天津大学

本科生毕业论文

学　　院	电气自动化与信息工程学院
专　　业	自动化
年　　级	**2015** 级
姓　　名	王昊
指导教师	曾明

2019 年 **6** 月 **12** 日

天津大学

毕业设计（论文）任务书

题目：基于深度学习的垃圾文字检测研究

学生姓名 王昊

学院名称 电气自动化与信息工程学院

专　　业 自动化

学　　号 **3015203171**

指导教师 曾　明

职　　称 副教授

一、原始依据（包括设计或论文的工作基础、研究条件、应用环境、工作目的等。）

文字，区别于图像和视频，有着更强的逻辑性和表达能力。随着计算机视觉研究的深入，分辨并利用图像中的文字信息越来越重要。从海量视频图像中检索文字可以极大提高人们的认知效率。所以，近年来自然场景下的文字检测成为了计算机视觉领域的热门话题之一。

而在垃圾智能分类领域，对垃圾图片上的文字进行提取并检测可以大大提高垃圾的分类效率。使用传统的特征检测分类时，由于玻璃碎片，塑料等材质往往太过相似，很容易被分错。而垃圾上的文字往往是最重要的信息，直接与垃圾的种类相关，可以避免因垃圾相似而错分类的情况。

场景文字识别研究与传统的文本文字检测的重要区别是需要将照片或视频中的文字识别出来，首先对照片中存在文字的区域进行定位，即找到单词或文本行的边界框。然后对定位后的文字进行识别，这样就能得到文字的端到端检测，是我们场景文字检测的最终目标。

近年来场景文字检测的发展主要分为两个阶段：一是基于传统的手工设计特征，二是基于深度学习的方法。学生需要在了解传统的目标检测方法后，运用深度学习的框架进行编程，在充分学习了解国内外相关工作成果后，自行选择基于目标检测和语义分割的方法，得到场景文字检测算法并使用实验室采集的标注垃圾图片数据集进行检测。

二、参考文献

[1] Yao, C. "Detecting texts of arbitrary orientations in natural images."*IEEE Conference on Computer Vision and Pattern Recognition* IEEE Computer Society, 2012:1083-1090.

[2] Kang, Le; Li, Yi and D. Doermann. "Orientation Robust Text Line Detection in Natural Images." *IEEE Conference on Computer Vision and Pattern Recognition* IEEE, 2014:4034-4041.

[3]Yin, XC; Pei, WY; Zhang, J et al. "Multi-Orientation Scene Text Detection with Adaptive Clustering." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 37.9(2015):1930-1937

[4] Jaderberg, M; Simonyan,K; Vedaldi, A et al. "Reading Text in the Wild with Convolutional Neural Networks." *International Journal of Computer Vision*116.1(2016):1-20.

[5] Zhang, Z; Shen, W; Yao, C and Bai, X. "Symmetry-based text line detection in natural scenes." *Computer Vision and Pattern Recognition* IEEE, 2015:2558-2567.

[6] Gupta, A; Vedaldi, A and Zisserman，A "Synthetic data for text localisation in natural

images." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.:2315-2324
[7] Tian, Z; Huang, W; He, T et al. ”*Detecting Text in Natural Image with Connectionist Text Proposal Network*.” *Computer Vision – ECCV 2016*. Springer International Publishing, 2016.9912:56-72.
[8] Ma, J; Shao, W; Ye, H et al. "Arbitrary-Oriented Scene Text Detection via Rotation Proposals." IEEE Transactions on multimedia(2017):3111-3122.
[9] Liu, Y, and L Jin. "Deep matching prior network: Toward tighter multi-oriented text detection." 30th IEEE Conference on computer vision and pattern recognition (CVPR 2017)*:3454-3461*.
[10] Zhang, Z; Zhang, C; Shen, W et al. "Multi-oriented text detection with fully convolutional networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016:4159-4167,
[11] Zhou, X; Yao, C; Wen , H et al. "EAST: An Efficient and Accurate Scene Text Detector."30th IEEE Conference on computer vision and pattern recognition (CVPR 2017):2642-2651.
[12]He, W; Zhang, XY; Yin, F et al. "Deep Direct Regression for Multi-Oriented Scene Text Detection."2017 IEEE International conference on computer vision (ICCV):745-753
[13] B Shi, X Bai, and S. Belongie. "Detecting Oriented Text in Natural Images by Linking Segments."30th IEEE Conference on computer vision and pattern recognition (CVPR 2017):3482-3490

三、设计（研究）内容和要求（包括设计或研究内容、主要指标与技术参数，并根据课题性质对学生提出具体要求。）

在了解前人工作的基础上，利用深度学习的方法，学习场景文字检测神经网络，使用垃圾图片数据集进行训练并最终评估辨识效果，调节网络使其适合垃圾文字检测。

毕设工作主要包括以下内容：

- 掌握 Python 语言与 TensorFlow，PyTorch 等深度学习框架。
- 了解目前国内外进行场景文字检测的方法，包括传统方法和应用深度学习的方法，重点掌握如 Faster R-CNN，SSD 和 YOLO 等经典目标检测算法，了解语义分割算法的原理。
- 学习场景文字检测中的重要问题：不规则，多角度，多长宽比文本框的定位，学习 Textboxes++，CTPN 等文字检测算法，同时思考如何解决文本框定位难题，参考 Pixel-anchor 等最新算法。
- 学习 CNN 神经卷积网络并用其实现已有的场景文字检测算法。
- 使用适合的场景文字检测算法，用垃圾图片数据集进行网络训练与测试，微调超参数和网络结构，使得算法在垃圾图片数据集上可以的得到较高的精确

率和召回率。

指导教师（签字）

年　月　日

审题小组组长（签字）

年　月　日

天津大学本科生毕业论文开题报告

课题名称	基于深度学习的垃圾文字检测研究		
学院名称	电气自动化与信息工程学院	专业名称	自动化
学生姓名	王昊	指导教师	曾明

一、课题的来源及意义

目标检测一直是计算机视觉领域十分重要的一个分支，对于计算机视觉系统来讲，可以快速准确地识别出图中目标并分类识别，是十分必要的。其中文字作为人类文明的载体，包含着十分丰富的语义信息，在目标检测中是不可忽略的信息来源。

在垃圾分类问题中，对于垃圾上文本的识别可以直接指明垃圾的种类，对于垃圾分类的准确度有着很大的提升。在使用文字检测算法对垃圾图片进行检测时，要考虑垃圾上的形变和残缺问题，选择适合的网络结构，并且调节超参数使算法可以在垃圾图片数据集上取得较好的结果。本课题以深度学习为基础，学习使用文字检测算法，并将其运用于垃圾图片数据集上，训练后进行验证，得到较高的F分数。

二、国内外发展状况

2014年加州大学伯克利分校的Ross B. Girshick提出R-CNN算法，这是深度学习首次运用在目标检测上，但算法仍有缺陷，比如计算量大，冗余计算等。

2015年微软研究院的何恺明等提出一种SPP-Net算法，针对卷积神经网络重复运算问题，在卷积层和FC层之间加入了池化层，来防止R-CNN对ROI进行剪裁变形，SPP-Net只用一次CNN，极大的提高的运算速度，但由于特征尺寸依然不同，识别效率依然不高。

2015年微软研究院的Ross B. Girshick又提出一种改进的Fast R-CNN算法，加入ROI pooling layer将所有区域变为统一大小的向量，将分类和位置回归放在深度神经网络同时实现，节省了存储空间，同时提高了准确率。

同年微软研究院的任少庆、何恺明以及Ross B Girshick等人又提出了Faster R-CNN算法，采用RPN替代选择性搜索，特点是先通过CNN，在feature map上经过RPN网络，之后经过ROI pooling进行分类，相比之前的算法，在提取ROI的部分也实现了深度学习，并且生成建议窗口和分类网络的CNN实现了特征共享，这是第一个真正意义上的深度学习目标检测算法。

2016年,北卡大学教堂山分校的Wei Liu等提出了SSD(single shot multibox detector）算法，结合了YOLO 和Faster R-CNN各自的优点，是现在目标检测的通用框架。

近年来，白翔教授团队 Textbox 算法是基于候选区域的方法中较为出众的方法，Textbox基于SSD框架，为了更好的检测文本，选取了不同纵横比的候选框，并改进了卷积核的大小来适应文本的感受野。由于场景中的文本框不可能总是水平的，又加入了分支进行多边形回归，优化为Textbox++。

2018年，云从科技提出了结合语义分割和目标检测的Pixel-anchor算法，通过两个模块的特征共享来实现较高的F分数。

同年,白翔教授团队发表在CVPR2018上的论文也结合了目标检测和语义分割，提出MaskTextSpotter算法，也获得了较高的F分数，可见结合语义分割和目标检测来进行文字识别是新的趋势。

三、研究目标与研究内容

研究目标：

以CNN网络为基本框架，学习并使用文字检测算法，使用垃圾图片数据集进行训练，调节超参数和网络结构使得网络可以准确的识别垃圾图片上的文字。

研究内容：

1. 实现基于语义分割和目标检测的场景文字识别算法，在中文标注数据集上进行训练，并理解网络工作原理。
2. 分析垃圾场景文本的特征，结合两种方法，在垃圾图片数据集上进行训练，并得到较好的性能，即较高的F值。

四、研究方法与手段

1. 采用Python语言作为算法实现语言，选择如PyTorch，TensorFlow，Caffe作为软件，自主选择神经网络框架，结合语义分割和目标检测算法，运用场景文字检测网络，使用垃圾图片数据集进行训练。
2. 阅读深度学习，目标检测及语义分割相关文献，理解并实现其中的算法，比较不同算法间的不同，分析各种算法优势的来源，领会各类算法中的核心思想；
3. 根据语义分割和目标检测的基本框架，学习先进的场景文字识别算法，结合垃圾图片文本的固有特征，构建并训练深度神经网络来识别场景文本。

五、进度安排

2019年1月26日-2月20日：阅读相关文献，了解场景文字识别的研究进展，了解所需学习的知识；

2019 年 2 月 20 日-3 月 5 日：学习深度学习，目标检测，语义分割等基础内容，同时阅读论文，了解最新场景文字识别算法的实现手段。
2019 年 3 月 5 日-4 月 15 日：搭建深度学习框架，使用文字检测算法在公开标注中文数据集上进行训练并评估效果，理解算法的运作原理。
2019 年 4 月 15 日-5 月 15 日：使用文字检测算法在垃圾图片标注数据集上进行训练，微调超参数使得网络可以在垃圾标注数据集上取得较高的 F 分数。
2019 年 5 月 15 日-6 月：论文撰写与修改，进行论文答辩。

六、参考文献

[1] Yao, C. "Detecting texts of arbitrary orientations in natural images." *IEEE Conference on Computer Vision and Pattern Recognition* IEEE Computer Society, 2012:1083-1090.
[2] Kang, Le; Li, Yi and D. Doermann. "Orientation Robust Text Line Detection in Natural Images." *IEEE Conference on Computer Vision and Pattern Recognition* IEEE, 2014:4034-4041.
[3]Yin, XC; Pei, WY; Zhang, J et al. "Multi-Orientation Scene Text Detection with Adaptive Clustering." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 37.9(2015):1930-1937
[4] Jaderberg, M; Simonyan,K; Vedaldi, A et al. "Reading Text in the Wild with Convolutional Neural Networks." *International Journal of Computer Vision*116.1(2016):1-20.
[5] Zhang, Z; Shen, W; Yao, C and Bai, X. "Symmetry-based text line detection in natural scenes." *Computer Vision and Pattern Recognition* IEEE, 2015:2558-2567.
[6] Gupta, A; Vedaldi, A and Zisserman, A "Synthetic data for text localisation in natural images." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.:2315-2324
[7] Tian, Z; Huang, W; He, T et al. "*Detecting Text in Natural Image with Connectionist Text Proposal Network.*" *Computer Vision – ECCV 2016*. Springer International Publishing, 2016.9912:56-72.
[8] Ma, J; Shao, W; Ye, H et al. "Arbitrary-Oriented Scene Text Detection via Rotation Proposals." IEEE Transactions on multimedia(2017):3111-3122.
[9] Liu, Y, and L Jin. "Deep matching prior network: Toward tighter multi-oriented text detection." 30th IEEE Conference on computer vision and pattern recognition (CVPR 2017)*:3454-3461*.
[10] Zhang, Z; Zhang, C; Shen, W et al. "Multi-oriented text detection with fully convolutional networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016:4159-4167,
[11] Zhou, X; Yao, C; Wen , H et al. "EAST: An Efficient and Accurate Scene Text Detector."30th IEEE Conference on computer vision and pattern recognition (CVPR 2017):2642-2651.

[12]He, W; Zhang, XY; Yin, F et al. "Deep Direct Regression for Multi-Oriented Scene Text Detection."2017 IEEE International conference on computer vision (ICCV):745-753
[13]B Shi, X Bai, and S. Belongie. "Detecting Oriented Text in Natural Images by Linking Segments."30th IEEE Conference on computer vision and pattern recognition (CVPR 2017):3482-3490
[14]B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010, 2010, pp. 2963–2970.
[15]L. Neumann and J. Matas, "A method for text localization and recognition in realworld images," in Computer Vision - ACCV 2010 - 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III, 2010, pp. 770–783.
[16]Y.Zhong,K.Karu,andA.K.Jain,"Locatingtextincomplexcolorimages,"Pattern Recognition, vol. 28, no. 10, pp. 1523–1535, 1995.
[17]K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12, pp. 1631–1639, 2003.
[18]J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," Neural Processing Letters, vol. 9, no. 3, pp. 293–300, 1999.
[19]K.WangandS.J.Belongie,"Word spotting in the wild," in Computer Vision-ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I, 2010, pp. 591–604.
[20]P. Shivakumara, T. Q. Phan, S. Lu, and C. L. Tan, "Gradient vector flow and grouping-based method for arbitrarily oriented scene text detection in video images," IEEE Trans. Circuits Syst. Video Tech, vol. 23, no. 10, pp. 1729–1739, 2013.
[21]C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, 2014, pp. 391–405.

选题是否合适： 是□ 否□

课题能否实现： 能□ 不能□

指导教师（签字）

年 月 日

选题是否合适： 是□ 否□

课题能否实现： 能□ 不能□

审题小组组长（签字）

年 月 日

摘　　要

垃圾处理问题是现代社会面临的重大环境问题之一，垃圾数量的增加与垃圾处理难度的上升使得处理垃圾成为了一大难题。而由于我国垃圾分类观念宣传的时间并不长，垃圾分类的观念不深入人心。在垃圾处理过程中，人工对垃圾进行分类消耗了大量的人力财力，导致了整个垃圾处理环节效率的低下，因此，对垃圾准确的进行自动分类，对环境保护具有重要意义。

由于不同种类的垃圾可能具有很相似的视觉特征，使用深度学习的分类算法很难将其区分开，故采用分类算法对垃圾进行分类的系统的准确率很难提高。而垃圾图片上普遍存在的一个强语义特征——文字，有希望成为一个突破口。当分类网络很难判定垃圾类别时，结合对垃圾图片上的文字检测与识别，可以准确的对垃圾进行分类。

本文针对垃圾上的文字检测问题，建立垃圾图片文字标注数据集，采用 PixelLink 场景文字检测算法，调整网络参数，调整网络结构，训练出了可以较为准确地检测垃圾上文字位置的深度学习模型。通过该模型，可以快速的检测出测试垃圾图片上的文字位置，并以包围框进行标注。同时本文采用的算法对多角度，多形变的文字也有不错的检测效果。

关键词：深度学习；神经网络；场景文字检测；垃圾标注数据集

ABSTRACT

The problem of garbage disposal is one of the major environmental problems which our modern society is facing. The increasing of quantity of garbage and the difficulty of garbage disposal make it a big issue to deal with garbage. However, due to the fact that the promotion of garbage classification concepts in China is not long, the concept of garbage classification is not deeply rooted in the hearts of the people. In the process of garbage disposal, manual classification of garbage consumes a lot of human and financial resources, resulting in low efficiency of the entire garbage disposal process. Therefore, accurate automatic classification of garbage is significant to environmental protection.

Since different types of garbage may have very similar visual features, it is difficult to distinguish them using a deep learning classification algorithm, so the accuracy of the system that classifies garbage using a classification algorithm is difficult to improve. And a strong semantic feature that is common on garbage pictures – text, hopefully becomes a breakthrough. When the classification network is difficult to determine the garbage category, combined with the text detection and recognition on the garbage pictures, the garbage can be classified accurately. In this paper, the paper detects the text detection problem on the garbage, establishes the garbage image text annotation data set, adopts the PixelLink scene text detection algorithm, adjusts the network parameters and network structure, then trains a deep learning model that can accurately detect the position of the text on the garbage. With this model, you can quickly detect the position of the text on the test garbage image and mark it with the bounding box. At the same time, the algorithm used in this paper has a good detection effect on multi-angle and multi-deformed texts.

Keywords: deep learning, neural network, scene text detection, garbage annotation dataset

目　　录

第一章　绪论

1.1　垃圾自动分类的重要性

伴随着科技的进步，人类产生的垃圾类别越来越多，处理的难度也越来越大。为了最大限度的保护环境，将不同种类的垃圾通过不同的方式来处理是必要的。但由于我国对垃圾分类的宣传与教育力度不大，垃圾要分类的理念并不被大众所接受，即使有垃圾分类的意识，也只有很少的人可以做到时刻注意。我国需要大量体力劳动者对垃圾进行手工分拣，才可以进行下一步的垃圾处理。手工分拣占用大量劳动力，同时进行垃圾分拣的小型分拣场位于市区内，容易对周边地区造成污染，产生卫生隐患。垃圾自动分类可以有效，快速地对垃圾进行分类，节约人力和时间，同时可以减少因垃圾分拣站对周边造成的环境影响，对保护环境具有重要意义。

1.2　分类算法的局限

目前主要通过垃圾图片训练分类网络，来对垃圾进行分类，但分类网络仅对整张图的输入通过概率进行类别判断，对于塑料瓶和玻璃瓶这些有着相似纹理特征的图片无法准确的进行判断。由于特征太过相似，有时人眼也无法准确分辨，导致适用于大多数分类网络的优化方法如增加网络深度，进行数据增广等也无法显著提高分类算法的准确率。因此需要提取其他垃圾图片上的其他特征来辅助分类算法进行判定。

1.3　垃圾图片文字检测的优势和挑战性

根据对垃圾特征的分析，对垃圾上的文字进行检测往往可以极大的减小垃圾分类的难度。为了引导消费者，出现在垃圾图片上的文字如“饮用水”，“啤酒”和“雪糕”等，直截了当地标明了商品类型[1]，而这些商品类型可以直接对应垃圾的类别“可回收塑料”，“玻璃制品”和“不可回收塑料”等。如此一来，再结合分类网络，可以快速准确地通过垃圾图片来判定垃圾所属类型。在商品包装设计时，商品生产厂家为了吸引消费者，往往会采用艺术字[2]，多角度，弯曲文本等复杂的文字表现手法，出现文字的位置也不尽相同。同时人们丢垃圾时，也会习惯的将商品包装折成一团，撕碎等，垃圾无法保持原来作为商品售卖时完整的形状。这使得对于垃圾上的文字检测，无法直接使用传统的 OCR（光学字符识别）。在处理垃圾文字检测问题时，需要选择针对于复杂场景的文字检测算法，同时调节网络参数来对其进行优化，使得算法可以在垃圾文字数据集上取得较好的效果。

1.4 国内外研究现状

目前仍没有专门针对于垃圾的文字检测算法，但由于垃圾文字属于自然场景文本的一种，很多自然场景文本检测算法可以直接采用。自然场景文本检测与传统的 OCR 不同，场景文本具有多角度，易变形，背景复杂，出现位置不确定的特点，所以无法应用传统的光学字符识别算法。目前以深度学习为主干网络的文字检测算法是场景文字检测领域的主流。最初，有团队尝试使用通用目标检测算法来实现文本检测如 SSD，YOLO 等，理论上可行，然而场景文本有其自己独特的特征，使用通用目标检测算法的效果不尽如人意，还是需要根据文本的固有特点来开发文本检测算法。基于深度学习的文字检测算法，大部分将文本检测当作通用目标检测的一个子任务，大体框架仍使用目标检测网络，如提出年代较早的 CTPN[3]，基于 SSD 通用目标检测算法的端到端文本检测算法 Textboxes[4] 和其改进版 Textboxes++，通过部件连接得到文本区域的 Seglink[5] 等。另外，还有一些团队提出在像素层面对文本区域进行提取，类似语义分割和实例分割网络，如本文使用的 PixelLink[6] 和 2017 年大放异彩的 EAST[7]; 半年前，有团队开始总结当前各种不同检测思路的优势，结合目标检测和语义分割的各自优点来构成文本检测网络, 如云从科技提出的 Pixel-Anchor[8]。

1）CTPN[3] 算法使用 CNN 和 RNN[9] 结合，由于文本通常可以被看作一个序列，可以通过前后文来进行文本位置确定，所以作者使用 LSTM[9] 循环网络，同时引入数学上的微分思想，先检测矩形文本段，在后处理过程中将其连接起来，但由于其构造文本候选区域的方式为垂直细长矩形的连接，本算法仅仅在检测水平文本区域时的表现较好，对于角度略大的文本框无法发挥作用。

2）EAST 由旷视科技公司发表于 CVPR2017，文中提出了一种高效且精确的文字检测器，EAST 采用全卷积主干网络，直接在整幅图片上对所有方向文本进行预测，省略了区域聚合和单词分割的步骤，同时着重设计 loss 函数和网络架构，在公开数据集 ICDAR2015 上取得了很好的效果。目前仍然被广泛采用，作为其他网络的基础。

3）Textboxes 算法由华科大白翔教授团队提出，从 SSD[10] 通用目标检测框架上改进而来。根据文本长宽比较大的特点，改变了检测框的形状，采用长条形的检测框。同时修改卷积核的大小，增加水平方向的感受野。Textboxes 网络为端到端网络，不仅对文本位置进行确定，还可以识别出文本内容。端到端算法中的文本识别又可以提高文本检测部分的检测效果。

4）Seglink[5] 于 2017 年发表于 CVPR，其主要思想是将文本拆分为文本片段，同时其中间具有连接关系，使用全卷积网络，同时加大网络深度，在不同卷积层的特征图上回归文本框，达到多尺度的目标检测（原文从 6 个不同卷积层进行检

测），最后通过融合规则对文本框进行融合，生成最终的结果。

5）PixelLink[6] 算法由浙江大学郑丹硕士发表于 CVPR2018，文中将文字检测问题当作实例分割来处理，同时连接思想来自于 Seglink，但 PixelLink 实现了像素层面的连接。将原本的卷积神经网络改为全卷积网络来得到大小为原图 1/2 或者 1/4 的大尺寸特征图，并在其上进行分类任务，在确定像素是否为文字后，判断邻域的连接关系，通过并查集的方法来聚合成文字区域，省去了回归文本框位置和进行 NMS（非极大值抑制）的过程，最后直接调用 opencv 里的内置函数生成带角度的旋转矩形框。

6）Pixel-Anchor[8] 方法由云从科技发表于 2018 年末，结合了像素分割和参考框回归的优点，可以认为是 EAST 和 Textboxes++ 的组合模型。该算法将 SSD[10] 和 EAST 分别作为 Pixel 部分和 Anchor 部分的基本框架，使用前者检测中等大小的文本，后者检测纵横比大或者占面积小，密集的文本，来得到检测结果。文中采取的方法有效的改善了 EAST 算法感受野小的问题，又引入 OHEM 避免了类别不均的问题。Pixel-Anchor 方法在 ICDAR 上取得了历史上的最好检测结果。虽然创新点不多，但证明了结合两种思想可以得到更好的检测结果，也给其他文字检测相关研究人员提供了新思路。

1.5 论文的主要工作安排

第一章介绍垃圾自动分类的重要性，分类算法的局限，垃圾图片文字检测的优势及面临的挑战，国内外自然场景文字检测算法的发展以及本文各章的工作安排。

第二章介绍本课题的具体内容，采用方法以及简述文字检测评估指标的方法和依据。

第三章简要介绍实现算法所需的操作系统环境，开发环境和网络框架等，同时介绍所选算法的基本原理。

第四章主要介绍本文使用的垃圾图片数据集的筛选，标注，四边形生成及出界自动更改等工作。

第五章主要介绍将 PixelLink 应用于垃圾图片数据库的实验过程，超参数调节，网络结构改并总结完成情况。

第六章主要总结实验过程中出现的各种问题，提出未来可能的优化方式。同时对未来可能的实际应用场景进行了简单描述。

第二章　课题的具体内容和技术指标

2.1　课题内容概述

课题实现的具体内容为：标注并构建垃圾文字数据库，了解各种不同的深度学习算法并选择适用于垃圾图片数据集的算法，并将其应用于垃圾数据库，训练模型并在测试图片集上测试模型效果，进行多次实验，调节参数与网络结构来实现更好的效果。其中垃圾图片中的文本使用文本框进行标识，文本框的表示方法有很多种，在 ICDAR2013 及之前的文本检测数据集中，文本框一直为水平矩形，但自然场景中的文字多存在倾斜角度，这一点在由谷歌眼镜进行拍摄的 ICDAR2015[11] 数据集中有着集中体现，所以大部分文本框需要用旋转矩形或四边形进行表示。垃圾与其他自然场景不同，其文本内容极有可能由于垃圾的不完整而发生形变，为了在标注时更加准确，应该使用四边形进行标注。选择算法应选择适合检测多方向文本的算法，同时要学习搭建卷积神经网络，可以在原网络模型的基础上做出创新。

2.2　文字检测评估指标

文字检测作为目标检测的一个子任务，使用召回率，精确率，F 值来评价文本检测算法的效果。其中召回率和精确率可以由检测结果直接计算得到，F 值则由召回率和精确率计算得到。

从理解的角度来讲，精确率高，则将其他区域误认为文字的几率就小。召回率高，则说明算法可以找到尽可能多的文字区域。一个好的深度学习模型，应该在这两个指标间做出权衡，多用 F 值来进行综合性的评价：

F 值 = 精确率 * 召回率 *2/（精确率 + 召回率）

可以看出只有精确率和召回率都比较高时才能得到好的 F 值，这个方法在评估文字检测算法的效果时被广泛采用同时效果很好，本文中的模型预测效果也使用以上指标进行评估。

由于与分类问题不同，目标检测类问题的预测结果很难用一个数值或概率来与真值进行比较，故需要根据任务的特点制定不同的匹配策略，同时不同的策略也使得精确率和召回率的计算方法有所不同。通常使用的文本检测算法评价方法有 IoU 算法[12] 和重叠矩阵匹配算法两种，其中 IoU 算法为一对一匹配，而重叠矩阵匹配可以兼容多种模式（一对一，一对多，多对一）。对于本课题中的文本框检测算法，是否可以将全部文本识别成一个整体也是算法效果好坏的一个评价标准，并且 IoU 匹配算法是应用于 ICDAR2015 数据集上检测效果的方

法，同时第四章的垃圾数据集全部以 ICDAR2015 格式构建，故评估算法的使用与 ICDAR2015 的一致。下面简要介绍 IoU 匹配算法的原理和计算公式。

IoU[12] 经常用于目标检测，其计算的是预测出的边框和 ground truth 的交集与并集的比值。

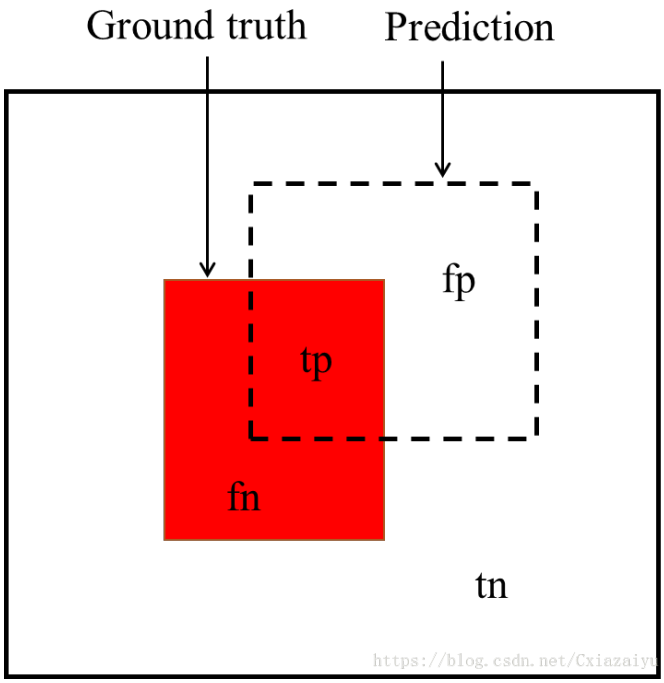

图 2-1　　IoU 定义图示

通用计算公式为：

$$\mathrm{m}\left(D_i, G_j\right) = \frac{\mathrm{Inter}\left(D_i, G_j\right)}{\mathrm{Union}\left(D_I, G_j\right)} \tag{2-1}$$

式中 Inter 项和 Union 项分别表示两个文本框 D_i 和 G_j 的交和并的面积。检测结果用 D (detected) 表示，真值用 G (ground truth) 表示。由于垃圾数据集按照 ICDAR2015 的格式进行标注，同时使用官方的评估脚本，所以区域计算方式同 ICDAR2015 和 ICDAR2013:

$$\begin{cases} \mathrm{Inter}\left(D_i, G_j\right) = \mathrm{Area}\left(D_i \cap G_j\right) \\ \mathrm{Union}\left(D_i, G_j\right) = \mathrm{Area}\left(D_i \cup G_{j)}\right. \end{cases} \tag{2-2}$$

其他数据集还有不同的处理方式，比如 ICDAR2003 中的并操作所得到面积需要得到包围目标的最小外接矩形框，并计算其面积。得到两个文本框之间的 IoU 后，使用阈值函数判断其是否匹配，公式为：

$$M(D_i, G)\begin{cases} 1, \text{ if } m(D_i, G) \geq th \\ 0, \text{ otherwise} \end{cases} \tag{2-3}$$

阈值 th 一般取 0.5。检测到的所有结果对真值的匹配程度，公式为：

$$N_{CD} = \sum_{i=1}^{|D|} M(D_i, G) \tag{2-4}$$

同理可求所有真值对检测结果的匹配程度。召回率，精确率和 F 值的计算为：

$$\begin{cases} \text{Recall} = \frac{N_{CG}}{|G|} \\ \text{Precision} = \frac{N_{CD}}{|D|} \\ \text{FScore} = 2\frac{\text{Recall*Precision}}{\text{Recall+Precision}} \end{cases} \tag{2-5}$$

IoU 的匹配策略很清晰合理，因此被广泛用于目标检测的评估，但缺点是只能考虑一对一的情况。

第三章　软件环境及算法简述

3.1　算法实现环境

实验室所用硬件为服务器一台，双路 GTX1080TiGPU 用来进行神经网络训练。本课题属于场景文字检测的子任务之一，所选系统，编程语言和深度学习框架应适用于大型神经网络的训练与测试。首先操作系统选择 Ubuntu16.04，同时使用 Python 语言作为编程语言，Ubuntu 作为轻量级开源系统，有着强大的稳定性和安全性，可以最大效率地发挥服务器硬件的性能，适合深度学习项目。深度学习框架有 TensorFlow, PyTorch 和 Caffe 三个主流框架，本课题选择 TensorFlow，因为 TensorFlow 作为 Google 主推的深度学习框架，有着庞大的第三方库支持，同时网络上教程和论坛交流很为活跃，可以方便的进行学习。

3.2　文字检测算法的取舍

本课题选择 Github 上的开源算法进行训练，初步认定 CTPN，Textboxes，EAST，PixelLink 等算法可以作为候选算法。其中 CTPN[3] 年代稍远，同时效果已经被其他算法超越，虽易于实现且经典，但是对于大倾斜角的文本检测效果很差，不考虑采用。Textboxes[4] 虽然效果很好，但是更适用于水平文本，结合垃圾图片上文本的特点，Textboxes 无法取得太好的效果。EAST 可以得到很好的效果，同时网络结构简单，但 EAST[7] 存在感受野较小的缺点，垃圾数据集上的文本有的横跨整张图片，应选择网络效果不受制于感受野，或者感受野大的算法。经过权衡，最终选择 PixelLink 算法来实现本课题。

3.3　PixelLink 算法简介

PixelLink 算法基于全卷积神经网络，主干网络完成预测单个像素（pixel 部分）是否为文本以及其邻域 8 个像素是否为连接（link 部分）关系这两个独立的任务，总共输出 18 张特征图。提取出特征图后，使用两个不同的阈值进行过滤，通过 link 部分的预测结果，利用并查集的方法将所有同属于一个文本实例的文本像素形成连通区域，每一个连通的子区域都是一个文本实例，完成实例分割。接着调用 openCV 函数 minAreaRect 来得到连通区域的最小外接框，即为所求文本框。PixelLink 相比于其他的文本检测算法，无需文本框回归的步骤，而是直接通过实例分割的结果得到了文本框，因此节省了不少算力和时间。

3.3.1 网络结构

本课题采用文字检测常用的 VGG16[13] 作为主干网络，VGG 于 ImageNet[14] 中取得优异成绩，因为其提出时间长，同时实现较为简单，可以快速有效地得到结果。但有很多近年来出现的新型卷积神经网络也可以取得很好的结果，可以在随后优化时选择更改主干网络来改善模型性能。同时考虑到要进行实例分割，使用 FCN（全卷积神经网络）[15] 版本的 VGG，网络构成图如3-1。

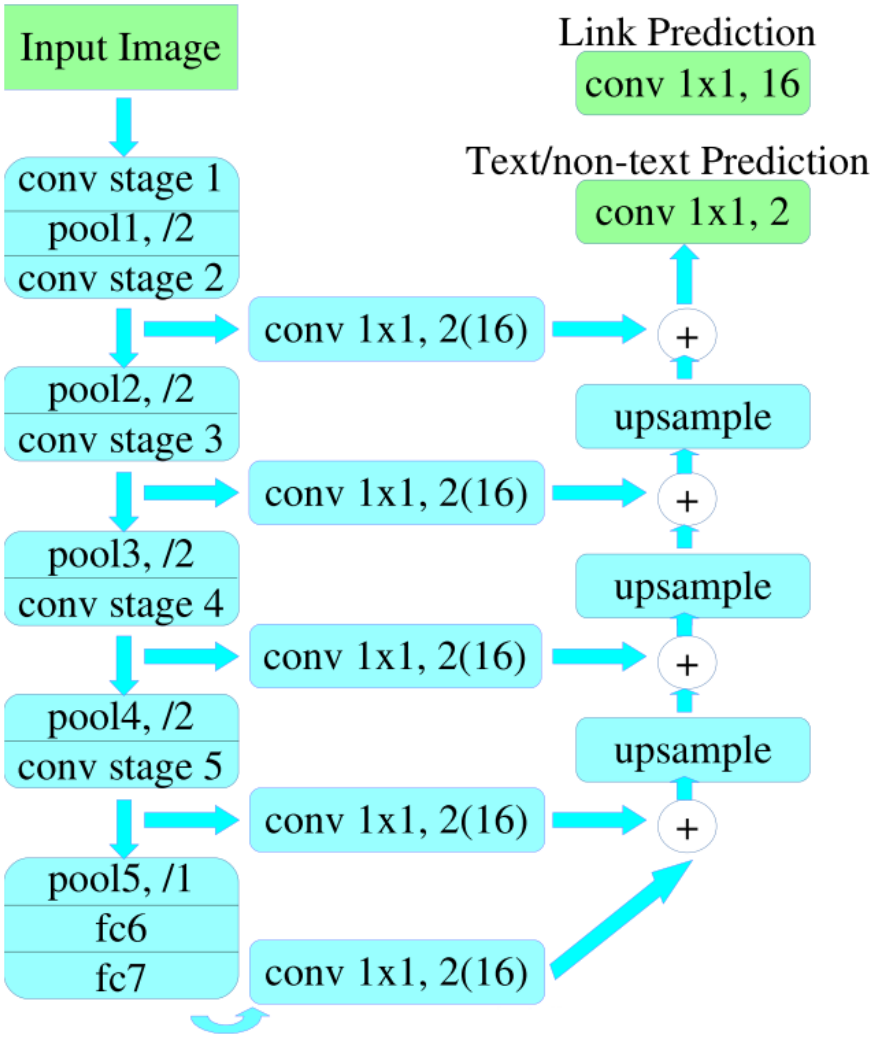

图 3-1　　PixelLink VGG16-2s 网络构成

VGG 由五个连续的卷积模块构成，每个卷积模块由两个卷积层和最大值池化层组成，VGG 采用两层 3*3 的卷积核而不是之前普遍采用的一层 5*5 卷积核，VGG 的两层 3*3 在感受野上与 5*5 一致，但增加了网络深度同时减少了计算量。由于 VGG 模型最初提出时被应用于图片分类任务，在最后的卷积之后还要跟随全连接层，并使用 Softmax 来输出概率值确定图片的类别。Softmax 是归一化指

数函数，可以将一组 N 维向量转换为一组可以代表概率值的 N 维向量，在分类任务中，常常被用于计算预测结果。Softmax 函数公式：

$$\sigma(\mathrm{z})_{\mathrm{j}} = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \tag{3-1}$$

其中 j 为图像类别，(z) 则为图片属于该类别的概率。除了最后的连接层，其余卷积层的激活函数都使用 ReLU[16], 表达式为：

$$\mathrm{Re\,LU(x)} = \begin{cases} x, \text{ if } x > 0 \\ 0, \text{ otherwise} \end{cases} \tag{3-2}$$

由于本课题算法要在像素级别上提取特征图来达到实例分割的效果，需要将 fc6 改为卷积层，使用 3*3 卷积核，同时 fc7 也转化为卷积层，使用 1*1 卷积核，保持特征图大小不变。

3.3.2 特征合并

在 CNN 网络对图片语义特征提取的过程中，越靠近输入层的卷积层提取的信息细节越多，但没有很强的语义信息；越靠近输出层的卷积层提取的信息越具有强语义特征，通过对各层的特征图进行特征合并，可以同时在特征图上保留细节与强语义信息，可以提高语义分割和实例分割的准确性。本课题采用全卷积网络所采用的特征合并方法，使用双线性插值的方法进行上采样（upsampling），将分辨率低的深层信息进行分辨率扩大，同时将经过上采样的各层特征图相加，得到最终结果。在 PixelLink 中，作者提出了两种特征融合方法，以 conv2_2 为最终大小时，特征图的大小为原图的 1/2，被称为 2s 模型；以 con3_3 为最终大小时，特征图的大小为原图的 1/4，被称为 4s 模型。2s 模型的构成如图3-1。

3.3.3 后处理

如3-1，网络输出有两个独立任务：文本预测和邻域像素连接预测。二者为二分类问题，故使用 Softmax 函数进行激活，文本预测需要 2 个通道，连接关系的预测需要 8*2=16 个通道，因为要判断与周围 8 个像素的连接关系。得到图像的文本预测图和连接关系预测图后，设定两个阈值对其进行二值化处理，这样就可以分别得到预测为正的像素图，和预测每个像素与其相邻像素的连接关系。接下来将若两个正文本像素之间连接关系为正，则两者可以得到一个连通子集。以这种方法遍历所有正文本像素，得到分割后的实例。得到实例后，为了可以同一目标检测输出的格式，需要生成矩形框来表示文本的预测位置。这一步直接调用 opencv[17] 中的 minAreaRect 函数完成，该函数可以输出一个带角度的矩形，可以通过 IoU 算法来评估算法效果。

3.3.4 真值计算

在 ICDAR2015 格式的训练集中，标注信息为文本框的四个点坐标，而本课题所用算法需要进行像素层面的预测，所以需要将文本框信息转换为像素级的信息。由于网络存在两个独立任务，所以真值也由两部分组成，分别为像素类真值和连接关系真值。根据像素真值将图片中像素分为两种，一种为文本像素，另一种为非文本像素。在本算法中，仅属于单一文本框的像素标为文本像素，不在文本框内的或同属于多个文本框的像素均为非文本像素。与语义分割不同，为了防止训练出的网络出现无法区分多个实例的情况，将不同文本框重叠部分的像素也标为非文本像素，虽然可能降低算法的召回率，但可以保证每一个文本实例都可以被独立检测出来。连接关系真值的计算原则是：当两个相邻文本像素同属于一个文本实例时，它们的连接关系为正，其余情况一律为负。

3.3.5 损失函数计算

损失函数由两部分组成，分别为像素分类和连接关系分类的损失函数，总体损失函数为其加权和，由于连接关系的计算基于文本像素，所以像素分类的重要性更大，故赋予其更高的权值，算法原作者使用 为 2.0，本文沿用作者的设定。损失函数表达式为：

$$\mathrm{L} = \lambda L_{\mathrm{pixel}} + L_{\mathrm{lin}\,k} \tag{3-3}$$

首先说明像素分类部分的损失函数，损失函数由交叉熵来表示，交叉熵公式如下所示：

$$\mathrm{H(p,q)} = -\sum_{\mathrm{x}} p(x)\log q(x) \tag{3-4}$$

上式为分类任务的交叉熵公式，在分割任务中要使用矩阵格式，如下式（3.7）由此可知，占像素面积大的文本将会对 loss 函数产生更大的影响，而我们是要检测出所有文本实例，无论大小，这样对小面积的文本是不公平的，因此 PixelLink 作者提出了一种可以赋给所有文本实例相同权重的方法，公式如下：

$$w_i = \frac{\sum_{i=1}^{N} s_i}{N s_i} \tag{3-5}$$

其中 N 为文本实例总个数，此公式将每个像素的权重设为其面积占总体文字面积的反比，这样处理后，交叉熵加和的结果中，各个不同文本实例的权重相同。根据此机制即可计算出整个图片各个像素的权值，得到一个和特征图等大的权重矩阵，损失函数表示为：

$$L_{\text{pixel}} = \frac{1}{(1+r)S} W L_{\text{pixel}_{-CE}} \tag{3-6}$$

其中 S 表示上面计算出的图中所有文本面积之和。L 矩阵为交叉熵矩阵：

$$L_{pixel_{-CE}}(u,v) = -\ln p_t(u,v) \tag{3-7}$$

第二部分是像素连接关系分类任务的损失函数，由于第二部分的损失仅在文本像素上计算，在连接关系预测中存在分类不平衡，文本实例内部的像素连接为正，边界位置的文本像素连接为负，文本实例内部的像素数量远远大于边界的文本像素，为了分类平衡，这一部分的损失函数计算也使用平衡交叉熵进行计算。首先，需要计算文本像素上正，负连接关系的交叉熵矩阵：

$$L_{link\_pos} = W_{pos\_link} L_{link\_CE} \tag{3-8}$$

$$L_{link\_neg} = W_{neg\_link} L_{link\_CE} \tag{3-9}$$

其中，L_(link_CE) 为交叉熵三维张量，表达式为：

$$L_{link_{CE}}(u,v,k) = -\ln p_t(u,v,k) \tag{3-10}$$

W_(pos_link) 和 W_(neg_link) 分别为正负连接的权值三维张量，计算如下：

$$W_{\text{poslink}}(u,v,k) = W(u,v) * (Y_{\text{link}}(u,v,k) == 1) \tag{3-11}$$

$$W_{\text{neglink}}(u,v,k) = W(u,v) * (Y_{\text{link}}(u,v,k) == 0) \tag{3-12}$$

其中 Y 为 ground truth，有了权值与交叉熵计算，像素邻域连接关系预测的损失函数为：

$$L_{link} = \frac{L_{link,pos}}{reduce\_\text{sum}\left(W_{poslink}\right)} + \frac{L_{link}, neg}{reduce\_\text{sum}\left(W_{neglink}\right)} \tag{3-13}$$

至此，网络的 loss 函数计算结束。

3.3.6 优化方法

SGD+Momentum[18] 为目前使用较为广泛也是易于实现的方法，本课题所使用的 Tensorflow 框架直接提供接口进行使用。所采用的方法更新参数的方式为：

$$\Delta\omega = \text{momentum} * \omega_{t-1} + lr * \text{grad}(L, \omega_t) \tag{3-14}$$

$$\omega_{t+1} = \omega_t - \Delta\omega \tag{3-15}$$

公式中 即为待优化参数，t 表示迭代的次数，L 为 loss，grad 为偏导数，lr 表示学习速率，momentum 表示动量参数。具体参数在算法运行时根据表现进行调整。

3.3.7 后过滤过程

PixelLink 算法的工作原理基于实例分割，将预测为正的像素互相连接，就不可避免的出现一些噪点，这些噪点通常面积不大，但数量会很多，这对于评估时精确率会产生很大的影响，因此需要在获得文本实例后，对其进行筛选过滤。PixelLink 原作者通过检测文本框的短边长度或面积大小来认定文本区域是否为噪点。当使用 ICDAR2015 训练数据集时，判断文本框短边长度小于 10 或者文本框总面积小于 300 时，将其认定为噪声并抛弃。这两个数字来自于官方文档，ICDAR2015 数据集上的 99% 真值文本框的短边长度都大于 10，99% 真值文本框的面积都大于 300，故使用这两个值进行过滤。

3.3.8 并查集概念介绍

PixelLink 算法中合成文本区域时要用到并查集方法, 本小节进行简要的介绍和举例。并查集为一种数据结构，常用于将不相交集合进行合并。初始化每个元素为一颗独立的树，合并时查找元素之间的关联关系，指向规定的父节点，查找时判断两个节点是否属于同一父节点，若同属于则有联系，将其合并。举例：设连接通路为（x, y）, 则含义为像素 x 和像素 y 同为文本像素且两个像素之间的连接关系为正。如现有（1，2），（2，4），（3，4），（3，5），（6，7），（6，9），（8，9）几个连接关系。进行并查集合并时，最后结果为：1，2，3，4，5 和 6，7，8，9 两个森林，在本论文任务中表现为两个独立的文本实例，实现实例分割。

第四章　　数据集构建与规范

4.1　数据及简介

本课题所使用的数据集为使用高清便携摄像头拍摄的单一背景图片，每张图片包含一件垃圾，分辨率为 900*960，格式为 png。最初数据集用来训练分类网络，故没有标注信息，同时所有图片被分为电子垃圾，玻璃制品，有害垃圾，厨余垃圾，金属垃圾，可回收纸质垃圾，不可回收纸质垃圾，织物，可回收塑料和不可回收塑料，由于本课题不涉及图像类别区分，故忽略其标签，将其统一整理在一起，共有 10624 张图片。图片实例4-1。

图 4-1　　垃圾数据集图示

4.2　数据集筛选

要对垃圾图片中的文本部分进行标注，首先要筛选总数据集，选出具有文字区域的图片，如部分厨余垃圾，织物，电子垃圾，撕掉标签的玻璃瓶等无文字部分，故应先进行筛选，不符合要求的图片如4-2 同时，为了可以让分类任务提高精确率，构建垃圾数据集时，对一些经常出现的垃圾进行了多角度拍摄，如4-3。

这种做法虽然可以对分类问题做到数据增广，但其中的文本内容完全一样，若在训练中这种图片存在过多，会加重过拟合程度，即训练出来的模型在训练集上表现良好，也可以取得低的损失函数值，但在测试集上表现很差。为了避免

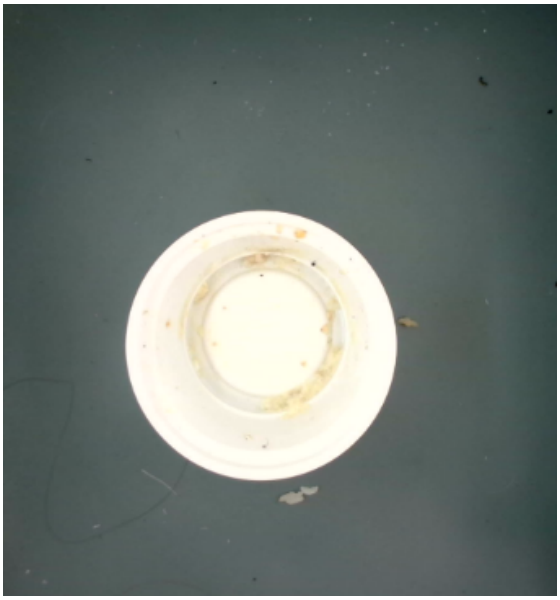

图 4-2　　无法进行标注文本的垃圾图片

图 4-3　　对同一垃圾的多角度拍摄

这种情况出现，在筛选图片时也要删掉这种对同一垃圾进行多角度拍摄的图片，仅保留一张即可。有了以上两个筛选标准，对数据集进行筛选后，得到了总数为 4255 的子集，并将它们分为训练集和测试集，为了得到更好的效果，同一类型的图片如塑料瓶，这种出现频率较高的图片必须同时在训练集和测试集中存在，否则将会引起严重的过拟合现象，经过筛选后，选择每一类垃圾的 15% 左右组成测试集，共有图片 600 张，其余 3655 张图片为训练集，比例为 1：6 左右。之后的工作为对这些图片进行文本框标注。

4.3 数据集标注

标注工具首先使用 MIT 开发的开源标注工具 labelme，安装运行之后发现无法对角度大的文本行进行标注。垃圾数据集采集时，物体在摄像头中的方向是随机的，所以任意角度的文本行都会存在，为了可以准确的进行标注，需要使用可以进行四边形标注的软件，最终选择精灵标注助手作为标注软件。标注时，文本框用四边形进行标注，最终输出 xml 文件，其文件结构如下：

```
<?xml version="1.0" ?>
<doc>
<path> 存储绝对路径 </path>
<outputs>
<object>
<item>
<name> text </name>
<polygon>
<x1> XXX </x1>
<y1> XXX </y1>
<x2> XXX </x2>
<y2> XXX </y2>
<x3> XXX </x3>
<y3> XXX </y3>
<x4> XXX </x4>
<y4> XXX </y4>
</polygon>
</item>
</object>
</outputs>
<time_labeled> 1552828448099 </time_labeled>
```

<labeled>t rue </labeled>

<size>

<width> 900 </width>

<height> 960 </height>

<depth> 3 </depth>

</size>

</doc>

若图中有多个文本框，则会有多个 item 存在，其中坐标即为标注的文本框四个角点的坐标，横坐标不仅限于（0，900）之间，纵坐标不仅限于（0，960）之间，标注时有可能出现负数和超过图片边界的点出现，这一点在后续处理垃圾数据时进行统一更改。文件中 text 即为标注的文本框标签，由于所有目标均为文本框，故该值在本课题中无实际作用。

4.4 数据集格式转化及规范

本文使用 ICDAR2015 效果评估脚本进行效果测试，标注的真值需要存为 ICDAR2015 的相同格式，其格式为：X1, y1, x2, y2, x3, y3, x4, y4, ### 其中 ### 为文本内容，由于本次课题不涉及文字识别的部分，仅输出文字的位置，故无需考虑。由 xml 转换为 txt 的算法过程如4-4。

1) 导入相关库，建立函数主体，输入参数为读取 xml 文件的目录和写入 txt 文件的目录

2) 打开 xml 目录并列举所有存在的文件

3) 设置变量，寻找所有 xml 后缀名文件并将其名称存成字符串格式

4) 对于上述变量，从第一个文件开始循环，同时以相同的名称建立 txt 文件，并存储在输入的 txt 目录中，并以写入方式打开。

5) 使用标准函数 ET.parse 解析 xml 文件，同时使用 getroot 得到所有子节点。

6) 对于所有找到的子节点，查找子节点名称为“item”，name 为所标注的类别名称，本文中为“text”，接着查找“polygon”字段，新建变量存储 4 个点坐标

7) 在新建的 txt 文件中写入 4 个坐标，继续第 6 步寻找其他文本框

8) 回到第 4 步，继续处理下一个 xml 文件

遍历列出的所有 xml 文件，至此处理完毕。处理过后的文件内容实例如下：

448, 472, 502, 384, 595, 434, 544, 528, text

294, 308, 338, 224, 364, 233, 320, 324, text

572, 569, 640, 472, 675, 502, 610, 584, text

370, 380, 386, 360, 434, 384, 418, 405, text

此时文件格式和 ICDAR2015 标注格式一致。输出 txt 文本框后在 ICDAR2015

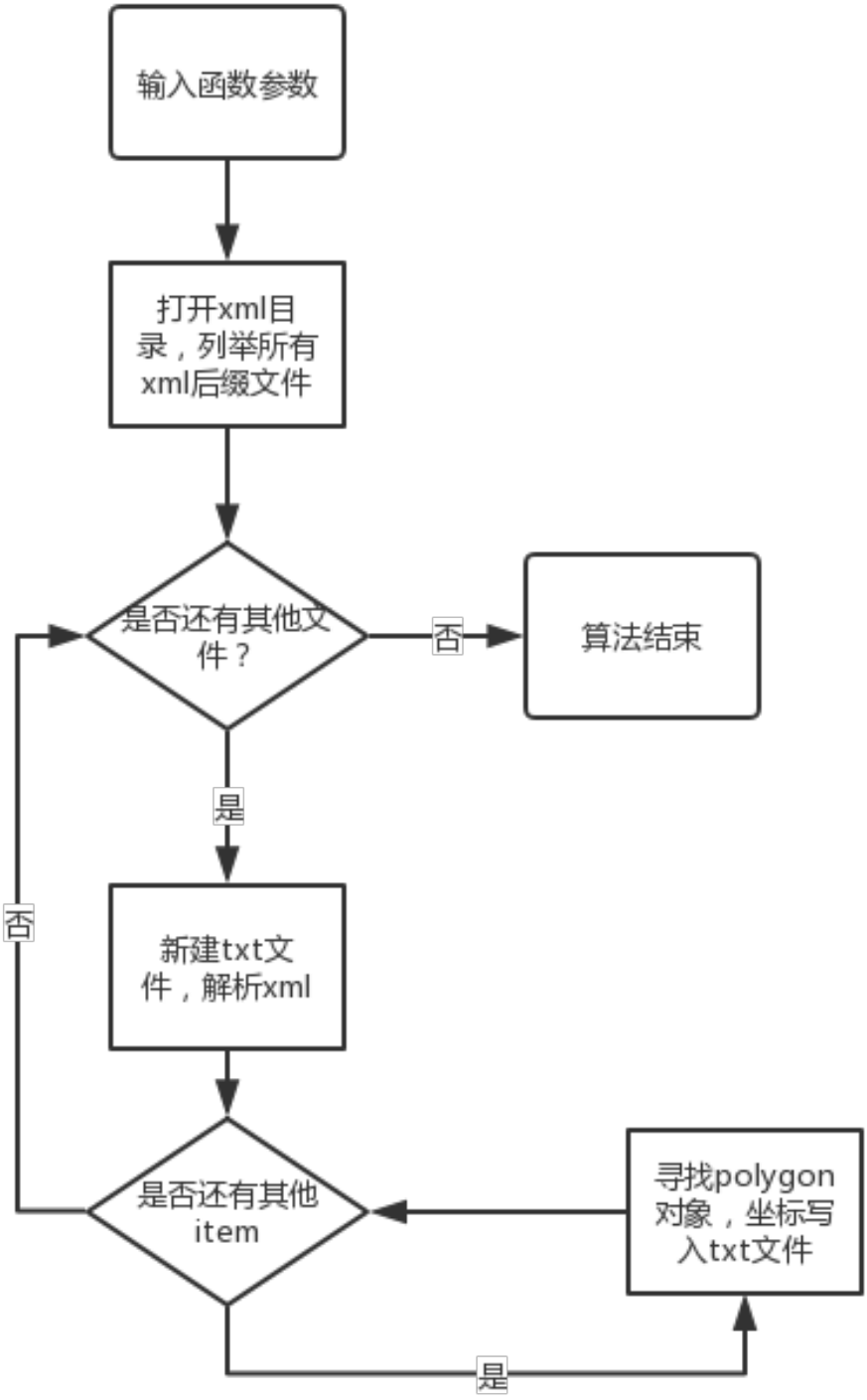
输入函数参数
打开xml目录，列举所有xml后缀文件
是否还有其他文件？
否
算法结束
是
否
新建txt文件，解析xml
是否还有其他item
寻找polygon对象，坐标写入txt文件
是

图 4-4　　格式转换流程图

的评估脚本上上传，运行 error，仔细分析后发现问题在于 ICDAR 脚本的计算方式，第一个点必须是位于左上角，同时几个点顺序为顺时针，否则计算面积为负，无法进行下一步 IoU 的计算。为解决这个问题，还需要对数据集进行规范，由于格式转换脚本中已经提取出各点值，所以无需另写程序，直接在上述脚本中修改即可。设计算法逻辑为：首先筛选横坐标，横坐标下的两个点为左边两个点，其余两点为右边两个点，左边两个点中纵坐标小的为第一个点，另外一个为第四个点；右边两个点中纵坐标小的点为第二个点，另外一个为第三个点。编写后使用 ICDAR2015 评估脚本进行测试，仍有面积为负的情况。发现在进行比较时，由于 xml 文件中的坐标为字符型（str），比较大小的逻辑和整型不同，导致各点坐标比较错误。之后又对逻辑进行了完善，始终有不符合要求的文本框存在，最终查找 Python 手册，找到了内置的 polygon 函数可以生成四边形对象。工作方式为，输入一组 4*2 数组，将每一行作为一个点的横纵坐标，按照顺时针顺序生成四边形。建立之后，配合 exterior.coords.xy 函数提取横纵坐标，间接地实现了自动排序。四个点坐标按照要求排列完毕。人工标注文本框的时候，不免出现标注出界的情况，例如横坐标出现负数或大于 900 的情况，虽然这种情况极少，但会影响 tensorflow 读取，导致丢掉出界的文本框，为了保证算法可以得到最好结果，尽可能多读取文本框，需要对所有文本框左边进行检查，如果出界则标在图片边缘。经过几次修改，最终流程图为4-5。

1）输入为之前算法提取的四个点坐标

2）建立 4*2 矩阵，将所有点坐标存入

3）调用函数 poly，将之前矩阵中的点作为四边形各点坐标

4）使用 exterior 提取横纵坐标

5）使用 for 循环判断坐标值，横坐标要在 [0,900] 之间，纵坐标要在 [0,960] 之间

6）使用 assert 确保所有坐标满足要求，若有一个不满足则程序停止输出 error

通过该程序可以顺利生成 txt 文件，同时所有文件可以通过 ICDAR2015 的评估程序，不再报错。为了使用 tensorflow 框架，还需要将 txt 数据集转换为 tensorflow 读取的 tfrecord 格式，由于将 ICDAR 格式数据集转换成 tfrecord 格式的工作为 tensorflow 下实现文本检测的第一步，有很多开源代码可以做到这一步，本课题选用 PixelLink 作者 Github 中附带的脚本实现，经过简单的更改路径和读取数据集，可以生成垃圾图片数据集的 tfrecord 数据集。

4.5 数据集 RGB 均值计算

在图像处理相关算法中，出于简便计算的目的，需要减去整个数据集图片的 RGB 均值，对所有图片进行像素值归一化后再输入网络。PixelLink 算法中直接

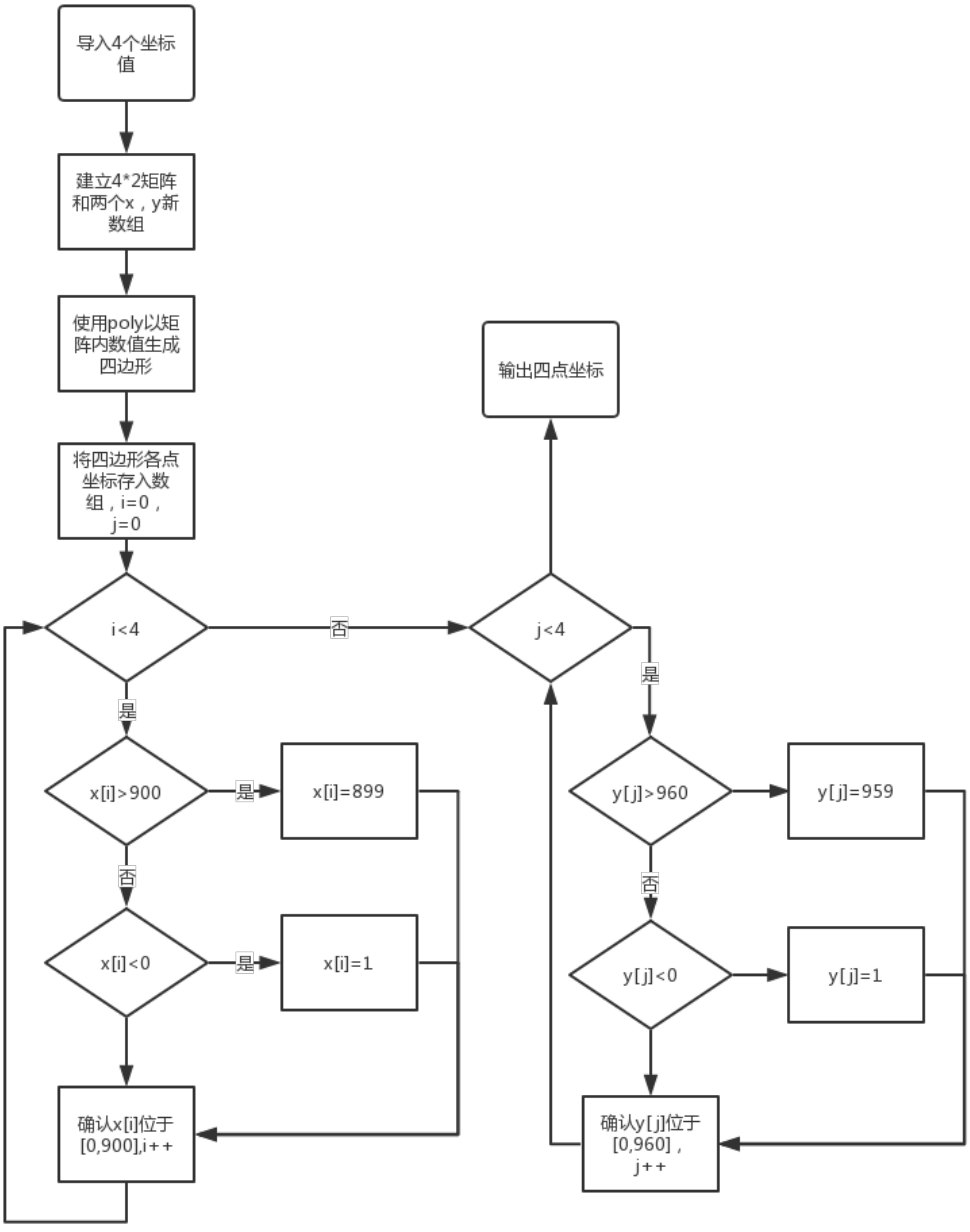

图 4-5　　标注数据规范化流程图

使用了 ICDAR2015 上的 RGB 均值进行计算，出于科学严谨性考虑，本课题进行了垃圾数据集的 RGB 均值计算，来代替默认的 ICDAR2015 数据库的图片像素均值。计算流程如下：

1）首先读取目标目录，存为列表

2）建立三个空矩阵用于存储 RGB 值，空矩阵为 1*n，n 为图片数量

3）循环读取图片，获得图片中的三个通道的均值，直接使用 mean( ) 函数计算均值。

4）所有图片读取结束，求取矩阵的均值，即为整个图片数据库的 RGB 均值

得到垃圾数据集的 RGB 均值为 129，133，125，ICDAR2015 的 RGB 均值为 123，117，104，两者有一定差距。

4.6 垃圾文本数据集数据增广

由于垃圾数据集图片数量不多，可以使用一些程序对图片进行随机的旋转，白化和裁剪，来增加模型的鲁棒性，具体实现方法与 SSD 的数据增广方法类似，共由四步组成。首先对图片进行随机旋转，在每张图片读取时，以某一概率旋转图片，旋转的角度在 90 度，180 度和 270 度[19] 之间选择。进行旋转后，对图片进行裁剪，裁剪区域随机，裁剪后仍为矩形图片，裁剪面积为原图的 0.1 到 0.9 之间，同时长宽比控制在 [0.5，2.0]。统一将所有图片分辨率整合到一个固定尺寸，该尺寸由输入的参数决定，默认大小为 512 * 512。最后，对 RGB 三通道色彩值进行扰动，同时随机更改亮度和饱和度。经过上述步骤，我们可以增加图片数量，由于对图片进行了 4 种不同的操作，很多在原图中完整的文本框被切割，这里我们认为剩余的文本框若是不到原文本的 20%，则忽略其中的文本，在 W 权重矩阵上的值为 0，不参与 loss 的计算。

第五章　PixelLink 实验

5.1　实验环境搭建与实现细节

实验环境的框架和系统部分已经有过介绍，本小节不再赘述，主要根据算法实现的要求列出所需软件库和 Python 版本的选择。Python 版本选择 2.7，虽然版本较老，但支持库较多。首先 Python 需要导入所有基础数学计算库，如 Numpy 等，简洁的做法为使用 anaconda 集成环境，将所有基础库全部导入。由于本课题需要使用 opencv 的部分功能，故需要安装 opencv-python。Tensorflow 选择最新版本即可。另外，由于服务器提供给多人使用，为防止不同版本库的冲突，所有实验需要在虚拟环境下进行。通过随机梯度下降结合动量的方式优化模型，默认权值衰减系数为 0.0005，动量系数为 0.9。在前 100 步学习中，默认学习率为 0.001，同时保存模型文件，在剩余的训练中采用 0.01 的学习率。所有训练图片尺寸更改为 512*512 分辨率进行训练。

5.2　原模型训练结果及参数调整

5.2.1　预训练模型测试

首先，在 Github 上作者放出的原工程中，提供有在 ICDAR 上训练过的模型，第一步先下载 4s 模型，直接进行测试，进行效果可视化后结果如5-1。将模型测

图 5-1　ICDAR2015 模型检测可视化结果

试结果进行评估，结果为：

“recall”：0.285，“precision”:0.103,“hmean”：0.151

可以看出，精确率仅有 0.1 左右，效果较差。

5.2.2 默认参数训练

既然固有模型的效果不好，便需要使用垃圾文字训练数据集从零开始训练模型。第一次训练使用论文原作者在 ICDAR2015 上所使用的文字阈值和连接阈值，分别为 0.6 和 0.9。全部使用前文提到的默认参数，学习率衰减设为 0.9999。经过 11 万步左右的训练后，loss 到达 1.1 左右。停止训练，读取模型进行测试，结果可视化如5-2。 可以看出部分图片的非文本区域会被误认为文本区域，也存

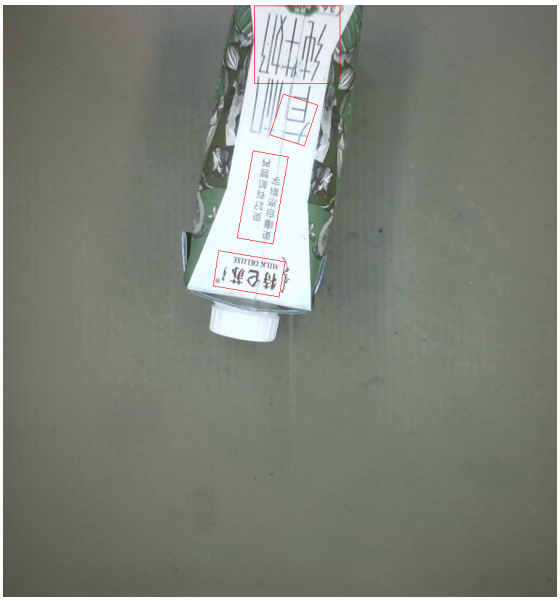

图 5-2　　默认参数测试可视化结果

在部分文本检测不全的问题。将生成的测试集 txt 文件压缩，使用 ICDAR 评估脚本进行测试，测试结果如下：

“recall ”: 0.347, “precision ”: 0.333, “hmean ”: 0.340

由于训练结果较好，继续进行训练，到 20 万左右时暂停，提取模型文件并测试结果：

“recall”：0.355，“precision”：0.346，“hmean”：0.351

可以看出效果有了一点提升，在此的基础上，进一步进行训练，同时观察 loss 是否有下降的趋势，在迭代到 40 万步左右时，loss 值基本稳定在 0.9 左右，无法继续下降，停止训练同时进行结果可视化和评估，可视化结果基本等同于 11 万步模型的效果，测试结果如下：

“recall”：0.349，“precision”：0.339，“hmean”:0.344

可以看出，超过 10 万的全局步数基本不会改善网络效果，而且训练到 40 万步时，随着训练的深入会有过拟合现象的发生。接下来应该考虑对调节参数进行优化，来得到更好的结果。

5.2.3 阈值对训练的影响

根据上一次实验测试集可视化呈现出来的结果，有些非文字区域被识别为文本，可以适当提高文字阈值，从之前的 0.6 改为 0.7，来更严格的筛选文本区域。同时有些区域文本并未完全连接在一起，应降低连接检测阈值设为 0.5，来尝试连接更多的文本像素。在先前 11 万步模型基础上继续进行训练，但由于服务器内存占用过大进行自动重启，仅训练到 14 万步，训练进程被迫停止，可视化结果与之前并无太大差异，使用评估脚本进行测试，测试结果如下：

“recall ”: 0.351, “precision ”: 0.34, “hmean ”: 0.350

可见效果有所改善，但存在一种可能，为之前模型训练不充分，故没有取得最佳效果，所以改变阈值是否有用仍有待探讨。两个阈值可以在 0.5-0.9 之间随意变动，要想求出最优阈值，至少需要 25 次实验，以每次实验训练到 15 万步为标准，每次实验需要 20 小时左右，由于时间关系，无法使用不同的阈值组合来进行多次实验，下一步为继续探究别的参数的影响。

5.2.4 学习率对训练的影响

由于学习率设置过大会出现梯度摇摆的情况，使得梯度无法到达极值点，将之前的 0.001 和 0.01 学习率改为 0.000001 和 0.0001，进行训练。训练到 25 万步左右，由于学习率过小，同时衰减率不变，使得最后学习率基本为 0，已经没有继续训练的必要，将模型进行测试，测试结果如下：

“recall ”: 0.233, “precision ”: 0.31, “hmean ”: 0.266

由于本次实验已经进行到 25 万步仍无法取得好的结果，故改小学习率对于本课题并无意义。0.001 和 0.01 的学习率只要在没有梯度爆炸的前提下可以使用。由于这个学习率已经足够大，不再做增加学习率的尝试。

5.2.5 输入图片尺寸对训练的影响

原文中，为了方便处理裁剪后的图片，将所有输入的图片格式同一处理为 512*512，这样的操作不仅可以节约时间，512 的大小也是大多数目标检测任务倾向于使用的分辨率。但高分辨率的训练图片往往可以取得更好的结果，故进行使用原尺寸 900*960 代替之前的 512*512 进行训练。由于显卡内存问题，需要将批尺寸改为 4 张，训练速度大约为 1.2s 一步，同时在运行不到一万步时，出现了损失函数计算为 nan，即损失函数不存在的问题，经检查，之前的训练没有出现这个问题，则不是学习率过高导致，是批尺寸过小导致的。由于小的批尺寸会导

致这个问题出现，同时无法在 1080ti 显卡上以 8 的批尺寸运行该算法，故仍使用 512*512 的分辨率进行实验。

5.3 优化网络结构及训练结果

前面的实验讨论了各个参数的影响，其中最优阈值需要进行多次实验，有可能大幅改善网络表现，而学习率选取合适，不需更改，图片尺寸若要修改需要更多台显卡同时工作，来保证批尺寸一定，不会出现过拟合。为了提高网络的表现，本小节基于对卷积神经网络的理解与学习，借鉴往年优秀的神经卷积网络模型，改变算法的主干网络，进行实验，来获得更好的实验结果。

5.3.1 增加网络深度及训练结果

首先可以进行的是多加卷积层，提取更深层次的语义特征。由 VGG[13] 原论文提供的模型与训练数据显示，采用 VGG19 的结构可以达到更好的效果，原 VGG 网络结构如图3-1所示，改动后的网络结构如5-3。 由于上采样部分与之前网络相同，不在图中进行体现。使用 tensorflow 中 slim 模块的 repeat 功能可以更改网络层数，在第四，五，六个卷积模块分别多加一层 3*3 卷积，更改网络完成后进行训练，每步训练时间增加到 0.8s，训练至 10 万步左右时停止训练，使用模型进行测试，评估结果为：

“recall”：0.366，“precision”：0.348，“hmean”：0.357

由此可见，虽然步数减少了，但由于网络层数的加深，效果比之前训练的略好，达到了目前最高的 f 值，说明适当加深层数对提高网络表现有所帮助。同时 VGG 原论文还指出，在继续加深网络层数时，会出现梯度爆炸和梯度消失等负面影响，并不能显著提高网络性能，需要用其他方法来进行网络优化，出于节省实验时间的考虑，不再继续尝试增加网络层数。

5.3.2 添加 Inception 模块及训练效果

在进行增加网络深度的尝试后，反观前几年的 ImageNet 比赛,GoogLeNet[20] 战胜了 VGG 获得了第一名，说明 GoogleNet 的网络模型有可以借鉴之处，对论文进行学习后，在原有网络增加了网络深度的基础上，增加 GoogLeNet 使用的 Inception 模块。Inception 模块如5-4所示 在 inception 模块中，在同一层使用不同尺寸大小的卷积核，形成感受野不同的特征图，同时 padding 参数设为“same”，使得不同卷积核运算得到的特征图大小一致，进行相加后输入至后续模块。GoogLeNet 在每一卷积层都使用了该模块，可以在不增加网络深度时，提高网络表现。本次实验效仿 GoogLeNet，在 block3 增加 Inception 模块进行尝试，改进后的主干网络如5-5。如图，保留之前三，四，五模块的三层卷积层不变，在第三个卷积模块增加两层使用 5*5 卷积核的卷积层，与之前的三层并联，进行相

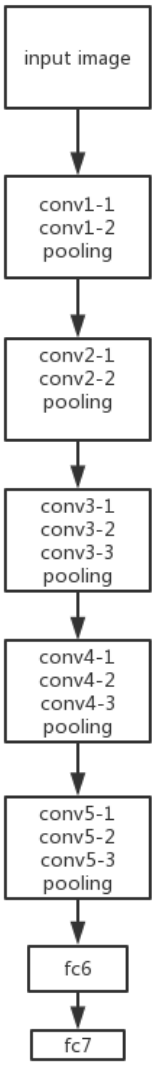

图 5-3　　更改后的类 VGG19 网络

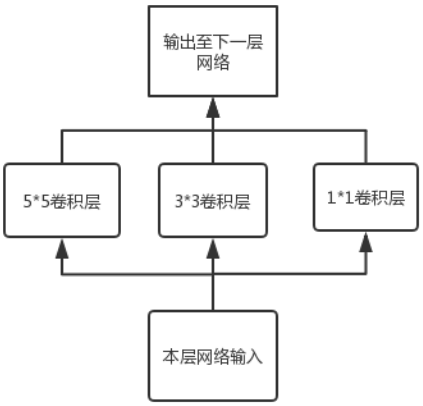

图 5-4　　inception 模块图示

加之后输入到池化层。更改结构后进行实验，参数为默认参数，训练时每一步的用时达到 1 秒左右，训练结果的可视化为5-6。 训练的结果评估为：

“recall”：0.379，“precision”：0.358，“hmean”：0.369

可见对网络结构进行复杂化后，可以发现召回率有着明显提高。

5.4　实验结果总结

至此已经进行了 9 次不同的模型训练，其中改变分辨率的实验由于损失函数计算为 nan 而没有成功，其余 8 次实验的各个评价指标如表5-1。

表 5-1　　实验结果总结表

实验组别	召回率	精确率	F 值
预训练模型	0.285	0.103	0.151
默认模型 11 万步	0.347	0.333	0.340
默认模型 20 万步	0.355	0.346	0.351
默认模型 40 万步	0.349	0.339	0.344
阈值更改模型 14 万步	0.351	0.340	0.350
19 模型 10 万步	0.366	0.348	0.357
19+Inception 模块 11 万步	0.379	0.358	0.369

可见，使用在 ImageNet 上效果更好的底层网络后，可以提高算法的综合表现，如不考虑每一步训练时长的影响，可以在 5 个卷积层增加 inception，可以继续提高模型表现。由于时间关系，暂不进行下一步的网络结构改变。

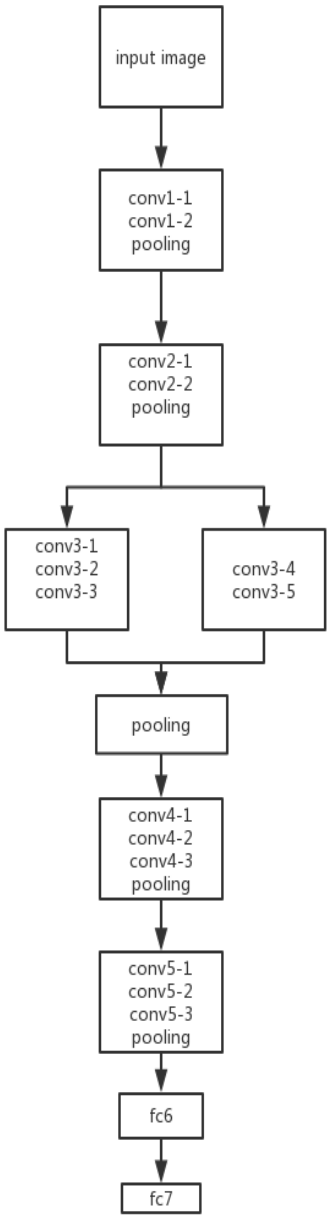

图 5-5　　增加 inception 模块的 VGG19 网络

图 5-6　　增加 Inception 模块结果可视化

第六章　总结与展望

6.1　实验的贡献与优点

本课题使用自己构建的垃圾文本图片数据集，结合 PixelLink 的方法，尝试检测垃圾文本上的所在位置，训练出模型后，可以在一秒之内对文本框进行检测，检测结果较为准确。同时经过参数优化和网络结构更改，改用了更加合理的阈值，同时尝试增加网络深度，增加类 GoogleNet 的 inception 模块，比使用原始的 VGG16 主干的 PixelLink 算法取得了更好的效果。

6.2　实验存在的问题及改进方法

本课题使用 PixelLink 算法，构建了垃圾图片数据集，并进行了多次实验，得到了不错的效果。垃圾上的重要文字基本可以被标注到，但课题仍有很大的改进空间，基本在两个方面，一是数据集方面，二是算法方面。

6.2.1　数据集改进方法

首先从基础来讲，数据集的图片数量还可以继续增加，更多的数据通常可以带来更好的效果和更鲁棒的网络，仅仅有 3655 张训练图片，不能保证一定不会出现过拟合情况，同时由于数据采集时的背景统一为灰色，对实验也有一定影响，如果可以，可以使用算法智能更改垃圾背景，同时增加垃圾图片的数量。第二，在对数据集进行人工标注时，该工作由不同人员完成，每个人负责一定数量的垃圾图片。每个人对于“可识别的文本”的定义不同，如饮料瓶背后的小字和各种食品包装上密密麻麻的说明性文字是否可以被认定为有效文本的这一问题，不同的人有不同的看法。如6-1是一张很容易引起歧义的图片。 同时不同图片上相同尺寸的文本不一定被标注出，当一张图中只有几个小尺寸文本时，往往可以引起标注者的注意，但当一张图中的大部分面积被几个大尺寸文本占据时，小尺寸文本有可能会被漏标。这种“双重标准”会使得算法的效果大打折扣，这也解释了为什么 PixelLink 原作者可以在 ICDAR2015 上取得 F 值高达 83.7 的好成绩，而同样的参数同样的主干网络，我们训练出的模型只有 30 几的 F 值。第三，垃圾数据集的构建不够系统和详细，很多数据集关键参数并没有经过计算，在运行网络时，常常只能使用 ICDAR2015 上的数据，而两个数据集在一些方面完全不同，这也导致了模型效果的下降。例如，在进行过滤噪声的操作时，采用和 ICDAR2015 一样的短边长度小于 10，面积小于 300 进行筛选。这个数值适合 ICDAR2015，但未必适用于垃圾图片文本数据集，相关数值应该经过计算，切实计算出 99% 的文本短边长度和 99% 的文本面积之后，替换原工程中的相关参数。

图 6-1　　有争议的图片举例

6.2.2　算法改进方法

第一，从调节参数上来讲，由于数据集不同，设置阈值和图像输入尺度等参数需要多次实验，才能选出适合垃圾图片文本数据集的相关参数，本课题虽然考虑到了这个问题，但无奈时间有限，同时两个阈值搭配起来至少有 25 种可能，一次次尝试时间不允许。同时由于实验室条件限制，当图片以原尺寸 900*960 输入时，每一次迭代的时间高达 1.5 秒，比尺寸更改到 512*512 的时间长了一倍，即使可以提高各项参数值，时间成本太高，同样时间内输入尺寸为 512*512 的网络可以迭代更多步数，使用更大尺寸输入的意义不大。同时由于在单显卡上进行训练，批尺寸设置为 8，导致算法出现过拟合的可能性较高。如有多显卡训练的条件，增加批尺寸很有可能可以取得更好的效果。第二，在主干网络方面，本课题使用 VGG16 主干网络进行训练，对主干网络的更改也仅限于加大网络深度和增加一些改进模块，效仿 VGG19 和 GoogleNet 的网络结构。在今后的完善中，可以使用其他的卷积神经网络模型，如残差网络等，通常可以得到更好的结果。第三，本课题采用的算法中的特征融合和语义分割方法均为 FCN 结构初次提出时采用的方法，如上采样使用双线性插值，特征融合采用特征图相加等。语义分割相关研究者提出了很多新颖且效果较好的新方法，以后可以尝试用 DeepLab，SegNet 等替换之前的方法。第四，在优化时，本课题沿用了传统的随机梯度下降配合动量的方法，今后可以使用其他高级的优化算法来进行权重优化，也许可以提高训

练进程中的收敛速度。第五，由于算法使用 ICDAR2015 的 IoU 匹配算法进行评估，有些文字区域被多个独立的文本框检测出来，但由于 IoU 匹配算法仅适用于一对一检测，导致众多文本框仅保留一个，对召回率造成了很大的影响。因此 ICDAR2015 的评估算法并不一定可以完全客观地对算法的效果进行评价。

6.3 未来应用展望

本课题的目的为实现垃圾文字识别的第一阶段——垃圾文字检测，搭配第二阶段的文字识别算法，可以识别出垃圾上的文字内容，并通过文字内容推断垃圾类别，从而实现辅助分类网络对难分垃圾进行分类的目的。虽然本课题的实现效果在使用 ICDAR2015 文字识别挑战的评价方法进行评估时，数值较差，但由于数据集不同，将本课题的结果与其他算法在官方数据集上取得的结果比较意义时不大的。从可视化结果来看，本课题成功的检测出了大多数的文本内容，这些已检测出的内容大部分为垃圾文本中最明显，最有指向性的文本。这些文本结合分类任务给的高概率类别，可以完成最终确定垃圾类别的任务。算法的实际应用场合可以为公共区域的智能分拣垃圾桶，垃圾桶内置摄像头，实时对放入垃圾进行拍摄，同时经过简单尺寸更改来压缩数据大小，并将数据送入服务器，服务器同时运行两种算法，一为分类算法，二为本文提供的文字检测算法，通过两个算法的综合，输出垃圾的类别，将类别传回垃圾桶的微处理器，垃圾桶做出反应对垃圾进行自动分类，将垃圾投入其所在类别的垃圾箱内。从拍摄到回传垃圾类别，整个过程可以在几秒内完成，这种系统在公共场合采用后，可以实现对公共场合投入垃圾箱的垃圾准确自动地分类，达到代替初步人工分拣的目的，一定程度上解决了特定场合的垃圾处理分类难，分类慢的问题。

参考文献

[1] 杨薇. 汉语语词在商品品牌名称中的运用 [J]. 语言文字应用, 2000, 4: 49.

[2] 陈阳. 字体设计在平面设计中的重要性研究 [J]. 文艺生活: 中旬刊, 2013 (8): 48–49.

[3] Tian Z, Huang W, He T, *et al*. Detecting text in natural image with connectionist text proposal network [C]. In European conference on computer vision, 2016: 56–72.

[4] Liao M, Shi B, Bai X, *et al*. Textboxes: A fast text detector with a single deep neural network [C]. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[5] Shi B, Bai X, Belongie S. Detecting oriented text in natural images by linking segments [C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2550–2558.

[6] Deng D, Liu H, Li X, *et al*. Pixellink: Detecting scene text via instance segmentation [C]. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[7] Zhou X, Yao C, Wen H, *et al*. EAST: an efficient and accurate scene text detector [C]. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017: 5551–5560.

[8] Li Y, Yu Y, Li Z, *et al*. Pixel-Anchor: A Fast Oriented Scene Text Detector with Combined Networks [J]. arXiv preprint arXiv:1811.07432, 2018.

[9] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述 [J]. 计算机学报, 2017, 40 (6): 1229–1251.

[10] Liu W, Anguelov D, Erhan D, *et al*. Ssd: Single shot multibox detector [C]. In European conference on computer vision, 2016: 21–37.

[11] Karatzas D, Gomez-Bigorda L, Nicolaou A, *et al*. ICDAR 2015 competition on robust reading [C]. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015: 1156–1160.

[12] Lucas S M, Panaretos A, Sosa L, *et al*. ICDAR 2003 robust reading competitions [C]. In Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings., 2003: 682–687.

[13] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014.

[14] Deng J, Dong W, Socher R, *et al*. Imagenet: A large-scale hierarchical image database [C]. In 2009 IEEE conference on computer vision and pattern recognition, 2009: 248–255.

[15] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015: 3431–3440.

[16] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks [C]. In Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011: 315–323.

[17] Kaehler A, Bradski G. Learning OpenCV 3: computer vision in C++ with the OpenCV library [M]. " O'Reilly Media, Inc.", 2016.

[18] Sutskever I, Martens J, Dahl G, *et al*. On the importance of initialization and momentum in deep learning [C]. In International conference on machine learning, 2013: 1139–1147.

[19] He W, Zhang X-Y, Yin F, *et al*. Deep direct regression for multi-oriented scene text detection [C]. In Proceedings of the IEEE International Conference on Computer Vision, 2017: 745–753.

[20] Szegedy C, Liu W, Jia Y, *et al*. Going deeper with convolutions [C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015: 1–9.

致　　谢

这一次的课题，虽然是我较为熟悉的领域，但是对我来说，还是很有挑战性，因为这是我第一次从构建数据库，学习算法，配置环境等一直到训练出模型，并跑出实验结果。毕业设计过程中，出现的很多困难都是我之前从没有遇到的，很多问题并没有在论文中得到体现。多亏了各位同学，老师的悉心教导和热情帮助，终于得到了较为理想的效果。我要向尽心帮助我的各位表示最诚挚的谢意。

首先，我要感谢我的论文指导老师曾明老师。曾明老师是我入门计算机视觉与模式识别的引路人。从论文的选题，实现，到最终撰写完成，曾明老师都给了我悉心的指导与关键性的建议。曾明老师发现新问题的敏锐直觉，夜以继日研究的忘我精神和严谨的学风都值得我认真学习。

其次，我要感谢电气自动化与信息工程学院的各位老师和领导，四年来，他们认真的讲解专业知识，负责的带领我进行实验，使我学到了扎实的基础知识和专业技能。更要忠心的感谢论文答辩组的孟庆浩老师，鲜斌老师，齐俊桐老师，曾明老师和任超老师对我论文各个部分的认真审阅，并提出宝贵的批评与意见，对我的论文完成起到了至关重要的作用。

最后，我要感谢家人对我的关怀和鼓励，以及所有一路上对我帮助和包容的同学和朋友。有了他们的包容和鼓励，我才能安心学习，顺利完成学业。

毕业在即，在今后的学习与工作中，我将始终铭记实事求是的校训，以曾在天津大学就读而骄傲自豪。