**一、环境配置：**

* Python3
* Keras
* sklearn，opencv，matplotlib, numpy
* GPU

**二、代码使用：**

**主要包括三个模块：**

**1. 使用VerifyCodeUtils.java通过调参，生成伪验证码图片；**

**2. 使用生成式对抗网络对真实验证码进行预处理（降噪操作）**。该部分代码保存在**tensorflow-cyclegan-master**文件夹中。

* **model**文件夹中为cycleGAN 结构代码；
* **build\_dataset.py**：对图片进行预处理，输入两组图片，输出.npy数据集文件用于后续训练;
* **cyclegan.py**: cyclegan模型代码;
* **example.py**: 主程序，输入两组npy数据集文件，输出结果对照图和模型。

**3. 使用CNN网络对降噪后的验证码进行识别。**该部分代码保存在**CNN\_keras-master/core**文件夹中。

* **utils.py**: 预处理，轮廓检测提取单个验证码字符，输入为3通道图片，如果是单通道，可加上代码newimage= np.expand\_dims(newimage, axis=2)转化为三通道,输出为提取的每个验证码字符的可视化；
* **train.py**: 模型训练，输入为图片文件夹地址和模型存放地址，输出为准确率及loss和得到的模型；
* **predict**：用于测试，输入为图片的地址和已经训练好的模型，输出为相应的图片预测结果。

**三、实验结果：**

* **共标注wiki验证码1500张。**
* **GAN模型：**使用975张作为训练集；降噪后图片保存在tensorflow-cyclegan-master/img12700文件夹。
* **CNN模型：**训练集：1152张，验证集：128张，测试集：143张

训练轮数：50

识别准确率：单个字符准确率: **98.13%**，整体准确率: **84.35%**

