



硕士毕业论文

(毕业设计)

题目：？？？

姓名：林丽

学院：信息科学与技术学院

系： 计算机科学系

专业：计算机科学与技术专业

年级：2014级

学号：23020141153152

指导教师：曲延云 职称：副教授

2017年4月10日

基于汉字组合特征的自然场景中文文本检测

**[摘要]** 自。

**[关键词]**自然场景文本检测

Chinese Scene Text Detection Based on Chinese Character

Combination Features

**[Abstract]** Scene text detection and recognition is a hot topic in the fields of computer vision. It has been widely used in intelligent navigation, image retrieval, vision assistant system, etc. Although there exist a lot of research activities on scene text detection, Chinese text detection is still remained as a challenging problem. In this paper, we proposed a text detection method based Chinese character combination features. Our contributions are as follows: 1) We proposed simple Chinese character combination features, based on which a Chinese components merging algorithm, a word analysis algorithm, and a two-layer filtering algorithm are implemented afterwards. 2) We proposed an integrated algorithm for Chinese text detection, and the experimental results demonstrate the availability of our approach. 3) We built a dataset for Chinese scene text detection and recognition to evaluate the performance of our algorithm. The recall of our approach is 81.068% and the precision is 85.978%.

**[Key Words]**Scene text detection, Chinese character combination features, context

# 绪论

## 研究背景

### 1.1.1 研究意义

# 目标检测方法

目标检测（Object detection），作为计算机视觉领域的一个重要分支，也是ILSVRC（ImageNet Large Scale Visual Recognition Challenge）比赛[[1](#_ENREF_1)]的任务之一

**目标检测**指的是给定输入图像和类别（一种或多种类别），找出图像中所有属于给定类别的目标的位置和所属类别。

根据目标检测的定义，只有目标的位置和类别同时符合要求才是正确的检测。如图2.1的上图（A1-A4）为给定类别为(“Steel drum”)的目标检测标准（A1）和检测结果（A2-A4）。图A1为GroundTruth，即要检测的目标类别为“Steel drum”,且目标位置为两个绿色框所在的矩形。图A3的绿色框类别判断正确但框太大，位置错误。图A4的橙色框找到的位置是正确的，但是类别判断错误。因此，只有图A2是位置和类别都正确的。【插图2.1】



图2.1 单类目标检测与多类目标检测

目标检测按照要求类别的多少，可以分为单类目标检测（也称为“**定位”**）和多类目标检测。如图2.1，上图为给定类别为(“Steel drum”)的单类目标检测标准和检测结果，下图为给定类别为(“Steel drum”, “Person”，“Microphone”，“Folding chair”)的四类检测标准和检测结果。单类检测只需找到要求的那类的所有目标，多类检测需要找到所有要求的类别的所有目标。

目标检测按照目标的不同，又可分为行人检测[[2](#_ENREF_2)]，人脸检测[[3](#_ENREF_3)]，交通标志检测[[4](#_ENREF_4)]，文本检测[[5-10](#_ENREF_5)]等等【插图2.2】。虽然不同子领域因为目标本身属性的特殊性，有各自的不同算法。例如，基于Viola and Jones的人脸检测算法[[3](#_ENREF_3)]，基于HOG-SVM的行人检测算法[[2](#_ENREF_2)]。然而，这些检测方法同属于目标检测的领域，殊途同归，许多算法的框架和本质都是相通的。比如，HOG-SVM除了可以用在行人，用在交通标志检测中也能取到很好的效果[[4](#_ENREF_4)]。本文的文本检测方法正是从通用的目标检测方法中得到启发，并将其迁移至文本检测领域。因此，本章着重探讨通用的目标检测方法——从传统方法到近几年基于深度的检测方法，通过目标检测方法的发展引出本文选择SSD作为核心算法的相关背景和原因。

## 传统的目标检测方法

传统的目标检测方法一般分为两大类，基于滑动窗口和基于区域的方法。

基于滑动窗口的方法就是用不同大小的窗口在原图上进行滑动，并用分类模型判断每一个窗口是否包含目标，最后把分类模型判断为正样本的检测结果进行一定的后处理，比如非极大值抑制等。这种方法需要用不同尺度，不同长宽比的窗口对图像的所有位置进行遍历，虽然尽可能降低了目标漏检的可能性，但时间复杂度高，产生的候选区域过多，给后续的分类模型过滤噪声步骤带来很大的计算压力。

基于区域的方法首先根据低级特征（比如，光强，颜色，梯度等）把图像的像素分成不同的区域，再用分类模型对这些候选区域进行判断，过滤其中的噪声区域。该类方法得到的候选区域比滑动窗口法得到的窗口少得多，因此在时间和实用性上比基于滑动窗口的方法更具有优势。另一方面，候选样本的减少以及使用的低级特征的鲁棒性差也极大限制了该类算法的性能。

两类方法各有优劣，其共同的部分在于特征表示与模型分类，主要不同在于基于区域的方法需要额外的候选区域提取算法来选择候选样本。近年来，候选区域提取，特征表示，模型分类这三个关键技术受到了越来越多的关注，发展迅速。

### 候选区域提取

文献[[11](#_ENREF_11)]对比了目标检测领域近几年有代表性的十二种候选区域提取算法，包括Bing [[12](#_ENREF_12)]， CPMC[[13](#_ENREF_13)]， Edge Boxes [[14](#_ENREF_14)]，Endres [[15](#_ENREF_15)]，Geodesic [[16](#_ENREF_16)]， MCG [[17](#_ENREF_17)]， Objectness [[18](#_ENREF_18)]， Rahtu [[19](#_ENREF_19)]， RandomizedPrim’s [[20](#_ENREF_20)]**，** Rantalankila [[21](#_ENREF_21)]**，**Rigor [[22](#_ENREF_22)]**，**Selective Search [[23](#_ENREF_23)]。文献[[11](#_ENREF_11)]中，Selective Search和Edge Boxes相较于其他算法，其性能和速度的综合表现更好，使用也更加广泛。Selective Search是使用图像分割技术得到初始的小区域，再根据临近区域的颜色，纹理，尺寸，填充等相似性用贪心算法进行合并。Edge Boxes是用一个框内包含的完整的轮廓个数来判断框内是否包含目标，其依据是如果一个框内包含目标，则框内应该包含足够多的完整的轮廓，而不会与框的边界有很多相交的边缘。

### 特征表示

图像特征对图像分类，目标检测，语义分割等计算机视觉技术非常重要，好的图像特征可以极大的提高算法的性能。在没有引入深度学习之前，几乎所有好的目标检测系统都是基于HOG（Hitogram of Gradient）[[2](#_ENREF_2)], SIFT（Scale-Invariant Feature Transform）[[24](#_ENREF_24)]这两大经典特征提取算法。HOG统计了图像的局部方向梯度的直方图，其核心思想是用光强梯度（或边缘方向）来描述物体的局部轮廓。SIFT通过在尺度空间中寻找极值点作为特征点，并用与位置，尺度，方向相关的描述子来表征目标。

### 模型分类

常用的分类模型有很多，比如支持向量机（Support Vector Machine，简称SVM）[[25](#_ENREF_25)], 随机森林（Random Forest）[[26](#_ENREF_26)]， AdaBoost [[27](#_ENREF_27)]，K近邻（K-Nearest Neighbors，简称KNN）[[28](#_ENREF_28)]，朴素贝叶斯（Naïve Bayes）[[29](#_ENREF_29)], 人工神经网络（Artificial Neural Network，简称ANN）[[30](#_ENREF_30)]等等。基于结构风险最小化的支持向量机，尤其是采用了核方法的SVM，由于其鲁棒性强而被广泛应用于各种检测算法中。

## 基于R-CNN框架的目标检测方法

近年来深度学习受到了计算机视觉领域的极大关注，短短几年内在各种不同的视觉任务中取得了巨大的突破。相比于传统方法使用大量人工设计的特征，深度神经网络可以通过层次化的多层网络自动学习出具有更强鲁棒性的高级特征与语义信息。在众多深度网络结构中，CNN（Convolutional Neural Network）是发展最早、最快的结构之一。2012年，CNN [[31](#_ENREF_31)]在ImageNet的图像分类比赛中大放异彩，拉开了CNN快速发展的帷幕。2014年，Ross B.Girshick [[32](#_ENREF_32)]使用了Region Proposal与CNN结合的方法（R-CNN）代替了传统目标检测中滑动窗口与手工设计特征的方法，提出了R-CNN框架，使得目标检测取得了巨大突破，并开启了基于深度学习目标检测的热潮。

### R-CNN

1. **R-CNN的主要思想**

R-CNN的主要思想是利用候选区域算法从图像中提取出一定量的候选区域，并将其分别输入到CNN网络中，进行特征提取和分类判别。

1. **R-CNN的算法流程**

图【插图】是R-CNN的算法流程图。其检测过程主要分为3步：

第一， 用Selective search提取2000个候选区域；

第二， 用7层CNN网络模型（5个卷积，2个全连接层）提取卷积特征；

第三， 用多类线性SVM对每一个区域进行分类；

第四， 用softmax分类器对每一个区域进行回归得到更精确的位置。

1. **R-CNN的实验效果**

R-CNN算法通过把Region proposal与CNN结合起来，在Pascal VOC 2012 [[33](#_ENREF_33)]的检测结果比当时业内最高的记录高出了30%【插入表格或图】。紧随其后的SPP [[34](#_ENREF_34)], Fast R-CNN [[35](#_ENREF_35)]， Faster R-CNN [[36](#_ENREF_36)]正是基于R-CNN的这种框架进行的改进，这些方法被称为是R-CNN系列的目标检测法。

### SPP

* 1. **R-CNN算法的问题**

**问题一**：需要对Selective Search提取到的每一个候选框（proposal）都用前向CNN进行特征提取，大量重叠区域的卷积重复计算，影响算法的实时性；

**问题二**：由于存在全连接层，CNN的输入图像要求统一大小。因此，每个proposal需要统一裁切（crop）或者缩放（warp）到相同尺度大小，如图【插图】，这在一定程度上造成了图像畸变，影响算法的性能。

* 1. **SPP解决R-CNN问题的思路**

针对问题一，SPP采用整张图像作为CNN的卷积模块的输入，得到全图的卷积特征图。再依据每个候选框在原始图像中的位置，在卷积特征图上取出对应的卷积特征。采用这种策略，整张图只需要一次卷积，避免了重叠区域的重复卷积计算。由于卷积计算是整个CNN中最耗时的部分，因此，采用共享卷积计算的方式可以大大提高计算速度。

针对问题二，SPP在卷积和全连接层之间引入了一个称为Spatial Pyramid Pooling的池化层。该池化层可以将任意维度的卷积特征转换成固定维度的特征。该方法不但可以避免图像畸变，而且满足全连接层输入大小固定的要求。

* 1. **SPP的关键技术——Spatial Pyramid Pooling**

Spatial Pyramid Pooling是一种特殊的池化层，类似于图像金字塔。它的特点是不管输入图像大小如何，输出是固定维度的（一维）特征向量。

如图【插图】，Spatial Pyramid Pooling的输入是任意大小的图像（图中是卷积特征图），设为w\*h\*c，w和h分别表示每张图像的宽度和高度，c表示图像的通道数（图中c = 256）。设Spatial Pyramid Pooling的层数为k （图中k = 3），将每层的每个通道的图像分成nk\*nk个bin（图中，n1 = 1，n2 = 2，n3 = 4），在图像的每个bin中进行池化（图中采用最大池化，也可以用平均池化），并把c个通道的nk\*nk个bin的池化结果串联成一个nk\*nk\*c的一维向量（如图中的256 = 1\*1\*256， 4\*256 = 2\*2\*256， 16\*256 = 4\*4\*256分别为第1层，第2层，第3层的特征向量）。最后把k层的特征向量依次串联作为Spatial Pyramid Pooling的输出结果（输入到后续的全连接层）。

* 1. **SPP和R-CNN的网络结构对比**

采用了Spatial Pyramid Pooling后的SPP与传统的R-CNN网络结构不同，如图【插图】。传统R-CNN中，需要对每个候选区域进行裁切或压缩后再分别输入到卷积层，全连接层得到每个候选区域的分类结果。而SPP则首先对整张图进行全卷积，再对每个候选区域提取卷积特征，并分别输入到Spatial Pyramid Pooling和全连接层，得到对应的分类结果。

### Fast R-CNN

1. **R-CNN和SPP算法的问题**

**问题一**：R-CNN和SPP都是多步训练。R-CNN和SPP都需要分三步训练，先微调CNN网络（softmax分类器）用来提取特征，再训练SVM进行分类，最后训练回归器（regressor）对bounding-box进行精调。

**问题二：**R-CNN和SPP的训练都非常耗时，需要大量存储空间。对于R-CNN，每张图的每个候选区域的特征都要单独计算，而且需要将结果写入磁盘。SPP虽然共享了卷积计算，一张图像只需一次卷积，消除了R-CNN的卷积计算冗余问题，但是仍然需要将特征写入磁盘，这不但降低了训练速度，而且针对大训练集，需要大量存储空间。

**问题三：**R-CNN的检测速度很慢，SPP虽然提高了检测速度，但是CNN网络的卷积层参数无法更新，影响模型整体性能。R-CNN对每个候选区域单独使用CNN提特征，因此速度很慢。SPP共享卷积进行加速，同时为了解决不裁切和压缩的情况下，可以在CNN中输入任意大小的图像，加入了Spatial Pyramid Pooling，但无法解决Spatial Pyramid Pooling之前的卷积层的参数更新问题，导致SPP只能更新全连接层的参数，限制了模型的整体性能。

1. **Fast R-CNN的解决思路**

针对问题一和问题二, Fast R-CNN采用的策略是把特征提取，分类，bounding box回归整合成一个多任务（multi-task）的问题。对于问题一， Fast R-CNN摒弃了之前先用softmax训练CNN来提取特征，再用提好的特征来训练分类器和回归器这种分阶段训练的思路。Fast R-CNN不再用SVM，而是直接用CNN网络的softmax作为最后的分类器，甚至直接用另一个softmax作为回归器，把两个softmax同时接入损失层作为一个多任务问题来学习。这样直接把特征提取，分类，位置回归集成到一个网络中，使得训练只需一步，非常简洁的端到端模式。对于问题二，因为R-CNN和SPP的特征提取和SVM分类是分开的，所以必须先提取好所有样本的特征，这个数据量一般情况下比较大需要放入磁盘中。当要训练SVM时，需要把特征从磁盘中读入再训练。而Fast R-CNN将特征提取、分类、位置回归三者整合到一个多任务问题后，训练整个CNN模型就是同时学习了特征提取、分类、位置回归的参数，一步到位，不存在用磁盘作为中间媒介存储特征的过程。

针对问题三，Fast R-CNN采用了ROI Pooling来处理任意大小候选区域的输入问题，同时引入了分层采样策略（Hierarchical sampling，也称为Mini-batch Sampling）解决卷积层的参数更新问题。SPP的Spatial Pyramid Pooling层采用的是多层金字塔结构，Fast R-CNN将它进行简化为单层，称为ROI Pooling层，使得网络结构和训练更加容易。SPP无法更新卷积层的根本原因是每批训练样本（ROI）来自于不同的图像，而每个ROI往往感受野很大，甚至可能是整张图像大小。也就是说大量的训练样本（ROI）无论在前向传播还是后向传播都要处理几乎整张图像大小的输入数据，因此SPP如果要更新整个网络的参数（尤其卷积部分），将非常低效。Fast R-CNN的分层采样正是解决了这个问题。分层采样指的是用分层的方法采集训练样本——每次用来做批量随机梯度下降更新参数的样本（【角标，说明梯度下降法】mini-batch Stochastic Gradient Descent，本文用的随机梯度下降都是这个方法，以下用SGD简单表示）。假设每批需要R（R = 128）个ROI样本，首先采样N张图像，然后每张图像采样R/N个ROI。由于N取得小（N = 2），因此每张图像采样的ROI区域很多（R/N = 128/2 = 64），而这些来自于同一张图像的ROI在前向传播和反向传播中都是计算共享的（卷积）。实际上，R-CNN和SPP的采样策略是N = R, R/N = 1的特殊情况，每个ROI都来源于不同的图像。共享计算的区域越少，重复计算就越多，算法效率越低。因此，Fast R-CNN的采样策略比R-CNN和SPP的快了64倍。分层采样的策略大大提升了模型的训练速度，使得整个网络包括卷积部分的参数更新成为可能。

1. **Fast R-CNN的关键改进**

除了多任务，ROI Pooling和层次化采样，Fast R-CNN还采用了Truncated SVD [[37](#_ENREF_37)]来压缩模型和加速训练。下面简要介绍下这四个关键的改进技术。

* **多任务损失**

多任务损失包括了每个训练样本（ROI）的分类损失和bounding box的回归损失（特征提取任务作为分类和位置回归的前置步骤，必须接分类或回归来指导训练，没有独立的损失）。分类损失使用了softmax损失（log损失），回归损失使用了L1-smooth损失，而不是普通的L2损失，因为回归问题是无边界的，为了避免梯度爆炸，L2损失会对学习损率产生微妙影响，而鲁棒性的L1损失会消除这种敏感性【插图，L1损失，L2损失，L1-smooth损失】。

【公式】如公式，Lcls是分类损失项， Lloc是回归损失项。P = （p0, …，pK）表示在样本（某一个ROI）在K+1类的分类概率。K为总的类别数目，K+1表示除了总分类数，还要加上一个背景类。U是类别的索引，背景类索引为0， 其他类索为1， 2，…K。

表示样本的bounding box的回归, k表示属于第k类（每个样本对每一类都有一个回归）的回归，(x, y)表示对bounding box的质心的位置回归，w, h分别表示对bounding box的宽度和高度的回归。V= 表示和样本匹配的groundTruth的bounding box。超参数用来控制分类损失和回归损失的权重，在文献[[35](#_ENREF_35)]中， = 1。

[u>=1] = 1, u>=1, else 0（用公式表示）



* **ROI Pooling层**

ROI Pooling是2.2.2中介绍的Spatial Pyramid Pooling的特例，是一个k = 1的单层金字塔结构的Spatial Pyramid Pooling。二者的对比如下图【插图】所示，此处不再赘述其原理（详细请参考2.2.2）。

**【插图，多个尺度和单个尺度的不同】**

* **层次化采样**

层次化采样的原理在本节的第（2）部分“Fast R-CNN的解决思路”中已经介绍过了，这里不再赘述，只简要说明层次化采样策略可能潜在的问题。层次化采样中每批样本有一半以上来自于同一张图像，这些样本可能是相关的，所以有可能降低训练的收敛速率。然而，文献[[35](#_ENREF_35)]中的实验却表明这个问题并不存在，Fast RCNN收敛的迭代次数比R-CNN更少。

* **Truncated SVD加速**

如图【插图】，文献[[35](#_ENREF_35)]通过实验表明，对于检测任务，由于ROI数目很多，因此算法有一半的时间是花在了全连接层上。Fast R-CNN利用了Truncated SVD分解把一个大的全连接层换成了两个小的全连接层，该方法不但减少网络参数，压缩了网络模型，而且使得检测算法更快（训练和测试）。

1. **Fast R-CNN的算法流程**

Fast R-CNN算法的检测步骤主要有两步：

第一，用候选区域提取算法（Selective search）得到候选区域集（ROI）；

第二，将图像和ROI输入到Fast R-CNN中，得到每个ROI的分类概率和bounding box的回归值，用阈值过滤后得到检测结果。

如图，【插图】是Fast R-CNN的网络框架，可以分为四步：

第一， 用ConvNet得到整张图像的卷积特征图；

第二， 用ROI在第一步得到的卷积特征图上进行映射，得到每个ROI对应的特征；

第三，将每个ROI的卷积特征用ROI pooling层池化得到固定长度的特征，并输入到后续的两个全连接层；

第四，将全连接层得到的特征分别输入到用来分类和位置回归的两个softmax层，得到分类的概率分布和bounding box的回归值。



1. **Fast R-CNN的实验结果**

Fast R-CNN使用了多任务损失，分层采样，ROI pooling，Truncated SVD等方法改进了R-CNN和SPP算法。和R-CNN相比，Fast R-CNN的训练速度快了9倍，测试速度快了213倍。和SPP相比，Fast R-CNN的训练速度快了3倍，测试速度快了10倍。且Fast R-CNN的检测精度比R-CNN和Fast R-CNN都更高。

### Faster R-CNN

1. **Fast R-CNN的问题**

在Fast R-CNN的检测流程中，候选区域生成算法（Selective Search）成为了整个算法的瓶颈——体现在提取算法本身是用CPU实现，速度慢，且得到的候选区域过多，增加了后续步骤的计算压力。而且，候选区域生成算法的存在使得整个检测流程分成了级联的两个部分，不但影响检测速度，而且不便于训练和优化。

1. **Faster R-CNN的解决思路**

文献[[36](#_ENREF_36)]提出了一个称为Region Proposal Network（RPN）的深度全卷积网络代替了Fast R-CNN的候选区域生成算法（Selective Search），用来生成候选的区域。并且，把RPN和Fast R-CNN整合成一个检测网络，该网络称为Faster R-CNN。RPN网络可以在GPU中完成计算，且提取的候选区域比Selective Search等候选区域生成算法更好（更少但能保证足够的召回），更重要的是，和Fast R-CNN能够联合训练，成为一个完整的端到端检测框架。

1. **Faster R-CNN的网络结构**

* **Faster R-CNN的网络结构和检测流程**

图【插图】是Faster R-CNN的网络结构图，主要是由RPN网络和Fast R-CNN网络组成。RPN网络和Fast R-CNN网络共享了前面的卷积部分。因此，整个检测流程可以分为三步：

第一，用共享的卷积网络部分得到卷积特征图；

第二，将卷积特征图输入到RPN网络，得到候选的区域集；

第三，将卷积特征图和候选区域集同时输入到Fast R-CNN中，得到检测结果。

* **共享的卷积网络**

RPN网络和Fast R-CNN网络共享卷积的网络结构从Zeiler and Fergus model（ZF）[[38](#_ENREF_38)]和Simonyan and Zisserman model（VGG-16）[[39](#_ENREF_39)]继承来的。ZF模型和VGG-16模型的网络结构如图【插图】所示。

1. **Faster R-CNN的RPN**

* **锚点（Anchor）**

锚点（Anchor）是RPN的核心概念。它受滑动窗口机制的启发。**Anchor**指的是用滑动窗口搜索**特征图**的每一个位置（或者说用一个卷积核对特征图的每个位置进行卷积），将与每一个位置相关联的具有一定尺度、长宽比的区域映射在原图像上的矩形区域。

RPN的输出要求是：当对一个给定大小的特征图进行预测时，必须输出固定个数的候选区域（不管该候选区域被判断是背景还是目标，都是一个候选区域，只是所属类别不同而已）。然而，由于目标可能存在图像中的任意位置，可能有各种大小的尺寸。因此，一种可能的办法是可以在图像的每一个位置上放置一些固定的不同尺度，不同大小的参考区域。不论图像上有没有目标，目标在哪里，这些参考区域的个数，位置，大小都始终不变。这些参考区域就叫做Anchor。有了这些Anchor，那么网络的学习过程实际上就是以下两步；

第一，把groundTruth和Anchor进行一一匹配，把正确匹配上的Anchor标为正样本，没有匹配上的Anchor标为负样本。每个正样本的Anchor不但可以记录与其匹配的groundTruth的类别信息，也可以记录该groundTruth与自己的bounding box的匹配误差；

第二，反复调整网络参数，使预测结果不但可以正确判断Anchor是正样本还是负样本，而且对于正样本，也能够拟合出第一步中正样本与对应的bounding box的匹配误差。当网络可以做到这两点，就说明模型学习结束。

如图，【插图】利用256个3\*3的滤波核对卷积特征图进行卷积（3\*3的滑动窗口），每个位置卷积后得到256维的特征，将特征分别输入到分类层（cls）和回归层（reg）得到分类结果与bounding box的回归结果。因为每个位置与k（k = 9）个anchor相关联（3种尺度，3种长宽比），每个anchor都对应了类别信息和位置信息。Faster R-CNN中为了简化，类别信息只包含了两类信息，是背景或者不是背景，每个anchor的每个类别都有对应的得分概率，故每个位置的类别信息共2k维。位置信息包括了bounding box的中心点坐标（水平和竖直），宽度和高度这4维信息，故每个位置的位置信息共4k维。

* **RPN的损失函数**

和Fast R-CNN一样， RPN的损失函数由两部分组成，分类损失和bounding box的回归损失。如公式【公式】所示，设一个mini-batch中和第i个anchor（设为anchor\_i）相关联的训练样本为ai，当ai是正样本时，记与ai匹配的groundTruth为ai\*。pi表示用网络预测ai是正样本（目标）的分类概率，pi\*表示ai实际是正样本（目标）或负样本（背景），当ai是正样本时，pi\*=1, 否则pi\* = 0。ti是ai与anchor\_i的bounding box的差值，Ti\*是ai\*与anchor\_i的bounding box的差值。Ti和ti\*都是四维的向量，分别对应bounding box的中心点的水平坐标x,竖直坐标y，bounding box的宽度w和高度h。ti和ti\*的具体计算如公式【公式】所示。在公式【公式】中，第一项是分类损失，第二个是bounding box的回归损失，前者使用Smooth L1损失，后者使用二类的Log损失。Ncls和Nreg分别是正则参数，表示对应的样本个数。用来平衡两项损失，文献[[36](#_ENREF_36)]中取10，使得二者相近。





RPN和Fast R-CNN的损失不同在于，Fast R-CNN回归器的输入特征是由任意大小的ROI得到的，也就是说所有尺度的ROI都共享回归器的参数，只有一组回归器参数。而RPN的回归器有k组，每一组回归器负责一种尺度和一种长宽比的anchor，k组回归器没有参数共享，各自独立。因此，虽然每个anchor的输入特征维度是固定的（每张feature map上的每个位置都是同一个用3\*3的卷积核进行卷积），但是仍然可以预测多尺度的bounding box。

1. **Faster R-CNN的训练**

由于Faster R-CNN的RPN和Fast R-CNN共享卷积部分的特征，因此这两个网络的训练都会更新卷积部分的参数，这需要有特别的联合训练方式而不是独立训练两个网络。文献[[36](#_ENREF_36)]中提出了三种训练方式，交替训练，近似联合训练，非近似联合训练。这里我们介绍Faster R-CNN最终使用的四步交替训练流程：

第一，使用ImageNet上预训练好的模型训练RPN；

第二， 以第一步得到的RPN的候选区域为ROI输入，使用ImageNet上预训练好的模型，训练Fast R-CNN；

第三，固定RPN和Fast R-CNN共享的卷积部分，训练RPN；

第四，以第三步得到的RPN的候选区域为ROI输入，固定RPN和Fast R-CNN共享的卷积部分，训练Fast R-CNN；

1. **Faster R-CNN的实验结果**

Faster R-CNN将RPN和Fast R-CNN通过卷积共享整合成一个完整的端到端检测网络，使用GPU实现整个检测系统，大幅度提升了检测速率，并进一步提高了检测精度。

### R-CNN系列方法的总结

如图【插图】， 从R-CNN到Faster R-CNN，目标检测的四大基本步骤——候选区域生成，特征提取，分类，bounding box的回归（位置精修），逐步被统一到一个深度网络框架中。从检测的训练、测试速度来看，每一次改进都是一次跳跃，直到最后完全在GPU中实现，没有了重复计算。检测精度也是如此，稳步上升。R-CNN发展到Faster R-CNN，最初的传统方法已经不可与之比拟了。可以说，R-CNN系列的方法将整个目标检测领域带入了一个新的阶段。尽管Faster R-CNN不是最好的检测方法，但后来发展起来的YOLO，SSD，R-FCN等Single stage的方法都是基于R-CNN系列进行的改进或者是受到的启发，足见R-CNN系列方法对目标检测领域的重要性和影响力之大。

## Single stage的目标检测方法

Faster R-CNN虽然是一个end-to-end的完整的检测框架，但是仍然是一个two-stage的思路，即先检测出候选区域，再对候选区域进一步分类和精调，这在无形中限制了训练和检测的速率。随Faster R-CNN之后发展起来的YOLO（You Only Look Once）[[40](#_ENREF_40)], SSD（Single Shot MultiBox Detector）[[41](#_ENREF_41)]都是基于R-CNN系列的改进，且都属于Single stage的方法。由于本文重点在SSD，本节关于YOLO将做简要介绍，详细请参考文献[[40](#_ENREF_40)]。

### YOLO

1. **YOLO的主要思想**

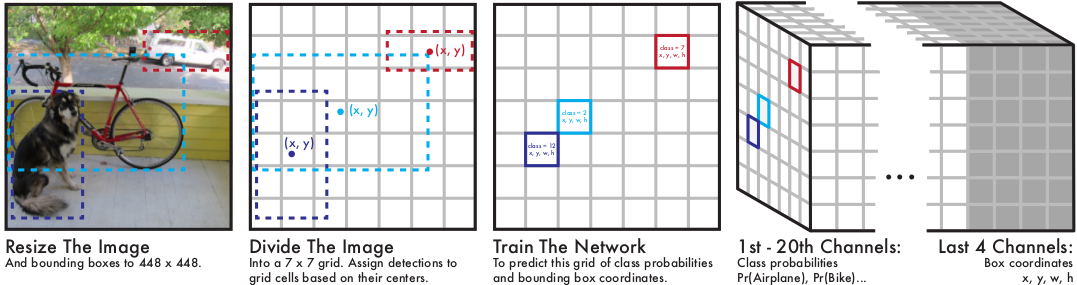
R-CNN系列的方法虽然经过了一系列的加速，但仍然无法达到实时的25fps的要求。影响速度的原因有很多，主要有以下几点：

第一，图像至少都要过两遍，第一遍找出候选区域，第二遍确定检测结果；

第二，为了保证性能，对输入图像大小有一定要求（例如，Faster R-CNN默认不小于600，虽然实际可以任意大小输入，但图像太小影响性能）；

第三，候选区域的数量偏多（例如，Faster R-CNN的默认是300）。

YOLO的主要亮点和目标就是快速，实时。YOLO延续了Faster R-CNN的端到端训练的特点，并把目标检测看作是一个回归问题，而且只需过一遍图像，这从它的名字You Only Look Once就可以看出来。它的设计思想是：把图像分割成多个网格，假如某个目标的中心落在其中某一个网格中，则该网格负责检测该目标的位置和类别。如图【插图】所示，图中的网格为7\*7大小，狗，自行车，小车的中心分别是紫色点，蓝色点，红色点。这三个点所在的网格负责输出紫色，蓝色，红色对应的矩形框，以及各自的分类信息（狗，自行车，小车）。



1. **YOLO的检测流程**

如图【插图】，YOLO的检测流程非常简单。只需三步：

第一，把原始图像缩放成固定大小；

第二，运行单个卷积网络；

第三，使用非极大值抑制后处理得到物体的位置和类别。



1. **YOLO的模型**

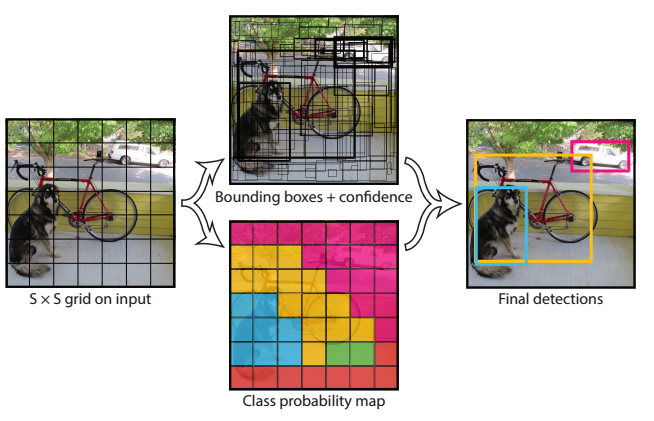
YOLO的模型如图【插图】所示。由于YOLO把检测看做是回归问题，因此，模型的主要任务就是回归目标的类别和位置。模型的逻辑流程如下：

第一，将缩放后的图像平均分割成S\*S的单元格；

第二，每个单元格负责预测B个候选框，每个框包括三类信息：位置信息，二分类信息，多分类信息。位置信息包括了bounding box的（x,y, w,h）共4个维度。（x,y）表示中心点坐标，w,h分别是宽度和高度。二分类信息用来判断框内是否包含了目标（不考虑具体是哪一类），只有1个维度。多分类信息提供了框内目标的所属类别，假设共有C类，则该信息为C个维度。由于YOLO假定以每个单元格为中心的目标只有一个，因此，在一个单元格上的B个候选框的多分类信息是共享的。也就是说，对于每个单元格，输出个数应为(4+1)\*B + C = 5B+C。对于测试，每个候选框的置信概率预测为公式【公式】，分类的总概率由二分类概率与多分类概率的乘积得到。

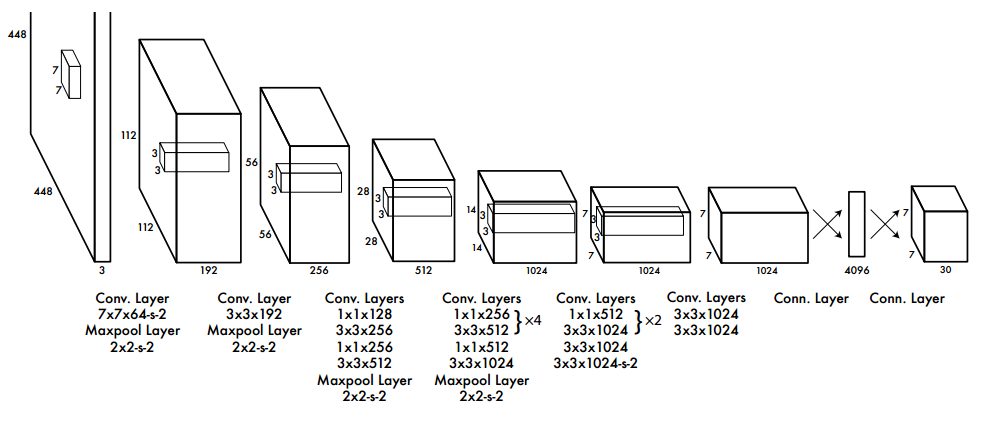


第三，最后的总输出层应有S\*S\*(5B+C)个单元。文献[[40](#_ENREF_40)]中，S = 7, B = 2, C = 20，输出单元为7\*7\*（4\*2+20）= 7\*7\*30 = 1470。



1. **YOLO的网络结构**

YOLO的网络结构参照了GoogleNet [[42](#_ENREF_42)]的设计。如图【插图】是YOLO的网络。可以看出，YOLO使用了24个级联的卷积层和2个全连接层。图中的多个3\*3卷积前接入1\*1的卷积是为了降维，这是参考了GoogleNet的经典Inception结构，如图【插图】。



1. **YOLO的实验结果**

YOLO的输入图像大小从Faster R-CNN的1000\*600减少到了YOLO的446\*446，候选区域的数量从300减少到了98，检测速度得到了巨大的提升（45fps），完全达到了实时的要求，但是精度比Fast R-CNN低一些。

1. **YOLO的优缺点**

* **优点**

第一，快速，已经达到实时的效果。把检测看作是一个回归问题，流程非常简洁；

第二，有全局的上下文信息。在网络结构中有对整张图像的全连接层操作，因此利用到了全局的上下文；

第三，泛化能力强。文献[[40](#_ENREF_40)]中用自然图像训练模型，并在艺术作品中测试，效果比其他的检测方法更好。

* **缺点**

第一，精度不够高，尤其对于小目标检测效果不好；

第二，由于网络结构中使用了全连接层，所以初始图像只能缩放到固定大小。另外，由于对初始图像缩放，可能造成了一定的形变，使得无法检测极端缩放比的目标；

第三，每一个单元格只预测一个类别，当多个物体中心落入同一个单元格时，无法被同时检测出。

### SSD: Single Shot MultiBox Detector

在保证精度的同时兼顾速度，一直以来都是目标检测的方向。在速度方面，YOLO一直遥遥领先，达到并超过了实时的基本要求，但是其精度差强人意，与state of the art有很大差距。另一方面，从精度来说，最为业内所追捧的当属R-CNN系列的Faster R-CNN，同样Faster R-CNN也因为被人所诟病的速度问题难以被投入实际应用。而在YOLO和Faster R-CNN之中取到完美平衡的正是SSD，它同时吸收了二者的精髓，Faster R-CNN的anchor box的思想，以及YOLO的回归和single stage的策略，并提出了多层特征融合的关键技术，很好的解决了YOLO的精度问题和Faster R-CNN的速度问题，创造了目标检测领域的新纪录。

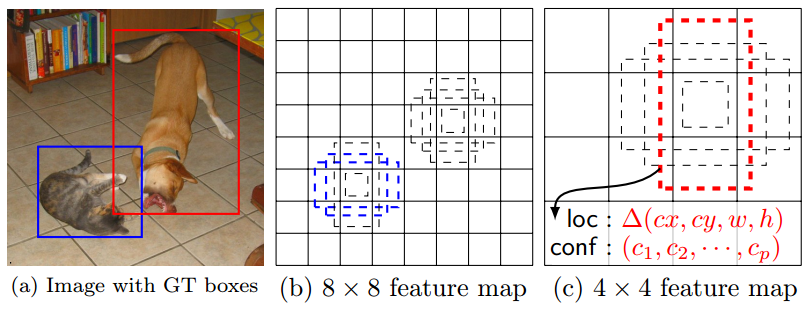
1. **SSD的检测流程**

SSD的检测比YOLO更简单，无需将图像缩放成固定大小，因为SSD网络是全卷积网络，可以接受任意大小的输入图像。



1. **SSD的框架**

SSD和YOLO一样，把目标检测看成是一个回归问题。与YOLO不同，SSD不是把原图上直接划分成网格，而是在特征图上，用滑动窗口提取每一个位置的特征，然后用特征回归得到目标的位置信息和类别信息。此外，SSD参考了Faster R-CNN的anchor思想，让每一个滑窗的位置对应了多个不同尺度，不同长宽比的anchor——在SSD中被称为Default Boxes，且Default Boxes不只应用于最后一个卷积特征层上，而是应用到多个不同大小的的特征层上，采用了多尺度的卷积特征融合来共同回归目标。



* **多尺度特征融合**

图像中的目标有大有小，好的目标检测算法应该能够检测出各种尺度的目标。为了可以检测出各种尺度的目标，方法之一是使用图像金字塔技术，将图像缩放为不同尺度大小，分别运用检测算法，最后把各种尺度下的结果进行融合。该方法中各种尺度需要单独处理，计算量大，而且耗时。SSD提出的多尺度特征融合不但能模拟出该效果，而且不同尺度间计算共享，大大提高了计算效率。文献[[43](#_ENREF_43), [44](#_ENREF_44)]中提出，低层特征有更多的细节，可以提高语义分割的质量，而文献[[45](#_ENREF_45)]中指出采用全局的上下文信息可以平滑分割结果。受这些方法的启发，SSD同时使用了低级特征图和高级特征图来检测。如图【插图】所示，SSD使用了大小和深度不同的六个特征层来进行预测，conv4\_3, conv7, conv8\_2, conv9\_2, conv10\_2, conv11\_2。每个特征层都使用不同的卷积模型（3\*3的卷积滤波核）进行预测。最后将多层特征图的预测结果进行融合作为检测输出。

如图【插图】，SSD的原图和groundTruth对应的bounding box如图（a）所示，图（b）和图（c）是两个大小分别为8\*8和4\*4的示例特征图，图上的矩形框分别表示Default Boxes在对应特征图上的映射（实际Default Boxes的大小符合原图尺寸，这里方便显示对Default Boxes进行了比例缩放）。从图中可以看出，不同大小的目标可能在不同尺度的特征图上回归得到（猫比较小，出现在8\*8的特征图上的蓝色框里。狗比较大，出现在4\*4的特征图上的红色框里）。由于SSD的多尺度特征图融合方法，使得不同大小的目标都可以检测到。

* **用于预测的卷积滤波核**

在YOLO和传统的分类方法[[31](#_ENREF_31), [46](#_ENREF_46)]中，都是使用全连接层+softmax进行预测结果（无论是分类结果，还是回归结果）。由于全连接层的参数个数与上一层卷积特征图的大小相关，因此，存在全连接层的网络要么固定输入图像大小，要么采用ROI pooling的策略将卷积特征图统一到固定特征维度，再与全连接层相连。实际上，除了采用全连接层+softmax的这种预测方式，文献[[43](#_ENREF_43)]给出了一个新的预测方式——使用卷积滤波+softmax也可以进行预测，而且可以对卷积的每个位置都得到一组预测值。假设需要预测的特征层有p个大小为m\*n的特征图（p也称为通道数），则使用一个3\*3\*p的滤波核对该特征层进行预测，由于每个位置都会得到一个预测值，因此最后会输出m\*n个预测值（对原图进行padding【脚注】）。也就是说，使用一个滤波核，可以得到一张和原图大小一样的预测图（response map）。如果使用多个滤波核，就可以得到多个预测图。假设是一个类别数为C的多分类问题，则可以用C个滤波核去预测，每个滤波核用来判断是否属于某一类的二分类问题，则C个滤波核就可以用来预测类别为C的多分类问题（需要接上softmax进行归一化）。SSD方法正是采用了这种卷积滤波核+softmax的方法进行预测。

* **Default Boxes**

Default Boxes和Faster R-CNN的**Anchor**类似。不同的地方在于，Faster R-CNN的Anchor只应用于最后一个卷积层上，而Default Boxes则是应用于多个层，且每一层的Default Boxes的尺度都不同。这里和Anchor一样，我们给出Default Boxes的定义。Default Boxes为用滑动窗口搜索**特征图**的每一个位置（或者说用一个卷积核对特征图的每个位置进行卷积），将与每一个位置相关联的具有一定尺度、长宽比的区域映射在原图像上的矩形区域。

如图【插图】，图（b）和图（c）上的矩形框分别表示Default Boxes在对应特征图上的映射（实际Default Boxes的大小符合原图尺寸，这里方便显示对Default Boxes进行了比例缩放）。从图中可以看出，不同大小的特征图，其defaut box的尺度不同（8\*8的矩形框比4\*4的矩形框小）。Default Boxes的具体尺度，长宽比选择将在后面详细介绍。这里我们分析下每张特征图对应的Default Boxes的个数以及用卷积滤波核预测得到的输出个数。

根据Default Boxes的定义，每一个Default Boxes都和特征图上的某一个位置相对应，且由于Default Boxes是由滑窗（或者卷积）方式在图像上放置，Default Boxes的个数与滑窗个数相关，也就是和特征图的大小相关。给定特征图大小和每个位置上Default Boxes的个数，则Default Boxes的总个数是固定的。假设特征图大小为m\*n，每个位置上Default Boxes个数为k，则总的Default Boxes个数为m\*n\*k。

在每个特征图的每个位置上，需要预测和每个Default Boxes相关的位置信息（位置形状信息？中心水平坐标，中心竖直坐标，宽度，高度，四个维度）以及该Default Boxes内可能含有某一类目标的概率值。假设总类别数为c, 则需要的滤波核个数为c+4。由于每种尺度，每种长宽比的Default Boxes需要采用不同的滤波核，故总滤波核个数为k\*(c+4)。利用这些滤波核在大小为m\*n的特征图上进行预测，将会输出m\*n\*k\*(c+4)个输出。

1. **SSD的网络结构**

* **SSD的网络结构**



SSD的网络结构如图【插图】所示，可以分解为5个部分：

第1部分，使用VGG-16的conv1~comv5\_3的13个卷积层；

第2部分，使用VGG-16的fc6, fc7，并将其改造成全卷积；

第3部分，增加了4个卷积模块，这4个卷积模块是由两个卷积层组成，分别是一个1\*1的卷积和一个3\*3的卷积；

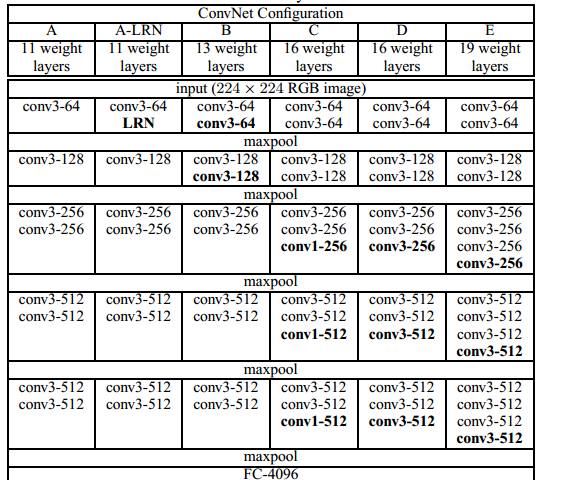
第4部分，在conv4\_3, conv7, conv8\_2, conv9\_2, conv10\_2, conv11\_2这6个卷积层后分别接上一个3\*3的卷积层作为分类层，并把6个分类层的结果融合起来，得到初始的检测结果；

第5部分，在初始的检测结果后接上非极大值抑制，得到最终的检测结果。

* **VGG-16的网络结构**

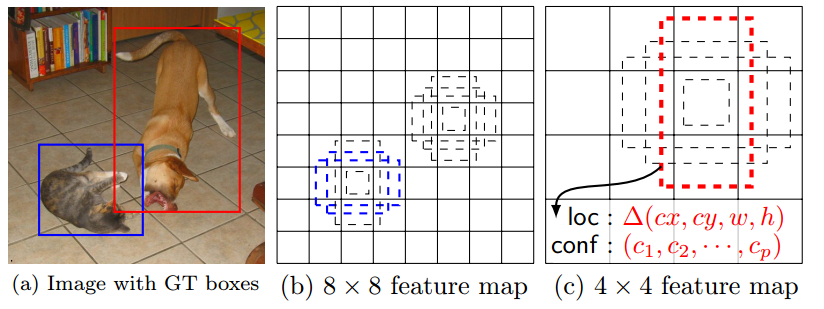
SSD中用到的经典的VGG-16的网络结构图和网络配置如图【插图】所示。





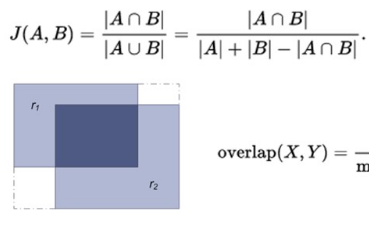
1. **SSD的训练**

和Faster R-CNN的Anchor一样，在训练SSD时，图像的原始groundTruth要被赋予到每层的Default Boxes上。如图【插图】所示，原始的groundTruth是图（a）中的红色和蓝色实线框。但实际训练中用到的是groundTruth Default Boxes，也就是图（b）和图（c）中所有虚线框。2个蓝色虚线框和1个红色虚线框是训练用到的正样本，其他的几个黑色虚线框是负样本。对于每一个正样本，以红色虚线框为例，它的groundTruth信息包括了位置回归信息，以及类别信息。位置回归信息指的是相匹配的groundTruth与groundTruth Default Boxes之间的误差，也就是红色实线框与红色虚线框之间的误差——分别在中心水平坐标，中心竖直坐标，宽度，高度，四个维度的差值。类别信息指的一个p维（假设共p类）的概率分布向量c1 + c2 +…+ cp = 1, 0<=ci<=1。当目标是第k类时，ck=1，ci = 0，I = 1,2,…,k-1, k+1, …p。groundTruth Default Boxes的类别信息指的是与该Default Boxes相匹配的groundTruth的类别。图【插图】中的红色虚线框对应的红色实线框内是狗，因此在类别概率分布上只有狗对应的类别为1，其他为0。采用这种方式把每一个Default Boxes和groundTruth对应后，就可以端到端得进行损失函数的计算和使用反向传播算法（Back propagation）进行参数更新了。



* **Default Boxes与groundTruth的匹配策略**

Jaccard overlap。对于两个矩形A和B，|A交B|指的是矩形A和矩形B相交区域（深蓝色矩形）的面积。|A并B|指的是包含矩形A和矩形B的最小外接矩形（虚线矩形）的面积。A和B的Jaccard overlap指的是|A交B|与|A并B|的比值。



每个Default Boxes（记为bi）与每个groundTruth（记为gj）进行匹配，当满足bi和gj的Jaccard overlap阈值大于0.5时，表明bi和gj匹配。由于groundTruth之间不会存在相互间Jaccard overlap大于0.5（大小相近，位置相近），而Default Boxes之间有可能存在（如图【插图】（b）中的两个蓝色虚线矩形框）。因此，每个bi只可能与一个gj相匹配，而每个gj可能与多个bi匹配。

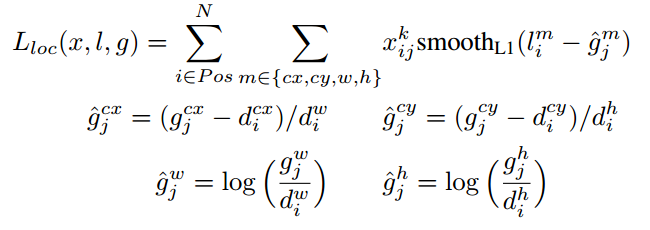
* **目标函数（损失）**

【由于每个Default Boxes已经和groundTruth进行匹配过（训练样本的准备），因此每个Default Boxes已经具有了groundTruth信息——包括类别信息和位置信息，类别信息包括该Default Boxes属于哪一类（包括背景类，如果是背景类，表示是负样本，其他都是正样本），位置信息主要指正样本与与其匹配的groundTruth的bounding box的差值，包括中心水平坐标，中心竖直坐标，宽度，高度，四个维度的差值。】

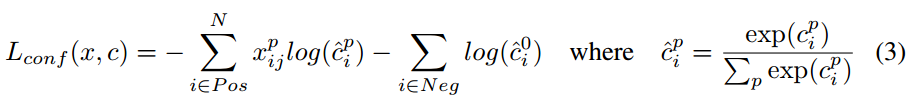
SSD的损失函数是MultiBox [[47](#_ENREF_47)]的多类扩展，它包含了位置回归损失（loc）和分类置信损失（conf）。如公式【公式】所示。用来平衡两类损失，通过交叉验证确定该值为1。



X和l表示的是对Default Boxes的类别和位置的预测。C和g表示Default Boxes的实际（groudTruth）类别和位置。在公式【公式】中，Lconv表示的位置回归损失，其采用的是smoothL1损失，具体定义如公式【公式】所示.i指的是所有Default Boxes的索引，j指的是所有groundTruth的索引。x\_ij\_p表示预测的第i个Default Boxes与第j个groundTruth在类别p上的匹配程度。根据上面的“Defaut box与groundTruth的匹配策略”，可以知道每一个groundTruth一定会至少有一个Default Boxes与其匹配，因此有。N表示匹配的Default Boxes的个数。



Lloc表示的是分类置信损失，采用的是在多类的softmax损失，其具体定义如公式【公式】所示。

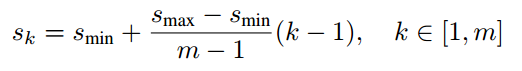


* **Default Boxes的尺度和长宽比选择**

Default Boxes和Anchor不同的地方在于它应用在不同层，大小不同的特征图上。虽然在深度网络中，不同层的特征图有不同大小的感受野，但SSD框架并没有让每层的Default Boxes与实际的感受野大小一一对应关联，而是另外设计了Default Boxes的布局（尺度和长宽比的分布），让不同的特征图可以检测不同大小的目标。SSD的Default Boxes的具体设计如下：

设用来预测的特征图来自于m层（m = 6），且每个位置的Default Boxes的初始长宽比有5种，

每层特征图的尺度：将最小（层数最深）的特征层尺度smin设为0.2，最大（层数最浅）的特征层尺度smax设为0.9，其他m-2层的尺度从smin-smax采用等间距分布。如公式【公式】所示。每层的尺度大小sk，k = 1, 2, …，m，如表【表格】所示。



设每层特征图的每个位置的Default Boxes的宽度和高度：对于第k层（1<=k<=m），Default Boxes的某个窗宽比ar对应的宽度wk和高度hk如公式【公式】。对于长宽比ar为1的情况下，增加一种Default Boxes的尺度，，因此实际上共有6种尺度，{1,1’, 2,3,1/2，1/3,}。每种尺度的长宽比和尺度对应如表【表格】所示。

，

每个Default Boxes的中心：为了保证每个Default Boxes的中心不在图像的边界上，对每个中心都进行了一定平移，如公式【公式】所示：



* **Hard Negative Mining**

由于图像中的目标往往只占图像的一个很小的比例，大部分图像区域是背景。如果按照上面的“Defaut box与groundTruth的匹配策略”，则大部分的Default Boxes都会被标定为负样本，这会造成正负样本的严重不平衡。因此，SSD使用了Hard Negative Mining策略——先按每个Default Boxes的最高置信损失（所有类别中取最高的）从高到低排序，然后选择排行靠前的Default Boxes样本，使得正负样本的比例保持在1:3。由于置信损失越大，说明和groundTruth越接近，即分类越正确，且位置误差越小的Default Boxes被优先选择。根据文献[[41](#_ENREF_41)]，采用该策略后可以使得训练速度加快并且训练更加稳定。

* **数据扩增**

为了使模型对各种尺度和形状的目标更加鲁棒，文献[[31](#_ENREF_31)]中使用了数据扩增来增加训练样本的丰富性。每一个训练图像由以下三种策略随机选择一个生成：

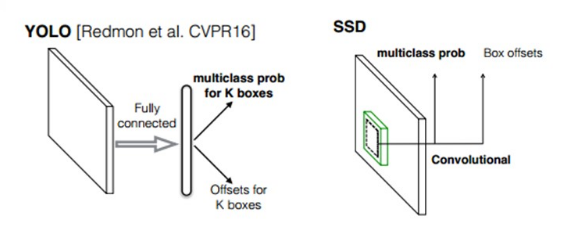
第一，使用整张原始输入图像；

第二，从原始图像中采样出一个图像块，使得与目标的最小Jaccard Overlap的大小值分别为0.1，0.3,0.5,0.7,0.9；

第三，从原始图像中随机采样出任意图像块。

采样的图像块大小是原始图像的0.1~1之间，长宽比在1/2~2之间。采样的图像块在训练前统一缩放成了固定大小，并且以0.5的概率进行水平翻转。此外，所有样本使用了文献[[48](#_ENREF_48)]中的图像测度的扭曲技术（比如亮度，对比度，色度，饱和度等按一定概率调整）。

1. **SSD与YOLO的对比**



SSD和YOLO的相同点：

第一，二者的网络都使用了已经预训练过的深层模型作为基本的卷积模块。SSD是使用了VGG16， YOLO是使用了GooleNet；

第二，从输入图像到得到检测结果都只用运行一个网络，都是single stage的方法；

第三，二者都使用了非极大值抑制进行后处理。

第四，二者的网络输出都同时包含了目标的位置，类别信息。

SSD和YOLO的不同点：

第一，SSD是一个全卷积网络，不存在全连接层，可以处理任意大小的输入；YOLO中存在2个全连接层，输入要求必须是固定大小，因此比SSD多一个缩放图像大小的步骤；

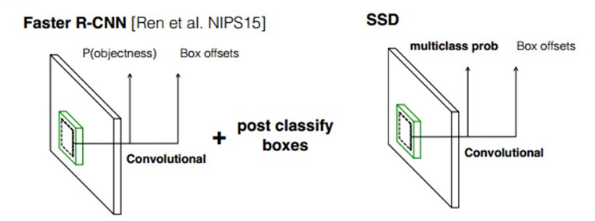
第二，SSD是在多个不同的特征图上预测结果，而YOLO的预测只在最后一个特征图上；

第三，从网络深度来看，YOLO比SSD更加深，这导致了YOLO模型也更大。这是因为YOLO使用的基本的深度卷积模型是GoogleNet（YOLO共24层），SSD使用的是VGG16（SSD共13+4+1=18层）；

第四，SSD需要预定义Default Boxes，并且把每个Default Boxes与groundTruth信息相关联，而YOLO是在原图像上划分网格，让每个网格负责中心落在网格里的groundTruth。YOLO的每一个网格关联的两个box共用一组类别信息，而对于SSD，同一个位置的每一个Default Boxes都拥有各自的类别信息，相互不共享。也就是说对于YOLO，当不同（类别的）目标的中心同时落在一个网格里时，无法被同时检测，但是SSD能检测出这种中心重叠的目标；

第五，从性能来说，YOLO更注重速度，速度比SSD更快一些，且输出的候选区域个数比SSD更少，而SSD同时兼顾速度和性能，性能比YOLO好很多；

1. **SSD与Faster R-CNN-的对比**



相同点：

第一， 二者的网络都使用了已经预训练过的深层模型作为基本的卷积模块。SSD是使用了VGG16， Faster R-CNN使用了VGG16和ResNet，事实上基本的卷积模块可以使用任何模型；

第二，网络都同时学习分类任务和位置回归任务，网络输出都包含了目标类别和位置信息；

第三，都预定义了Anchor或者Default Boxes，groundTruth都要和Anchor或者Default Boxes进行关联；

不同点：

第一，Faster R-CNN由RPN和Fast R-CNN（post classify boxes）两步构成，而SSD只需一步即可完成检测；

第二，Faster R-CNN的Anchor只用在单个特征层上，SSD的Default Boxes应用在多个特征层上，二者在尺度和长宽比的设置上也有所不同；

第三， Faster R-CNN的RPN和SSD虽然很相似，但是RPN的分类只是一个二分类，判断是否是目标，而没有判断具体是哪一类；

第四，Faster R-CNN需要将RPN和Fast R-CNN进行四步迭代训练，而SSD的训练只需一步端到端的训练；

第五，Faster R-CNN的Fast R-CNN模块有全连接层，虽然因为ROI pooling策略使得Fast R-CNN可以输入任意大小的候选区域集。但SSD整个网络是全卷积的，训练和优化更为容易；

## 对比总结与扩展延伸

### 对比与总结

1. **各方法关键技术总结**

目标检测方法从R-CNN系列发展到Single stage的SSD，回顾几大方法的关键创新，R-CNN将CNN首次成功应用到目标检测中，SPP提出了SPP来处理任意大小的ROI输入问题，Fast R-CNN将SPP简化成ROI pooling并且引入了多任务损失将位置回归和分类两大问题统一到一个框架中，Faster R-CNN提出了Anchor和RPN网络，YOLO将目标检测当做一个回归问题求解，SSD提出了Default Boxes和多尺度特征融合。以上的每一个方法都基于前人，并超越前人，其改进的核心思想主要在两个方面：

第一，整合多个步骤和任务，流程整体化；

第二，不同步骤，不同任务计算共享；

【表格1】方法名字，年份，论文发表，作者，关键技术，

【表格2】方法名字，是否开源，代码框架，代码语言，代码网址

1. **各方法性能对比**

技术的发展源于需求，正是性能和速度的要求推动了目标检测方法朝着当今的路线和趋势进步和发展。在保证精度的同时兼顾速度，是目标检测的本质需求。只有性能和速度双重发展，才能让目标检测方法从科学理论走向实际应用。从R-CNN到SSD，每个方法都在性能或者速度上有了巨大的提高，这一点从表【表格】和图【插图】中可以看出。在这些方法中，从性能和速度综合上来看，SSD是最具竞争力的方法。这也是本文选择用SSD作为核心方法的关键原因。

**【表格】方法名字，速度，性能，数据库**

**【画图】**

### 扩展延伸

本章介绍的方法是当前目标检测领域比较有代表性的算法，而且这些方法与本文的核心算法SSD有着密切的关联，SSD直接或者间接参考了这些算法的思想。事实上，除了本章介绍的这些，目标检测领域还有其他经典算法，甚至有些是本节方法的升级版，由于和本文方法关系不大，本章没有进行介绍，这里仅简单列出方法的名字和关键技术，详细请查阅本文的参考文献。

【表格】

YOLO2, R-FCN， MultiBox，ION，OHEM，PVANet, CRAFT

## 本章总结

本章介绍了近年来目标检测方法的发展历程，从传统的目标检测方法，到基于R-CNN框架的系列方法——R-CNN，SPP，Fast R-CNN，Faster R-CNN，再到Single stage的YOLO，以及本章重点介绍的本文核心算法SSD。从各大方法的对比和改进思路，来理解SSD方法设计的灵感来源，以及它所具备的其他方法没有的优势。

# 基于SSD的水平文本检测



## SSD与文字检测

### 目标检测与文字检测

目标检测涵盖的应用很多，文字检测作为其中的一个分支，指的是专门针对目标为文字的一类检测方法。通用的目标检测包括范围很广，文字只是其中一个小的类别。在文献[[41](#_ENREF_41)]中，SSD不只在类别数只有20的PASCAL VOC [[33](#_ENREF_33)]的目标检测数据库上进行测试，更将它用在类别数更多的COCO [[49](#_ENREF_49)]数据集（91类）和ILSVRC DET [[1](#_ENREF_1)]（200类），并取得了很好的效果。这证明了SSD是一种通用的目标检测框架——对文字检测也可能适用。

尽管如此，但文字作为一个特殊的类别，和普通的行人，车辆，人脸，动物等特定目标检测具有很大的差异性。表 3 - 1是PASCAL VOC的20类的类别信息，图 3 - 1是MS COCO的91类的类别信息。可以看出，无论是在类别少的PASCAL VOC，COCO，还是类别数较多的ILSVRC DET库上，都没有“文字”这一类。因为从实际场景来说，文字往往不是作为独立的目标个体出现，而是依附在特定的媒介上，例如各种交通工具，建筑，服装，家具，食品包装，交通标志牌等等。生活中随处可见的地方都会出现文字，文献[[50](#_ENREF_50)]中指出，即使是随意收集的图像库，其中就至少有一半的图像包含文字。文字作为一种书写符号，没有固定的形态，其颜色，尺度，形状，语言，布局等具有非常多样的差异性。文字的这种特殊的媒介依附性、分布广泛性、和多样性是其他的普通目标类所没有的。因此，将通用的目标检测算法应用于文字检测是一个非常大胆和创新的想法。

表 3 - 1：PASCAL VOC 的20类类别名字

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| aeroplane | bicycle | bird | boat | bottle |
| bus | car | cat | chair | cow |
| dining table | dog | horse | motorbike | person |
| potted plant | sheep | sofa | train | tv/monitor |

图 3 - 1：MS COCO的11个大类，91个小类名字和图标

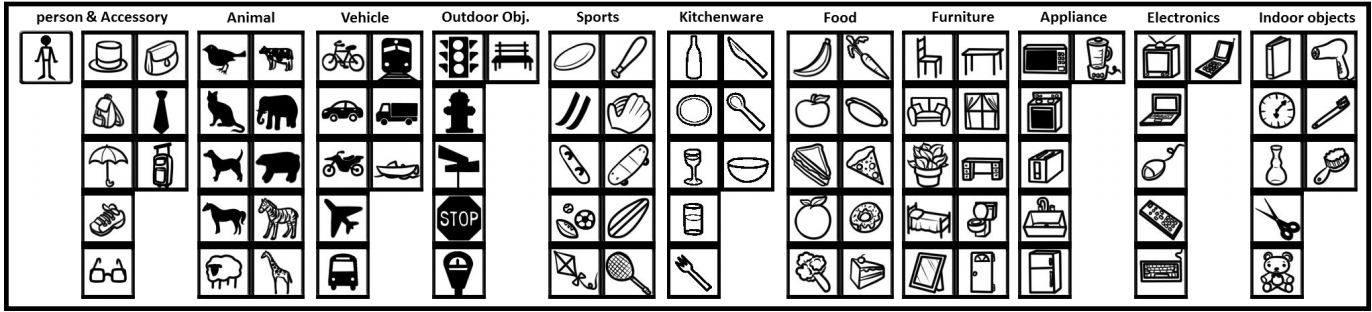


表 3 - 2 ILSVRC DET的200类类别名字

|  |  |
| --- | --- |
| **类序号** | **类别名字** |
| **1-100** | accordion, airplane, ant, antelope, apple, armadillo, artichoke, axe, babybed, backpack, bagel, balancebeam, banana, bandaid, banjo, baseball, basketball, bathingcap, beaker, bear, bee, bellpepper, bench, bicycle, binder, bird, bookshelf, bow, bowtie, bowl, brassiere, burrito, bus, butterfly, camel, canopener, car, cart, cattle, cello, centipede, chainsaw, chair, chime, cocktailshaker, coffeemaker, computerkeyboard, computermouse, corkscrew, cream, croquetball, crutch, cucumber, cupormug, diaper, digitalclock, dishwasher, dog, domesticcat, dragonfly, drum, dumbbell, electricfan, elephant, facepowder, fig, filingcabinet, flowerpot, flute, fox, frenchhorn, frog, fryingpan, giantpanda, goldfish, golfball, golfcart, guacamole, guitar, hairdryer, hairspray, hamburger, hammer, hamster, harmonica, harp, hatwithawidebrim, headcabbage, helmet, hippopotamus, horizontalbar, horse, hotdog, iPod, isopod, jellyfish, koalabear, ladle, ladybug, lamp |
| **101-200** | laptop, lemon, lion, lipstick, lizard, lobster, maillot, maraca, microphone, microwave, milkcan, miniskirt, monkey, motorcycle, mushroom, nail, neckbrace, oboe, orange, otter, pencilbox, pencilsharpener, perfume, person, piano, pineapple, pingpongball, pitcher, pizza, plasticbag, platerack, pomegranate, popsicle, porcupine, powerdrill, pretzel, printer, puck, punchingbag, purse, rabbit, racket, ray, redpanda, refrigerator, remotecontrol, rubbereraser, rugbyball, ruler, saltorpeppershaker, saxophone, scorpion, screwdriver, seal, sheep, ski, skunk, snail, snake, snowmobile, snowplow, soapdispenser, soccerball, sofa, spatula, squirrel, starfish, stethoscope, stove, strainer, strawberry, stretcher, sunglasses, swimmingtrunks, swine, syringe, table, tapeplayer, tennisball, tick, tie, tiger, toaster, trafficlight, train, trombone, trumpet, turtle, tvormonitor, unicycle, vacuum, violin, volleyball, waffleiron, washer, waterbottle, watercraft, whale, winebottle, zebra |

### 已有算法的问题

在本文方法之前，Zhuoyao Zhong等人[[51](#_ENREF_51)]将Inception [[42](#_ENREF_42)]与RPN [[36](#_ENREF_36)]相结合，成功将Faster R-CNN用于文字检测，证明了通用目标检测算法用于文字检测的可行性，但在性能和速度上仍存在明显不足。与本文方法最相似的是Xiang Bai等人[[52](#_ENREF_52)]提出了TextBoxeses的方法——对SSD进行了修改，使其可以用于文字检测，并达到了业内领先水平。但是，该方法存在以下问题：

第一，预训练样本太大，使用了80万的Synth Text [[8](#_ENREF_8)]数据库进行预训练。训练样本大对硬件的内存，计算能力要求更高，更重要的是，训练时间比较长（在文献[[52](#_ENREF_52)]中虽然只使用了一天的时间预训练，但是样本的利用率低，epoch【脚注说明epoch】不到2。如果要更充分利用训练样本，则训练时间将大大增加）；

第二，文献[[52](#_ENREF_52)]中没有探究Default Boxes设置的合理性，例如将每个位置上的Default Boxes长宽比设置为[1,2,3,5,7,10]这几种长宽比。该方式使得搜索空间变大（原始SSD只有5种长宽比，本文方法只使用了3种），增大了候选区域的个数。更重要的是，无法将Default Boxes的设置方法推广到其他类型的目标检测问题中；

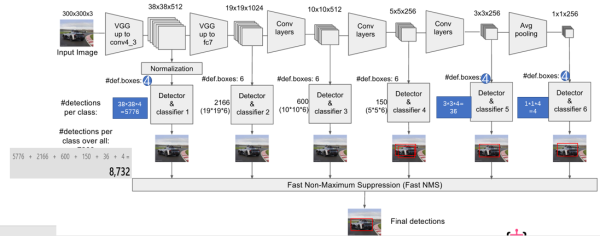
第三，TextBoxeses使用了多尺度输入的方式来提高检测效果，但速度慢了很多；

第四，TextBoxeses只在英文数据库上验证方法，并没有证明其在多语言上的可行性；

第五，TextBoxeses只能处理水平文本检测，无法直接用于倾斜文本的检测。

### 本文改进SSD的思路

## 本文方法的流程图



如图【插图】是本文方法的流程图，主要包括以下几步：

第一，将图像缩放成四种不同的尺度，500\*500， 700\*500，700\*700， 700\*300

第二，对每种尺度的图像单独运行网络进行图像检测；

第三，将不同尺度的检测使用“基于投票的多尺度融合机制”进行后处理。

## Default Boxes的改进

根据SSD的Default Boxes的设置经验，初始我们将Default Boxes设置如表 3 - 3所示。

表 3 - 3：初始Default Boxes的尺度和长宽比设置

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Default Boxes设置** | **conv4\_3** | **fc7** | **conv6\_2** | **conv7\_2** | **conv8\_2** | **conv9\_2** | **说明** |
| **尺度(%)** | 10~20 | 20-37 | 37-54 | 54-71 | 71-88 | 88-100 | 第1层大小: 10~20 第2层至第6层大小: 20~100 |
| **长宽比** | [1, 4, 8] | [1, 4, 8] | [1, 4, 8] | [1, 4, 8] | [1, 4, 8] | [1, 4, 8] | 实际要考虑k和1/k两种，这里为了简化，只写出k |

然而，通过实验，我们发现效果并不理想。表格【表格】是我们对每一个特征层的预测结果的统计，可以发现\*\*\*。图【插图】\*\*\*。综上可以得出以下结论；

第一，不同特征层上的目标分布不平均，导致某些层漏检和误检较多；

第二，对于小目标，漏检情况比较严重；

为了能够设置合理的Default Boxes的尺度和长宽比，我们通过对训练样本进行分析，来确定文字的尺寸，长宽比设置的合理范围，并将这些分析统计值作为Default Boxes的参考值，利用分析和实验相结合的方式来最终确定Default Boxes的设置，并总结出了可以用于其他目标检测问题的宝贵经验。

### 训练样本与各特征层预测分析

对任何一个学习问题，样本都是至关重要的。而有一个问题常常被忽略掉，那就是对样本的分析。即使面对同一类问题，不同应用背景的数据往往有很大的差异性。因此，分析训练样本，无论是通过观察还是利用统计分析的手段，都应该是模型设计的第一步。

图 3 - 2、图 3 - 3、图 3 - 4分别是ICDAR2013-train库上文字的宽度、高度和面积相对于图像大小的统计情况，从图中可以看出，文字的面积占图像面积的比例平均在3%左右，文字的高度与图像高度的比值平均在9%左右，文字的宽度与图像宽度的比值平均在24%左右。图 3 - 5是ICDAR2013-train库上文字的长宽比（宽度/高度）的比例分布情况，从图中可以看出，文字的长宽比主要分布在0~10中间，平均在4左右。通过统计分析，我们可以知道文字在图像中大部分比较小，且文字的长宽比偏向于水平的长条形。

图 3 - 2：ICDAR2013-train库上文字的宽度相对于图像宽度的比例分布（%）

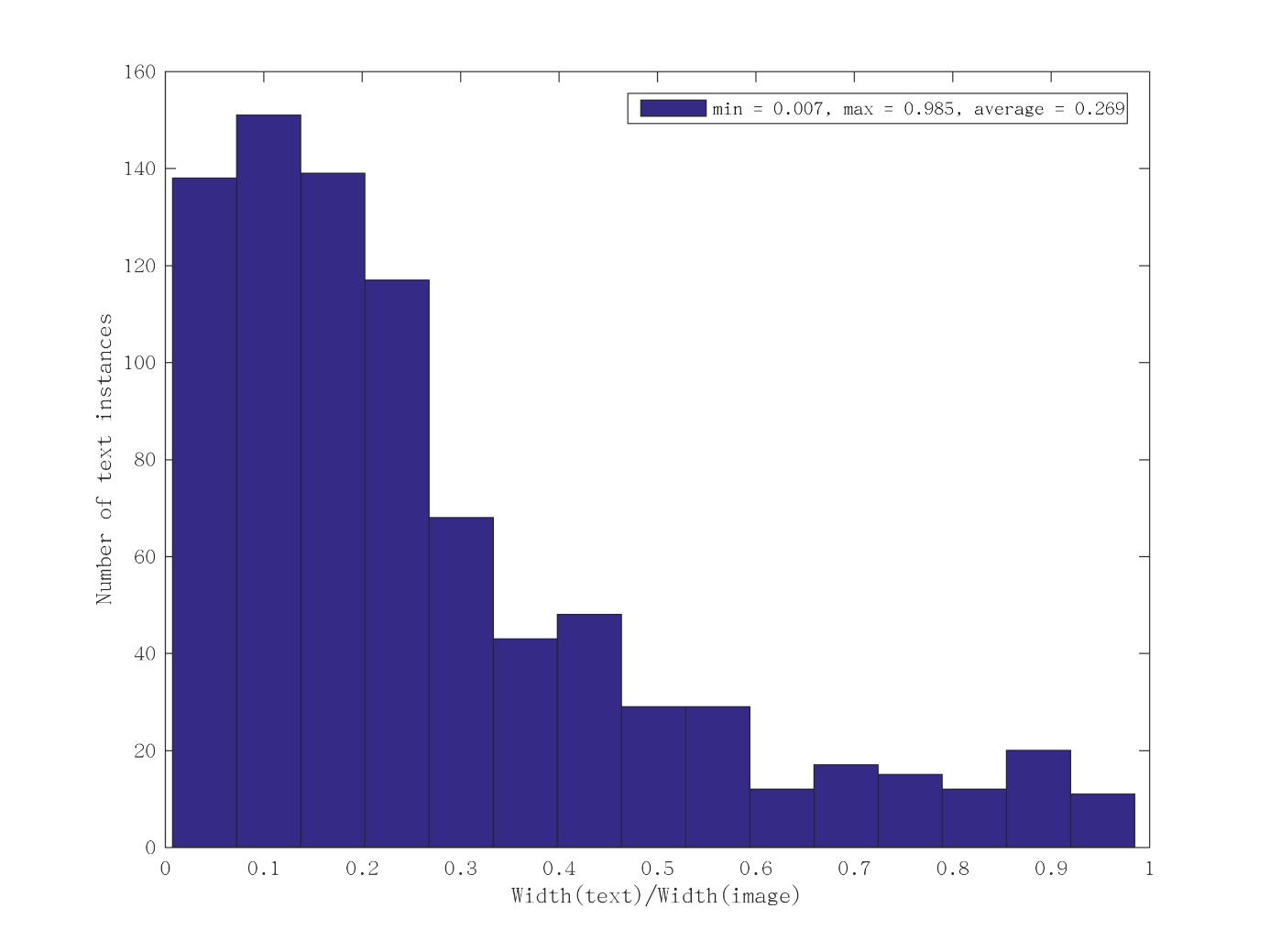


图 3 - 3：ICDAR2013-train库上文字的高度相对于图像高度的比例分布（%）

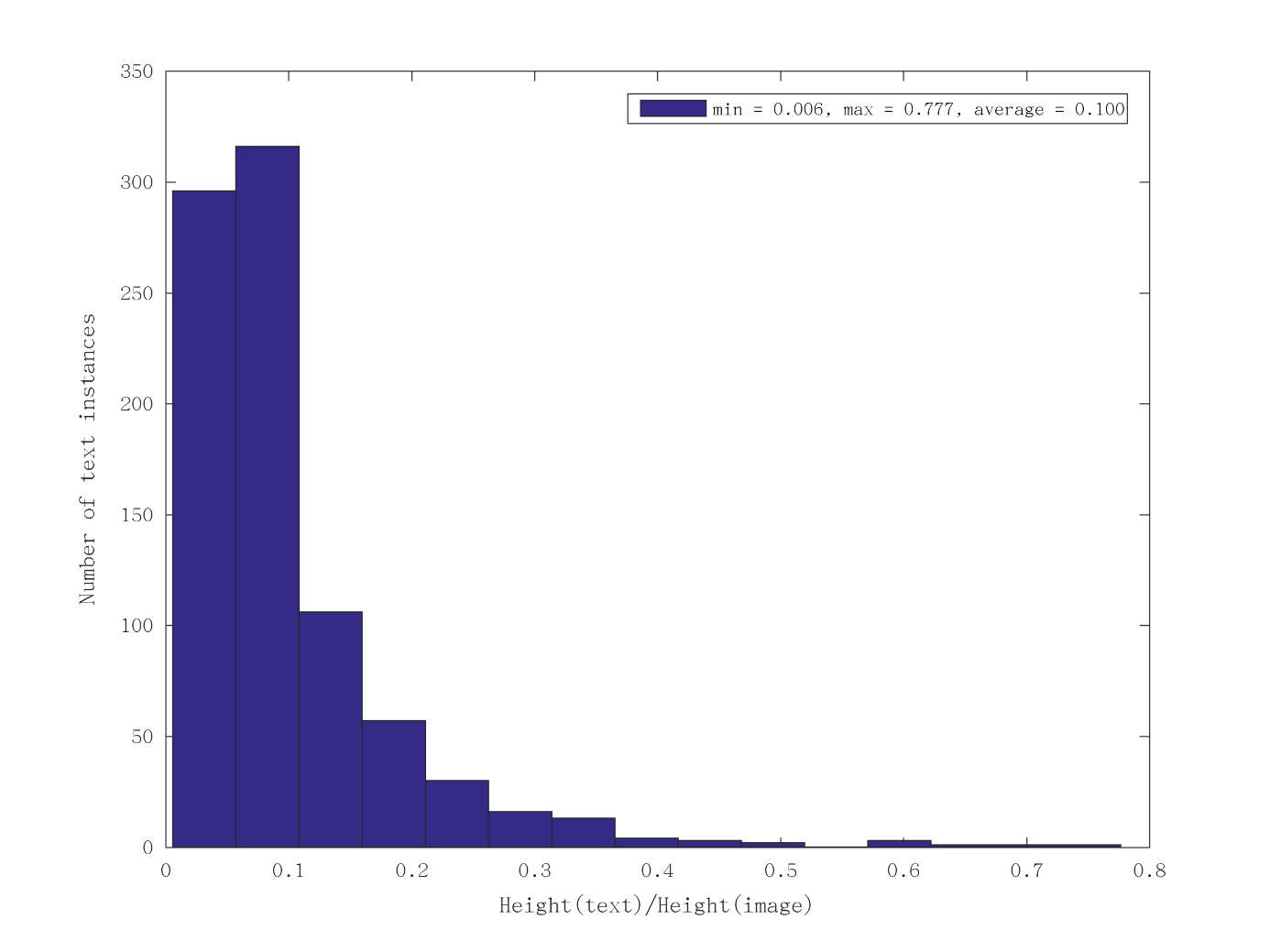


图 3 - 4：ICDAR2013-train库上文字的面积相对于图像面积的比例分布（%）

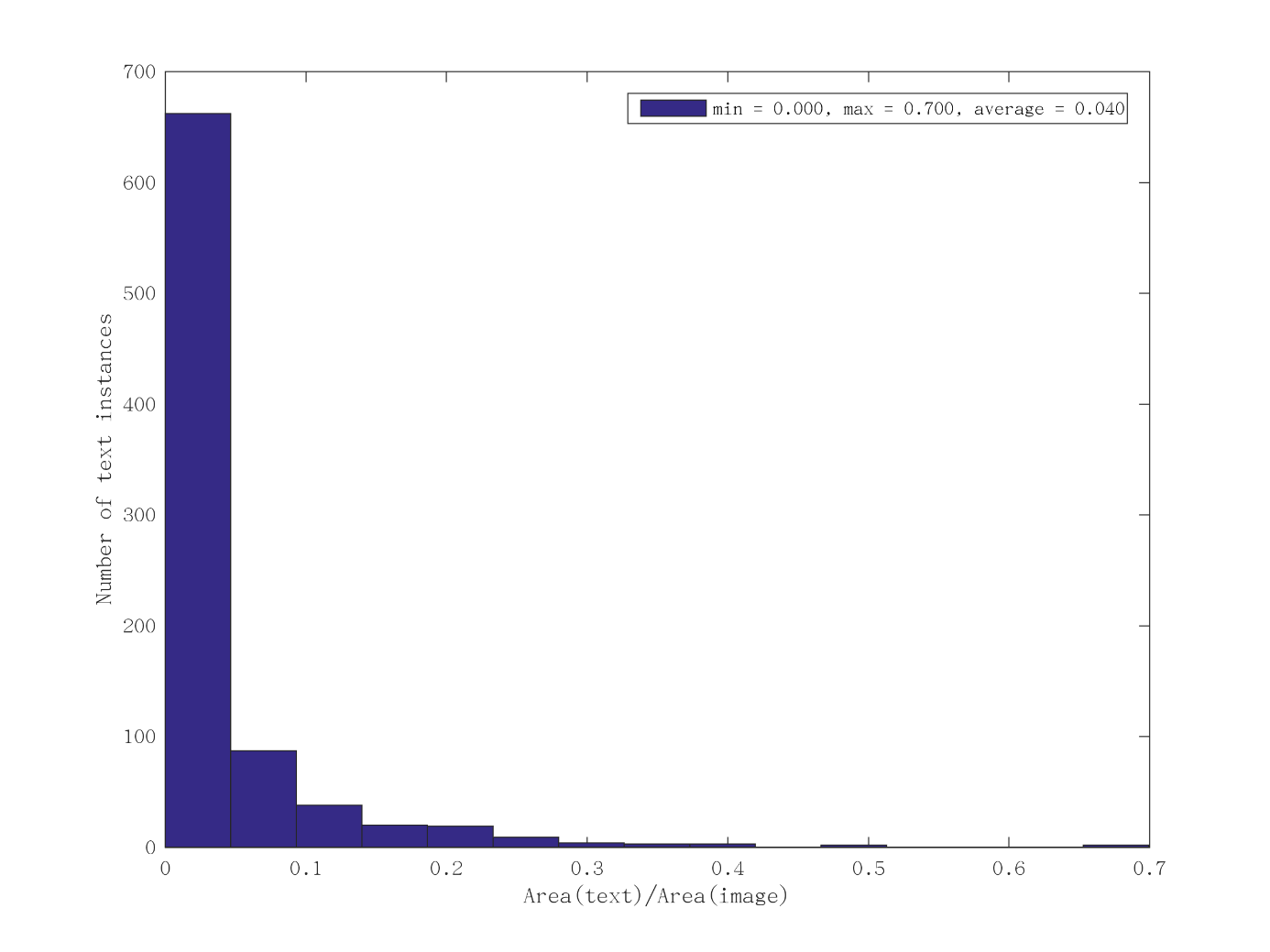
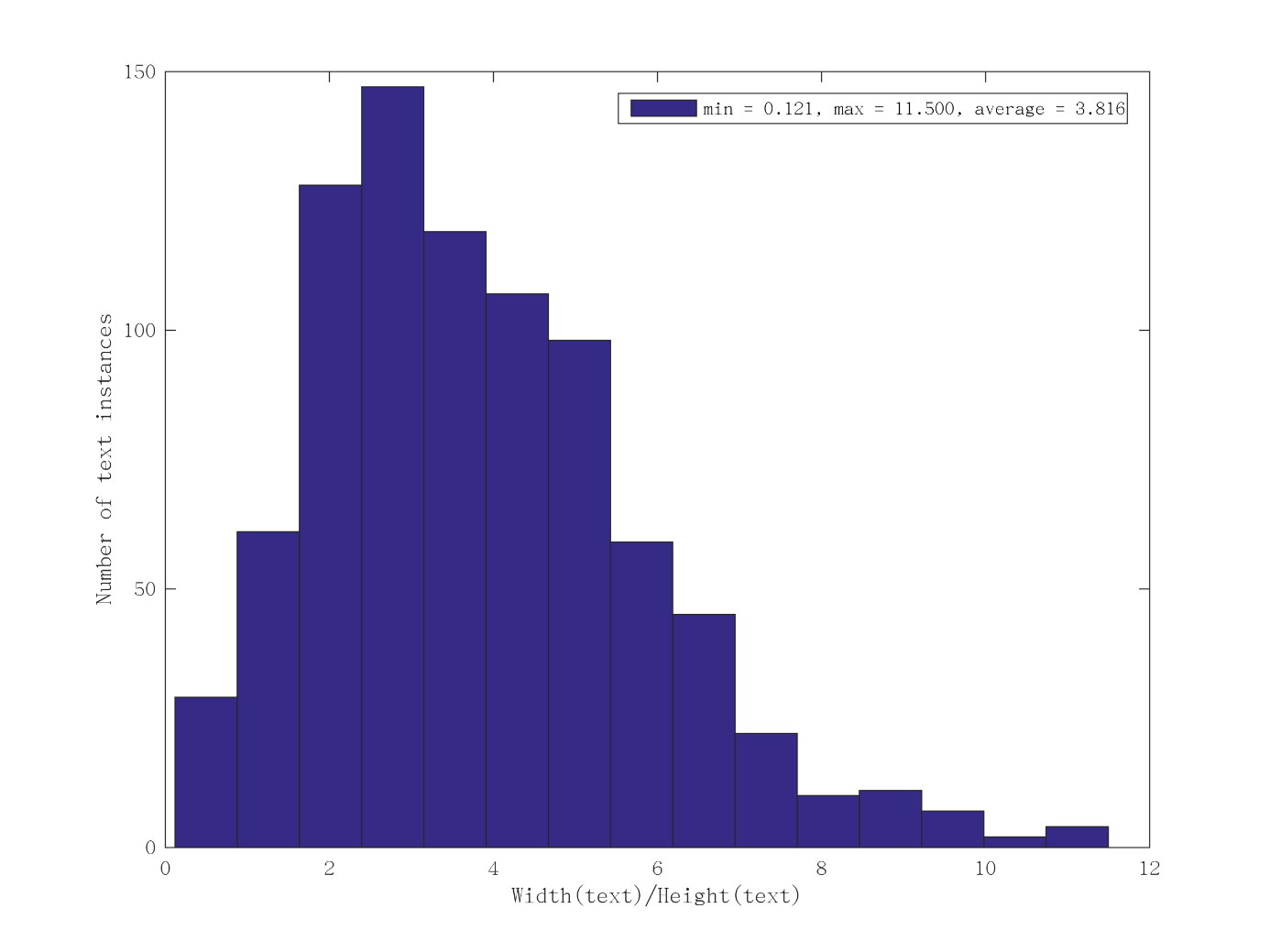


图 3 - 5：ICDAR2013-train库上文字的长宽比（宽度/高度）的比例分布（%）



### Default Boxes的设置

通过训练样本的分析，结合预测结果的统计，我们发现初始的Default Boxes在尺度的设置上，问题主要有两个：

第一，对于浅层（conv4\_3）的特征层，Default Boxes的尺度设置过大，超过了小目标的尺度大小，因此导致了小目标的大批量漏检；

第二，对于深层（conv8\_2， conv9\_2）的特征层，Default Boxes的尺度设置过大，几乎没有目标落在深层特征层对应的尺度范围内，因此深层特征层在提高召回率方面没有任何用途，反而还因引入了噪声而降低了整体的准确率；

第三，对于中间层（fc7，conv6\_2，conv7\_2）的特征层，由于目标的尺度大部分落在中间层的特征层对应的Default Boxes的尺度范围内，因此大部分目标都由这几层检测到，而这些层参数有限。因此，本文采用了两种思路来解决以上问题。

第一种是将每层的Default Boxes的尺度与各层对应的感受野大小相关联，称为感受野配置。假设原图大小为和，和分别为图像的宽度和高度。第层的特征层图像为和，则第层的感受野大小简单计算为， 该层的Default Boxes的尺度设置为，。

本文第二种方法采用了直观的分析，称为经验配置。将初始的Default Boxes的设置进行修改，降低了浅层的Default Boxes的尺度，以及将中间层的Default Boxes尺度分散到深层特征层上，使得落在各层上的目标更加均衡。

表 3 - 4是初始Default Boxes的尺度设置，以及和本文的两种改进版的Default Boxes的尺度设置情况。通过3.6.2实验表明，本文的第二种方法比初始设置和第一种方法更好，具体实验结果和分析详见3.6.2。

表 3 - 4：三种配置的Default Boxes的每层尺度设置对比（%）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **配置名字** | **配置说明** | **conv4\_3** | **fc7** | **conv6\_2** | **conv7\_2** | **conv8\_2** | **conv9\_2** |
| **初始配置** | 第1层: 10~20 第2~6层: 20~90均匀变化 | 10~20 | 20-37 | 37-54 | 54-71 | 71-88 | 88-100 |
| **感受野配置** | 通过感受野大小计算，图像大小为300 | 3-6 | 6-12 | 10-20 | 20-40 | 33-66 | 66-100 |
| **经验配置** | 第1层: 5~10 第2~6层: 10~70均匀变化 | 5~10 | 10-25 | 25-40 | 40-55 | 55-70 | 70-85 |

## 基于投票的多尺度融合

多尺度输入方式在图像分类[[31](#_ENREF_31)]上使用非常广泛。即使是对于已经采用了多种特征融合的SSD来说，输入图像的尺度对文字检测结果影响仍然很大，这主要是由于在训练和测试时输入图像的低分辨率造成（训练一般使用300或512）。为了进一步提高检测结果，我们采用了多尺度输入，即将输入图像缩放成不同尺度，对每一种尺度进行单独的文本检测，最后利用基于投票的多尺度融合技术将不同尺度的检测结果结合起来。多尺度输入方式以牺牲速度来提高性能，需要在速度和性能中间取得平衡，而且，多尺度输入的关键一点在于后处理方式具有良好的选择策略和高效的算法性能。一般的方法是采用简单的非极大值抑制，本文提出了基于投票的多尺度融合来代替普通的非极大值抑制，可以更充分利用多个尺度的综合信息进行决策。

### 非极大值抑制

非极大值抑制（Non-Maximum Suppression，简称NMS）是一种局部极大值搜索算法，它指的是在一个集合中找到所有的局部极大值，即每一个极大值都应该比它的所有领域值都大。在目标检测中，同一个目标常常会输出多个相近的检测结果，如图【插图】中所示，因此，常常需要用NMS来抑制多余的结果，选择最优的结果进行输出。

二维矩形的非极大值抑制算法流程如算法 3- 1所示。

算法 3- 1：二维矩形的非极大值抑制算法

|  |  |
| --- | --- |
| 算法名称 | 二维矩形的非极大值抑制算法 |
| 算法输入 | 1. 矩形区域集合，为输入的矩形区域个数，每个矩形区域有五个维度，表示的中心点的水平和竖直坐标，表示的宽度，表示的高度，表示的置信分数  2. 置信分数阈值  3. 矩形重叠区域阈值 |
| 算法输出 | 矩形区域集合，为输出的矩形区域个数， |
| 算法思路 | 步骤一：过滤置信分数低于阈值的矩形区域，并按照置信分数进行降序排列；  步骤二：计算每两个矩形区域的Jaccard Overlap；  步骤三：对于重叠的矩形，消除置信分数低的矩形 |
| 伪代码 |  |

### 基于投票的多尺度融合

非极大值抑制对于检测结果中重复区域比较少的，且不同检测结果差异不大的情况下效果很好。然而，对于多尺度输入，由于不同尺度的检测结果差异性可能相差很大，如图【插图】此时对于非极大值抑制，只是暴力选择分数较高的方式不够鲁棒，无法处理异常结果，而且没有充分利用到多个尺度的综合信息。为了解决这个问题，本文提出了基于投票的多尺度融合。

基于投票的多尺度融合主要思想是将每种尺度的检测结果进行投票，最后利用投票的综合得分进行非极大值抑制。基于投票的多尺度融合算法如算法 3- 2所示。它和非极大值抑制不同点在于：

第一，非极大值抑制的置信分数阈值针对的是每个尺度下的每个单独的检测区域，而基于投票的多尺度融合则针对的是不同尺度下重合的所有区域的综合值。前者每个检测区域的分数相互独立，非极大值抑制只是从中选择最高分数作为最终检测结果。而后者的检测区域的分数采用累积投票，重合的（不同尺度下的）区域对最终结果都有贡献；

第二，非极大值抑制在进行“重合区域的最优区域选择”前利用置信分数阈值过滤了低分的区域，而基于投票的多尺度融合则是先进行“重合区域的最优区域选择”，再用置信分数过滤低分区域。

算法 3- 2：二维矩形的多尺度融合算法

|  |  |
| --- | --- |
| 算法名称 | 二维矩形的多尺度融合算法 |
| 算法输入 | 1. 矩形区域集合，为输入的矩形区域个数，每个矩形区域有五个维度，表示的中心点的水平和竖直坐标，表示的宽度，表示的高度，表示的置信分数  2. 置信分数阈值  3. 矩形重叠区域阈值 |
| 算法输出 | 矩形区域集合，为输出的矩形区域个数， |
| 算法思路 | 步骤一：按照每个矩形区域的置信分数进行降序排列；  步骤二：计算每两个矩形区域的Jaccard Overlap；  步骤三：对于重叠的矩形，消除置信分数低的矩形，**累计所有重叠矩形的投票得分；**  步骤四；在**更新得分后**的矩形区域集合中，置信分数低于阈值的矩形区域。 |
| 伪代码 |  |

图【插图】与图【插图】是使用非极大值抑制与基于投票的多尺度融合算法的效果对比，可以看出，基于投票的多尺度融合算法对于异常噪声的鲁棒性更强。本章的3.6.6给出了具体的实验对比结果。

## 数据扩充

在文献[[31](#_ENREF_31)]中，训练数据的扩充都使得算法的性能得到了很大提升。众所周知，深度学习是一类以海量数据为输入基础的“数据饥渴”型算法，这是因为深度学习算法往往具有很多参数需要调整，必须使用大量的数据才能得出可概括的模型，大量数据是为算法提供良好训练集的关键。除了数据集的数量，数据集的质量对模型训练而言更为重要。杂乱无章，存在大量噪声干扰的数据用来做训练集，即使数据量再大，对训练也毫无作用，只可能降低模型的精度。

为了扩充训练数据库，除了ICDAR2013的训练集样本，本文还加入了COCO\_Text [[50](#_ENREF_50)]和SCUT\_FORU [[53](#_ENREF_53)]的样本，最终的训练数据库如表 3 - 5所示。

表 3 - 5：本文数据库cocoICDAR13SCUT训练数据集大小及图像来源

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **数据库** | **原始数据集** | | | | | | **挑选为训练库的图像数** |
| **训练集** | | **测试/验证集** | | **总数据集(有标注）** | |
| **图像数** | **单词数** | **图像数** | **单词数** | **图像数** | **单词数** |
| **ICDAR2013** | 229 | 849 | 233 | 462 | 462 | 1311 | 229 |
| **COCO-Text** | 43686 | 118309 | 10000 | 27550 | 53686 | 145859 | 1790 |
| **SCUT\_FORU** | 1200 | 3840 | 515 | 1581 | 1715 | 5421 | 1560 |
| **本文训练库（cocoICDAR13SCUT）** | | | | | | | **3579** |

1. **ICDAR2013数据库**

ICDAR2013是对ICDAR2003和ICDAR2011数据集的标注进行一定的修改和增删。ICDAR是ICDAR Robust Reading Competition中用到的官方数据集，从2003公布到现在，一直是文字检测领域研究工作者常用的算法性能评估数据库。它的训练集有229张，测试集有233张，共462张图像。本章所有的实验都是在该库的测试集上进行的评估。

1. **COCO-Text数据库**

COCO-Text是基于MS-COCO[[49](#_ENREF_49)]的数据库建立起来的专门针对自然场景文字检测与识别的数据库，也是迄今为止该方向最大的一个数据集。它同时也是ICDAR2017的Robust Reading Competition [[54-56](#_ENREF_54)]的官方比赛库之一。它共有63686张图像，其中43686张为训练集，10000张作为验证集，10000张作为测试集，测试集没有公开的标注信息。每个文字区域除了包含bounding box和文字内容，还包含了易读性（易读，难读），类别（机器印刷，手写或者其他），语言（英语，非英语），文字内容编码（utf8 string）四种额外的属性。

图 3 - 6为COCO-Text的中每张图像的文字实例个数统计，可以看出该库有约一半的图像中不含任何文字。图 3 - 7为COCO-Text的中每张图像中的文字实例所占图像大小的面积比例的分布统计，可以看出大部分分实例所占的图像比例都小于0.25%，说明大部分图像的文字都很小。该库与ICDAR2013类型的数据库差异很大，需要对其进行筛选过滤才能作为训练样本进行扩充。

图 3 - 6：COCO-Text与其他数据库的每张图像单词个数比较[[50](#_ENREF_50)]

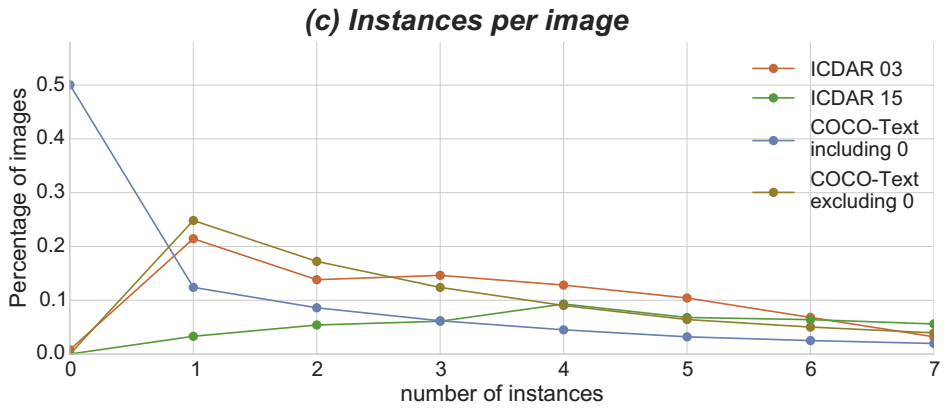
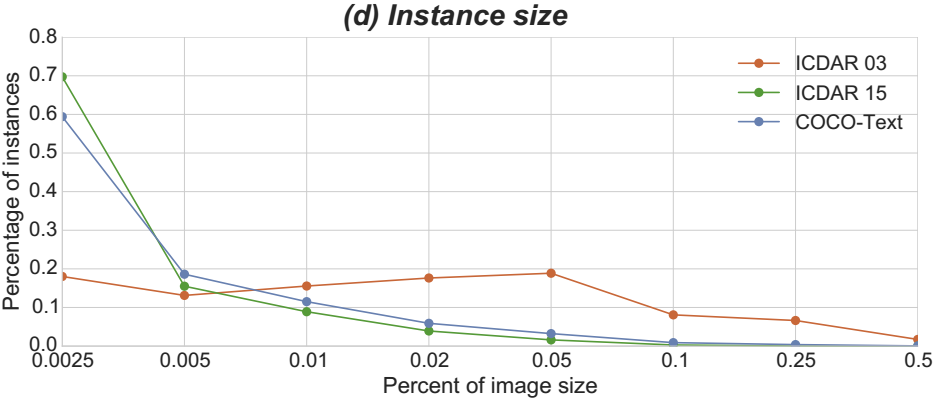


图 3 - 7：COCO-Text与其他数据库的每张图像中的单词所占图像大小的面积比例比较[[50](#_ENREF_50)]



筛选过程总共分为两步，第一步为程序过滤，利用COCO-Text的属性信息使用程序进行过滤，将大部分不符合要求的样本过滤掉；第二步为人工过滤，对遗留的样本进行快速人工筛选。过滤的图像示例如图【插图】所示，筛选过后COCO-text的训练库仅剩1790张图像，将其加入到扩充数据集中。

第一步程序过滤包括：

* 过滤不包含文字的图像；
* 过滤易读性差的图像；
* 过滤非英语图像；

第二步人工过滤原则为：

* 过滤人工合成图像（不是在自然场景图像下拍摄，而是后期通过图像或视频软件添加到图像中，大部分添加文字为广告商标等）；
* 过滤文字区域太小，分辨率太低的图像；
* 过滤文字区域扭曲，形变严重的图像；
* 过滤标注遗漏或标注错误的图像；

1. **SCUT\_FORU数据库**

SCUT\_FORU数据集包含了English2k和Chinese2k两个数据集。如表 3 - 5所示，English2k共有1715张图像，Chinese2k共有2216张图像。该库除了包含单词级的bounding box和单词内容标注，还包括了字符级的bounding box和字符标准，本文只用了其中的English2k数据集的单词级标注。

该库与ICDAR2013相比，图像大小更小，且文字区域在图像中的比例更大，而且更清晰和明显，唯一的问题在于部分图像的标注遗漏或者标注错误，如图【插图】所示。本文采用人工快速过滤的方式将少量标注遗漏和错误的图像过滤，剩余1560张图像，将其加入到扩充数据集中。

## 实验与分析

### 实验环境

1. **硬件配置**

本文的实验都是基于表 3 - 6所示的环境配置，值得一提的是，本文使用的GPU是Geforce GTX 1080，和TextBoxes[[52](#_ENREF_52)]中使用的Titan X在计算能力和显存等多方面都不同，因此在模型的训练，测试时间对比上存在差异。对此，我们将TextBoxes的模型在本文的硬件上进行测试，在同一硬件上对比了本文方法和TextBoxes的检测时间，但训练时间无法进行直接对比。

表 3 - 6：实验环境配置

|  |  |
| --- | --- |
| **操作系统** | Ubuntu 16.04 |
| **内核版本** | 4.4.0-71-generic |
| **CPU** | Intel(R ) Core（TM ) i7-6800K CPU @3.40GHz |
| **GPU** | GeForce GTX 1080 |
| **编程语言** | Python, C++, Matlab |

1. **评价指标**

本文采用了两种评价指标，一种是ICDAR2013 [[56](#_ENREF_56)]，另一种是DetEval [[57](#_ENREF_57)]。二者与ICDAR2003 [[54](#_ENREF_54)]指标都差不多，存在细微差异。这里简单介绍评价标准中的召回率（Recall,简称R），准确率（Precision,简称P），与综合得分（F-measure,简称F）的计算方法。

召回率指的是在GroundTruth中，正确检测的目标个数所占的比例。准确率指的是在检测的结果中，正确检测的目标个数所占的比例。如表 3 - 7所示，检测结果的个数为TP+FP，GroundTruth的个数为TP + FN，则Recall和Precision的计算公式如下：

，

判断一个检测结果是否正确检测的条件在于：是否存在一个GroudTruth使得该GroundTruth与检测结果的Jaccard Overlap值大于0.5，如果是，则表示该检测结果检测正确，且该GroundTruth被正确检测到，否则相反。

表 3 - 7：二类检测评价指标

|  |  |  |
| --- | --- | --- |
| **Detection GroundTruth** | **正** | **负** |
| **正** | True Positive (**TP**) | False Negative (**FN**) |
| **负** | False Positive (**FP**) | True Negative (**TN**) |

综合得分F-measure的定义如下所示，其中参数用来调节召回率Recall和准确率Precision的比重，一般取0.5。只有当Recall和Precision都很高的时候，F-measure取值才会高。因此，F-measure越高，表示检测中大部分是文字，且大部分文字都被检测出。



### Default Boxes尺度大小的影响

**表1. 不同尺度配置的Default Boxes的大小情况**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **groupName** | **实验组说明** | **conv4\_3** | **fc7** | **conv6\_2** | **conv7\_2** | **conv8\_2** | **conv9\_2** |
| **Origin** | **初始配置** | 10~20 | 20-37 | 37-54 | 54-71 | 71-88 | 88-100 |
| **Filed** | **感受野配置** | 3-6 | 6-12 | 10-20 | 20-40 | 33-66 | 66-100 |
| **Prior** | **经验配置** | 5~10 | 10-25 | 25-40 | 40-55 | 55-70 | 70-85 |

表2. 不同尺度配置

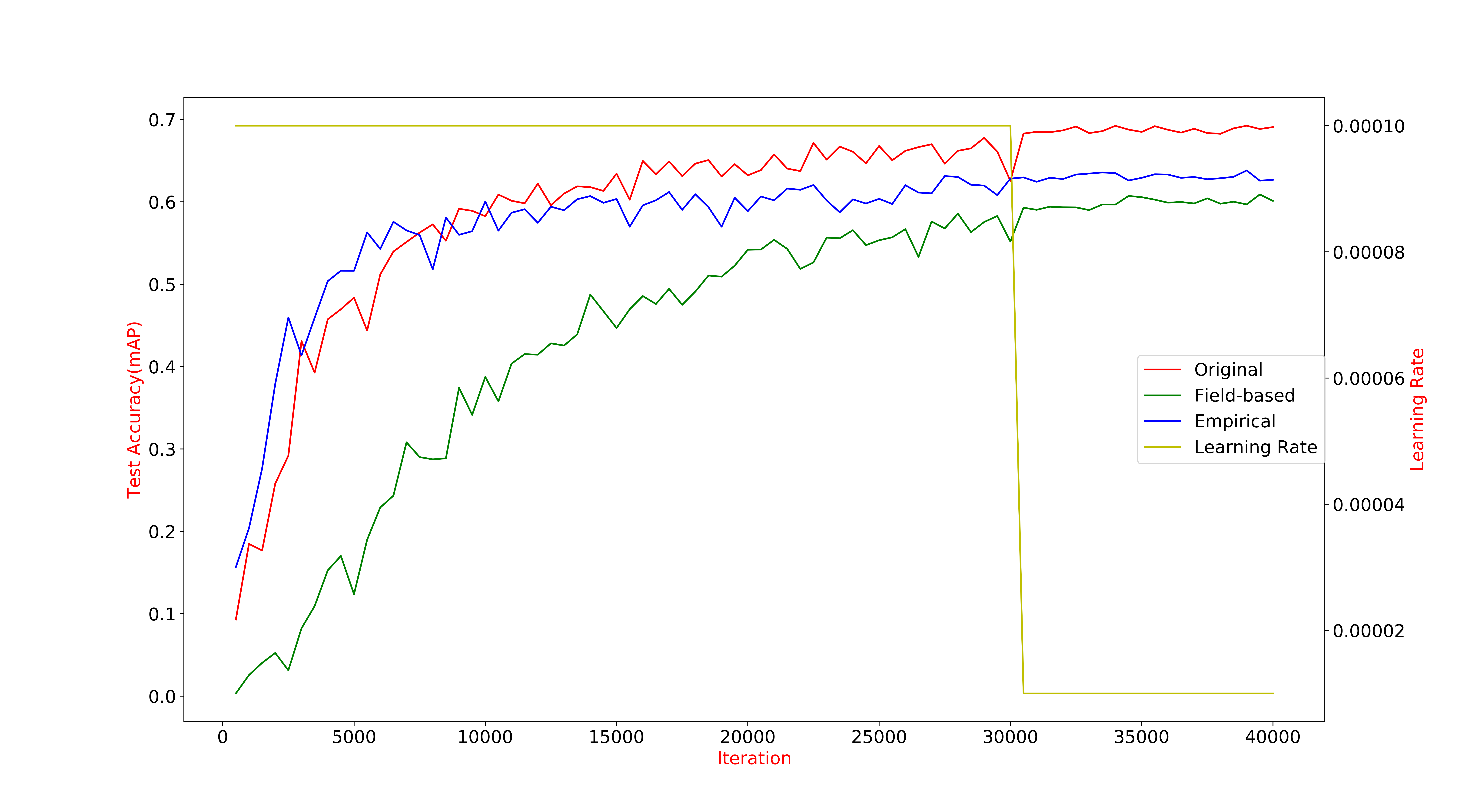


图1不同尺度下模型训练的过程

本实验比较了Default Boxes的尺度大小对实验效果的影响，表【表格1】表示不同尺度配置的Default Boxes的大小情况，图【插图】表示不同尺度下模型训练的过程，表【表格2】表示不同尺度下模型的性能对比。其中，初始配置组为原始SSD的尺度，感受野配置组表示采用感受野为参考计算的尺度，经验配置组表示按照3.3.3中的方式得到的尺度。从图中和表中可以看出两点：

第一， 经验配置组的效果最好，证明了采用分析训练数据的分布方式来确定Default Boxes尺度大小的科学性；

第二，经验配置组和感受野配置组的效果都比感受野配置组更好。这可能是因为几个作为预测的特征层感受野大小变化是指数级变化，前几层太小，而后几层太大，使得不同尺度的Default Boxes在不同层上的数量分布不均，这可以从表2中看出来，大部分的目标都在\*\*几层上检测到，而\*\*几层几乎没有目标。

### Default Boxes长宽比的影响

针对Default Boxes的长宽比，本文共做了两组实验。第一组采用了3种尺度的长宽比（1, k, 1/k），（实际每个位置的Default Boxes个数有4个，因为长宽比为1的多增加了一种尺度），对不同的k值进行了比较；第二组采用了更多尺度的长宽比，并对长宽比的不同组合进行了对比。

表【表格】表示两组实验用到的Defautl Box的尺度，表【表格】表示两组实验用到的Defautl Box的长宽比，图【插图】和图【插图】是分别是两组实验的模型训练的对比图。表【表格】是两组实验的性能对比。

### 数据扩充

### 训练尺度的影响

### 多尺度输入融合的影响

### 学习率的影响

### 文本线检测的影响

* + 1. **性能可视化**
    2. **样本大小的敏感性**

## 本章总结

# 基于SSD的倾斜文本检测

## 汉字组合特征

# 实验结果分析

## 文字检测的三种粒度

## 数据库

## 字符级，单词级，文本线级的不同库上的对比结果（表格，pr曲线，时间）

# 总结与展望

## 总结

自然场景文本检测是计算机视觉领域的研究热点之一。自然场景中的文字蕴含着丰富的信息，在很多领域有着重要的应用。本课题旨在对自然场景中的中文文本进行检测和定位。本文主要利用基于多通道的最大稳定极值区域融合技术提取候选的文本区域。然后对文本区域进行了基于组合特征的文本检测，包括汉字部件合并，词组分析，和非文本滤波三个步骤。最后，采用机器学习的思想，利用改进的方向梯度直方图（HOG）和基于卷积神经网络（CNN）的无监督特征分别训练支持向量机（SVM），对候选区域进一步滤波。本文的主要贡献如下：

1.搭建了适用于中文文本检测识别的自然场景中文文本识别的数据库；

2.提出了汉字组合特征，并基于该特征，设计了相关的汉字部件合并算法、词组分析算法和两层非文本过滤算法；

3. 提出了对候选区域进行分类时应该结合其周围相似区域的性质来判断的思想（上下文信息），并在实验中成功运用该思想提升了实验结果，

4. 提出了面向自然场景中文文本检测的完整算法，并通过实验验证了算法的有效性。

## 展望

本文的研究工作虽然取得了一定的研究成果，但是仍然存在许多需要改进的地方，未来的工作将在以下方面开展：

1. 扩建自然场景中文文本识别数据库CCSV；

2. 结合其他基于连通分量的方法或者基于区域的方法。本文采用的是连通分量分析方法，这种方法在候选区域生成时一般都存在漏检的问题，因此可以将几种连通分量方法结合起来（比如与ER结合等）；

3.改进基于组合特征的相关算法。基于组合特征设计的算法参数均由人工调整，没有很强的自适应性。可以结合无监督学习的思想，将这些算法改进成可以自动学习的模型；

4. 卷积神经网络结构的改进。本文考虑卷积神经网络的复杂性，只利用到了卷积神经网络的特征提取方法。因此未来工作可以对卷积神经网络的训练结构进行简化后使用；

5. 设计端到端的中文文本检测识别系统。将中文文本的检测与识别结合起来，设计完整的端到端中文文本检测识别系统；

6. 将中文的文本检测算法推广到英文上，使算法具备更广泛的使用价值。

# 科研成果

**参与的科研活动：**

1.2013年国家级大学生创新创业训练计划项目: 基于场景文本检测识别的地图智能定位研究（项目负责人)。

**发表论文:**

1. **Li Lin**, Yanyun Qu, Weimin Liao. "Structure Context Clues for Chinese Text Detection". in International Conference on Internet Multimedia Computing and Service(ICIMCS). 2014.(EI检索,已收录)

2. YAN-YUN QU, MENG-JIE LIAO, YAN-WEN ZHOU,TIAN-ZHU FANG, **LI LIN**, and HAI- YING ZHANG. "IMAGE SUPER-RESOLUTION BASED ON DATA-DRIVEN GAUSSIAN PROCESS REGRESSION". in International Conference on Intelligence Science and Big Data Engineering(IScIDE). 2013. pp.513-520 (EI检索).

3. Weiwei Wu, Yanyun Qu, Xiaoqing Yang, and **Li Lin**. "Traffic Sign Detection based on Camera Imaging Apriority". in Pacific-Rim Conference on Multimedia(PCM). 2013. pp.720-729 (EI检索).

# 致谢语

在本篇论文完成的同时，我首先要感谢我的导师曲延云老师。在曲老师的悉心指导下，我的论文才能如此顺利的完成。曲老师的负责、认真、严谨的作风对我的影响很深，以至于在我整个论文的撰写过程中都是带着这种精益求精的态度去做。在此我由衷感谢曲老师给我的帮助和指导，我相信这些习惯、思想会陪伴我一辈子。

另外，我想感谢研究生学长廖威敏同学。他从我接触这个领域开始就一直帮助我，带着我一步步往前走。对于我的问题，学长总会悉心解答，他不但在具体思路和编程中给予我指导，而且在生活中也像一个大哥哥一样帮助我。在这里，我衷心感谢学长。

我还想感谢实验室的所有同学，李建敏、廖梦杰、宋书阳、周燕稳、吴伟伟、卢畅、杨思颖。每当我想起大家都在实验室里一起奋斗的时候就很感动。其中，尤其感谢李建敏学长和吴伟伟学长，他们在我的论文和大学生创新性实验项目中提供了很多帮助。感谢你们！

我要感谢我们的创新性实验小队的所有成员，柳冰莹、刘勇顺、罗道文、王冰、车玉鑫。没有大家，就没有这个项目，感谢大家一直坚持我们的团队。

感谢我的舍友，感谢你们一路陪伴我，支持我！

最后，感谢我的家人，你们是我坚持的动力，谢谢你们给予的支持！

# 参考文献

[1] RUSSAKOVSKY O, DENG J, SU H, et al. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.

[2] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]// Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, 2005 CVPR 2005 IEEE Computer Society Conference on. IEEE,1: 886-893.

[3] VIOLA P, JONES M. Robust real-time object detection[J]. International Journal of Computer Vision, 2001, 4(34–47).

[4] GREENHALGH J, MIRMEHDI M. Real-time detection and recognition of road traffic signs[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(4): 1498-1506.

[5] JADERBERG M, VEDALDI A, ZISSERMAN A. Deep features for text spotting[C]// Deep features for text spotting. European conference on computer vision. Springer: 512-528.

[6] YAO C, BAI X, LIU W, et al. Detecting texts of arbitrary orientations in natural images[C]// Detecting texts of arbitrary orientations in natural images. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE: 1083-1090.

[7] NEUMANN L, MATAS J. Real-time scene text localization and recognition[C]// Real-time scene text localization and recognition. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE: 3538-3545.

[8] GUPTA A, VEDALDI A, ZISSERMAN A. Synthetic data for text localisation in natural images[C]// Synthetic data for text localisation in natural images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2315-2324.

[9] NEUMANN L, MATAS J. Text localization in real-world images using efficiently pruned exhaustive search[C]// Text localization in real-world images using efficiently pruned exhaustive search. Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE: 687-691.

[10] HE T, HUANG W, QIAO Y, et al. Text-attentional convolutional neural network for scene text detection[J]. IEEE Transactions on Image Processing, 2016, 25(6): 2529-2541.

[11] HOSANG J, BENENSON R, DOLL R P, et al. What makes for effective detection proposals?[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 38(4): 814-830.

[12] CHENG M-M, ZHANG Z, LIN W-Y, et al. BING: Binarized normed gradients for objectness estimation at 300fps[C]// BING: Binarized normed gradients for objectness estimation at 300fps. Proceedings of the IEEE conference on computer vision and pattern recognition. 3286-3293.

[13] CARREIRA J, SMINCHISESCU C. Cpmc: Automatic object segmentation using constrained parametric min-cuts[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(7): 1312-1328.

[14] ZITNICK C L, DOLL R P. Edge boxes: Locating object proposals from edges[C]// Edge boxes: Locating object proposals from edges. European Conference on Computer Vision. Springer: 391-405.

[15] ENDRES I, HOIEM D. Category-independent object proposals with diverse ranking[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 36(2): 222-234.

[16] KR HENB HL P, KOLTUN V. Geodesic object proposals[C]// Geodesic object proposals. European Conference on Computer Vision. Springer: 725-739.

[17] ARBEL EZ P, PONT-TUSET J, BARRON J T, et al. Multiscale combinatorial grouping[C]// Multiscale combinatorial grouping. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 328-335.

[18] ALEXE B, DESELAERS T, FERRARI V. Measuring the objectness of image windows[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 34(11): 2189-2202.

[19] RAHTU E, KANNALA J, BLASCHKO M. Learning a category independent object detection cascade[C]// Learning a category independent object detection cascade. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE: 1052-1059.

[20] MANEN S, GUILLAUMIN M, VAN GOOL L. Prime object proposals with randomized prim's algorithm[C]// Prime object proposals with randomized prim's algorithm. Proceedings of the IEEE International Conference on Computer Vision. 2536-2543.

[21] RANTALANKILA P, KANNALA J, RAHTU E. Generating object segmentation proposals using global and local search[C]// Generating object segmentation proposals using global and local search. Proceedings of the IEEE conference on computer vision and pattern recognition. 2417-2424.

[22] HUMAYUN A, LI F, REHG J M. RIGOR: Reusing inference in graph cuts for generating object regions[C]// RIGOR: Reusing inference in graph cuts for generating object regions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 336-343.

[23] UIJLINGS J R, VAN DE SANDE K E, GEVERS T, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154-171.

[24] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[25] CORTES C, VAPNIK V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297.

[26] HO T K. The random subspace method for constructing decision forests[J]. IEEE transactions on pattern analysis and machine intelligence, 1998, 20(8): 832-844.

[27] FREUND Y, SCHAPIRE R E. A desicion-theoretic generalization of on-line learning and an application to boosting[C]// A desicion-theoretic generalization of on-line learning and an application to boosting. European conference on computational learning theory. Springer: 23-37.

[28] ALTMAN N S. An introduction to kernel and nearest-neighbor nonparametric regression[J]. The American Statistician, 1992, 46(3): 175-185.

[29] RUSSELL S, NORVIG P, INTELLIGENCE A. A modern approach[J]. Artificial Intelligence Prentice-Hall, Egnlewood Cliffs, 1995, 25: 27.

[30] BHADESHIA H H. Neural networks in materials science[J]. ISIJ international, 1999, 39(10): 966-979.

[31] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]// Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 1097-1105.

[32] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 580-587.

[33] EVERINGHAM M, VAN GOOL L, WILLIAMS C K, et al. The pascal visual object classes (voc) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.

[34] HE K, ZHANG X, REN S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[C]// Spatial pyramid pooling in deep convolutional networks for visual recognition. European Conference on Computer Vision. Springer: 346-361.

[35] GIRSHICK R. Fast r-cnn[C]// Fast r-cnn. Proceedings of the IEEE International Conference on Computer Vision. 1440-1448.

[36] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[C]// Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 91-99.

[37] XUE J, LI J, GONG Y. Restructuring of deep neural network acoustic models with singular value decomposition[C]// Restructuring of deep neural network acoustic models with singular value decomposition. Interspeech. 2365-2369.

[38] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks[C]// Visualizing and understanding convolutional networks. European conference on computer vision. Springer: 818-833.

[39] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:14091556, 2014.

[40] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]// You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 779-788.

[41] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]// SSD: Single shot multibox detector. European Conference on Computer Vision. Springer: 21-37.

[42] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1-9.

[43] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]// Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3431-3440.

[44] HARIHARAN B, ARBEL EZ P, GIRSHICK R, et al. Hypercolumns for object segmentation and fine-grained localization[C]// Hypercolumns for object segmentation and fine-grained localization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 447-456.

[45] LIU W, RABINOVICH A, BERG A C. Parsenet: Looking wider to see better[J]. arXiv preprint arXiv:150604579, 2015.

[46] SERMANET P, EIGEN D, ZHANG X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:13126229, 2013.

[47] ERHAN D, SZEGEDY C, TOSHEV A, et al. Scalable object detection using deep neural networks[C]// Scalable object detection using deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2147-2154.

[48] HOWARD A G. Some improvements on deep convolutional neural network based image classification[J]. arXiv preprint arXiv:13125402, 2013.

[49] LIN T-Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C]// Microsoft coco: Common objects in context. European Conference on Computer Vision. Springer: 740-755.

[50] VEIT A, MATERA T, NEUMANN L, et al. Coco-text: Dataset and benchmark for text detection and recognition in natural images[J]. arXiv preprint arXiv:160107140, 2016.

[51] ZHONG Z, JIN L, ZHANG S, et al. Deeptext: A unified framework for text proposal generation and text detection in natural images[J]. arXiv preprint arXiv:160507314, 2016.

[52] LIAO M, SHI B, BAI X, et al. TextBoxes: A Fast Text Detector with a Single Deep Neural Network[J]. arXiv preprint arXiv:161106779, 2016.

[53] 张树业. 深度模型及其在视觉文字分析中的应用[D]. City: 华南理工大学, 2016.

[54] LUCAS S M, PANARETOS A, SOSA L, et al. ICDAR 2003 Robust Reading Competitions[C]// ICDAR 2003 Robust Reading Competitions. ICDAR. Citeseer,2003: 682.

[55] SHAHAB A, SHAFAIT F, DENGEL A. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images[C]// ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE: 1491-1496.

[56] KARATZAS D, SHAFAIT F, UCHIDA S, et al. ICDAR 2013 robust reading competition[C]// ICDAR 2013 robust reading competition. Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. IEEE: 1484-1493.

[57] WOLF C, JOLION J-M. Object count/area graphs for the evaluation of object detection and segmentation algorithms[J]. International Journal of Document Analysis and Recognition (IJDAR), 2006, 8(4): 280-296.