



- 首页
- Web开发
- Windows开发
- 编程语言
- 数据库技术
- 移动平台
- 系统服务
- 微信
- 设计
- 布布扣
- 其他
- 数据分析

首页 > 其他 > 详细

请输入关键词

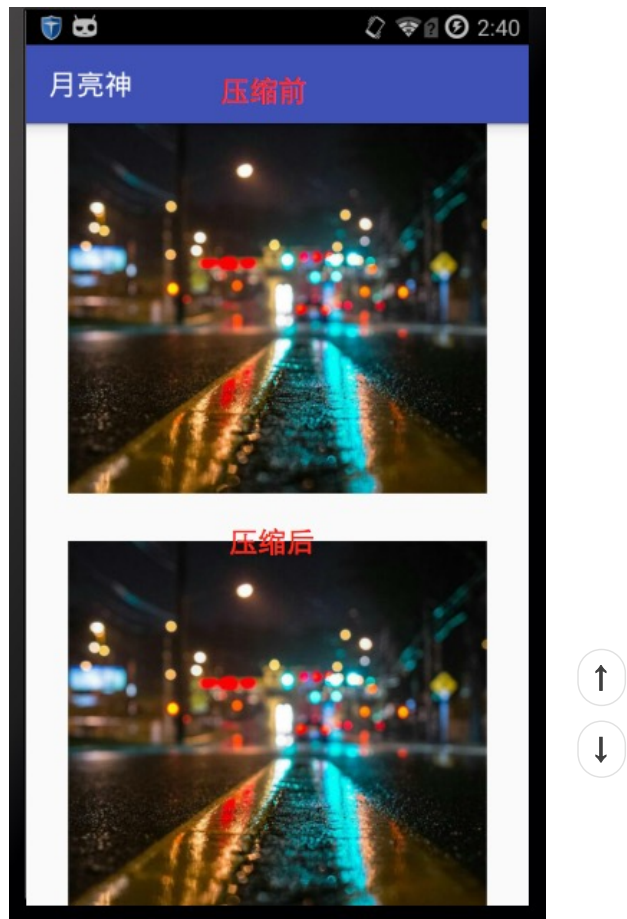
搜索

功能这么齐全的图片压缩类，还有谁？

时间：2015-12-04 18:45:36    阅读：59    评论：0    收藏：0    [\[点我收藏+\]](#)

标签：des android class log com http si 使用 it

效果图：



压缩日志

com.pengkv.moon I/--->: 原尺寸:1215\*1080  
com.pengkv.moon I/--->: 最终压缩比例:3倍/新尺寸:405\*360

工具特点

- \* 可以解析单张图片
- \* 可以解析多张图片
- \* 处理了压缩过程中OOM
- \* 处理了部分手机照片旋转问题
- \* 压缩后存储在缓存中，并可以清理
- \* 压缩后返回缓存路径，方便上传

分享档案

[更多>](#)

- 2016年01月21日 (4123)
- 2016年01月20日 (2689)
- 2016年01月19日 (1727)
- 2016年01月18日 (1394)
- 2016年01月17日 (1079)
- 2016年01月16日 (3369)
- 2016年01月15日 (2283)
- 2016年01月14日 (1538)
- 2016年01月13日 (1905)
- 2016年01月12日 (1563)

文章周排行

[更多>](#)

1. 利用Wireshark截取数据包，并对数据包进行解析 2014-05-10
2. 史上最详细的CocoaPods安装教程 2014-10-23
3. solr的schema官方介绍 2014-10-24
4. 大数据竞赛平台——Kaggle 入门 2014-12-15
5. zepto的tap事件的点透问题的几种解决方案 2015-03-02
6. Androd Toolbar 的简单使用（转） 2015-03-03
7. Bootstarp学习（二十三）模态弹出框（Modals） 2015-03-04
8. Robotium自动化测试框架实用教程（图） 2015-03-14
9. 十六进制转八进制 2015-03-31
10. 常用浏览器user\_agent大全 2015-05-10

最新新闻

[更多](#)

1. 三星Gear VR印度正式发布 售价约合人民币800元 2016-01-21
2. 英国核潜艇还在运行Windows XP 2016-01-21
3. 这架无人机可载重61kg 创造了吉尼斯世界纪录 2016-01-21
4. 微软：iOS应用移植到Windows系统So easy 2016-01-21
5. 北京站Wi-Fi全覆盖 盒饭自动贩售微信支付 2016-01-21
6. 央视记者体验零下39℃ 开水瞬间化白雾 冻梨轻松变铁锤 2016-01-21
7. 匹凸匹去年预亏2000万元 业务推进不理想 2016-01-21

- \* 可以从缓存路径读取出Bitmap，方便展示
- \* 封装在2个类里，方便调用

6-01-21

8. 新西兰最大的银行仍在运行Windows XP 2016-

01-21

9. 华为荣耀Holly 2 Plus意外曝光 2016-01-21

10. Fender宣布推出自己的耳塞式耳机 2016-01-21

## 使用方法

```
ImageCompressUtil.compressImageList(this, photos, new ImageCompressUtil.ProcessImg
ListCallBack() {
    @Override
    public void compressSuccess(final List<String> imgList) {
        //imgList为压缩后图片的路径
    }
});
```

## 压缩工具类

```
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Environment;
import android.text.TextUtils;
import android.util.Log;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import utils.ImageRotateUtil;

/**
 * Created by pengkv on 15/12/2.
 * 图片压缩工具类
 */
public class ImageCompressUtil {

    private static List<String> mImageList = new ArrayList<>(); // 临时图片集合
    private static String mImagePath = ""; // 单个临时图片
    public static String cachePath = "";
    public static int reqWidth = 320;
    public static int reqHeight = 480;

    //压缩单张图片方法
```

```

public static void compressImage(final Context ctx, final String filePath, final ProcessIm
gCallback callBack) {
    mImagePath = ""; //清空路径

    new Thread(new Runnable() {
        @Override
        public void run() {
            //如果路径是图片，则进行压缩
            if (isImage(filePath)) {
                mImagePath = compress(ctx, filePath);
            }
            callBack.compressSuccess(mImagePath);
        }
    }).start();
}

```

*//压缩图片集合方法*

```

public static void compressImageList(final Context ctx, final List<String> fileList, final Pr
ocessImgListCallBack callBack) {
    mImageList.clear(); //清空集合
    if (fileList == null || fileList.isEmpty()) {
        callBack.compressSuccess(mImageList);
        return;
    }

    new Thread(new Runnable() {
        @Override
        public void run() {
            String tempPath = "";
            for (String imagePath : fileList) {
                if (isImage(imagePath)) {
                    tempPath = compress(ctx, imagePath);
                    mImageList.add(tempPath);
                }
            }
            callBack.compressSuccess(mImageList);
        }
    }).start();
}

```

*//图片压缩的方法*

```

public static String compress(Context ctx, String filePath) {

    if (TextUtils.isEmpty(filePath))

```

```
return filePath;

File file = new File(filePath);
if (!file.exists())//判断路径是否存在
    return filePath;

if (file.length() < 1)//文件是否为空
    return null;

File tempFile = getDiskCacheDir(ctx);
String outImagePath = tempFile.getAbsolutePath(); // 输出图片文件路径

int degree = ImageRotateUtil.getBitmapDegree(filePath); // 检查图片的旋转角度

//谷歌官网压缩图片
final BitmapFactory.Options options = new BitmapFactory.Options();
options.inJustDecodeBounds = true;
BitmapFactory.decodeFile(filePath, options);
options.inSampleSize = calculateInSampleSize(options, reqWidth, reqHeight);
options.inJustDecodeBounds = false;
Bitmap bitmap = BitmapFactory.decodeFile(filePath, options);

// 旋转:这步处理主要是为了处理三星手机拍的照片
if (degree > 0) {
    bitmap = ImageRotateUtil.rotateBitmapByDegree(bitmap, degree);
}

// 写入文件
FileOutputStream fos;
try {
    fos = new FileOutputStream(tempFile);
    bitmap.compress(Bitmap.CompressFormat.JPEG, 70, fos);
    fos.flush();
    fos.close();
    bitmap.recycle();
} catch (FileNotFoundException e) {
    e.printStackTrace();
    return filePath;
} catch (Exception e) {
    e.printStackTrace();
    return filePath;
}
```

```

    return outImagePath;
}

/**
 * 计算压缩比例值
 * 按照2、3、4...倍压缩
 *
 * @param options 解析图片的配置信息
 * @param reqWidth 所需图片压缩尺寸最小宽度
 * @param reqHeight 所需图片压缩尺寸最小高度
 * @return
 */
public static int calculateInSampleSize(BitmapFactory.Options options, int reqWidth, int
reqHeight) {
    final int picheight = options.outHeight;
    final int picwidth = options.outWidth;
    Log.i("--->", "原尺寸:" + picwidth + "*" + picheight);

    int targetheight = picheight;
    int targetwidth = picwidth;
    int inSampleSize = 1;

    if (targetheight > reqHeight || targetwidth > reqWidth) {
        while (targetheight >= reqHeight && targetwidth >= reqWidth) {
            inSampleSize += 1;
            targetheight = picheight / inSampleSize;
            targetwidth = picwidth / inSampleSize;
        }
    }

    Log.i("--->", "最终压缩比例:" + inSampleSize + "倍/新尺寸:" + targetwidth + "*" + targ
etheight);
    return inSampleSize;
}

//图片集合压缩成功后的回调接口
public interface ProcessImgListCallBack {
    void compressSuccess(List<String> imgList);
}

//单张图片压缩成功后的回调接口
public interface ProcessImgCallBack {

```

```
void compressSuccess(String imgPath);
}

/**
 * 获取文件后缀名
 *
 * @param fileName
 * @return 文件后缀名
 */
public static String getFileType(String fileName) {
    if (!TextUtils.isEmpty(fileName)) {
        int typeIndex = fileName.lastIndexOf(".");
        if (typeIndex != -1) {
            String fileType = fileName.substring(typeIndex + 1).toLowerCase();
            return fileType;
        }
    }
    return "";
}

/**
 * 判断是否是图片
 *
 * @param fileName
 * @return 是否是图片类型
 */
public static boolean isImage(String fileName) {
    String type = getFileType(fileName);
    if (!TextUtils.isEmpty(type) && (type.equals("jpg") || type.equals("gif")
        || type.equals("png") || type.equals("jpeg")
        || type.equals("bmp") || type.equals("wbmp")
        || type.equals("ico") || type.equals("jpe")))) {
        return true;
    }
    return false;
}

/**
 * 将压缩后的图片存储在缓存中
 */
public static File getDiskCacheDir(Context ctx) {
    if (Environment.MEDIA_MOUNTED.equals(Environment.getExternalStorageState())
        || !Environment.isExternalStorageRemovable()) {

```

```
        cachePath = ctx.getExternalCacheDir().getPath();
    } else {
        cachePath = ctx.getCacheDir().getPath();
    }

    String uniqueName = System.currentTimeMillis() + "_tmp.jpg";
    return new File(cachePath + File.separator + uniqueName);
}

/**
 * 清理缓存文件夹
 */
public static void clearCache(Context ctx) {
    File file = new File(cachePath);
    File[] childFile = file.listFiles();
    if (childFile == null || childFile.length == 0) {
        return;
    }

    for (File f : childFile) {
        f.delete(); // 循环删除子文件
    }
}

/**
 * 从图片路径读取图片
 *
 * @param imagePath
 * @return
 */
private Bitmap decodeFile(String imagePath) {
    Bitmap bitmap = null;
    try {
        File file = new File(imagePath);
        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inJustDecodeBounds = false;
        FileInputStream fis = new FileInputStream(file);
        bitmap = BitmapFactory.decodeStream(fis, null, options);
        fis.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return bitmap;
}
```

```
}
```

## 图片旋转工具类

```
import android.graphics.Bitmap;
import android.graphics.Matrix;
import android.media.ExifInterface;

import java.io.IOException;

/**
 * Created by pengkv on 15/12/2.
 * 图片旋转工具类
 */
public class ImageRotateUtil {

    /**
     * 读取图片的旋转的角度
     *
     * @param path 图片绝对路径
     * @return 图片的旋转角度
     */
    public static int getBitmapDegree(String path) {
        int degree = 0;
        try {
            // 从指定路径下读取图片，并获取其EXIF信息
            ExifInterface exifInterface = new ExifInterface(path);
            // 获取图片的旋转信息
            int orientation = exifInterface.getAttributeInt(ExifInterface.TAG_ORIENTATION,
                    ExifInterface.ORIENTATION_NORMAL);
            switch (orientation) {
                case ExifInterface.ORIENTATION_ROTATE_90:
                    degree = 90;
                    break;
                case ExifInterface.ORIENTATION_ROTATE_180:
                    degree = 180;
                    break;
                case ExifInterface.ORIENTATION_ROTATE_270:
                    degree = 270;
                    break;
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



```

    return degree;
}

/**
 * 将图片按照某个角度进行旋转
 *
 * @param bm 需要旋转的图片
 * @param degree 旋转角度
 * @return 旋转后的图片
 */
public static Bitmap rotateBitmapByDegree(Bitmap bm, int degree) {
    Bitmap returnBm = null;

    // 根据旋转角度，生成旋转矩阵
    Matrix matrix = new Matrix();
    matrix.postRotate(degree);

    try {
        // 将原始图片按照旋转矩阵进行旋转，并得到新的图片
        returnBm = Bitmap.createBitmap(bm, 0, 0, bm.getWidth(), bm.getHeight(), matrix,
true);
    } catch (OutOfMemoryError e) {
        e.printStackTrace();
        // 当内存溢出时，利用递归进行重新旋转
        while (returnBm == null) {
            System.gc();
            System.runFinalization();
            returnBm = rotateBitmapByDegree(bm, degree);
        }
    }
    if (returnBm == null) {
        returnBm = bm;
    }
    if (bm != returnBm) {
        bm.recycle();
    }
    return returnBm;
}
}

```

## 最后小结

可以更改类中的压缩参数，对图片压缩的质量进行调整

功能这么齐全的图片压缩类，还有谁？

标签 : des android class log com http si 使用 it

赞

(0)

踩

(0)

举报

评论

一句话评论 ( 0 )

共0条

发表评论

友情链接

兰亭集智 国之画 cnbeta CSDN 博客园 百度统计 站长统计

阳和移动开发 汇智网 易捷博客网 天码营 HarriesBlog 程序员客栈 情怀 科学看世界 吾吾织梦 PHP教程