

# Mobile Device Oriented Image Scaling for Reducing Memory Consumption in storing in Android

Rupali Sachin Vairagde  
Computer Engineering Department  
Pune Institute of Computer Technology  
Pune, India

Rekha A. Kulkarni  
Computer Engineering Department  
Pune Institute of Computer Technology  
Pune, India

---

**Abstract:** In Android operating system, efficient memory consumption is an important feature for better performance. It is very important to efficiently use and manage the internal and external memory space present inside the mobile operating system. Various techniques has been used and implemented to reduce the memory usage in android. One of the technique for better utilization of memory is image compression using scaling technique in android. As per the previous work, image compression techniques are widely used in the field medical science for transparency of reports and effective treatment practices. Other studies didn't show good results and facing problems such as data loses, poor visibility, large image size and incompatibility issues with the devices and the size of application developed for image compression is too large. Here in this proposed system, we are developing an android mobile application where device oriented image scaling and user oriented image scaling for different screen sizes can be done to reduce the memory consumption of android mobile devices without data loss and visibility.

**Keywords:** Image Scaling; Android OS; Application; memory consumption; mobile device; storage management;

---

## 1. INTRODUCTION

Nowadays we are facing the increasing use of images in many parts of our life. 3D vision systems, satellites, cameras, medical equipments etc. All of these equipments use or produce image for different purposes. For example the images, for competitive examination or profile pictures we need to upload the compressed image[1]. We have to save image on mobile device or when transmit them, then because of the limitation in disk space and channel bandwidth we almost always need image compression for decreasing the size of data which must be save or transmit[1]. There are several methods for image compression based on the criteria and conditions.

Some of these criteria are compression ratio, compression quality, compression time. Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from android [13]. The JPEG method is more often used for photographs, while the PNG method is designed to work well in online viewing applications like web browsers so it is fully streamable with a progressive display option. The PNG (Portable Network Graphics) file format was created as the free, open-source successor to GIF. Android SDK multiple drawable directories exist for different screen resolutions. There are low, medium, and high DPI specific directories, drawable-ldpi, drawable-mdpi, drawable-hdpi respectively. This allows you to create images at different DPI to enhance the appearance of your application.

## 2. RELATED WORK

Image compression is very vast area where new concepts and methods are used day-by-day. Android JPEG2000 is a image compression engine or tool which helps in transplanting a image on smart phone based on Android to enhance efficiency

of image compression compared with traditional compression. They use the NDK for the compression. A Quantization approach is proposed for the image compression. This method is useful for enhancement of compression quality when each kind of neural network is used to compress the image.

In Android operating system image compression techniques are widely used in the field medical science for transparency of reports and effective treatment practices. Other studies didn't show good results and facing problems such as data loses, poor visibility, large image size and incompatibility issues with the devices and the size of application developed for image compression is too large.

## 3. IMAGE COMPRESSION TECHNIQUE

Image compression addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements [15].

### 3.1 Advantages of Image Compression

1. It provides a potential cost savings associated with sending less data over mobile phone network.
2. It not only reduces storage requirements but also overall execution time.
3. It also reduces the probability of transmission errors since fewer bits are transferred.
4. It also provides a level of security against illicit monitoring.
5. It is important to efficiently use and manage the internal and external memory space present inside the mobile operating system.

The image compression techniques are broadly classified into two categories:

1. Lossless Technique
2. Lossy Technique

In lossless compression techniques, the original image can be perfectly recovered from the compressed (encoded) image. In Lossy compression techniques, it provides much higher compression ratios than lossless schemes. lossy method can produce a much smaller compressed file than any known lossless method, while still meeting the requirements of the application.

Lossy methods are most often used for Compressing sound, images or videos. The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file). Lossily compressed still images are often compressed to 1/10th their original size. In this project we used the lossy compression method it will provide the good

compression ratio. Our objective is to reduce the size of file and save the more memory & reduced the consumption in storing in android. Now days we are using lots of application it will consume lots of space & in browsing if we used compressed image it take less time & less bandwidth to transfer the data. In Android it store the images in DCIM folder and create thumbnails for images it will take the more space to store files that's why rather than lossless method the lossy method is more beneficial in our project.

#### 4. PROPOSED SYSTEM ARCHITECTURE

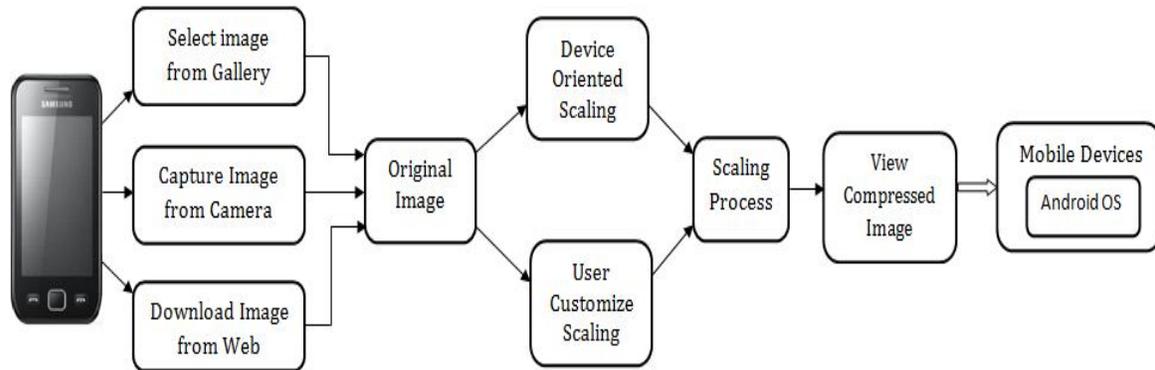


Fig 1. Proposed System Architecture

In proposed System architecture for Device Oriented Image Scaling for reducing memory consumption in storing in Android. Image Compression in Android for Mobile System it will run android application on any mobile device where the android operating system is present then Input of Images for application it will take Images from Gallery of Mobile Phones, Image capture from Camera and Images downloaded from WAP or Internet.

Either select single image or multiple Images for Image compression or scaling, for Single Image it will show the file size of the Image i.e original image stored in mobile device and the resolution of the image and the extension or type of the image for ex. jpeg or .png. Our objective is to reduce the file size of the image and save more memory it will reduced the memory consumption in storing in android for different screen devices. So, that after selection of image user will decide two ways for reducing the size of image with the process of scaling in android.

#### 4.1 Device Oriented Scaling

In Device oriented scaling it depends upon the different screen sizes and densities of the individual devices. The system performs scaling and resizing to make your application work on different screens, it fit the screen on their devices.

#### 4.2 User Customize scaling

In User Customize scaling it depends upon the user what is the scaling factor to compress the image. We can scale the image to reduce the size of images as per the requirement of the user. But, it will reduce the quality of the image. We need

to decide the scaling factor it will not reduced the quality of the image.

The image compression tool for android mobile devices it finally result it into the compressed image and it displayed all the details for example size of the original image and size of the compressed image, Compression ratio(CR). Android by default it will store the type .jpeg it will already compressed but in our proposed work we reduced the size of .jpeg image & .png image for multiple screen devices. Finally the resulted compressed image displayed on the mobile screen[8].

#### 5. IMAGE SCALING IN ANDROID

When you may want to use an image but it is too big to use it causing a memory problem for the devices. The way around the problem is to resize the image and scaling of Image is needed. In computer graphics, image scaling is the process of resizing a digital image.[10] Scaling is a non-trivial process that involves a trade-off between efficiency, smoothness and sharpness[16].

In android for images scaling it will consider three factors height of the image, width of the image, resolution of image it measured in a DPI. DPI is simply the “Dots Per Inch” in your image[11]. That simply means it's the measure of the resolution of your image based on the number of pixels or printer dots per inch. In android it considers the definition for the image properties is [6]:

**Resolution** = number of pixels available in the display, scale-independent pixel = **sp**

**Density** = how many pixels appear within a constant area of the display, dots per inch = **dpi**

**Size** = amount of physical space available for displaying an interface, screen's diagonal, **inch**

**Density-independent pixel** = virtual pixel that is independent of the screen density, **dp**

### 5.1 Density Classes in Android:

Class	Name	Density	Factor
Ldpi	Low Density	120 dpi	sp = 3/4 * dp
Mdpi	Medium Density	160 dpi	sp = dp
Hdpi	High Density	240 dpi	sp = 1.5 x dp
Xhdpi	Extra High Density	320 dpi	sp = 2 x dp
xxhdpi	Extra Extra High Density	480 dpi	sp = 3 x dp
xxxhdpi	Extra Extra Extra High Density	640 dpi	sp = 4 x dp

Fig. 2 Density Classes in Android

In our proposed work we adjust the DPI for multiple Screen and it will not loss the quality of image because we resize the image. resizing an image changes the size of its pixels, not its number of pixels[12]. We're not increasing or decreasing the number of pixels. The final output for the proposed system looks like this in a mobile device of android:



Fig. 3. Result of Image Compression Tool

## 6. MATHEMATICAL MODEL

Let S be the System such that,

$S = \{\text{Input, DD, NDD, SI, Sc, CR, Mt, Mavi, Macc, Succ, Failure, Output}\}$

Input = {P1, P2, P3} = {Number of Images from Gallery, Camera & WAP}

P1= {Image Capture from Camera type is .jpeg or .png}

P2= {Image is Downloaded from web in Mobile Devices type is .jpeg or .png}

P3= {Image is already present in a Gallery in jpeg or png}

DD = Deterministic Data  
 = CI ∈ AI

Where, CI = Required Images requested by client to compress

AI = Available Images in Mobile Device

NDD = Non-Deterministic Data  
 = CI ∈ AI

= Required Images resources are not available.

SI = {Size, Resolution, DPI} = {Single Image Properties from P1, P2, P3}

Sc = {De\_Or, Ur\_Cu} = {Device oriented Scaling, User Customized Scaling}

CR = Compression Ratio  
 = Original Image / Compressed Image

Mt = {1.....n} = Total Memory

Mavi = {1.....n} = Available Memory

Mocc = {1.....m} = Occupied Memory

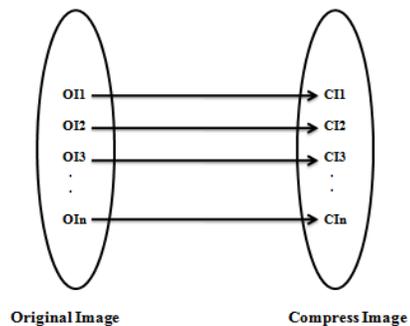
$$Mavi = \sum_{i=1}^n Mt - \sum_{i=1}^m Mocc$$

Output: Image is compressed and Save the more Memory of Android Devices.

Success: Reduced the file size of Image & reduced consumption of memory.

Failure: The File size of the image is not reduced & more consumption of memory.

### 6.1 Functional Mapping of Original Image and Compressed Image



## 7. CONCLUSION

The increasing use of images in many parts of our life for example in multimedia technology and for online environment we used digital image & that required image we need compress. Image compressions for decreasing the size of data which must be save or transmit. It is important to efficiently use and manage the internal and external memory space present inside the mobile operating system. In this project we used the lossy compression method it will provide the good compression ratio. Our objective is to reduce the size of image file and save the more memory & reduced the memory consumption in storing in android. Now days we are using lots of application for mobile devices it will consume lots of memory space, if we compressed image it take less time , less space & less bandwidth to transfer the data for that in proposed work the android mobile application is developed.

## 8. ACKNOWLEDGMENTS

I would like to express my gratitude to all those who helped me during the writing of this paper. A special thanks to my guide Prof. Rekha A. Kulkani for her valuable suggestions & with the reference material of great value. Finally I wish to extend my thanks to my family & all my friends for support.

## 9. REFERENCES

- [1] Hui-yang wang., xiang-dong, Study of jpeg2000 on Android , School of Tnfonnation and Communication Engineering, Beijing University of Posts and Telecommunications, proceedings of IEEE, 2012
- [2] Seyed Mehdi Moghadas, Hossein Pourghasem , New Image Compression Algorithm using Proposed Quantization Approach, 2011 International Conference on Pattern Analysis and Intelligent Robotics 28-29 June 2011, Putrajaya, Malaysia.
- [3] Robert D. Dony, Simon Haykin, Neural Network Approaches to Image Compression, Proceedings of the IEEE, vol. 83, no. 2, february 1995.
- [4] Lalit Kumar , Kanchan Sharma , Web Based Novel Technique for Watermarking Colour Images on Android Mobile Phones , International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 7, July 2013 ISSN: 2277 128X .
- [5] Miss. Jagtap R.S.,Mrs. Sadalage J.A, Image Processing as Android Application, IJSRT , MAY - JUNE 2013
- [6] DeepalinKayande and urmila shrawankar , Performance Analysis for Improved RAM utilization for Android Application, Software Engineering (CONSEG), 2012 CSI Sixth International Conference.
- [7] <http://stefan222devel.blogspot.in/2012/10/android-screen-densities-sizes.html>
- [8] [http://en.wikipedia.org/wiki/Image\\_file\\_formats](http://en.wikipedia.org/wiki/Image_file_formats)
- [9] <http://developer.sonymobile.com/2011/0627/how-to-scale-images-for-your-ndroid-application/>
- [10] [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)
- [11] [http://www.serif.com/appresources/HPX4/Tutorials/engb/photoplus\\_tutorials/photo\\_image\\_size.htm](http://www.serif.com/appresources/HPX4/Tutorials/engb/photoplus_tutorials/photo_image_size.htm)
- [12] <http://www.ccbirding.com/dcsig/howto/dpi/dpi.pdf>
- [13] <http://www.webdesignerdepot.com/2010/02/the-myth-of-dpi/>
- [14] <http://argillander.wordpress.com/2011/11/24/scale-image-into-imageview-then-resize-imageview-to-match-the-image/#comment-52>
- [15] <http://www.higherpass.com/Android/Tutorials/Working-With-Images-In-Android/3/>
- [16] <http://searchciomidmarket.techtarget.com/definition/image-compression>
- [17] [http://en.wikipedia.org/wiki/Image\\_scaling](http://en.wikipedia.org/wiki/Image_scaling)