

# 机器学习与人工智能-实验报告

17211239 黄铭杰

## 一、基本信息：

1. 相关论文：DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks
2. Dataset 来源：Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring
3. 作者：Orest Kupyn et al.
4. 原作者 Git: <https://github.com/KupynOrest/DeblurGAN>
5. 我的修改 Git: <https://github.com/dadapapahhh/DeblurGAN>

## 二、实验目的

了解和学习机器学习实际应用的编程实现与应用

## 三、实验原理

### 1. 文章研究的问题：

图像的模糊过程可以用以下的映射来表示： $IB = k(M) * IS + N$ （ $IB$  是模糊图像、 $IS$  为清晰图像、 $k$  为“模糊核”、 $N$  为加性噪声）。本文致力于解决的是盲解模糊的问题：即是在模糊核未知的情况下由模糊图像  $IB$  得到清晰图像  $IS$ 。

### 2. 文章的方法：

本文将清晰和模糊图像之间的转化当作 image-image translation 问题的一种，采用生成对抗网络（GAN）来由模糊图像生成清晰的图像。

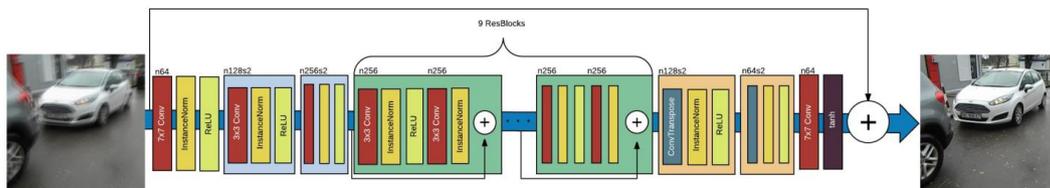
### 3. GAN:

即是生成对抗网络（Generative adversarial networks），由 Goodfellow 等人提出。GAN 由生成器和判别器两部分构成，在本文中生成器的任务是接收一个模糊图片然后生成一个对应的清晰的图片，生成器要努力产生一个很“真”的图片来骗过判别器使其无法分

辨图片是生成器生成的清晰图片还是本来就真实存在的清晰图片；判决器的任务是尽力得分辨出哪些图片是生成器生成的“假”清晰图片哪些是本来就存在的“真”清晰图片。

生成器和判决器之间形成了一种对抗关系，在进行数据集的训练的时候，它们依靠这种对抗关系互相学习从而提高了双方的能力，达到了训练的目的。

其中生成器的结构如下：



#### 4. Gopro 数据集：

数据集的作者采用 240fps 的高速摄像机获取图片，由于摄影帧率高曝光时间短不容易模糊可以得到清晰的图片；将相邻的几帧图片叠加在一起相当于等效得增加了曝光时间从而可以合成得到模糊的图片，这样就产生了“模糊-对应的清晰图片”的数据集。

### 四、实验环境配置

1. 选用安装了 NVIDIA GPU 的计算机 + 安装 CUDA
2. 安装 Pytorch、以及其它所需的 python 模组
3. 下载论文采用的 GoPro dataset
4. 将代码 pull 到本地
5. Windows 10 x64 + PyCharm

### 五、实验内容

1. 代码按照 comment 里的帮助内容做相关的修改使其得以在我的环境下运行：

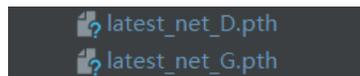
修改	修改
dadapapahh committed 15 hours ago	@@ -1,3 +1,4 @@
Merge pull request #131 from rahu/vign...	1 +import torch
1111frame28.jpg	2 from torch.autograd import Variable
GUI.py	3 from collections import OrderedDict
GUI1.py	4 import util.util as util
Update	@@ -33,8 +34,11 @@ class testModel(BASEMODEL):
Unknown committed Mar 24, 2019	self.image_paths = input['A_paths']
checkpoints/experiment...latest_net_D.pth	
Update	def test(self):
Unknown committed Mar 19, 2019	self.real_A = Variable(self.input_A, volatile=True)
data/base_dataset.py	self.fake_B = self.netG.forward(self.real_A)
data/custom_dataset_data_loader.py	# self.real_A = Variable(self.input_A, volatile=True)
Update	#self.fake_B = self.netG.forward(self.real_A)
Unknown committed Mar 19, 2019	with torch.no_grad():
Update Version	self.real_A = Variable(self.input_A)
Jonathan committed Mar 15, 2019	self.fake_B = self.netG.forward(self.real_A)
Updates	
Jonathan committed Mar 15, 2019	# get image paths
The Dropbox weights link.	def get_image_paths(self):
Rahul Vigneshwaran K committed Feb 8, 2...	
Merge pull request #105 from rimchang...	

2. 按照 readme.md 提示执行相关的代码进行训练:

```
python datasets/combine_A_and_B.py --fold_A D:\A00ML\blurred_sharp\blurred_sharp\blurred --fold_B D:\A00ML\blurred_sharp\blurred_sharp\sharp --fold_AB D:\A00ML\DeblurGAN\mydata
```

```
python train.py --dataroot D:\A00ML\1\DeblurGAN\train_data_self --learn_residual --resize_or_crop crop --fineSize CROP_SIZE 720
```

得到相关的模型文件:



3. 用训练好的模型文件, 运行以下代码输入我们的模糊图片, 然后进行解模糊测试, 得到输出结果见下模块模块结果分析。

```
python test.py --dataroot D:\A00ML\1\DeblurGAN\Test_picture\test --model test --dataset_mode single --learn_residual
```

## 六、结果分析



(a) (b)

(c) (d)

图 2-1: a、c 为模糊模糊图片, b 和 d 分别为解模糊后对应的生成图片, 其中 a 来自 gopro 数据集 c 为我自行拍摄 (1/20s iso:80 on vivo iqoo) 的图片

从观感上两张模糊图片在经过处理后清晰度都有所提高，其中来自 gopro 数据集的 a 图的变化较为明显，清晰度提高较为显著；我自己拍摄的模糊图 c 解模糊的效果一般，与图 a 对比鲜明。

经过多番比较（还有其他的图片碍于空间没有放上来），分析得该方法在用作训练的 gopro 数据集上的解模糊效果很好，甚至保留了很多的细节，能还原大部分的内容，在我自己拍摄的照片上实现效果略差；我认为可能是我拍摄的照片与测试数据集之间存在一定差异的原因，同时也说明训练出来的模型的泛化能力较差，在实际应用中我认为针对多种场景采用不同的训练模型可能会极大改善效果，碍于机器算力有限，我并没有实施，只是提出设想。