1、Drawable → Bitmap

public static Bitmap drawableToBitmap(Drawable drawable) {

Bitmap bitmap = Bitmap

.createBitmap(

drawable.getIntrinsicWidth(),

drawable.getIntrinsicHeight(),

drawable.getOpacity() != PixelFormat.OPAQUE ? Bitmap.Config.ARGB\_8888

: Bitmap.Config.RGB\_565);

Canvas canvas = new Canvas(bitmap);

// canvas.setBitmap(bitmap);

drawable.setBounds(0, 0, drawable.getIntrinsicWidth(),

drawable.getIntrinsicHeight());

drawable.draw(canvas);

return bitmap;

}

2、从资源中获取Bitmap

Resources res=getResources();

Bitmap bmp=BitmapFactory.decodeResource(res, R.drawable.pic);

3、Bitmap → byte[]

private byte[] Bitmap2Bytes(Bitmap bm){

ByteArrayOutputStream baos = new ByteArrayOutputStream();

bm.compress(Bitmap.CompressFormat.PNG, 100, baos);

return baos.toByteArray();

}

4、byte[] → Bitmap

private Bitmap Bytes2Bimap(byte[] b){

if(b.length!=0){

return BitmapFactory.decodeByteArray(b, 0, b.length);

}

else {

return null;

}

}

5、保存bitmap

static boolean saveBitmap2file(Bitmap bmp,String filename){

CompressFormat format= Bitmap.CompressFormat.JPEG;

int quality = 100;

OutputStream stream = null;

try {

stream = new FileOutputStream("/sdcard/" + filename);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

Generated by Foxit PDF Creator © Foxit Software

[http://www.foxitsoftware.com](http://www.foxitsoftware.com/) For evaluation only.

e.printStackTrace();

}

return bmp.compress(format, quality, stream);

}

6、将图片按自己的要求缩放

// 图片源

Bitmap bm = BitmapFactory.decodeStream(getResources()

.openRawResource(R.drawable.dog));

// 获得图片的宽高

int width = bm.getWidth();

int height = bm.getHeight();

// 设置想要的大小

int newWidth = 320;

int newHeight = 480;

// 计算缩放比例

float scaleWidth = ((float) newWidth) / width;

float scaleHeight = ((float) newHeight) / height;

// 取得想要缩放的matrix参数

Matrix matrix = new Matrix();

matrix.postScale(scaleWidth, scaleHeight);

// 得到新的图片

Bitmap newbm = Bitmap.createBitmap(bm, 0, 0, width, height, matrix,

true);

// 放在画布上

canvas.drawBitmap(newbm, 0, 0, paint);

相关知识链接：<http://www.eoeandroid.com/thread-3162-1-1.html>

7、bitmap的用法小结

BitmapFactory.Options option = new BitmapFactory.Options();

option.inSampleSize = 2; //将图片设为原来宽高的1/2，防止内存溢出

Bitmap bm = BitmapFactory.decodeFile("",option);//文件流

URL url = new URL("");

InputStream is = url.openStream();

Bitmap bm = BitmapFactory.decodeStream(is);

android:scaleType：

android:scaleType是控制图片如何resized/moved来匹对ImageView的size。ImageView.ScaleType /

android:scaleType值的意义区别：

CENTER /center 按图片的原来size居中显示，当图片长/宽超过View的长/宽，则截取图片的居中部分

显示

CENTER\_CROP / centerCrop 按比例扩大图片的size居中显示，使得图片长(宽)等于或大于View的长

(宽)

CENTER\_INSIDE / centerInside 将图片的内容完整居中显示，通过按比例缩小或原来的size使得图片

长/宽等于或小于View的长/宽

Generated by Foxit PDF Creator © Foxit Software

[http://www.foxitsoftware.com](http://www.foxitsoftware.com/) For evaluation only.

FIT\_CENTER / fitCenter 把图片按比例扩大/缩小到View的宽度，居中显示

FIT\_END / fitEnd 把图片按比例扩大/缩小到View的宽度，显示在View的下部分位置

FIT\_START / fitStart 把图片按比例扩大/缩小到View的宽度，显示在View的上部分位置

FIT\_XY / fitXY 把图片 不按比例 扩大/缩小到View的大小显示

MATRIX / matrix 用矩阵来绘制，动态缩小放大图片来显示。

//放大缩小图片

public static Bitmap zoomBitmap(Bitmap bitmap,int w,int h){

int width = bitmap.getWidth();

int height = bitmap.getHeight();

Matrix matrix = new Matrix();

float scaleWidht = ((float)w / width);

float scaleHeight = ((float)h / height);

matrix.postScale(scaleWidht, scaleHeight);

Bitmap newbmp = Bitmap.createBitmap(bitmap, 0, 0, width, height, matrix,

true);

return newbmp;

}

//将Drawable转化为Bitmap

public static Bitmap drawableToBitmap(Drawable drawable){

int width = drawable.getIntrinsicWidth();

int height = drawable.getIntrinsicHeight();

Bitmap bitmap = Bitmap.createBitmap(width, height,

drawable.getOpacity() != PixelFormat.OPAQUE ? Bitmap.Config.ARGB\_8888

: Bitmap.Config.RGB\_565);

Canvas canvas = new Canvas(bitmap);

drawable.setBounds(0,0,width,height);

drawable.draw(canvas);

return bitmap;

Generated by Foxit PDF Creator © Foxit Software

[http://www.foxitsoftware.com](http://www.foxitsoftware.com/) For evaluation only.

}

//获得圆角图片的方法

public static Bitmap getRoundedCornerBitmap(Bitmap bitmap,float roundPx){

Bitmap output = Bitmap.createBitmap(bitmap.getWidth(), bitmap

.getHeight(), Config.ARGB\_8888);

Canvas canvas = new Canvas(output);

final int color = 0xff424242;

final Paint paint = new Paint();

final Rect rect = new Rect(0, 0, bitmap.getWidth(), bitmap.getHeight());

final RectF rectF = new RectF(rect);

paint.setAntiAlias(true);

canvas.drawARGB(0, 0, 0, 0);

paint.setColor(color);

canvas.drawRoundRect(rectF, roundPx, roundPx, paint);

paint.setXfermode(new PorterDuffXfermode(Mode.SRC\_IN));

canvas.drawBitmap(bitmap, rect, rect, paint);

return output;

}

//获得带倒影的图片方法

public static Bitmap createReflectionImageWithOrigin(Bitmap bitmap){

final int reflectionGap = 4;

int width = bitmap.getWidth();

int height = bitmap.getHeight();

Matrix matrix = new Matrix();

matrix.preScale(1, -1);

Bitmap reflectionImage = Bitmap.createBitmap(bitmap,

0, height/2, width, height/2, matrix, false);

Bitmap bitmapWithReflection = Bitmap.createBitmap(width, (height + height/2),

Config.ARGB\_8888);

Canvas canvas = new Canvas(bitmapWithReflection);

canvas.drawBitmap(bitmap, 0, 0, null);

Paint deafalutPaint = new Paint();

Generated by Foxit PDF Creator © Foxit Software

[http://www.foxitsoftware.com](http://www.foxitsoftware.com/) For evaluation only.

canvas.drawRect(0, height,width,height + reflectionGap,

deafalutPaint);

canvas.drawBitmap(reflectionImage, 0, height + reflectionGap, null);

Paint paint = new Paint();

LinearGradient shader = new LinearGradient(0,

bitmap.getHeight(), 0, bitmapWithReflection.getHeight()

+ reflectionGap, 0x70ffffff, 0x00ffffff, TileMode.CLAMP);

paint.setShader(shader);

// Set the Transfer mode to be porter duff and destination in

paint.setXfermode(new PorterDuffXfermode(Mode.DST\_IN));

// Draw a rectangle using the paint with our linear gradient

canvas.drawRect(0, height, width, bitmapWithReflection.getHeight()

+ reflectionGap, paint);

return bitmapWithReflection;

}

}