
Glimpse

A content-aware cropping library for Android.

Give the right first impression with just a glimpse! Instead of center cropping images blindly, leverage Glimpse's eye to catch the right spot.



Setup

Add to top level *gradle.build* file

```
1 allprojects {
2     repositories {
3         maven { url "https://jitpack.io" }
4     }
5 }
```

Add to app module *gradle.build* file TensorFlow lite recommends most developers omit the x86, x86_64, and arm32 ABIs. This can be achieved with the following Gradle configuration, which specifically includes only armeabi-v7a and arm64-v8a, which should cover most modern Android devices.

```
1
2 android {
3     aaptOptions {
4         noCompress "tflite"
5         noCompress "lite"
6     }
7 }
```

```
7     defaultConfig {
8         ndk {
9             abiFilters 'armeabi-v7a', 'arm64-v8a'
10        }
11    }
12 }
13
14 dependencies {
15     implementation 'com.github.the-super-toys.glimpse-android:glimpse-
        core:0.0.5'
16
17     //only required for glide extensions
18     implementation 'com.github.the-super-toys.glimpse-android:glimpse-
        glide:0.0.5'
19
20     //only required for coil extensions
21     implementation 'com.github.the-super-toys.glimpse-android:glimpse-
        coil:0.0.5'
22
23     implementation 'org.tensorflow:tensorflow-lite:0.0.0-nightly'
24 }
```

Usage

Init Glimpse in your Android app

```
1 class YourApp : Application() {
2     override fun onCreate() {
3         super.onCreate()
4         Glimpse.init(this)
5     }
6 }
```

Use Glimpse without extensions

First compute image's focal points by calling `Bitmap.findCenter` extension function.

```
1 val original = (imageView.drawable as BitmapDrawable).bitmap
2 val (x, y) = original.findCenter()
```

Then supply those focal points alongside with the width and height values of the target `ImageView` to `Bitmap.crop` extension function.

```
1 val cropped = original.crop(x, y, imageView.layoutParams.width,
    imageView.layoutParams.height)
2 imageView.setImageBitmap(cropped)
```

Use Glimpse with Glide extension

If you are using Glide for image loading you can just add `GlimpseTransformation` to `GlideRequest` builder:

```
1 GlideApp.with(context)
2     .load(url)
3     .diskCacheStrategy(DiskCacheStrategy.RESOURCE)
4     .transform(GlimpseTransformation())
5     .into(imageView)
```

It is recommended to set `diskCacheStrategy(DiskCacheStrategy.RESOURCE)` to cache the cropped bitmap by Glimpse, otherwise focal points will be calculated every time the image is displayed.

Use Glimpse with Coil extension

If you are using Coil for image loading you can just add `GlimpseTransformation` to Coil's `LoadRequestBuilder` builder:

```
1 imageView.load(url) {
2     crossfade(true)
3     placeholder(R.drawable.image)
4     transformations(GlimpseTransformation())
5 }
```

What about other image loaders such as Picasso and Fresco?

We tried to ship Glimpse with both Picasso and Fresco extensions but we were not able to find the right api.

We opened an issue on Picasso repository and a labeled Fresco question on Stack Overflow asking for guidance, but sadly we did not find much support. If you know how to assemble Glimpse cropping functionality to either lib without disrupting their original pipeline, please open an issue to review together the proposal. We really want to offer support for all the popular loading image libraries!

Java callers

If you're using Java you can take a look at this file to see how Glimpse api looks for Java callers.

Sample app

: [sample-app](#) module showcase Glimpse usage by making use of Glide extensions. The application is also published on Google Play.

Is Glimpse ready for production?

Actually, it depends. Ideally you should not use Glimpse to crop the same image over and over. Even if you use Glide extension which caches the output transformation, the underlying calculation will be performed in every android client. Thus, if possible, send to your cloud solution the x and y focal points of the calculated crop alongside the image when user uploads content.

We timed the crop calculation on an OnePlus 6 and in average it takes 30 milliseconds. Depending on the device and your specific use case Glimpse may or may not be ready for production. In any case we encourage you to provide feedback to help us make Glimpse a mature library.

How much increase APK's size?

If you're already using TensorFlow lite in your app, adding Glimpse costs only the size of the model, which is 148 KB. But chances are that you are not using it, so you have to take into account also the size of TensorFlow lite, which is about 300 KB.

Let's go DEEPER (The Deep Learning Model)

If you want to know how Glimpse works in a more deeper lever, take a look at this repository, where you can see the architecture and implementation of the Deep Learning model used by Glimpse. We explain how everything works in a much more technical level, and with all details, so you can reproduce it on your own, and improve it if you want.

But if you do, don't hesitate to submit a Pull Request to that repository so we can keep improving Glimpse!!