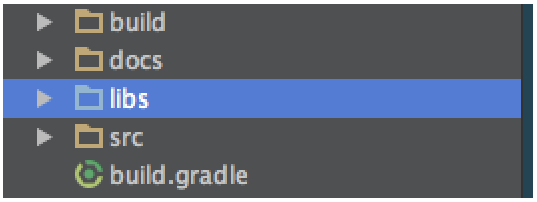
IDExtraction Documentation

- **OpenCV 2.411 Setup Instructions**
  http://opencv.org/platforms/android/

- **Here are the instructions for adding a IDExtraction jar file as a library to a module:**

1. Create a 'libs' folder in the top level of the module directory (the same directory that contains the 'src' directory)



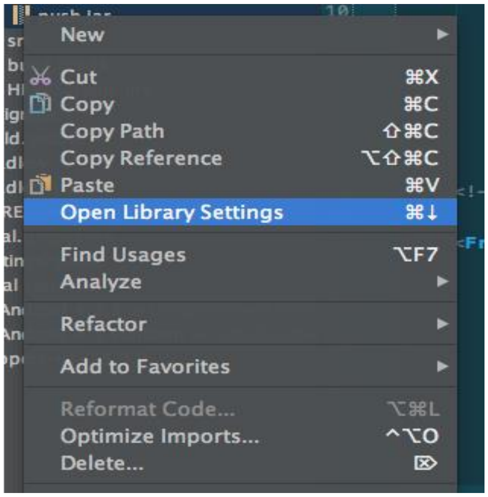
2. In the build.gradle file add the following so that your dependencies closure has

```
dependencies {
  // ... other dependencies
  compile files('libs/<your jar's name here>')
}
```

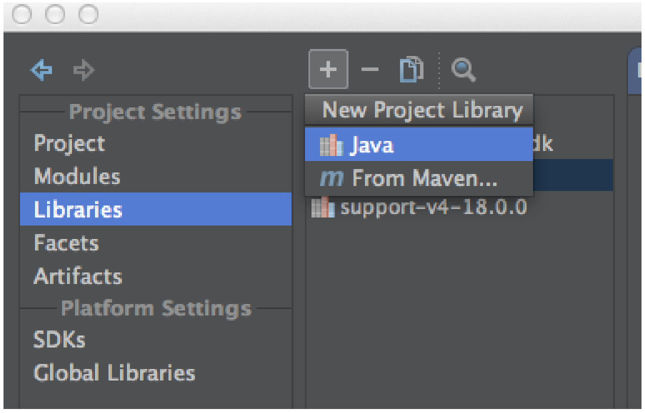
3. Android Studio should have already setup a gradlew wrapper. From the command line, navigate to the top level of your project (the directory that has a gradlew file).
   Run. /gradlew assemble. This should compile the project with the library. You may need to fix errors in your build.gradle file as necessary.

4. In order to have Android Studio recognize the local jar files as libraries for support while coding in the IDE, you need to take a few more steps:

   4.1. Right click on the module in the left-hand panel and choose Open Library Settings.



4.2. On the left panel of the dialog, choose Libraries.
4.3. Click the + sign above the panel second from the left -> Java

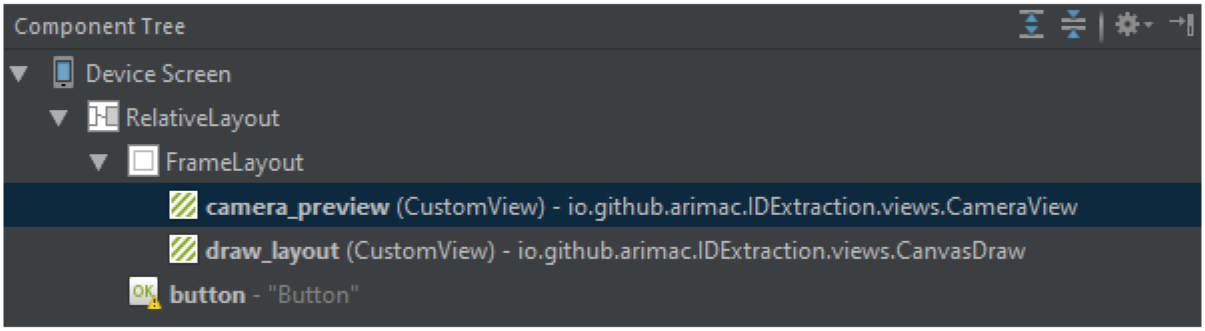


4.4. Select your local jar and add it to the project.

5. You may need to run the above. /gradlew command one more time

- **Configure the MainActivity layout**

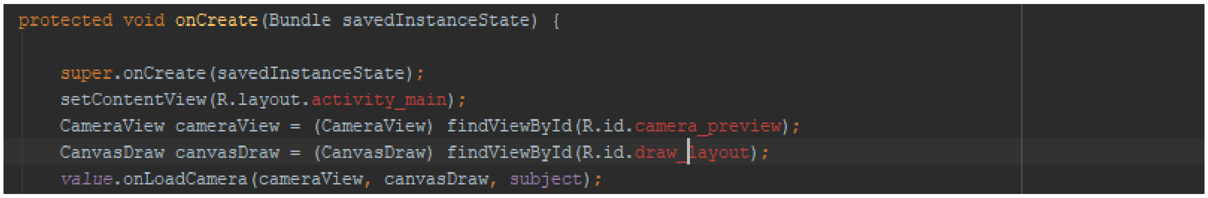
1. As the first step Create a **FrameLayout,** its layout width and height must be match parent.
2. Create two **CustomView** components for camera preview and rectangle preview and one Button for Image capturing purposes.



```
Component Tree
▼ Device Screen
  ▼ RelativeLayout
    ▼ FrameLayout
        camera_preview (CustomView) - io.github.arimac.IDExtraction.views.CameraView
        draw_layout (CustomView) - io.github.arimac.IDExtraction.views.CanvasDraw
      button - "Button"
```

3. After that, attach **CameraView** and **CanvasDraw** classes to above components as shown in the picture.

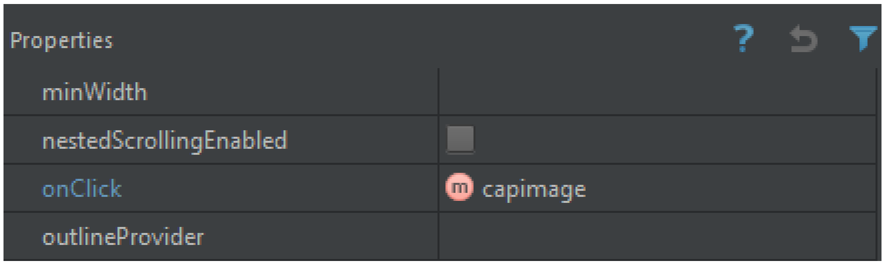
4. Your Custom Views must be named as **camera_preview** and **draw_layout**. And add last code line into your **onCreate()** method. **(activity_main is your main class)**



```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    CameraView cameraView = (CameraView) findViewById(R.id.camera_preview);
    CanvasDraw canvasDraw = (CanvasDraw) findViewById(R.id.draw_layout);
    value.onLoadCamera(cameraView, canvasDraw, subject);
```

5. In the Button properties, you can call **capimage ()** method as **onClick** event of Button.

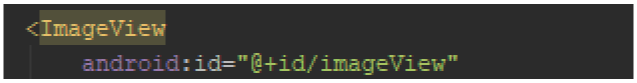


Properties

minWidth	
nestedScrollingEnabled	□
onClick	capimage
outlineProvider	

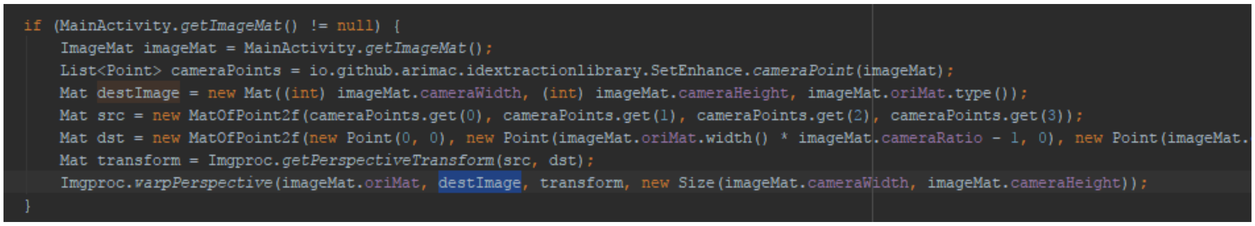
- **Configure the ImageViewActivity layout**

1. Create a new ImageView to display captured image and ImageView ID must be “imageView”.



```
<ImageView
    android:id="@+id/imageView"
```

2. As the next step, please add these lines into onCreate() method of **ImageViewActivity.class** for perspective correction of image.



```
if (MainActivity.getImageMat() != null) {
    ImageMat imageMat = MainActivity.getImageMat();
    List<Point> cameraPoints = io.github.arimac.idextractionlibrary.SetEnhance.cameraPoint(imageMat);
    Mat destImage = new Mat((int) imageMat.cameraWidth, (int) imageMat.cameraHeight, imageMat.oriMat.type());
    Mat src = new MatOfPoint2f(cameraPoints.get(0), cameraPoints.get(1), cameraPoints.get(2), cameraPoints.get(3));
    Mat dst = new MatOfPoint2f(new Point(0, 0), new Point(imageMat.oriMat.width() * imageMat.cameraRatio - 1, 0), new Point(imageMat.
    Mat transform = Imgproc.getPerspectiveTransform(src, dst);
    Imgproc.warpPerspective(imageMat.oriMat, destImage, transform, new Size(imageMat.cameraWidth, imageMat.cameraHeight));
}
```

3. After proceeding these steps, you’re free to do any image processing enhancemnt using **IDExtraction.JAR** library.

IDExtraction Reference

1. Io.github.arimac.IDExtraction
    - Controller
        - EnhanceController

- cameraPoint(ImageMat)

takes one argument which is included in Io.github.arimac.IDExtraction.models.ImageMat

Returns an ArrayList<> which consists of the four Camera Points.

- brightnessController(Mat sourceMat,double alpha,bouble beta)

this method is used to enhance the brightness of the image by multiplying each pixel of the image with an alpha value and adding another beta value to it and therefore it takes three arguments.

Pixel_Value=Pixel_Value*alpha+beta

Parameters
sourceMat It is destination image.
alpha It is optional scale factor.
beta It is optional delta added to the scaled values.

Returns an image matrix(Mat) after brightness process is happened.

- contrastController(Mat sourceMat,Mat destMat)

this method is used to enhance the contrast of an image using histogram equalization and takes two arguments.

Paramaeters
sourceMat It is 8-bit single channel source image.
destMat It is the destination image.

Returns an image matrix(Mat) after contrast process is happened.

- gaussianController(Mat sourceMat,Mat destMat,Size kernalSize,double sigmaX)

this method is used to blur image using Gaussian kernel and takes four arguments.

Parameters
sourceMat It is source image.
destMat It is destination image.
kernalSize It is Gaussian kernel size.
sigmaX It is Gaussian kernel standard deviation in X direction.

Returns an image matrix(Mat) after Gaussian blur process is happened.

- sharpnessController(Mat sourceArray,double alpha,Mat destArray,double beta,double gamma,Mat destMat)

this method is used to increase the sharpness of the image and takes six arguments.

Parameters
sourceArray It is first input array.
alpha It is weight of the first array elements.
destArray It is second input array of the same size and channel number as src1.
Beta It is weight of the second array elements.
gamma It is scalar added to each sum.
destMat It is output array that has the same size and number of channels as the input arrays.

Returns an image matrix(Mat) after sharpness process is happened.

- saveCapturedImage(Bitmap img)

this method is used to save image in External Storage and takes one argument.

Parameters
img Bitmap image to be stored

- overallEnhancement(Mat imageToEnhance,Mat mask,Mat sampledImage)

this method is used to enhance the image contrast and sharpness with in normal range. It takes three arguments.

Parameters
ImageToEnhance It is the result image matrix.
mask It is mask matrix and stresses which colour mask wants to be enhance
sampledImage It is an image matrix determines the result image size.

Returns an image matrix(Mat) after overall enhancement process is happened.

- hsvEnhancementController(Mat srcImage, Mat desImage)

this method is used to enhance the full colour image using image contrast channels which are called saturation(S) and value(V). It takes two arguments.

Parameters
srcImage It is source image matrix
destImage It is destination image matrix

Returns an image matrix(Mat) after overall HSVenhancement process is happened.

- noiseCancellationThresholdMethodController(Mat srcImage)

this method is used to denoise the colored image using threshold mask.

Parameters
srcImage it is an input 8-bit 3-channel image.

Returns an image matrix(Mat) after noiseCancellation process is happened.

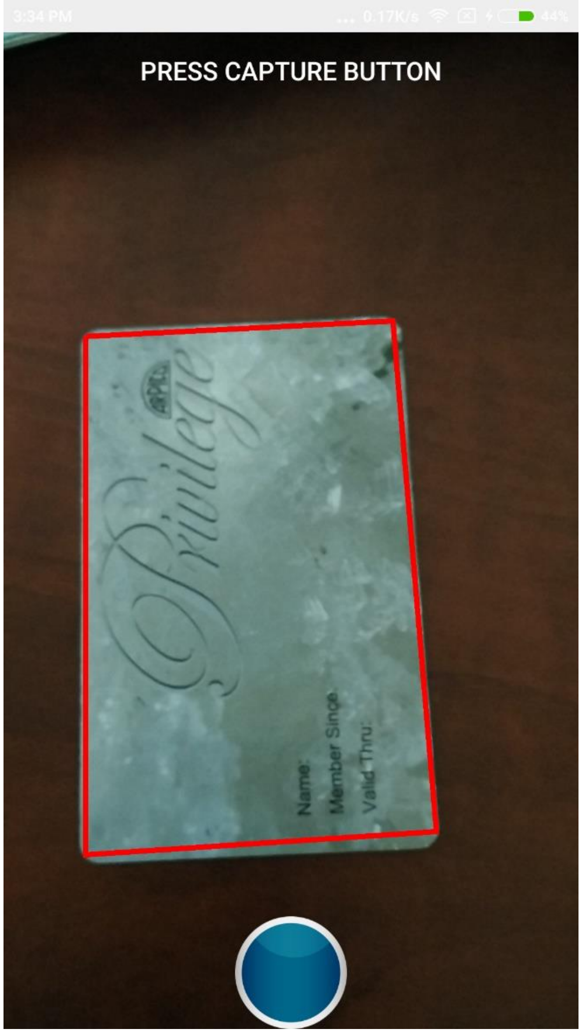
- fastNoiceCancellationController(Mat srcImage,float h,float hColour,int templateWindowSize,int searchWindowSize)

this method is used to denoise colored image in different color spaces. Also, this method converts image into CIELAB color space and then separately denoise L and AB components with different h parameter.

Parameters
srcImage it is an input 8-bit 3-channel image.
destImage It is an output image matrix with same type as srcImage
h It is parameter regulating filter strength for luminance component
hColour It is same as h but for color components
templateWindowSize It is size in pixels of the template patch that is used to compute weights. Should be odd value
searchWindowSize It is size in pixels of the window that is used to compute weighted average for given pixel. Should be odd

Returns an image matrix(Mat) after image denoising process is happened.

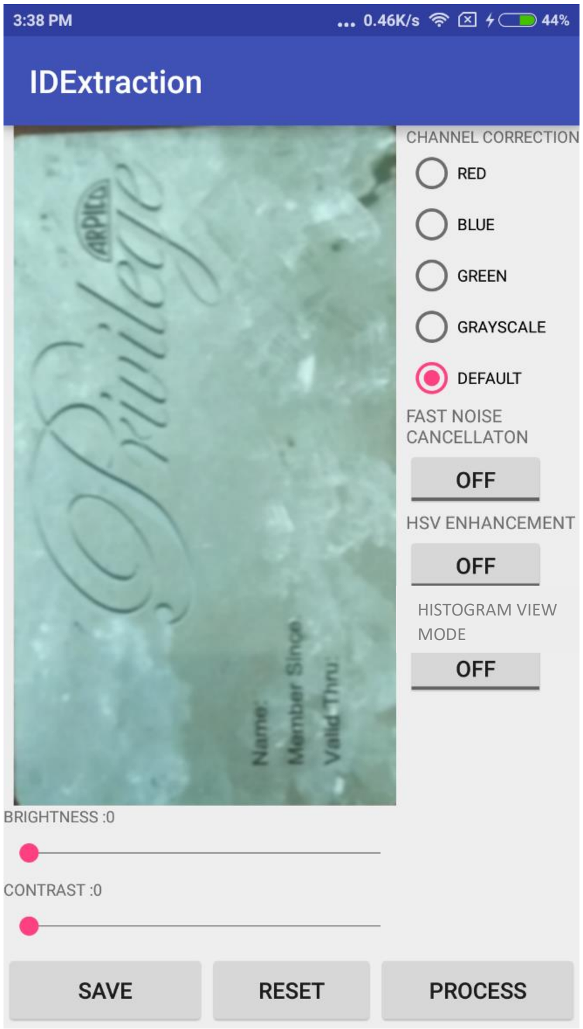
Sample Android Application



3:34 PM ... 0.17K/s 44%

PRESS CAPTURE BUTTON

Figure 1-ID Detection and Capturing



3:38 PM ... 0.46K/s 44%

IDExtraction

CHANNEL CORRECTION

○ RED

○ BLUE

○ GREEN

○ GRAYSCALE

● DEFAULT

FAST NOISE CANCELLATON: OFF

HSV ENHANCEMENT: OFF

HISTOGRAM VIEW MODE: OFF

BRIGHTNESS :0

CONTRAST :0

Figure 2 - Image Processing Intent

- User should capture clear image as much as possible. After the capture button is pressed, image will be automatically enhanced.
- As the second step, user will get image processing control panel to give desired values.
- After changing the properties of image, press PROCESS button to view new image.
- RESET- Load and preview default image.
- SAVE – Save proceed image in external storage source.