

南开大学

本科生毕业论文（设计）

中文题目：低分辨率下的文本定位识别研究与实现

外文题目：Scene Text Detection and Recognition Under
Low Resolution

学 号：1510742

姓 名：陶某某

年 级：2015 级

专 业：计算机科学与技术

系 别：计算机科学与技术

学 院：计算机学院

指导教师：王某 副教授

完成日期：2019 年 5 月

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

这是摘要部分。

关键词：换行与冒号对齐

Abstract

This is abstract part.

Key Words:

目录

摘要	I
Abstract	II
目录	III
第一章 绪论	1
第一节 背景意义	1
第二节 毕设内容	1
第二章 相关工作	2
第一节 节 1	2
2.1.1 子节 1	2
2.1.2 数学公式，上标引用，绘图示例	2
第二节 文本检测识别性能评价方法	3
2.2.1 表格环境	3
2.2.2 项目编号使用	3
2.2.3 算法伪代码	3
第三章 实验结果	5
第一节 数据集	5
第二节 自动评测	5
第三节 图像展示	5
第四章 总结和展望	7
参考文献	8
致谢	9
附录 A	10
A.1 XXX 的实现	10
A.2 自动评测实现	10

第一章 绪论

每一章章节之间说明论文写作逻辑

第一节 背景意义

选题的背景意义。

第二节 毕设内容

新想法 > 算法复现 > 跑模型

可展示 + 工作量（代码量）

可读性强的论文，往往能从摘要看做了什么，从目录看怎么写的。

第二章 相关工作

论文一般从相关研究，实验方法，实验结果，总结展望几个方面展开，不过为了直观明显，建议不要按这种抽象的方式组织，替换成具体工作。

第一节 节 1

2.1.1 子节 1

子节内容。

2.1.2 数学公式，上标引用，绘图示例

Laplacian pyramid^[1] 第 K 层定义：

$$L_K = G_K - UP(G_{K+1}) \otimes filter \quad (2.1)$$

公式按章标号，即章号. 公式号。

绘图先定义结点样式和相对位置关系，然后画线或者箭头，也有丰富的样式可以选择。用 `\ref{pyramid}` 引用，括号里是图像的标签，在图像环境中声明，效果为：如图1所示：

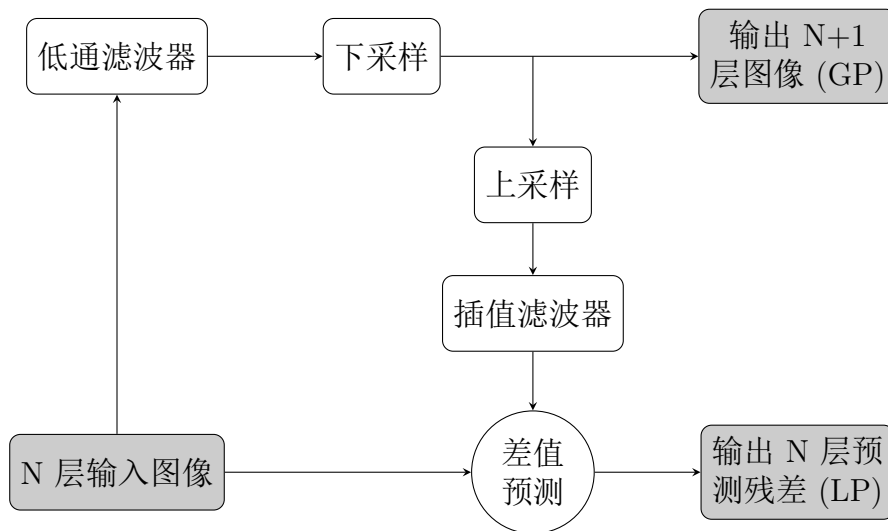


图 1: 高斯金字塔 (GP) 和拉普拉斯金字塔 (LP) 的采样流程

第二节 文本检测识别性能评价方法

2.2.1 表格环境

一种更牛逼的方式是用表格工具生成表格，但我觉得表格工具有点麻烦，不直观，用 excel 制表截图插入更方便，不过这样 caption 显示的是图 num: 而不是表 num:，建议还是用表格环境更好。

真实情况	预测结果	
	正例	反例
正例	TP(真正例)	FN (假反例)
反例	FP(假正例)	TN (真反例)

表 1: 分类混淆矩阵

2.2.2 项目编号使用

- (1) 编号括号、数字的样式都有具体格式，没仔细看
- (2) 项目 2
- (3) 项目 3

2.2.3 算法伪代码

用 algorithm 环境制作算法伪代码非常方便，基本的语法是输入输出，statement 语句，函数块，控制语句等。基本实现效果如算法1。

算法 1 基于小波空间域的超分辨率算法

输入: 低分辨率图像 lr , 超分辨率倍数 $Factor$

输出: 高分辨率图像 hr

```

1: function Super-resolution( $lr, Factor$ )
2:    $hr_t \leftarrow \text{UpSampling}(lr, Factor)$ 
3:    $hr_g \leftarrow \text{GaussianFilter}(hr_t)$ 
4:   for ( $i=0; i<2; i++$ ) do
5:      $lr_i \leftarrow \text{DownSampling}(hr_g, Factor)$ 
6:      $\text{ReconstructionError}(i) = lr - lr_i$ 
7:      $Eh(i) \leftarrow \text{UpSampling}(\text{ReconstructionError}(i))$ 
8:      $hr(i+1) = Eh(i) + hr(i)$ 
9:   end for
10:   $hr \leftarrow hr_2$ 
11: end function

```

关于 tikz 绘图包，可以像 ps 一样设置图层绘制更加复杂的图，不过用到的只有基本要素，下面再给一例：

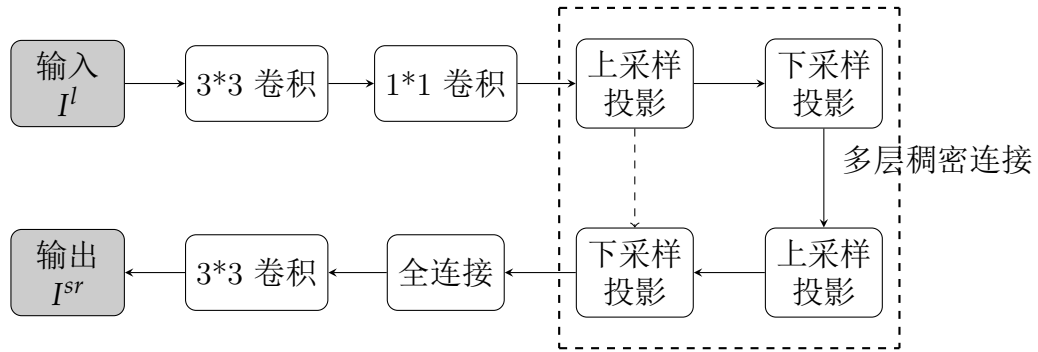


图 2: DBPN 网络结构

第三章 实验结果

论文写作逻辑。

第一节 数据集

数据集多多益善，也可以作为展望中的一部分。

第二节 自动评测

在数据处理时。可以写一个自动评测 python 脚本来进行数据分析，例如用 python os 包 (文件目录访问读写以及代码中 cmd 指令的执行)，skimage 包 (提供丰富图像操作以及计算图像评价指标)，CSV 和 Pandas 包 (保存分析处理评价指标，数值分析)，matplotlib 包绘制结果对比图，实现的自动评测脚本过程如图3：在得到 CSV 文件后可以直接用 office Excel 进行数据分析，通过自动评

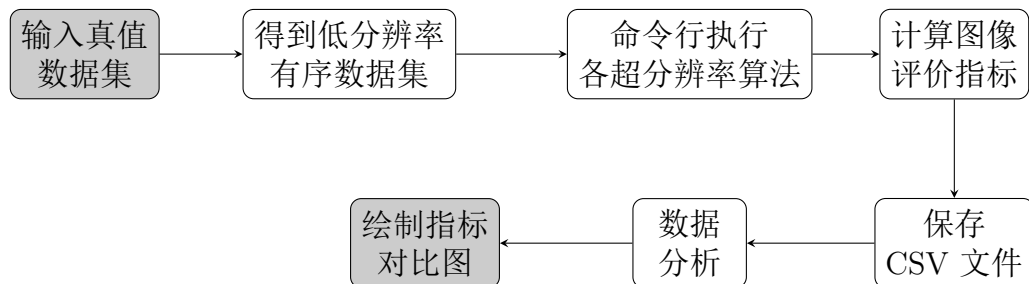


图 3: 自动评测流程

测可以减少重复输入测试，大大减少模型运行，数据测得，结果分析的时间，提高工作效率。

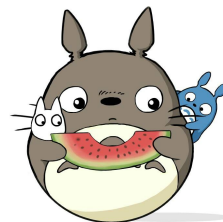
第三节 图像展示



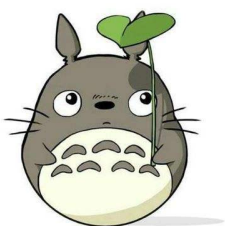
吃萝卜的龙猫



头上有老鼠的龙猫



吃西瓜的龙猫



雄性龙猫



雌性龙猫



大龙猫

图 4: 各种龙猫

第四章 总结和展望

总结论文得到的结论成果，存在的问题和可能的解决办法。

参考文献

- [1] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,”
IEEE Transactions on communications, vol. 31, no. 4, pp. 532–540, 1983.

致谢

这里是致谢部分：

毕设是学习的过程，感谢这次毕设机，让我看到了自己的不足，同时也让我学到了很多东西。

附上代码体现代码工作量 xxx 的实现如下 (c++ opencv):

```
#include <iostream>

cout<<"helloworld"<<endl;

return 0;

}
```

评测部分主要函数简要实现如下 (python):

```
#调用时使用process(gtpath,'wsd','raisr',.....)
def process(gtpath,*args):
#对真值图像进行下采样和重编号命名:
    paths = glob.glob(gtpath+'/*.jpg')
    paths.sort()
    for file in paths:
        (filepath, tempfilename) = os.path.split(file)
        (filename, extension) = os.path.splitext(tempfilename)
        img = cv2.imread(file, 1)
        factor=2
        size = (int(img.shape[1]/factor),int(img.shape[0]/factor))
        shrink = cv2.resize(img, size, interpolation=cv2.INTER_LINEAR)
        cv2.imwrite("LR/"+filename+".jpg",shrink)
        for i in range(len(args)):
#cmd执行python脚本
            os.popen('python ' + args[i]+'/' +args[i] + '.py')
#评测对比PSNR,输入为真值路径以及图像数量, SSIM方法类似
def evalpsnr(gtpath, num):
    out = open('psnr.csv', 'w', newline='')
    csv_write = csv.writer(out, dialect='excel')
    csv_write.writerow(['IMG#', 'WSD', 'RAISR','SRGAN','DBPN'])
    gt = skimage.io.imread(gtpath + '/' + str(j) + '.jpg')
    for j in range(num):
        wsd_img = skimage.io.imread('wsd_result'+str(j) + '.jpg')
        wsd_psnr = skimage.measure.compare_psnr(gt, wsd_img)
        #ssim = skimage.measure.compare_ssim(gt, test, multichannel=True)
        (.....计算其他方法method_psnr)
        csv_write.writerow([j, wsd_psnr,raisr_psnr,srgan_psnr,dbpn_psnr])
    out.close()
#使用pandas库对CSV求最大值, 最小值, 平均值等聚合统计
def analyse(path):
    df = pandas.read_csv(path+"/psnr.csv")
```

```
print(df)
#求某一方法处理后下psnr最佳和最差对应图像编号和psnr均值
print(df['WSD'].idxmin(),df['WSD'].idxmax(),df['WSD'].mean())
#matplotlib.plot绘制psnr对比折线图,横坐标为图像编号,纵坐标为PSNR
#不同方法标注不同颜色
x = np.linspace(1, num, num)
print(x)
l1 = plt.plot(x, df['WSD'], 'blue')
l2 = plt.plot(x, df['RAISR'],'purple')
l3 = plt.plot(x, df['SRGAN'], 'green')
l4 = plt.plot(x, df['DBPN'],'yellow')
plt.title('PSNR')
plt.xlabel('IMAGE')
plt.ylabel('PSNR')
plt.legend((l1[0], l2[0],l3[0],l4[0]), ('WSD',
    'RAISR', 'SRGAN', 'DBPN'))
plt.show()
```
