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

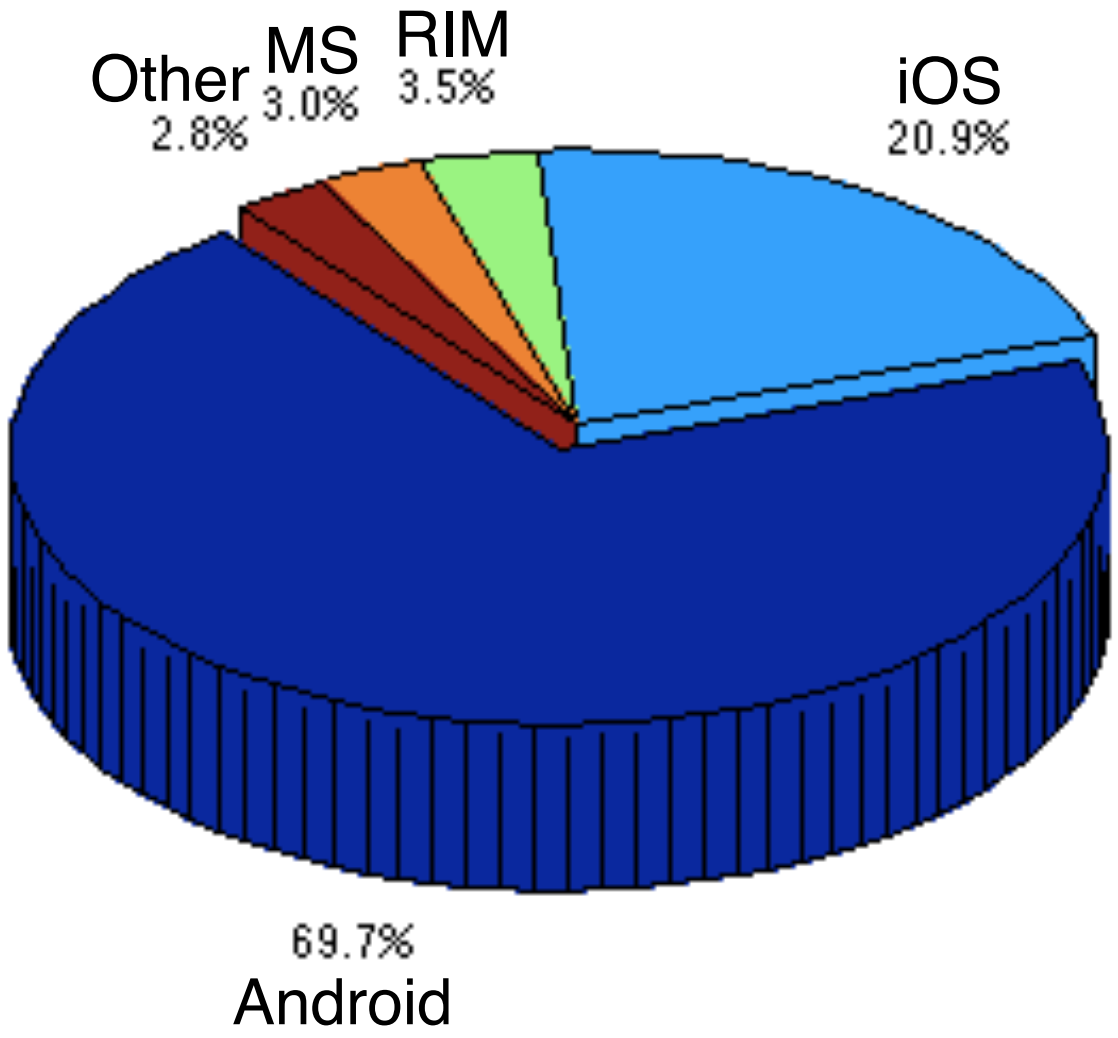
Mobile landmark recognition

Mobile product recognition

Spotlight on Android

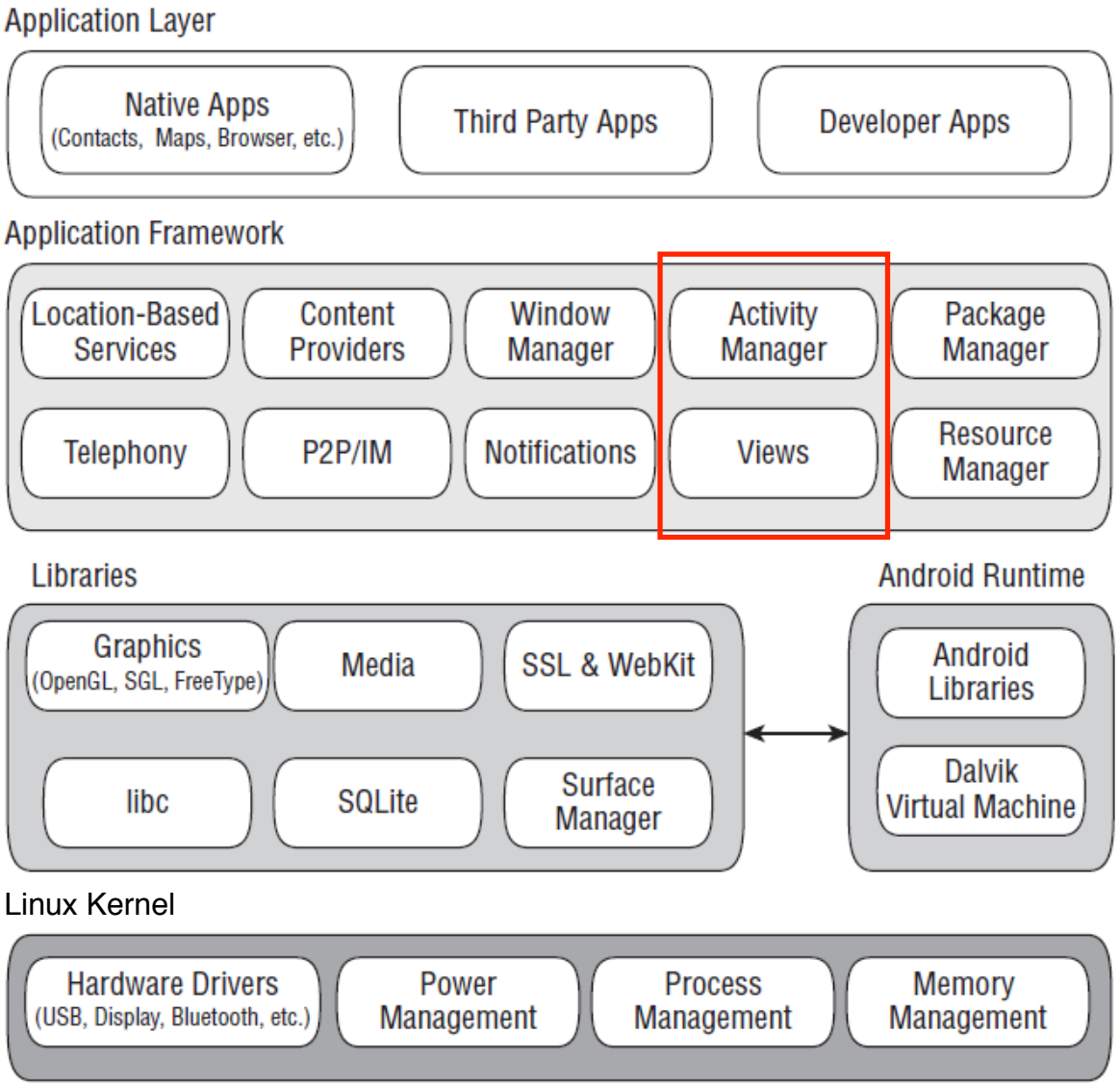
- Open source mobile platform developed by Google
- Supported and maintained by Open Handset Alliance
  - 13 mobile operators
  - 22 handset manufacturers
  - 20 semiconductor companies
  - 17 software makers
- Uses an open-source kernel and a virtual machine designed for mobile hardware
- Commands the largest share of the smartphone market in the world
- Google Play contains over 800k apps

Distribution of worldwide smartphone sales in Q4 2012



World Smartphone Sales to End Users
Gartner (Q4 2012)

Android software stack

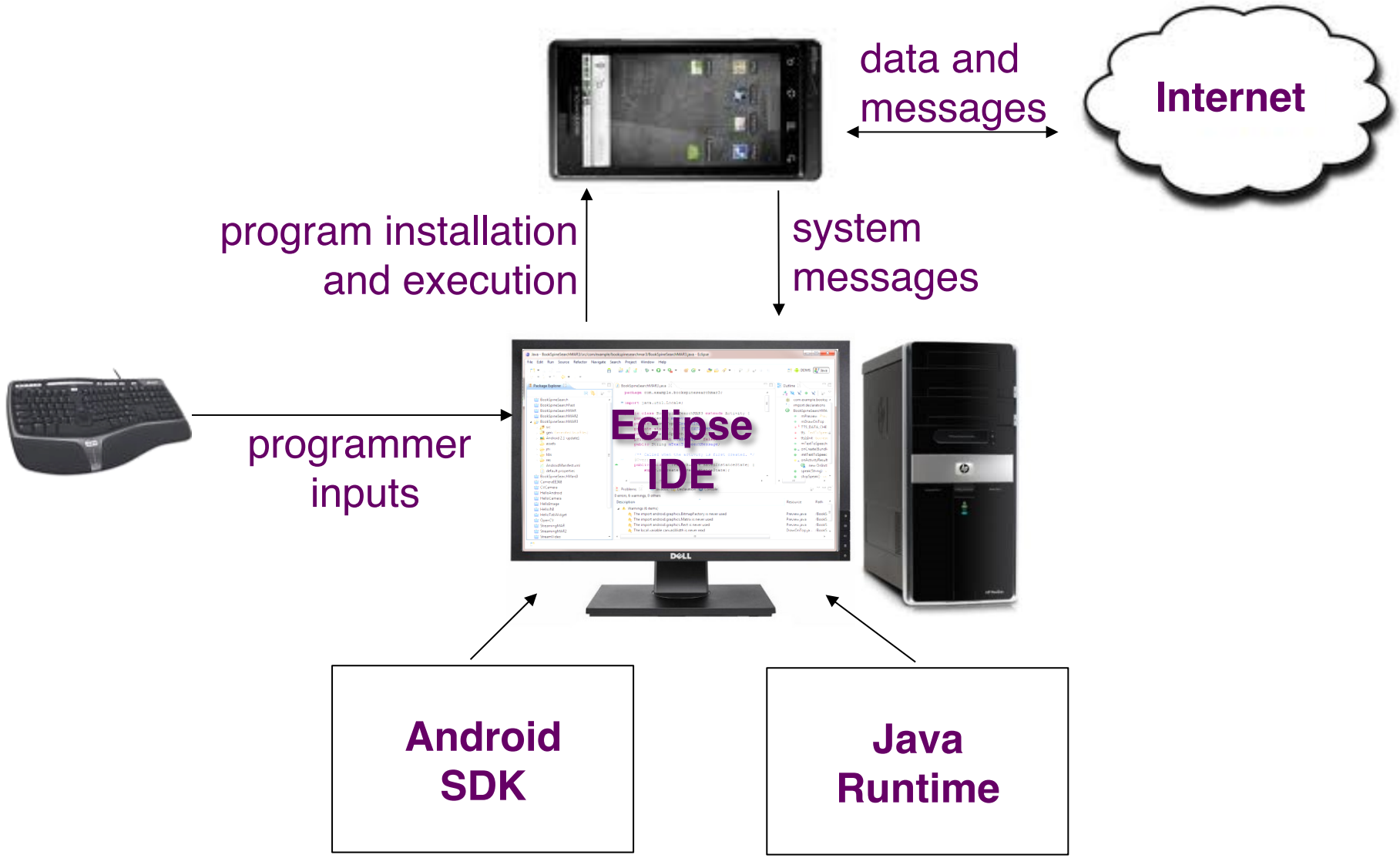


Professional Android Application Development

Programming Android applications in Java

- Android encourages high-level app development
- Android uses Java as the main programming language
- Android inherits basic classes from conventional Java
  - String, Container, Math, IO, Network, …
- Android also adds new classes specific to mobile devices
  - Camera, Telephony, Map, GPS, Speech, …

Android development with Eclipse

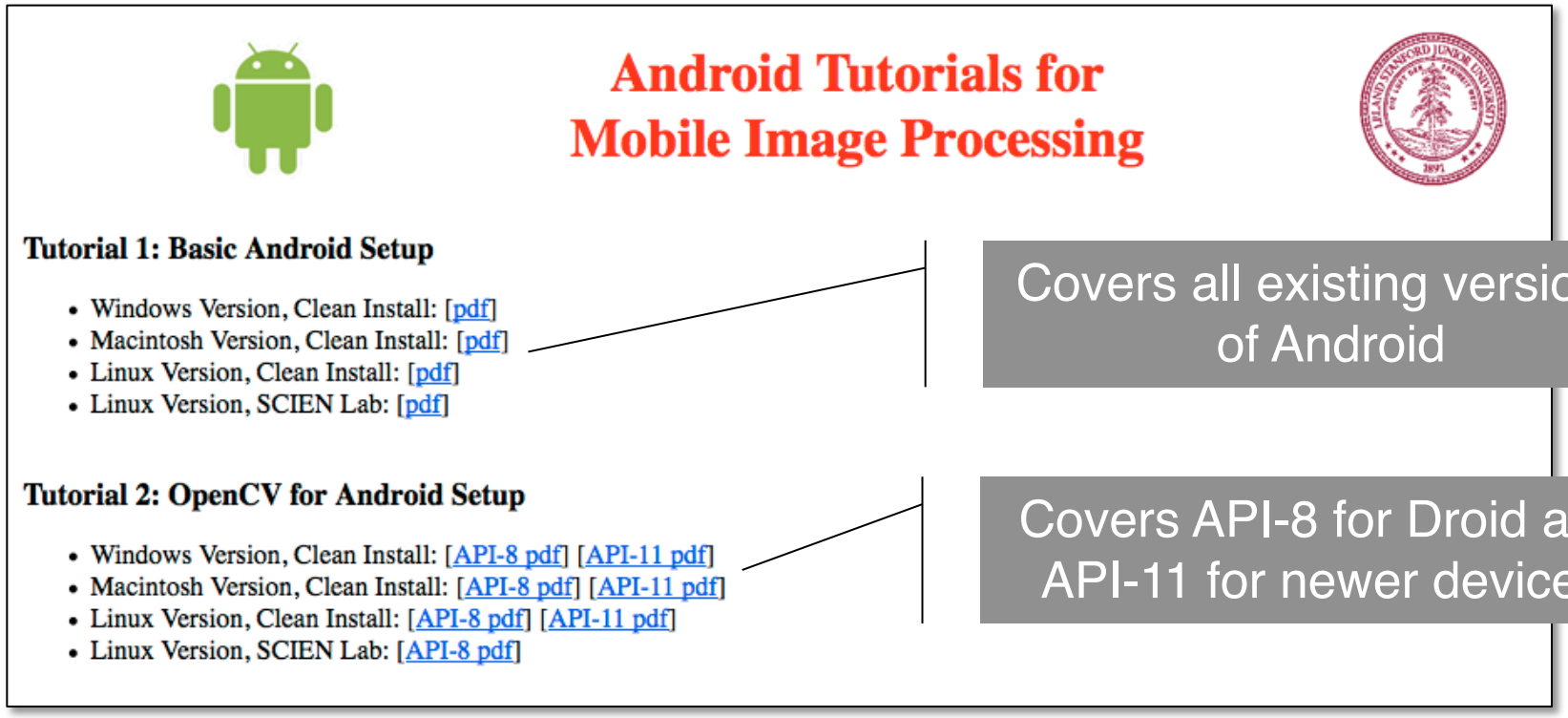


???

Class tutorials for mobile image processing

- Android tutorials designed specifically for mobile image processing applications in the EE368/CS232 class
- Tutorial #1: Basic Android Setup
  - Image processing-oriented introduction to Android
  - Explains how to download and install the different software packages (SDK, Eclipse) on your own computer
  - Shows how to build and run viewfinder augmentation apps in real time
- Tutorial #2: OpenCV for Android Setup
  - Builds on core skills developed in Tutorial #1
  - Explains how to download and install OpenCV for Android
  - Shows how to build viewfinder apps that detect circles and lines, detect feature keypoints, track feature keypoints, perform locally adaptive thresholding, detect human faces, …

Class tutorials for mobile image processing



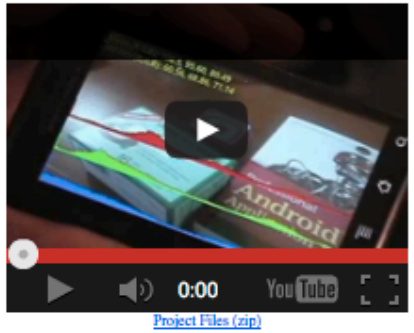
http://ee368.stanford.edu/Android

Class tutorials for mobile image processing

Histogram Equalization

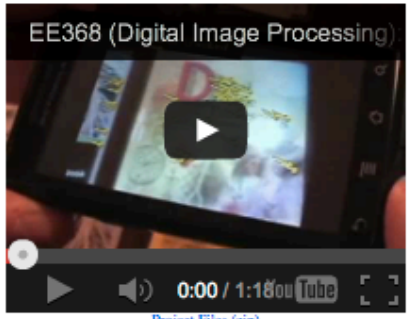
Project Files (zip)

Color Histograms



Project Files (zip)

Feature Tracking

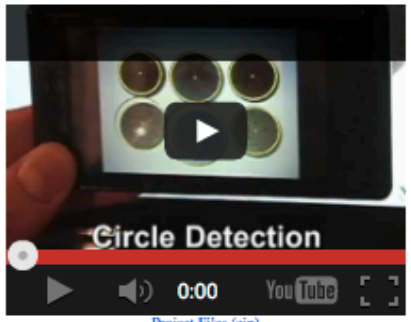


Project Files (zip)

Locally Adaptive Binarization

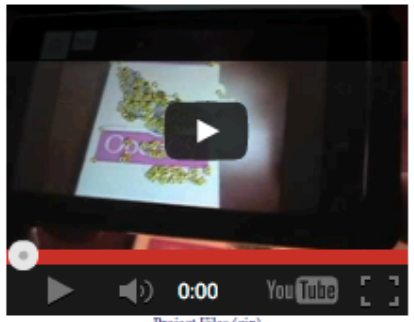
Project Files (zip)

Edges, Lines, and Circles



Project Files (zip)

Local Feature Keypoints



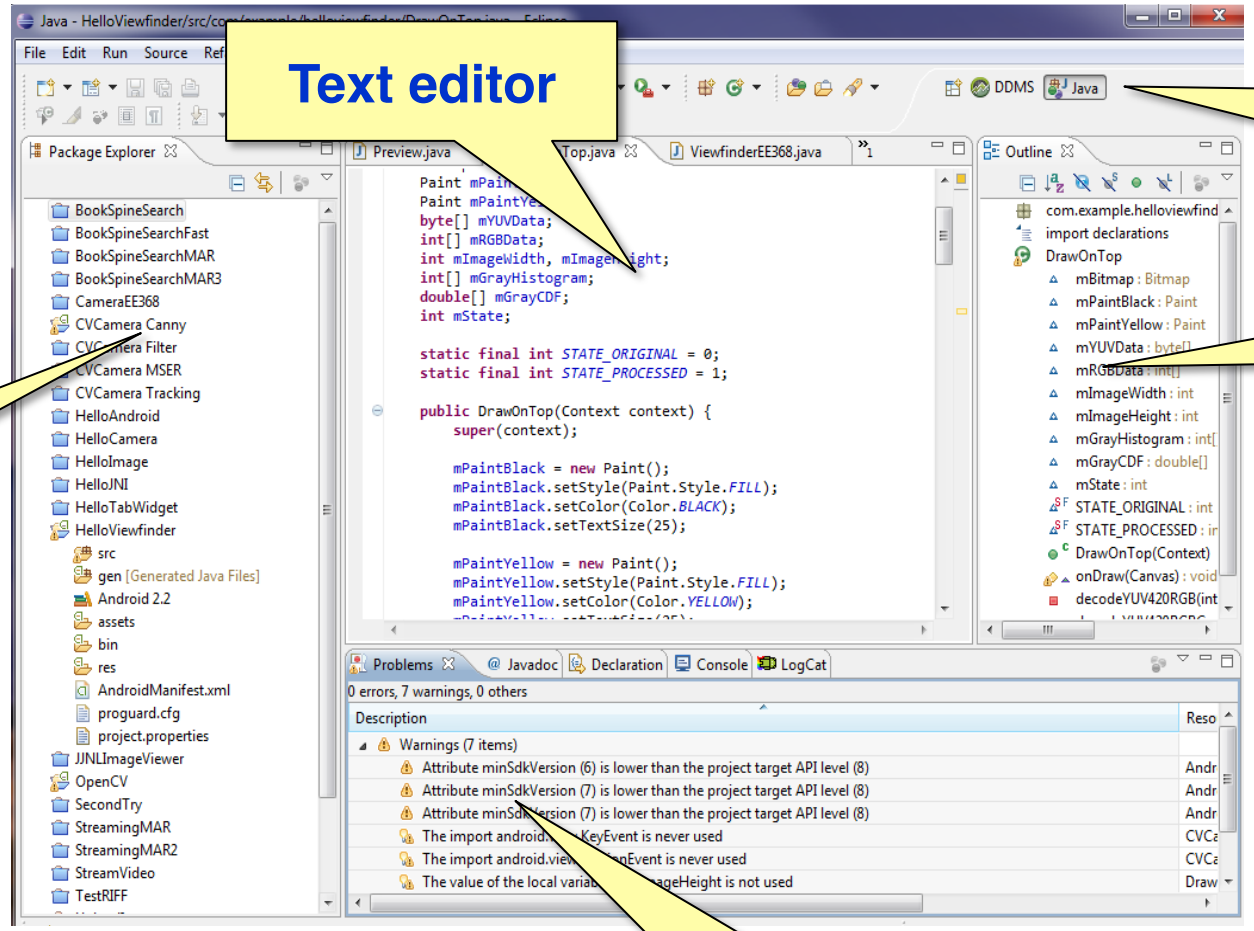
Project Files (zip)

Human Face Detection

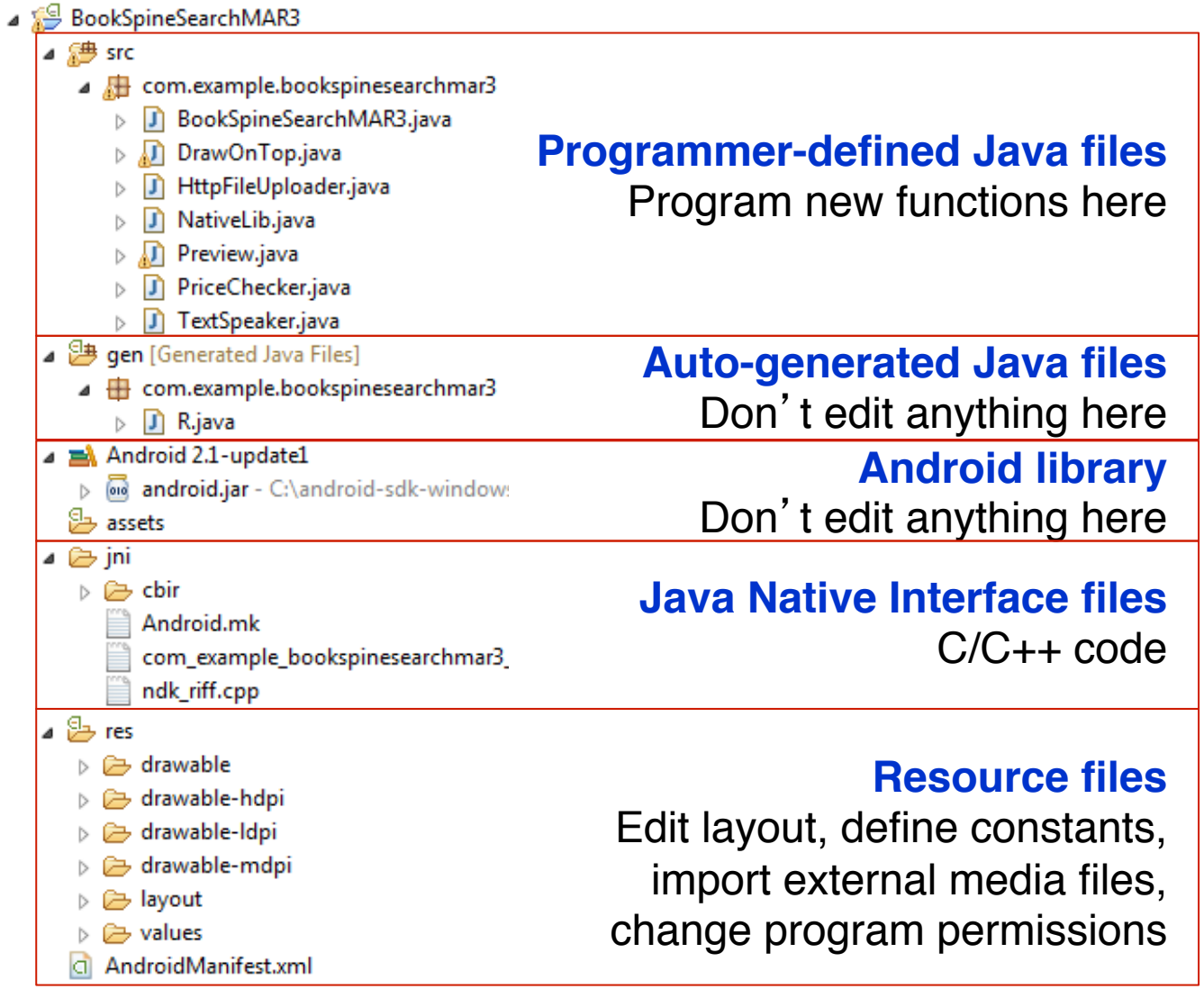
Project Files (zip)

http://ee368.stanford.edu/Android

Eclipse integrated development environment



Structure of an Android project

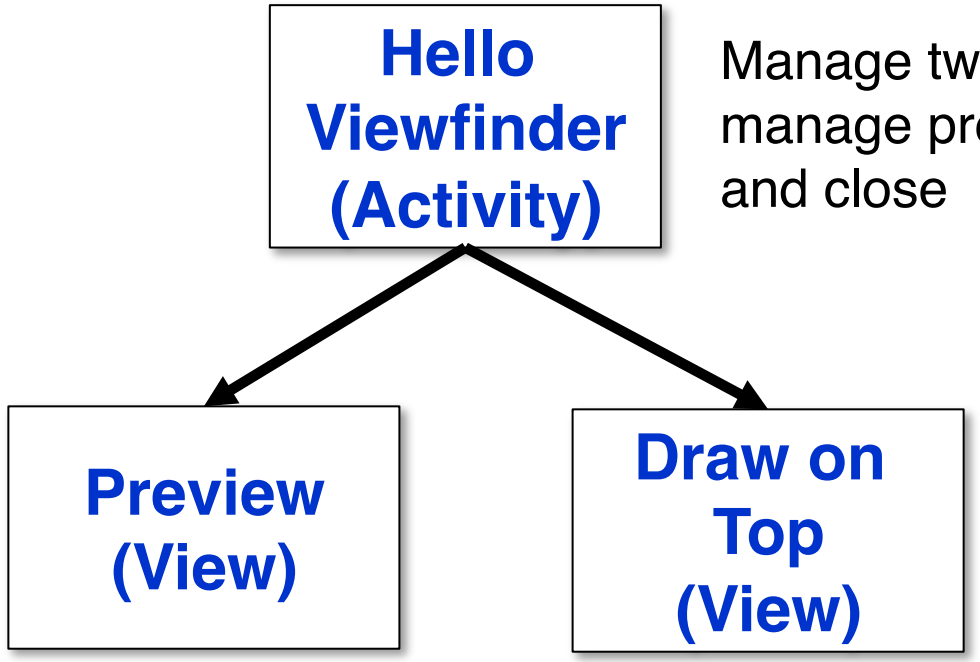


???

“Hello Viewfinder” project

- Goals of this project
  - Learn how to create a simple Android project
  - Learn how to display viewfinder frames
  - Learn how to process viewfinder frames
- Full source available on class website:
  - http://ee368.stanford.edu/Android/HelloViewfinder

“Hello Viewfinder” class hierarchy



Main activity class

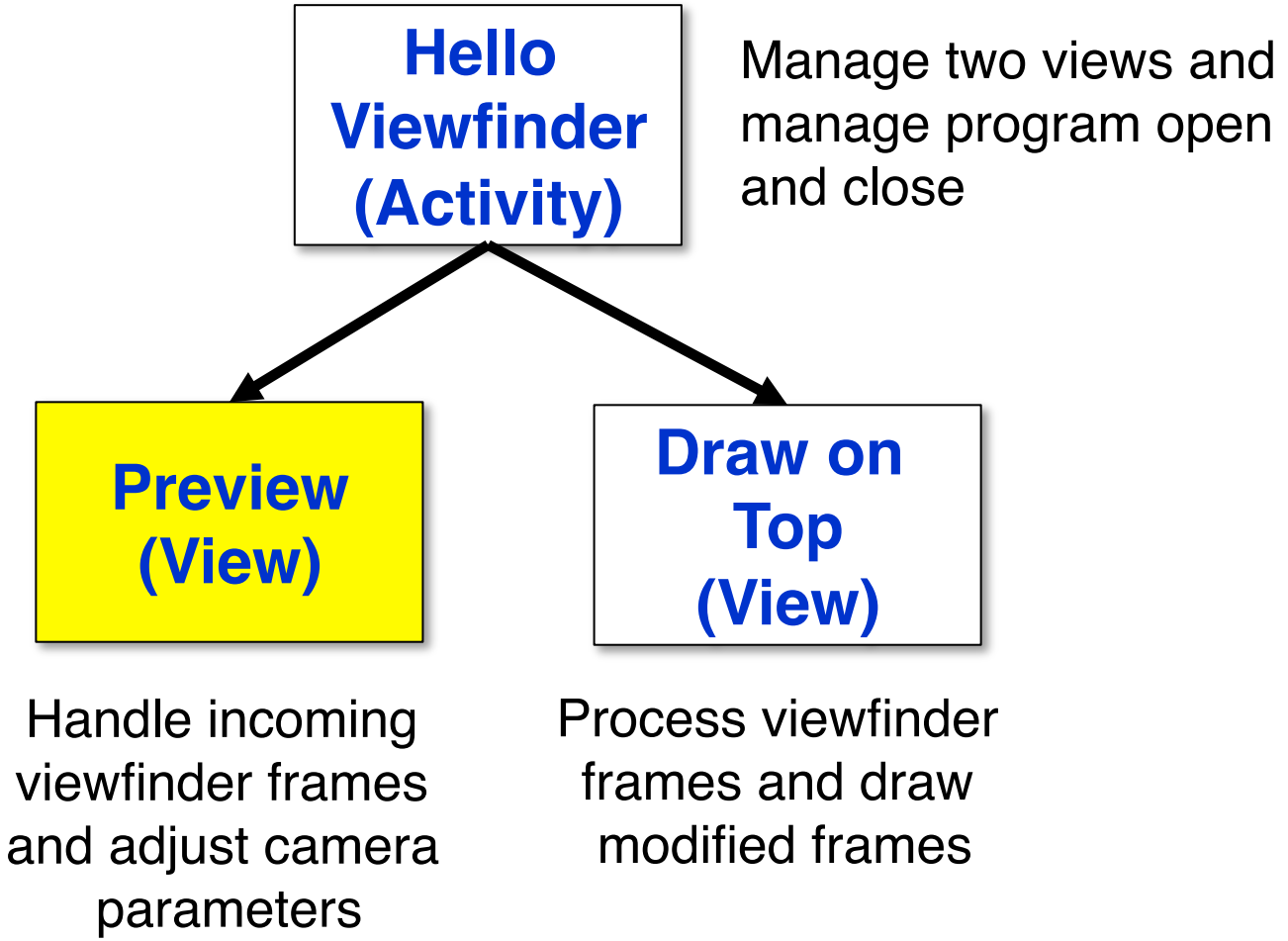
```
public class HelloViewfinderActivity extends Activity {
    private Preview mPreview;
    private DrawOnTop mDrawOnTop;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Hide the window title and set full screen
        getWindow().setFlags(... Full Screen Parameters ...);
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        // Create Preview and DrawOnTop
        mDrawOnTop = new DrawOnTop(this);
        mPreview = new Preview(this, mDrawOnTop);
        setContentView(mPreview);
        addContentView(mDrawOnTop, ... Layout Options ...)
    }
}
```

Make this class inherit the properties of an Activity

Called when the activity is first created

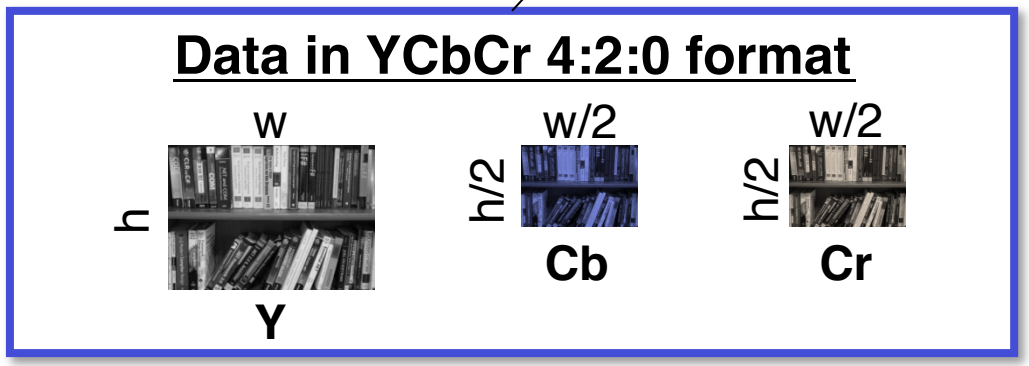
Create two children objects for displaying frames

“Hello Viewfinder” class hierarchy



Preview class: viewfinder frames go down two paths


```
myCamera.setPreviewCallback(new PreviewCallback() {
      public void onPreviewFrame(byte[] data, Camera camera)
      { ... Pass data to DrawOnTop class ... }
});
```



Preview class: toggle states via touch screen

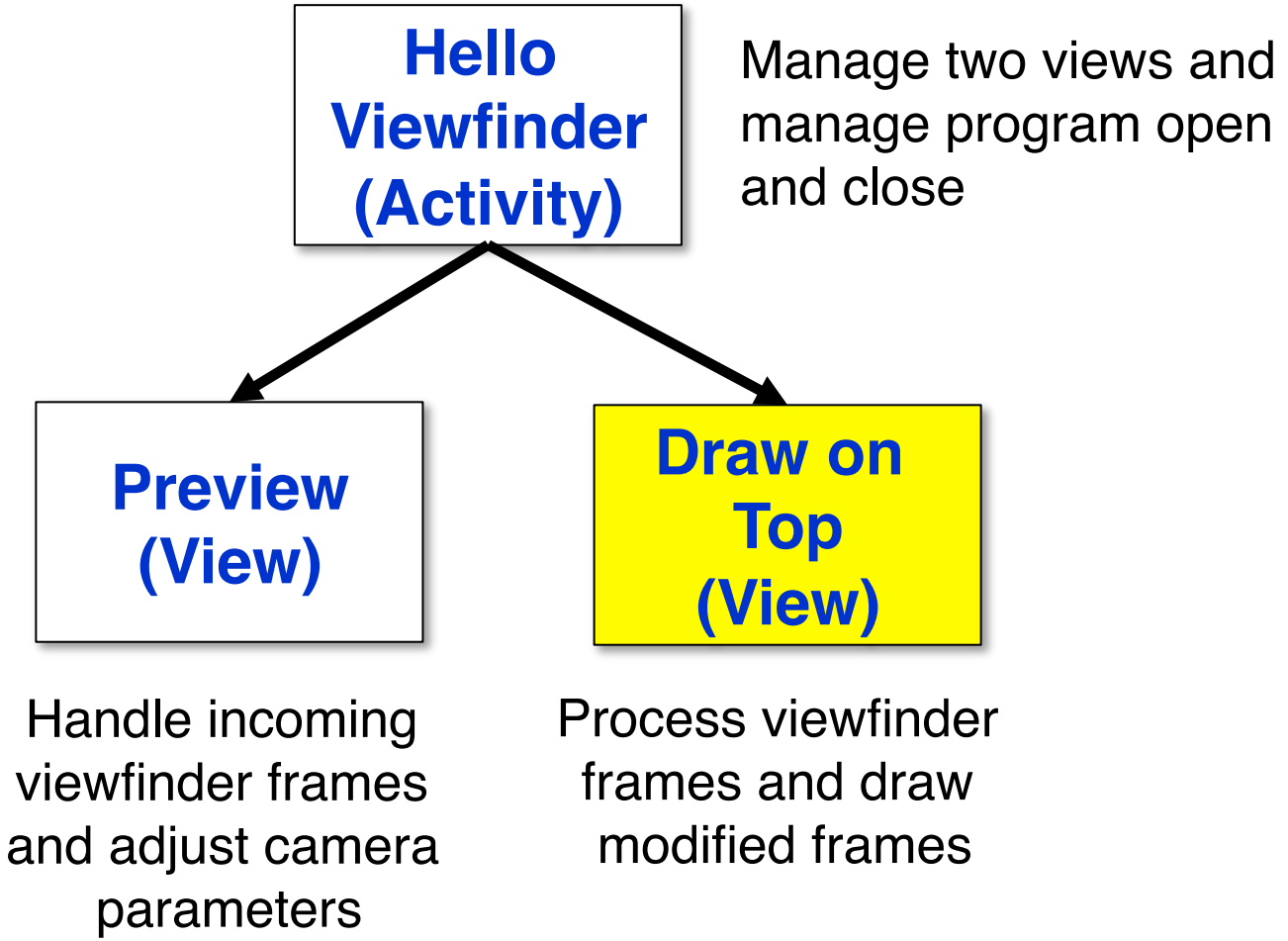
```
// Define on touch listener
this.setOnTouchListener(new OnTouchListener() {
   public boolean onTouch(View v, MotionEvent event)
   {
       if (mDrawOnTop.mState == DrawOnTop.STATE_ORIGINAL)
        {
          mDrawOnTop.mState = DrawOnTop.STATE_PROCESSED;
        }
        else if (mDrawOnTop.mState == DrawOnTop.STATE_PROCESSED)
        {
          mDrawOnTop.mState = DrawOnTop.STATE_ORIGINAL;
        }
        return false;
   }
});
```

Define an anonymous touch listener object

If in original state, toggle to processed state

If in processed state, toggle to original state

“Hello Viewfinder” class hierarchy



Draw-on-top class: process viewfinder frames

```
// Called whenever a repaint is requested
protected void onDraw(Canvas  canvas)
{
    ...
    // Convert from YCbCr to RGB
    if (mState == STATE_ORIGINAL)
         decodeYCbCr420RGB(mRGBData, mYCCData, mWidth, mHeight);
    else
         decodeYCbCr420RGBHistEq(mRGBData, mYCCData, mWidth, mHeight);

    // Draw bitmap
    mBitmap.setPixels(mRGBData, 0, mWidth, 0, 0, mWidth, mHeight);
    Rect src = new Rect(... Size parameters ...);
    Rect dst = new Rect(... Size parameters ...);
    canvas.drawBitmap(mBitmap, src, dst, mPaintBlack);
    ...
}
```

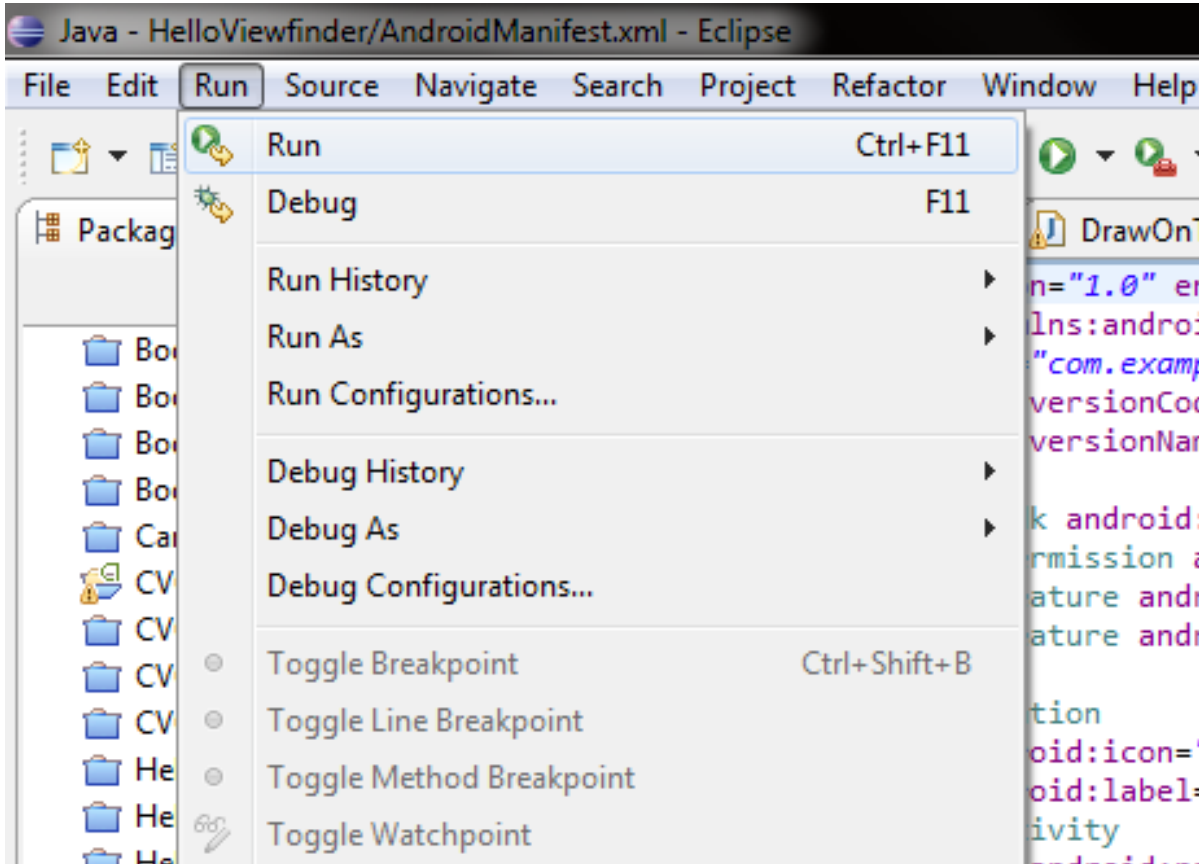
Called whenever this view is repainted

Decode frame with or without hist. eq.

Draw decoded frame in new layer

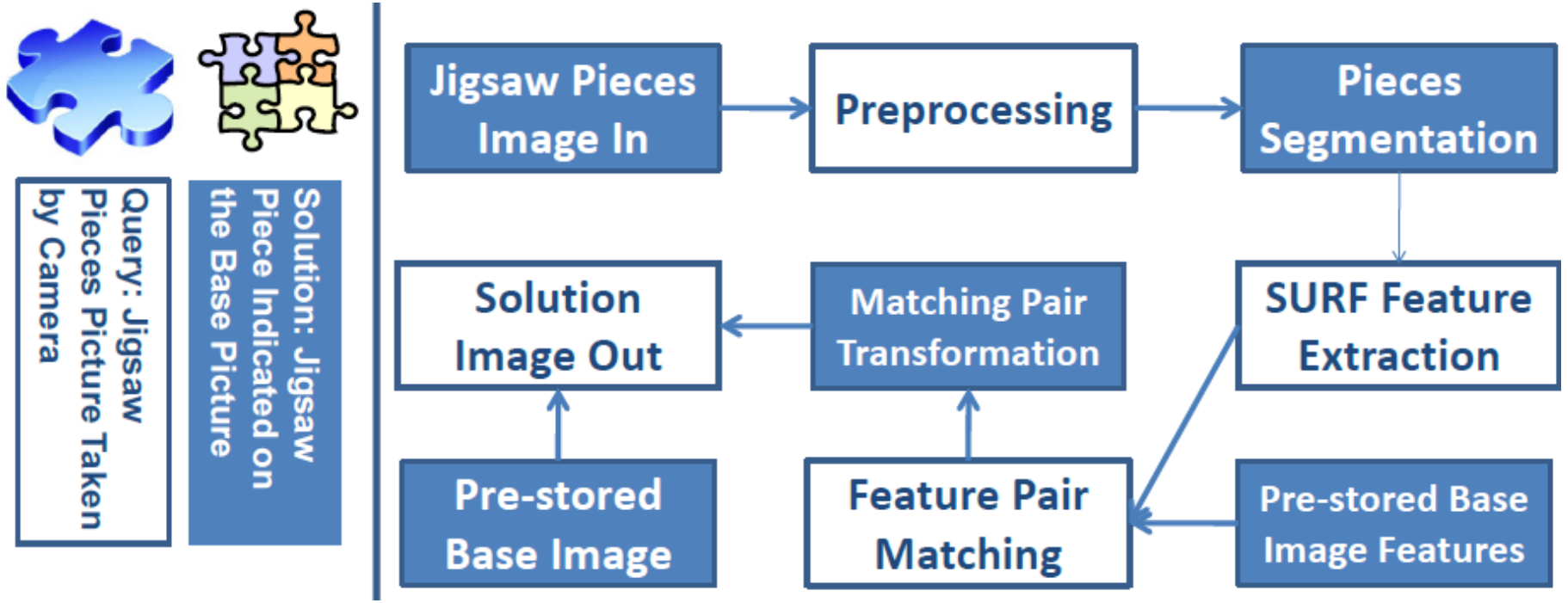
???

Running the program on an Android device



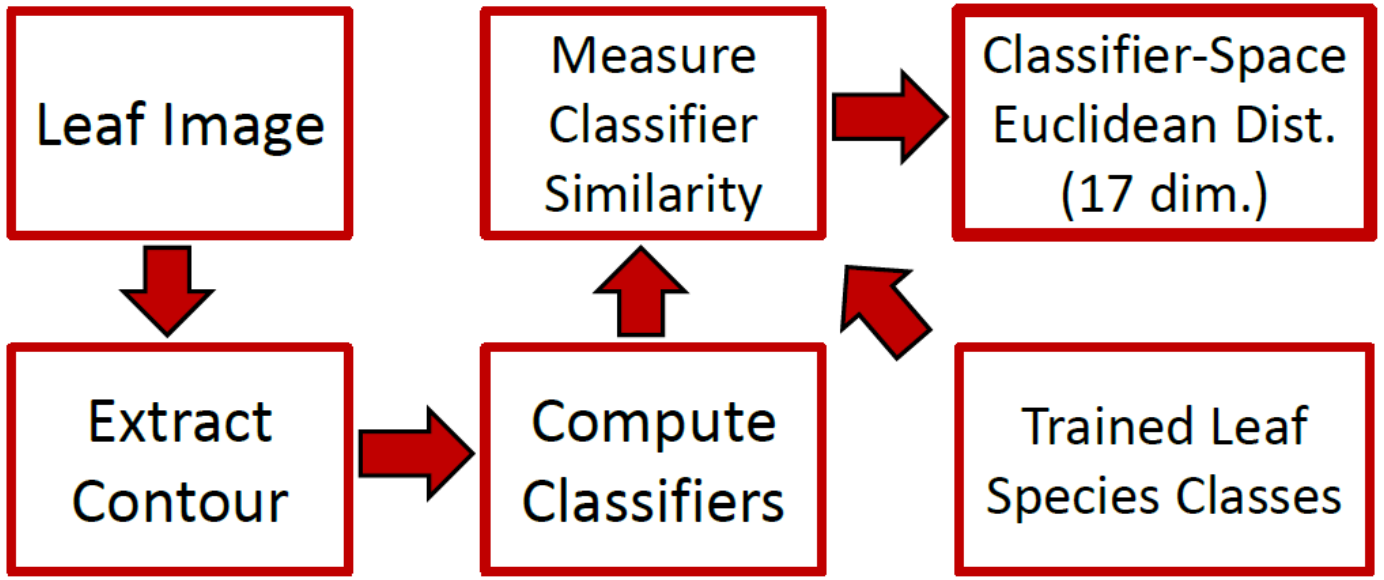
“Hello Viewfinder” application running on device

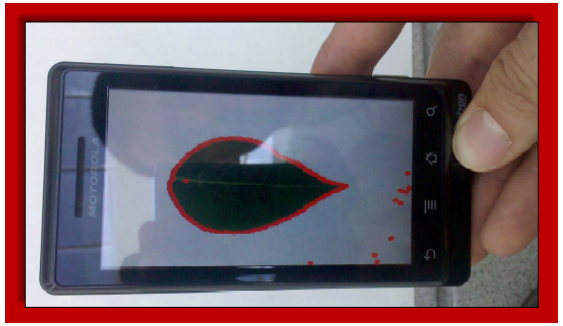
Class project: jigsaw puzzle solver

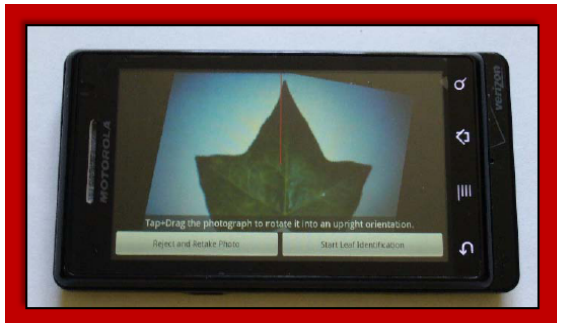


L. Liang and Z. Liu, Spring 2010
http://ee368.stanford.edu/Project_10

Class project: plant leaf classification







D. Knight, J. Painter, and M. Potter, Spring 2010
http://ee368.stanford.edu/Project_10

???