



# 안드로이드 이미지 로드

이미지 라이브러리 Glide 참고

[bumptech/glide](https://github.com/bumptech/glide)  
<https://github.com/bumptech/glide>

**사용하지 않음** ← 기능 사용 하지 않음 그냥 참고용

일반적인 Glide의 경우에는 이미지의 경우 큰 문제 없이 사용 가능 하지만 현재 사용 중인 앱의 경우 이미지는 올리는 크기 및 규격이 정해져 있지 않음 또한 리사이징을 통해서 올린 경우 이미지가 틀어지고

해당기기의 최대 텍스처를 기준으로 이미지를 로드 하다 보니 이미지 열화가 발생 하드웨어 가속을 끈 경우에는 해당 화면이 프레임 수가 낮아져 프레임이 떨어져서 보임

현재 최선책인 [주로 고화질 이미지] 긴 이미지의 경우 web를 통한 코드 변경]

## 큰 이미지 로드를 웹뷰로 하려고 하는 이유

모바일에서 주로 사용하는 jpg, png, gif 등 이미지는 압축된 포맷을 사용. 하지만 Android에서 사용하는 Bitmap은 압축된 포맷이 아니며 압축되지 않은 탃에 압축을 풀 때 사용 되는 시간과 CPU리소스를 아끼는 것이 가능하지만 대신 많은 메모리를 소비. Bitmap이 소비하는 메모리의 양은 아래와 같이 계산할 수 있습니다.

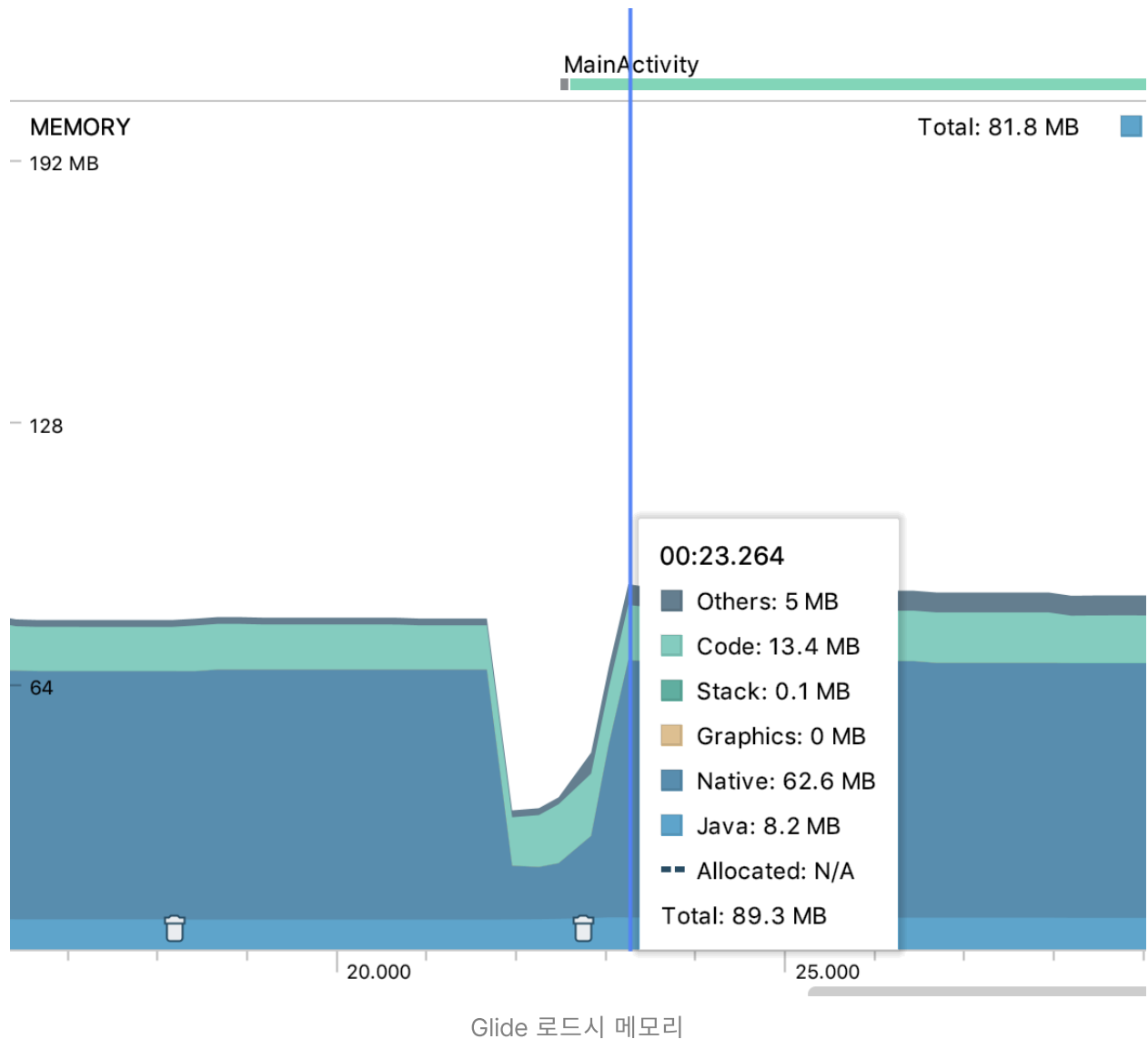
$(width) * (height) * (bits) / 8 = \text{메모리 크기}$

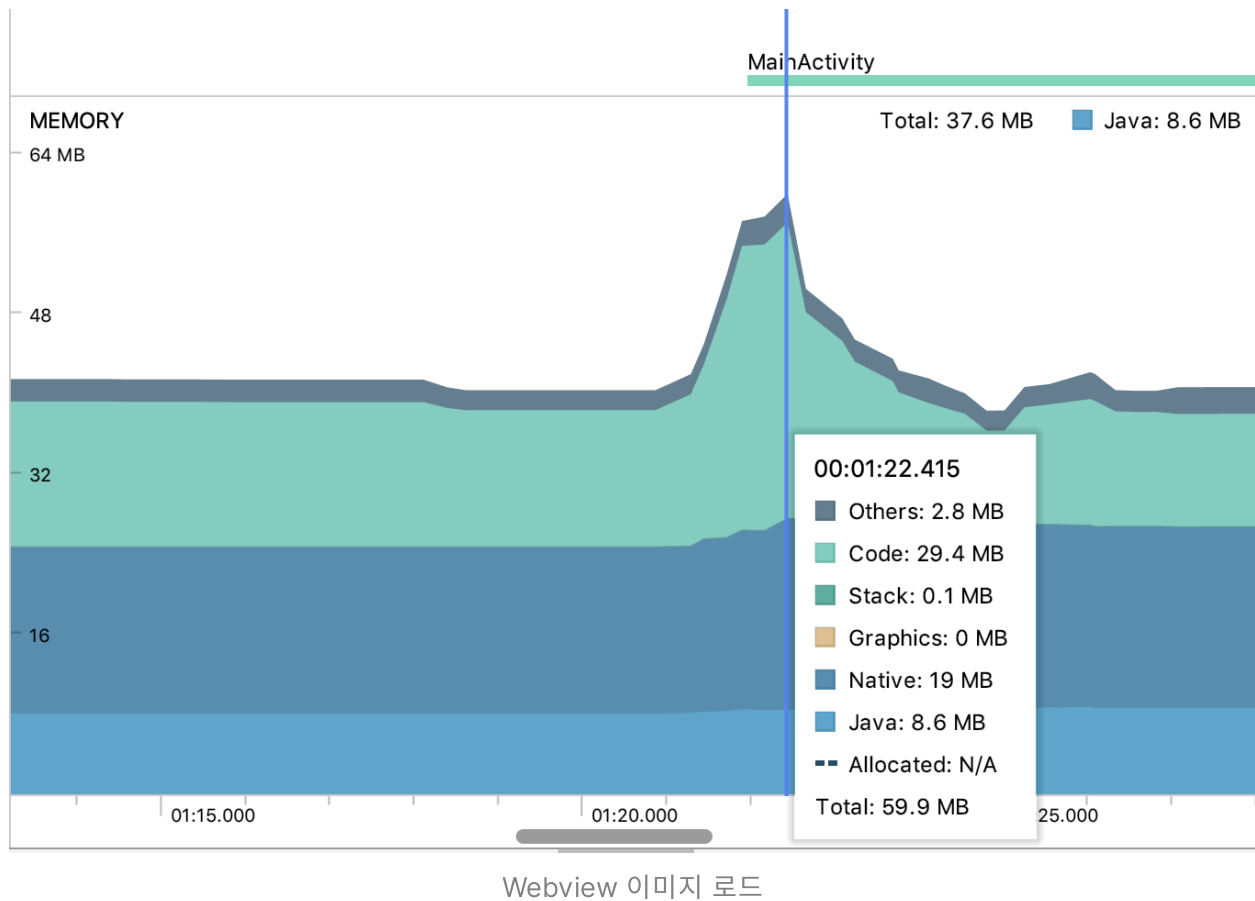
V50의 경우 기본적으로 4032 x 1920 해상도의 사진을 촬영합니다. 이 이미지를 픽셀당 32bit(ARGB\_8888)와 16bit(RGB\_565)로 Bitmap 로드할 경우 차지하는 메모리는

$4032 * 1920 * 32 / 8 = 30,965,760 \text{ bytes}$

$4032 \times 1920 \times 16 / 8 = 15,482,880$  bytes

각각 30MB, 15MB가량을 차지. 이 경우 V50의 최대 메모리 사이즈에 대입 하여 보면 앱당 대략 8장, 16장을 한 번에 로드 할 수 있다고 계산 가능 . 물론 메모리를 사용하는 건 이미지 만이 아니기 때문에 실제로는 그보다 더 적을 것. 하지만 Bitmap 이미지를 원본 사이즈로 로드 할때 앱의 메모리에 얼마나 큰 부담을 주는지는 정확하게 알 수 없음 이 앱의 올리는 이미지의 크기는 규격이 없기 때문에 따라서 이미지의 전체 크기가 1을 기준으로 이미지를 로드를 시키는 것을 웹뷰로 이미지를 호출 처리함





같은 크기의 이미지 호출 시 webview 로드 시 메모리를 덜 사용함

**결론 : 더 좋은 방법이 있으면 갈아 타는 것을 추천하고 일단 webview를 통한 이미지 호출을 하는 것이 제일 효율적인 방법이라고 봄**

## 사용 방법

### kotlin 코드

```
val style = "<style>img{height: auto;max-width: 100%;}</style>"
val tag = "<img src=\"\${이미지 url}\"/>"
```

```
val body = "<body style='margin:0;padding:0; overflow:scroll;*>$tag</body>"
val settings = webview.settings
settings.javaScriptEnabled = true
settings.domStorageEnabled = true
webview.loadData("$style$body", "text/html", "utf-8")
```

## Java 코드

```
webview.getSettings().setJavaScriptEnabled(true);
webview.getSettings().setDomStorageEnabled(true);
String style = "<style>img{height: auto;max-width: 100%;}</style>";
String tag = "<img src=\"\" + 이미지 url + \"\">";
String body = "<body style='margin:0;padding:0; overflow:scroll;*>" + tag + "</body>";
String loadURL = style + body;
webview.loadData(loadURL, "text/html", "utf-8");
```

단점 : 이미지가 전체 로드 전 까지는 빈 공백 화면이 보이며, 반드시 html 태그가 필요함

장점 : 이미지에 대한 메모리 사용하지 않아 프레임 수가 유지 됨 [20~30 정도의 프레임 유지], 위에 설명한 그대로 큰 문제 없이 언제든지 크기에 상관 없이 로드 가능