得到相似度集 S_t

 更新相似度集: $S = S \cup S_t$

 更新区域集: $R = R \cup r_t$

End

从所有的区域 R 中抽取目标定位 boxes: L



实验四： Selective search

保持多样性的策略

多种颜色空间：考虑RGB、灰度、HSV等

多种相似度度量标准：既考虑颜色相似度，又考虑纹理相似度、尺寸、重叠情况等。

通过改变阈值初始化原始区域， 阈值越大， 分割的区域越少。



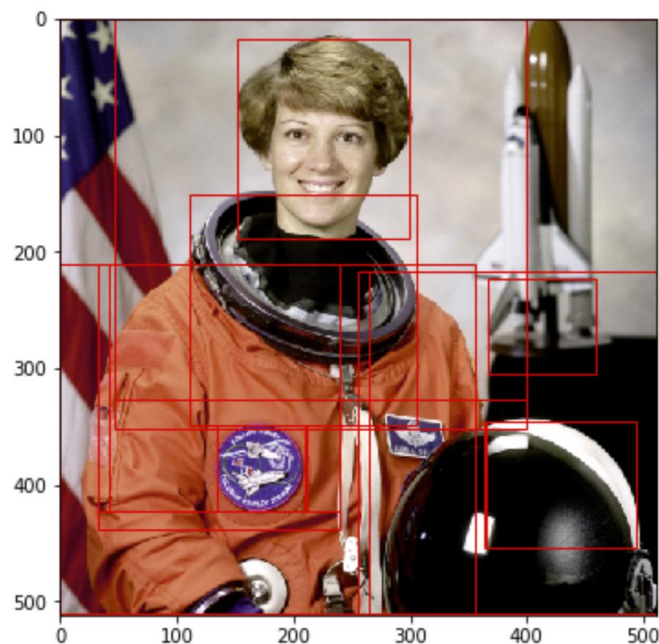
实验四： Selective search

给区域打分（选）

通过上述步骤，我们可以得到很多区域，但并不是每个区域作为目标的可能性都是相同的，我们要进行筛选，可以根据自己的需要选择多少个区域。

选择性搜索性能评估（选）

通过算法计算得到的包含物体的Bounding Boxes与真实情况（ground truth）的窗口重叠越多，算法性能就越好。



参考:

Selective Search for Object Recognition (IJCV)

Segmentation As Selective Search for Object Recognition (ICCV)

<https://www.koen.me/research/selectivesearch/>

初始的分割区域方法 《Efficient Graph-Based Image Segmentation》



实验要求:

1. 要求自己编程实现**Selective Search**算法;
2. 开发语言为**python**或者**C/C++**;
3. 要求提交报告和代码。

