

Mobile Image Processing

- Part 1: Introduction to mobile image processing on Android
- Part 2: Real-time augmentation of viewfinder frames
- Part 3: Utilizing optimized functions in the OpenCV library

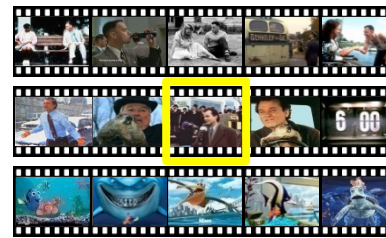


Recognizing video at a glance



(1) User snaps a photo of screen.

(3) User resumes video on the phone.



(2) Our system identifies video and frame within the video.

Recognizing video at a glance



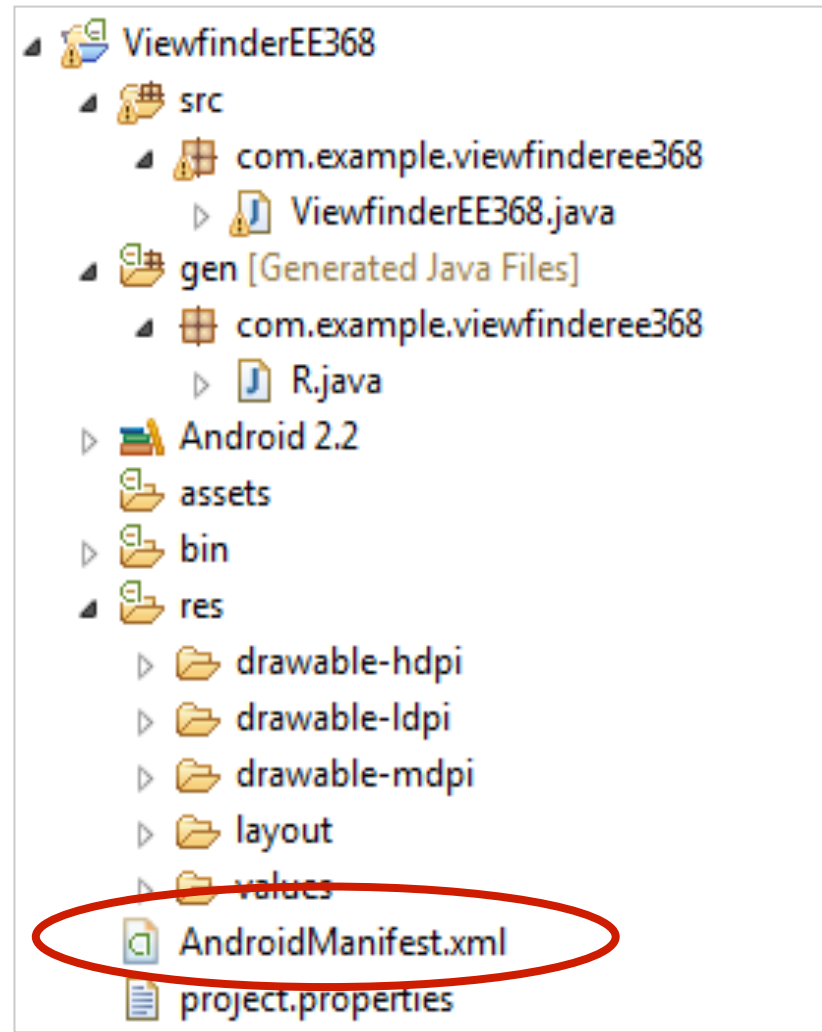
Human face detection

???

“Color Histograms” project

- Goals of this project
 - Learn how to access frames from the viewfinder
 - Learn how to modify camera parameters
 - Learn how to augment the viewfinder frames
- Full source available on class website:
 - <http://ee368.stanford.edu/Android/ViewfinderEE368>

Android manifest file



Android manifest file

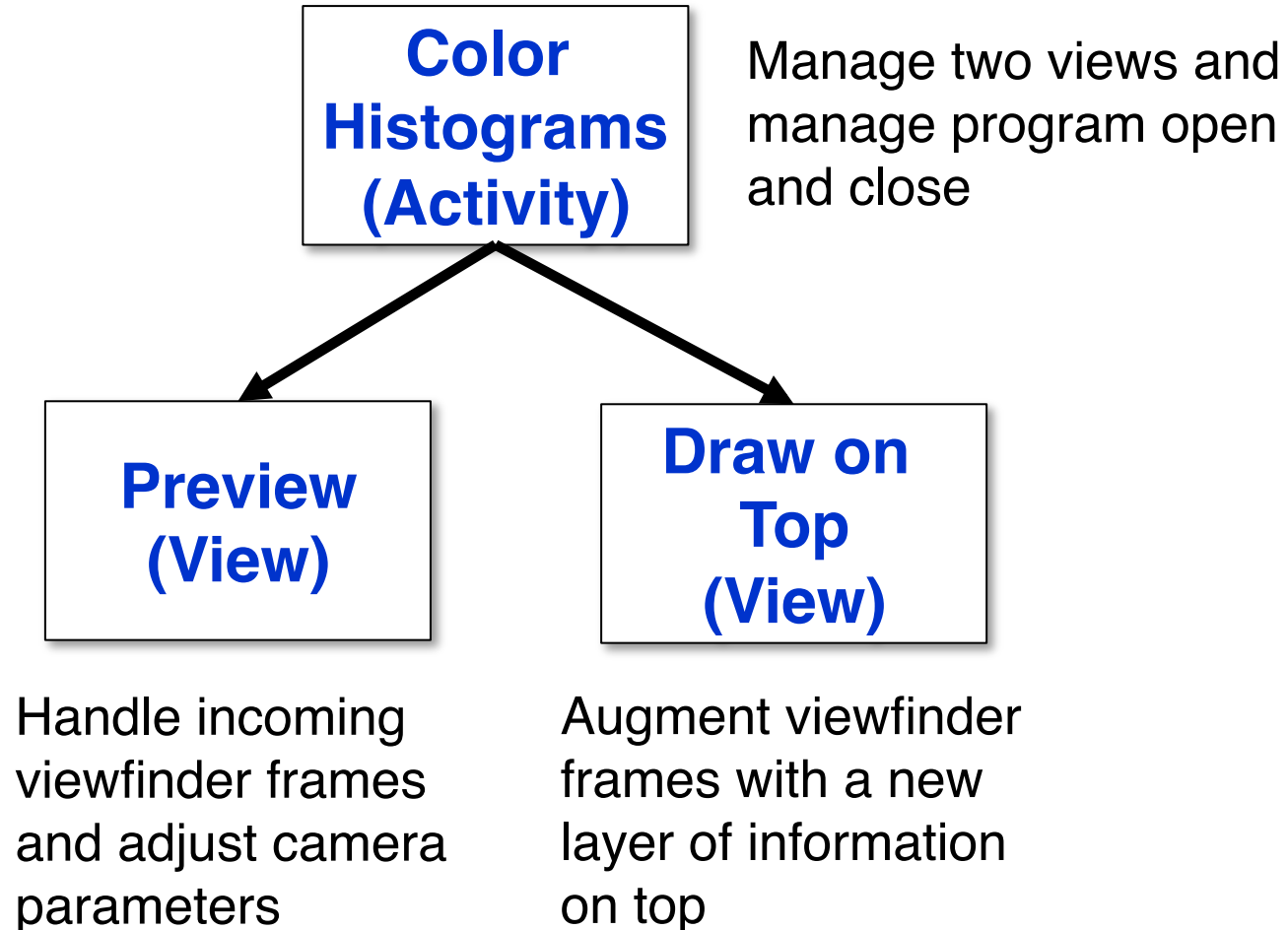
```
...  
<application android:icon="@drawable/ic_launcher"  
             android:label="@string/app_name">  
<activity android:name=".ViewfinderEE368"  
          android:label="@string/app_name"  
          android:screenOrientation="landscape"  
          android:theme="@android:style/Theme.NoTitleBar">  
<intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
</intent-filter>  
</activity>  
</application>  
  
<uses-sdk android:minSdkVersion="8" />  
<uses-permission android:name="android.permission.CAMERA" />  
...
```

Set landscape orientation
for viewfinder app

Set this activity as the
application's main activity

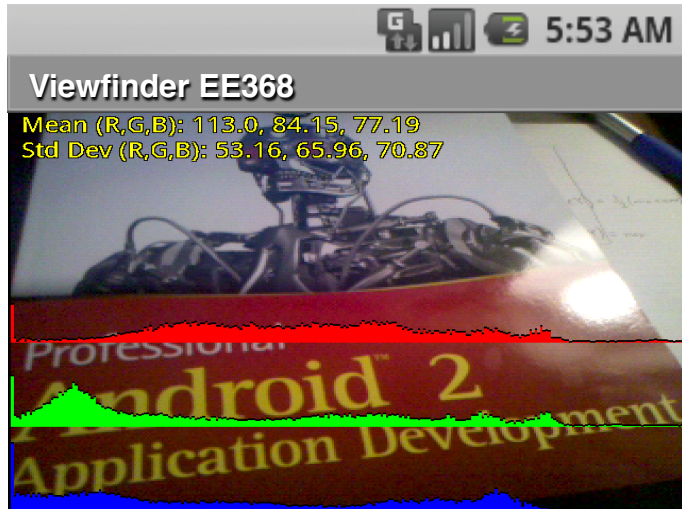
Declare permission to use
the device's camera

“Color Histograms” class hierarchy



Viewfinder class: full screen mode

Icon and title bars visible



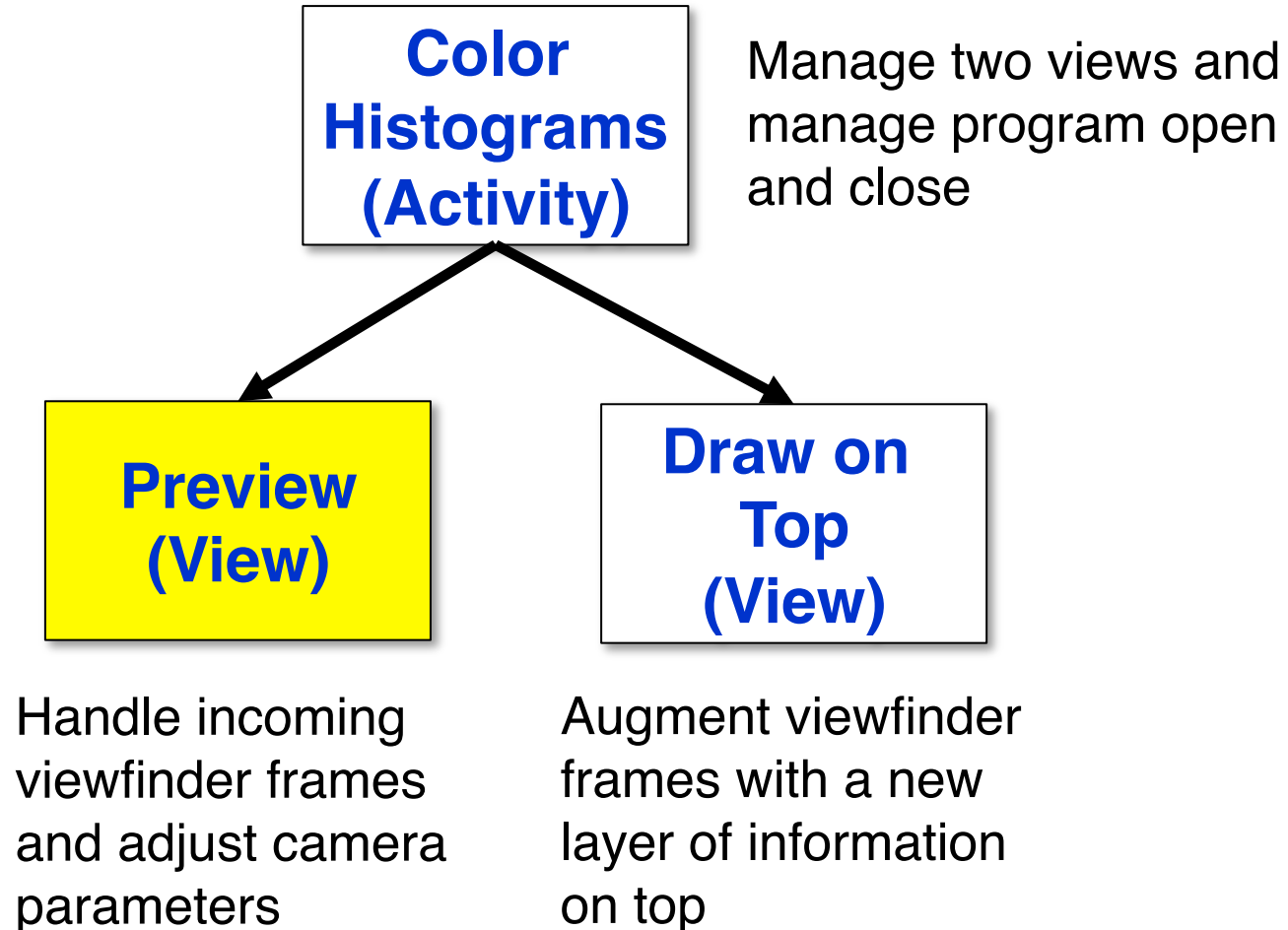
Icon and title bars hidden



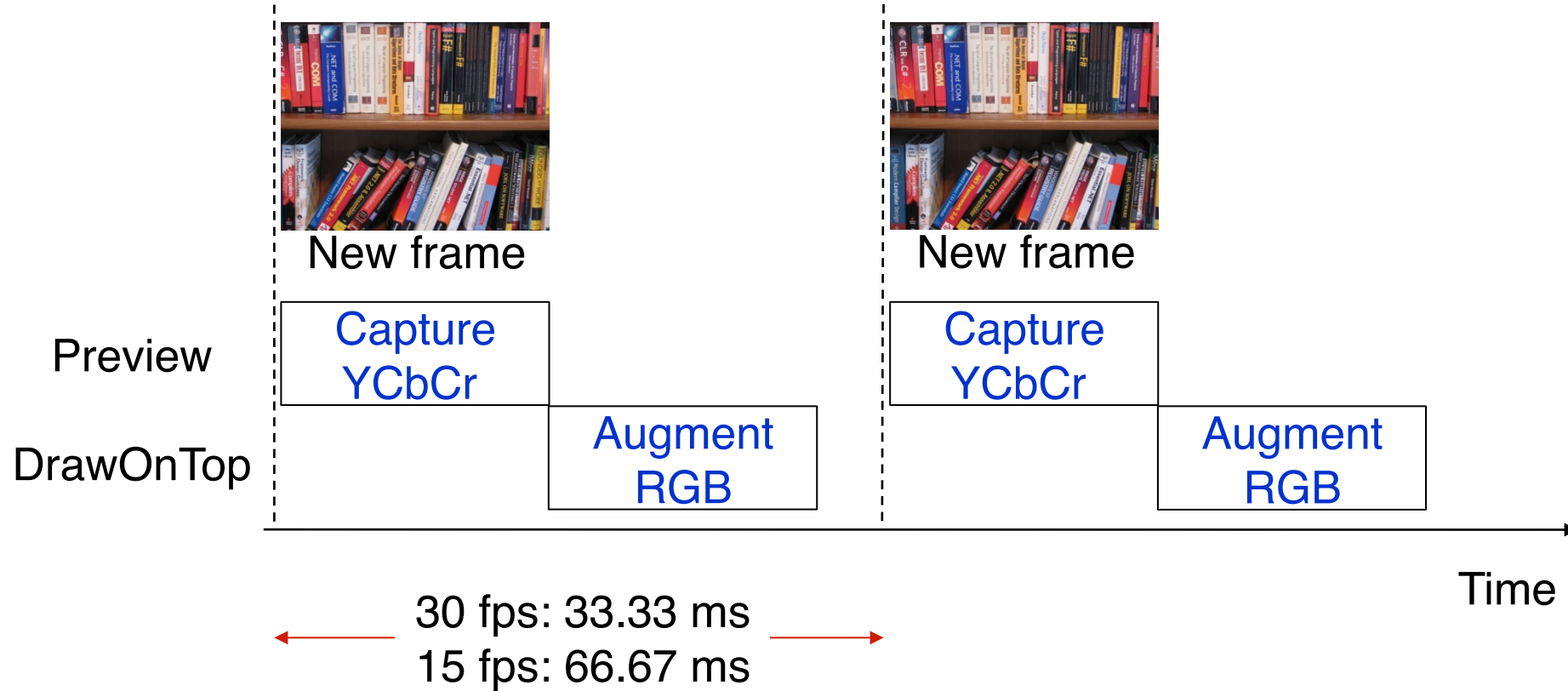
```
public void onCreate(Bundle savedInstanceState) {  
    ...  
    getWindow().setFlags(  
        WindowManager.LayoutParams.FLAG_FULLSCREEN,  
        WindowManager.LayoutParams.FLAG_FULLSCREEN  
    );  
    requestWindowFeature(Window.FEATURE_NO_TITLE);  
    ...  
}
```

???

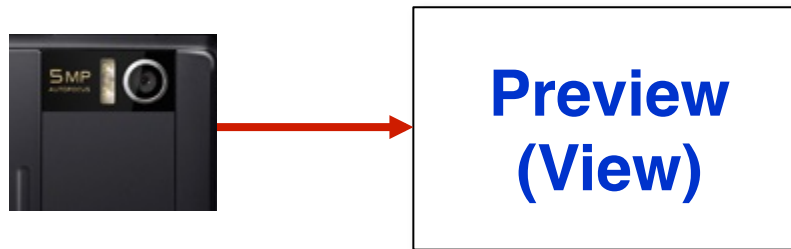
“Color Histograms” class hierarchy



Timeline of events on the mobile device

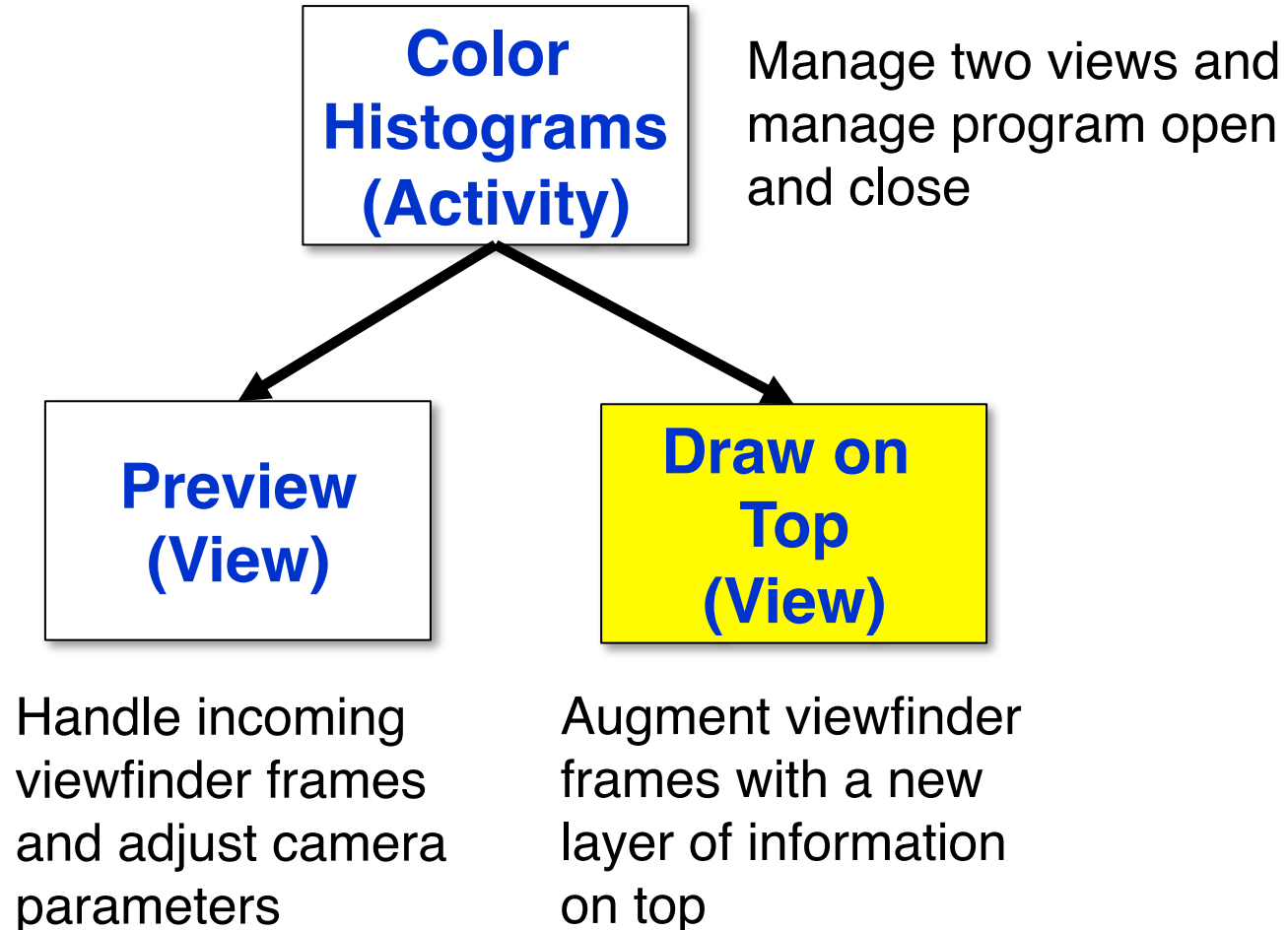


Preview class: set camera parameters

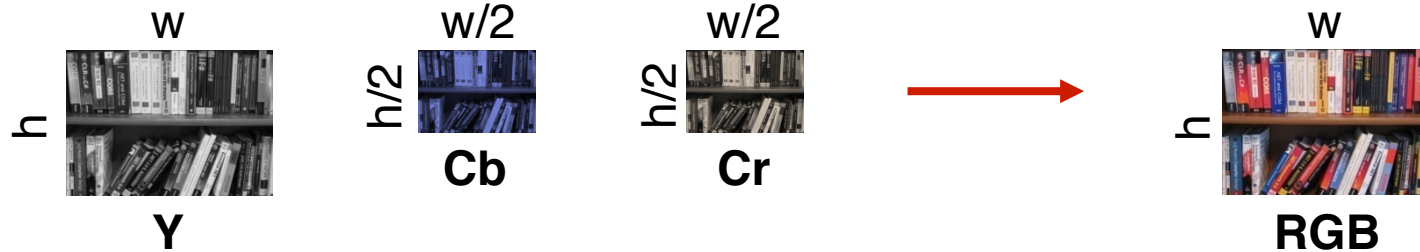


```
Camera.Parameters parameters = myCamera.getParameters();  
parameters.setPreviewSize(640, 480);  
parameters.setPreviewFrameRate(15);  
parameters.setSceneMode(Camera.Parameters.SCENE_MODE_NIGHT);  
parameters.setFocusMode(Camera.Parameters.FOCUS_MODE_AUTO);  
myCamera.setParameters(parameters);
```

“Color Histograms” class hierarchy



Draw on Top class: YCbCr to RGB conversion



$$R = \frac{298.08 \cdot Y}{256} + \frac{408.58 \cdot Cr}{256} - 222.921$$
$$G = \frac{298.08 \cdot Y}{256} - \frac{100.29 \cdot Cb}{256} - \frac{208.12 \cdot Cr}{256} + 135.58$$
$$B = \frac{298.08 \cdot Y}{256} + \frac{516.41 \cdot Cb}{256} - 276.84$$

Draw on Top class: create some paint brushes

```
myPaintRed = new Paint();  
myPaintRed.setStyle(Paint.Style.FILL);  
myPaintRed.setColor(Color.RED);  
myPaintRed.setTextSize(25);
```

```
myPaintGreen = new Paint();  
myPaintGreen.setStyle(Paint.Style.FILL);  
myPaintGreen.setColor(Color.GREEN);  
myPaintGreen.setTextSize(25);
```

```
myPaintBlue = new Paint();  
myPaintBlue.setStyle(Paint.Style.FILL);  
myPaintBlue.setColor(Color.BLUE);  
myPaintBlue.setTextSize(25);
```



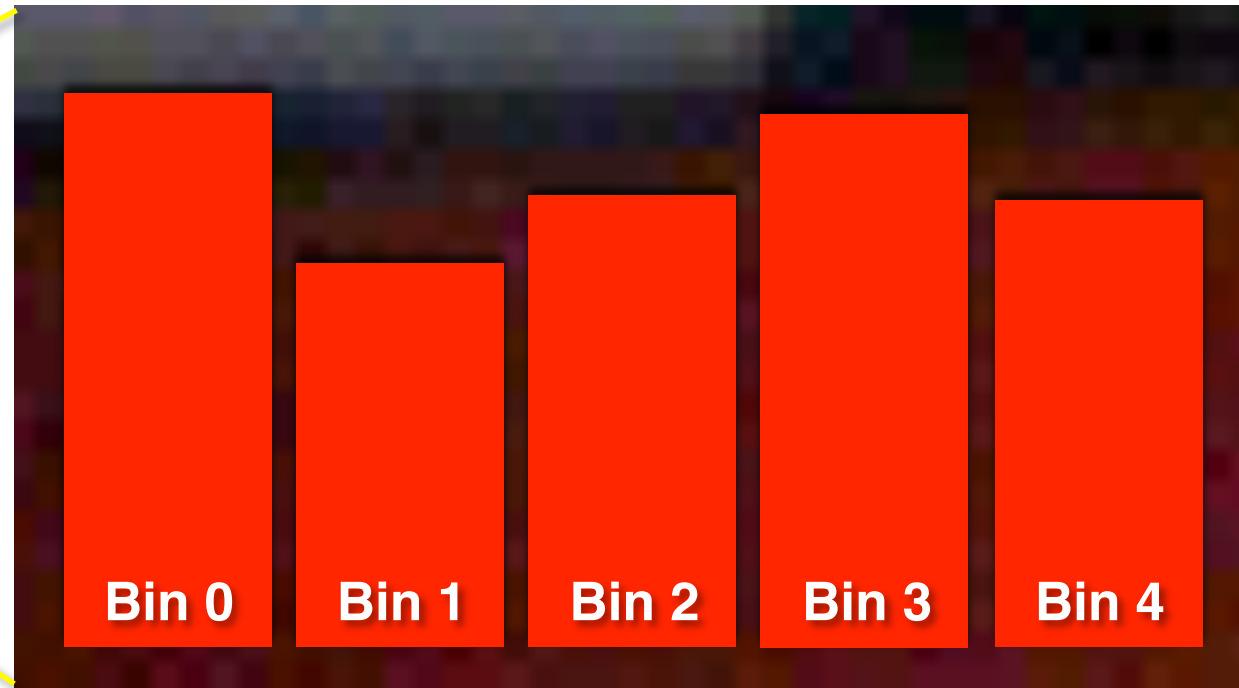
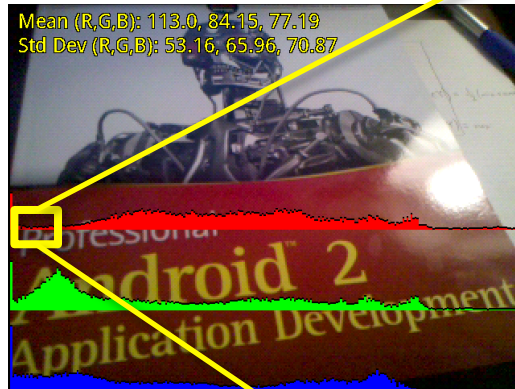
Draw on Top class: write text on canvas



Mean (R,G,B): 113.0, 84.15, 77.19
Std Dev (R,G,B): 53.16, 65.96, 70.87

```
String meanStr = "Mean (R,G,B): " +  
    String.format("%.4g", imageRedMean) + ", " +  
    String.format("%.4g", imageGreenMean) + ", " +  
    String.format("%.4g", imageBlueMean);  
myCanvas.drawText(meanStr, x, y, mPaintYellow);
```

Draw on Top class: draw histograms on canvas



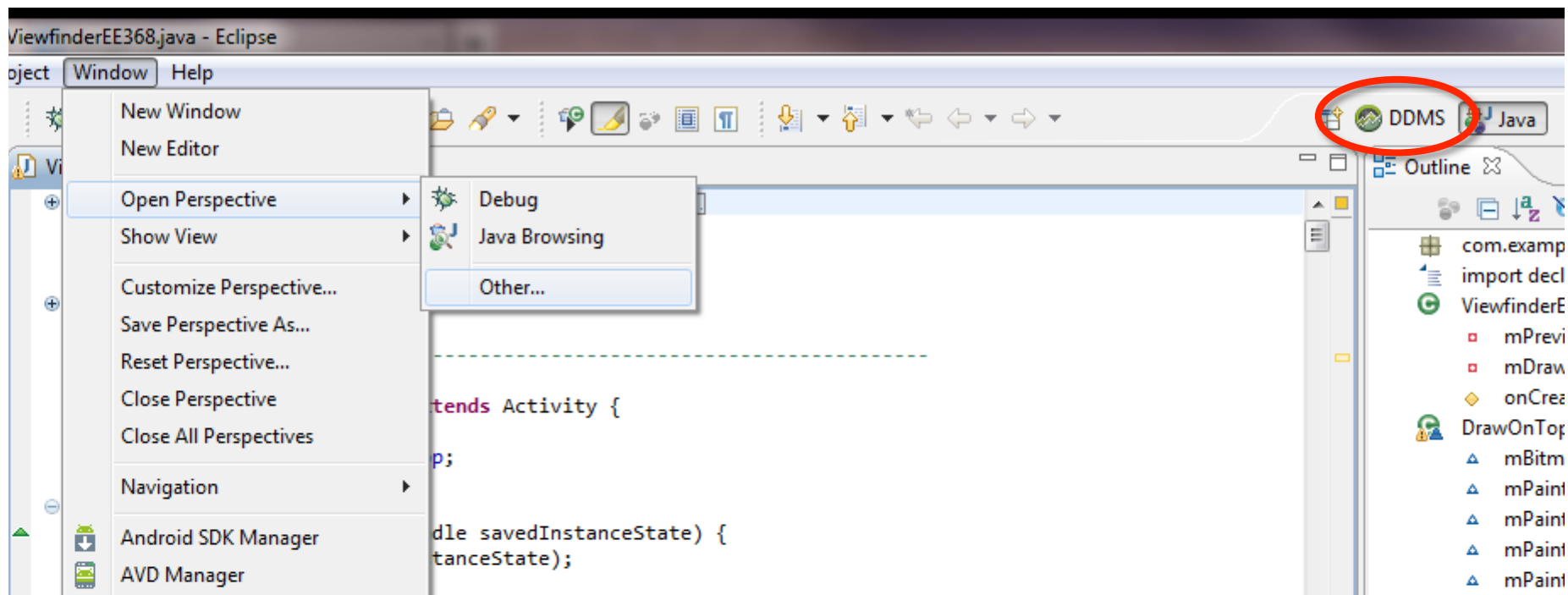
```
// Draw red rectangle  
Rect rect = new Rect(  
    left x, top y, right x, bottom y  
);  
myCanvas.drawRect( rect, myPaintRed );
```

“Color Histograms” app running on device



???

Real-time debugging with DDMS perspective



Real-time debugging with DDMS perspective

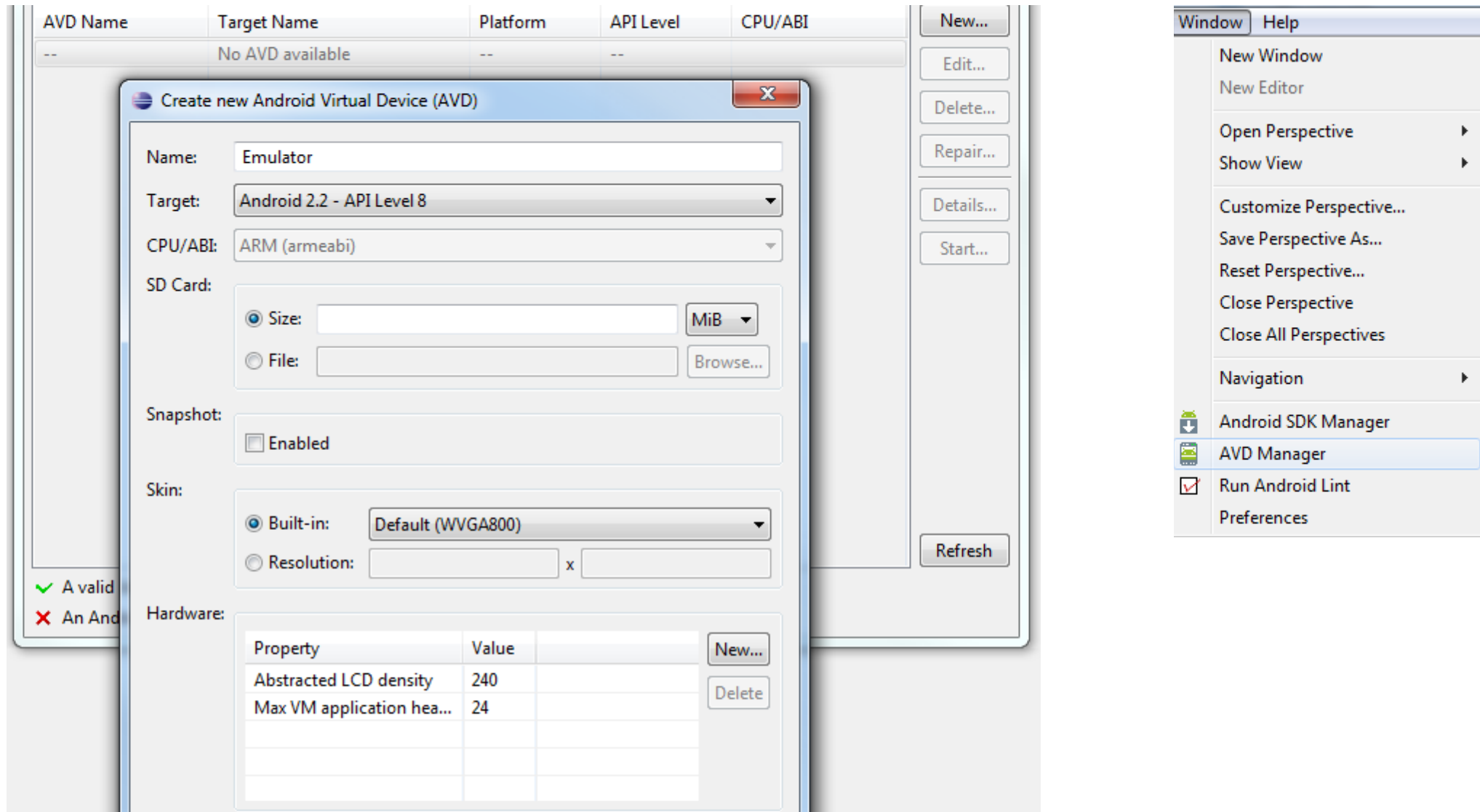
The screenshot displays the DDMS (Dalvik Debug Monitor Service) interface. It is divided into several panes:

- Devices:** A table listing connected devices. A yellow callout box labeled "Device" points to the first row.
- File Explorer:** A pane showing the file system structure of the selected device. A yellow callout box labeled "File system on device" points to the directory listing.
- Emulator Control:** A pane for controlling the device's telephony status, including voice and data settings.
- LogCat:** A pane showing the system log. A yellow callout box labeled "Messages on the device" points to the log entries.

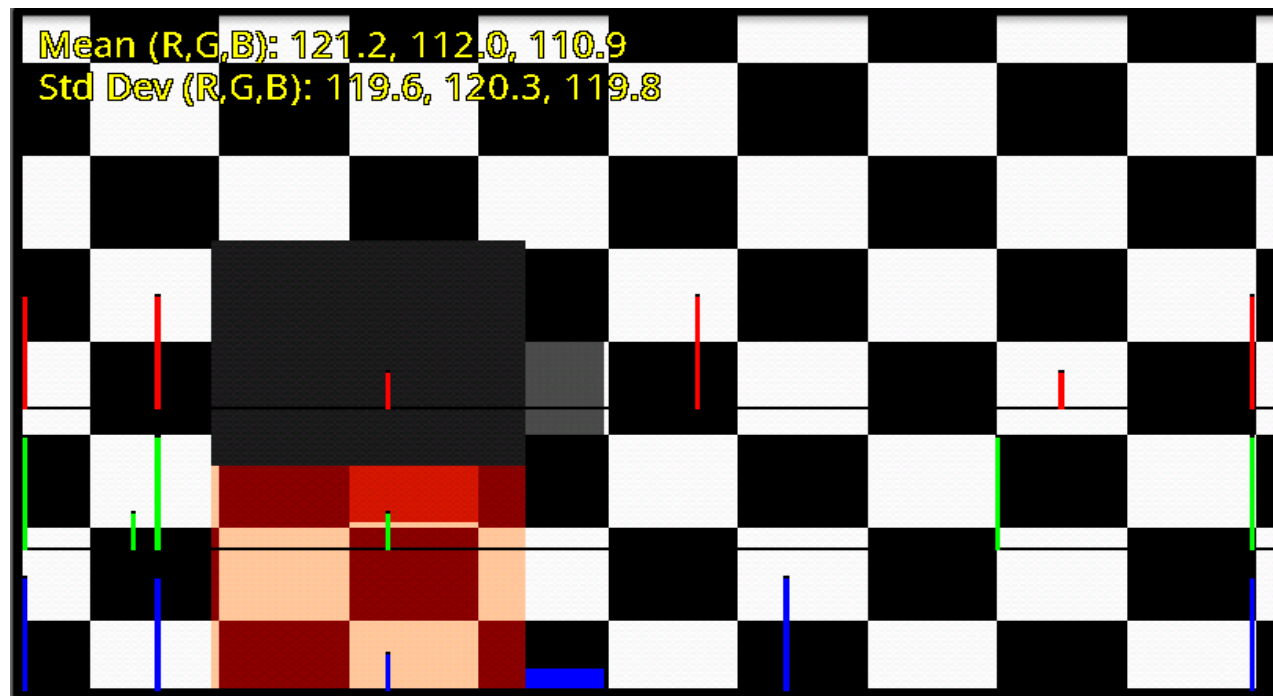
Name	Size	Date	Time	Perm
data		2010-03-16	17:32	drwx
sdcard		2010-04-05	12:54	d---r
system		2010-04-05	12:52	drwx

Time	pid	tag	Message
04-08 09:23...	I 9414	dalvikvm	Debugger th
04-08 09:23...	I 9414	ActivityThread	Publishing
04-08 09:23...	I 9414	ActivityThread	Publishing
04-08 09:23...	I 9414	ActivityThread	Publishing

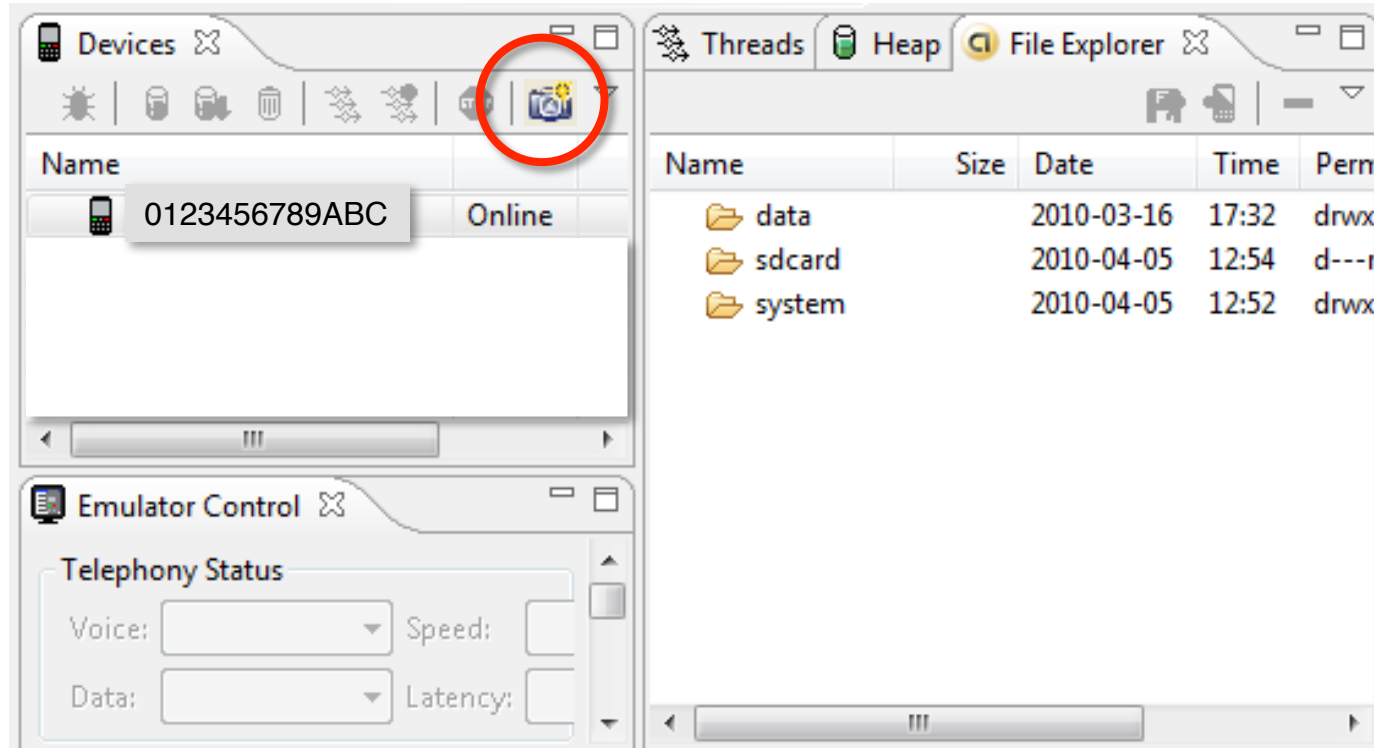
Creating an Android emulator



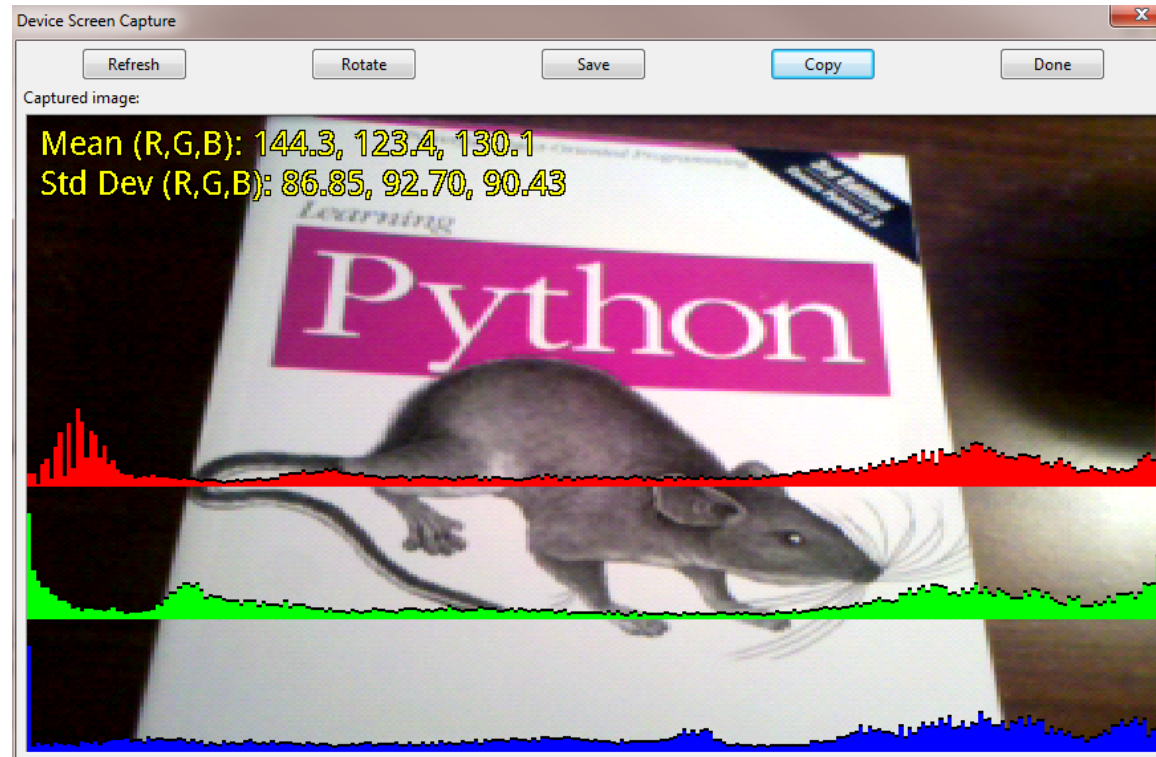
“Color Histograms” app running on emulator



Taking a screenshot of the device

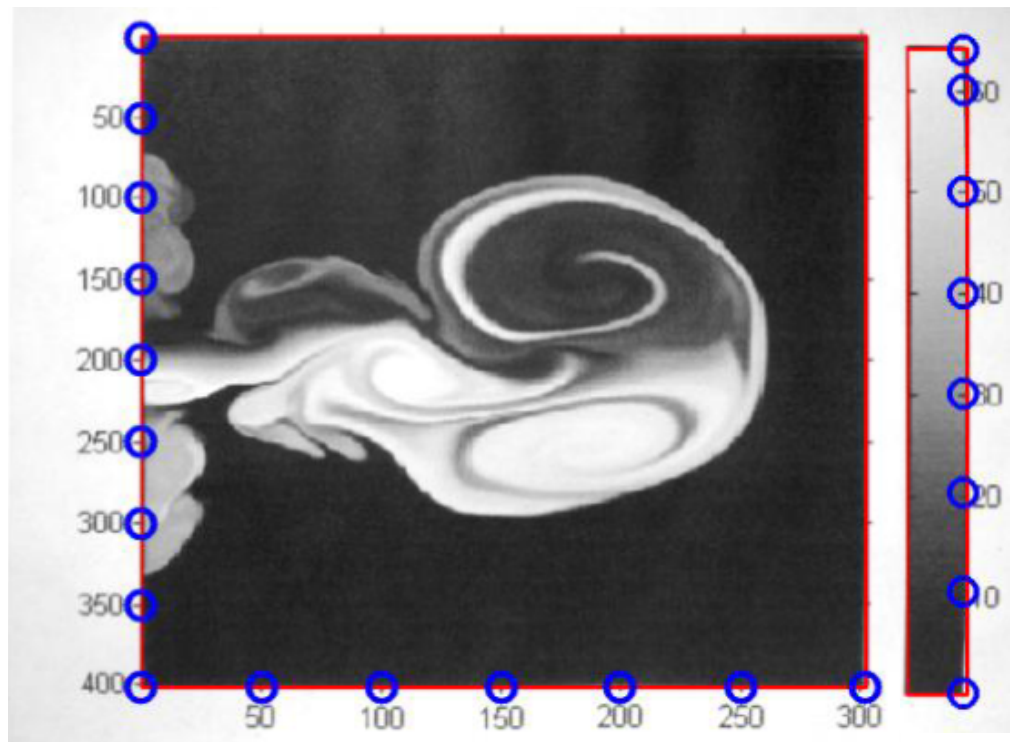
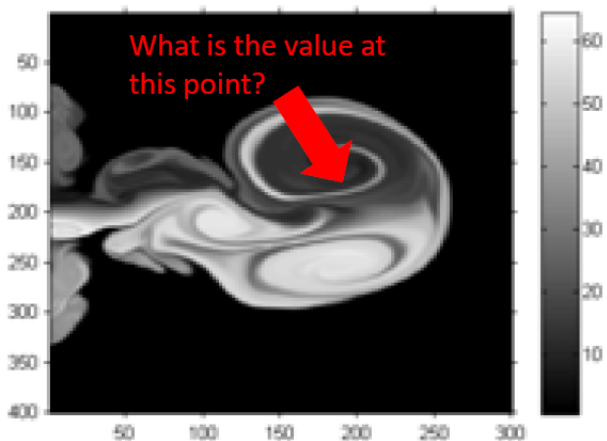


Taking a screenshot of the device



???

Class project: mobile graph reader



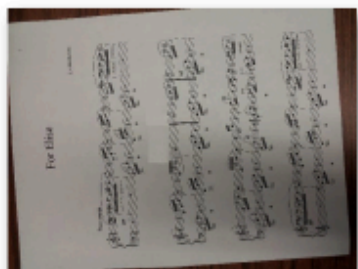
Segmentation of Image

Detection of Tick Marks

OCR of Text Labels

T. Jou, W. Ni, J. Su, Spring 2011
http://ee368.stanford.edu/Project_11

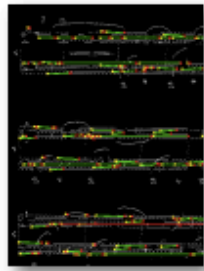
Class project: optical music recognition and playback



Captured Image of music staff



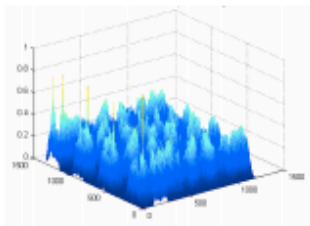
Upright Image after rotation



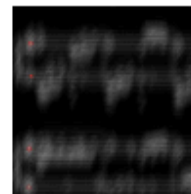
Horizontal Line Detection



Clef Template



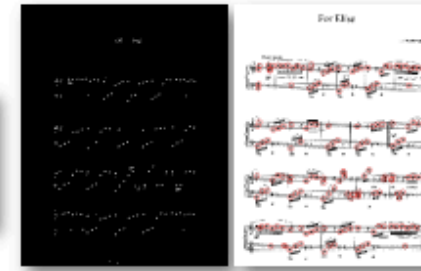
Clef Peak Detection



Matched Clefs



Top & Bottom Stave with red marks



Detected Note Heads



S. Dai, C.-W. Lee, Y. Tian, Spring 2012
http://ee368.stanford.edu/Project_12

???