# 针对正方教务验证码最最最最简易的一个 OCR 字库训练

作者：lonelyant

原文链接：https://ld246.com/article/1557729391582

来源网站：链滴

# 代码

废话不多说，上来先贴代码。

代码原型来自于互联网，时间很久了已经忘了从哪里copy的了，侵删。

```
package ocr;

import com.sun.deploy.net.HttpResponse;
import org.apache.http.HttpEntity;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;

import javax.imageio.ImageIO;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.*;
import java.util.*;
import java.util.List;

public class Test {
    private static Map<BufferedImage, String> trainMap = null;
    private static int index = 0;
    //验证码地址
    public static String getImageUrl = "http://221.232.159.27/CheckCode.aspx";
    public static String srcPath = "img/";
    public static String trainPath = "train/";
    public static String tempPath = "temp/";

    public static int isBlue(int colorInt) {
        Color color = new Color(colorInt);
        int rgb = color.getRed() + color.getGreen() + color.getBlue();
        if (rgb == 153) {
            return 1;
        }
        return 0;
    }

    public static int isBlack(int colorInt) {
        Color color = new Color(colorInt);
        if (color.getRed() + color.getGreen() + color.getBlue() <= 100) {
            return 1;
        }
        return 0;
    }

    public static int isWhite(int colorInt) {
        Color color = new Color(colorInt);
        if (color.getRed() + color.getGreen() + color.getBlue() > 600) {
            return 1;
        }
    return 0;
}
/** * 去除背景，二值化 * @param picFile * @return * @throws Exception */
public static BufferedImage removeBackgroud(String picFile)
        throws Exception {
    BufferedImage img = ImageIO.read(new File(picFile));
    img = img.getSubimage(5, 1, img.getWidth()-5, img.getHeight()-2);
    img = img.getSubimage(0, 0, 50, img.getHeight());
    int width = img.getWidth();
    int height = img.getHeight();
    for(int x=0; x<width; x++){
        for(int y=0; y<height; y++){
            if(isBlue(img.getRGB(x, y)) == 1){
                img.setRGB(x, y, Color.BLACK.getRGB());
            }else{
                img.setRGB(x, y, Color.WHITE.getRGB());
            }
        }
    }
    return img;
}
/** * 按自己的规则分割验证码 * @param img * @return * @throws Exception */
public static List<BufferedImage> splitImage(BufferedImage img)
        throws Exception {
    List<BufferedImage> subImgs = new ArrayList<BufferedImage>();
    int width = img.getWidth()/4;
    int height = img.getHeight();
    subImgs.add(img.getSubimage(0, 0, width, height));
    subImgs.add(img.getSubimage(width, 0, width, height));
    subImgs.add(img.getSubimage(width*2, 0, width, height));
    subImgs.add(img.getSubimage(width*3, 0, width, height));
    return subImgs;
}
/** * 载入训练好的字摸 * @return * @throws Exception */
public static Map<BufferedImage, String> loadTrainData() throws Exception {
    if (trainMap == null) {
        Map<BufferedImage, String> map = new HashMap<BufferedImage, String>();
        File dir = new File("train");
        File[] files = dir.listFiles();
        for (File file : files) {
            if (file.getName().startsWith("."))continue;
            map.put(ImageIO.read(file), file.getName().charAt(0) + "");
        }
        trainMap = map;
    }
    return trainMap;
}
/** * 识别分割的单个字符 * @param img * @param map * @return */
public static String getSingleCharOcr(BufferedImage img,
                                      Map<BufferedImage, String> map) {
    String result = "#";
    int width = img.getWidth();
    int height = img.getHeight();
    int min = width * height;
    for (BufferedImage bi : map.keySet()) {
        int count = 0;
        if (Math.abs(bi.getWidth()-width) > 2)
            continue;
        int widthmin = width < bi.getWidth() ? width : bi.getWidth();
        int heightmin = height < bi.getHeight() ? height : bi.getHeight();
        Label1: for (int x = 0; x < widthmin; ++x) {
            for (int y = 0; y < heightmin; ++y) {
                if (isBlack(img.getRGB(x, y)) != isBlack(bi.getRGB(x, y))) {
                    count++;
                    if (count >= min)
                        break Label1;
                }
            }
        }
        if (count < min) {
            min = count;
            result = map.get(bi);
        }
    }
    return result;
}
/** * 验证码识别 * @param file 要验证的验证码本地路径 * @return * @throws Exception */
public static String getAllOcr(String file) throws Exception {
    BufferedImage img = removeBackgroud(file);
    List<BufferedImage> listImg = splitImage(img);
    Map<BufferedImage, String> map = loadTrainData();
    StringBuilder result = new StringBuilder();
    for (BufferedImage bi : listImg) {
        result.append(getSingleCharOcr(bi, map));
    }
    ImageIO.write(img, "PNG", new File("result/" + result + ".png"));
    return result.toString();
}

/** * 下载验证码 * @param url */
public static void downloadImage(String url) {
    CloseableHttpClient httpClient = HttpUtils.getHttpClient();
    for(int i=0; i<10; i++){
        HttpGet getMethod = new HttpGet(url);
        CloseableHttpResponse response = null;
        try {
            response = httpClient.execute(getMethod);
            if("HTTP/1.1 200 OK".equals(response.getStatusLine().toString())){
                HttpEntity entity = response.getEntity();

                InputStream is = entity.getContent();
                OutputStream os = new FileOutputStream(new File(srcPath+i+".png"));
                int length = -1;
                byte[] bytes = new byte[1024];
                while((length = is.read(bytes)) != -1){
                    os.write(bytes, 0, length);
                }
                os.close();
                }
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    /** * 机器训练 * @throws Exception */
    public static void trainData() throws Exception {
        File dir = new File("img/");
        File[] files = dir.listFiles();
        for (File file : files) {
            if (file.getName().startsWith("."))continue;
            System.out.println(files.length + file.getName());
            BufferedImage img = removeBackgroud("img/" + file.getName());
            List<BufferedImage> listImg = splitImage(img);
            if (listImg.size() == 4) {
                for (int j = 0; j < listImg.size(); ++j) {
                    ImageIO.write(listImg.get(j), "PNG", new File("temp/"
                            + file.getName().charAt(j) + "-" + (index++)
                            + ".png"));
                }
            }
        }
    }

    public static void testOCR() throws Exception{
        for (int i = 0; i < 10; ++i) {
            String file = "img/" + i + ".png";
            System.out.println(file);
            String text = getAllOcr(file);
            System.out.println(i + ".png = " + text);
        }
    }

    /** * @param args * @throws Exception */
    public static void main(String[] args) throws Exception {
        //downloadImage(getImageUrl);
        //trainData();
        //testOCR();
    }
}
```

代码中用到的HttpClient的代码也顺手贴上来吧

```
package ocr;

import org.apache.http.config.Registry;
import org.apache.http.config.RegistryBuilder;
import org.apache.http.config.SocketConfig;
import org.apache.http.conn.socket.ConnectionSocketFactory;
import org.apache.http.conn.socket.PlainConnectionSocketFactory;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.conn.ssl.TrustStrategy;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.DefaultHttpRequestRetryHandler;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.impl.conn.PoolingHttpClientConnectionManager;
import org.apache.http.ssl.SSLContextBuilder;

/**
 * Created by Ant on 2017/8/15.
 */
public class HttpUtils {

    public synchronized static CloseableHttpClient getHttpClient() {
        CloseableHttpClient httpClient;
        PoolingHttpClientConnectionManager cm = getCM();
        httpClient = HttpClients.custom()
                .setConnectionManager(cm)
                .setConnectionManagerShared(true)
                .setRetryHandler(new DefaultHttpRequestRetryHandler(0, false))
                .build();
        return httpClient;
    }

    private static PoolingHttpClientConnectionManager getCM(){
        PoolingHttpClientConnectionManager cm = null;
        try {
            SSLContextBuilder builder = new SSLContextBuilder();
            builder.loadTrustMaterial(null, (TrustStrategy) (x509Certificates, s) -> true);
            SSLConnectionSocketFactory sslsf = new SSLConnectionSocketFactory(builder.build(),
ew String[]{"SSLv2Hello", "SSLv3", "TLSv1", "TLSv1.2"}, null, NoopHostnameVerifier.INSTANCE)

            Registry<ConnectionSocketFactory> registry = RegistryBuilder.<ConnectionSocketFac
ory>create()
                    .register("http", new PlainConnectionSocketFactory())
                    .register("https", sslsf)
                    .build();
            cm = new PoolingHttpClientConnectionManager(registry);
            cm.setMaxTotal(200);//max connection
            cm.setDefaultMaxPerRoute(50);
            cm.setDefaultSocketConfig(SocketConfig.custom().setSoTimeout(15 * 1000).build());
        } catch (Exception e) {
            e.printStackTrace();
        }
        return cm;
    }

}
```

# 大致流程

训练的大概思路就是，从验证码服务器拉取验证码图片，人工识别验证码，去背景切片后加入字库。

## 拉取验证码图片

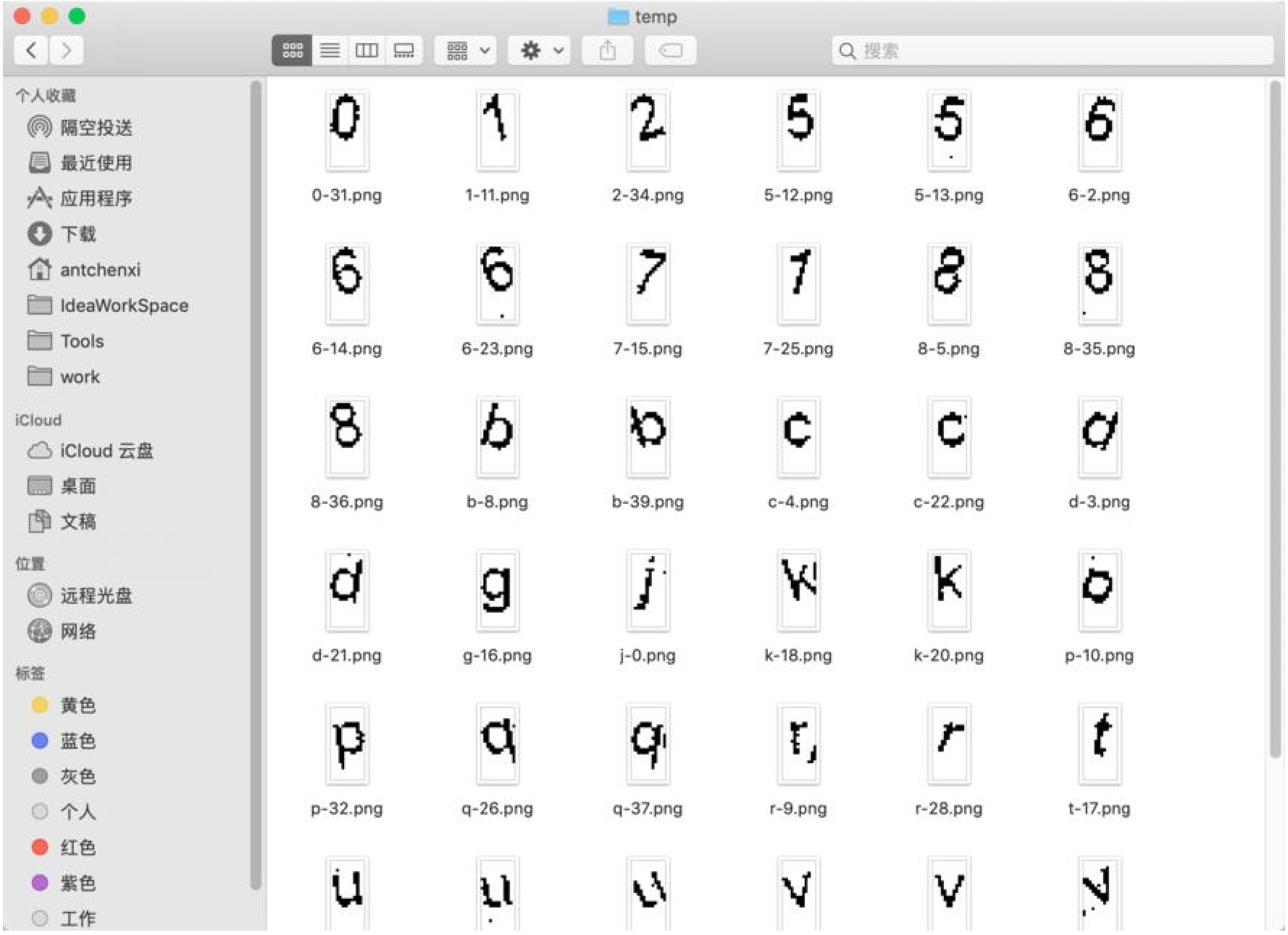
新建 img temp result train 四个文件夹
运行 downloadImage方法，验证码会保存到 img 文件夹下

## 人工识别验证码

分别将文件重命名为验证码内容，对于不能准确确定的验证码，建议删除。

## 去背景、切片

运行 trainData 方法，temp 文件夹中会出现如下切片

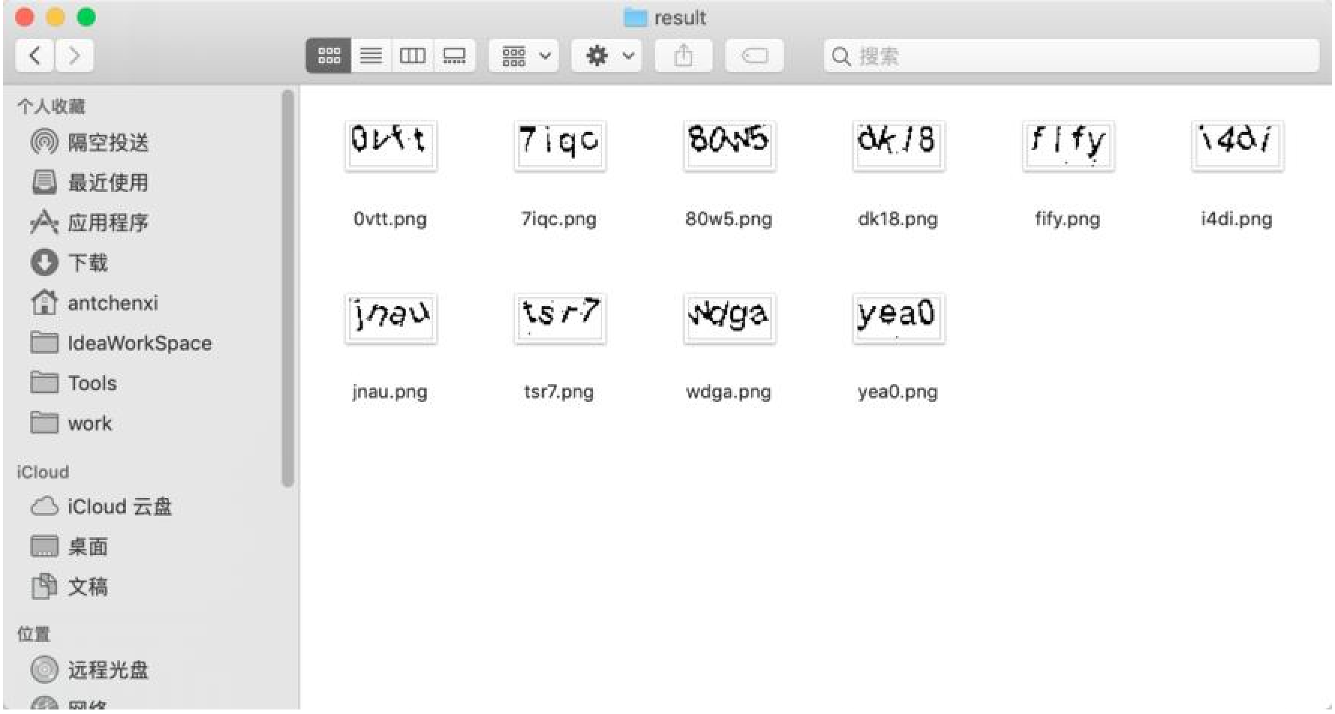


大概看一眼，把类似于p-10这种不太准确的删掉，之后将所有文件移入 train 文件夹，完成一轮训练。

## 验证

以上流程多走几遍，当字库达到一定规模的时候就可以验证一下字库的准确率了

运行 testOCR 方法，在 result 文件夹下可以看到输出结果



可以看到，正确率尚可。

这套字库需要的可以直接拿去使用 train.zip

关于字库的使用，可以参考我的另一篇文章 正方教务系统Java免验证码登录抓取成绩

原文链接：针对正方教务验证码最最最最简易的一个 OCR 字库训练