获得圆角图片

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

int w = bitmap.getWidth();

int h = bitmap.getHeight();

Bitmap output = Bitmap.createBitmap(w, h, Config.ARGB\_8888);

Canvas canvas = new Canvas(output);

final int color = 0xff424242;

final Paint paint = new Paint();

final Rect rect = new Rect(0, 0, w, h);

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

public static Bitmap createReflectionImageWithOrigin(Bitmap bitmap) {

final int reflectionGap = 4;

int w = bitmap.getWidth();

int h = bitmap.getHeight();

Matrix matrix = new Matrix();

matrix.preScale(1, -1);

Bitmap reflectionImage = Bitmap.createBitmap(bitmap, 0, h / 2, w,

h / 2, matrix, false);

Bitmap bitmapWithReflection = Bitmap.createBitmap(w, (h + h / 2),

Config.ARGB\_8888);

Canvas canvas = new Canvas(bitmapWithReflection);

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

Paint deafalutPaint = new Paint();

canvas.drawRect(0, h, w, h + reflectionGap, deafalutPaint);

canvas.drawBitmap(reflectionImage, 0, h + 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, h, w, bitmapWithReflection.getHeight()

+ reflectionGap, paint);

return bitmapWithReflection;

}