9th Central and Eastern European Software Engineering Conference in Russia - CEE-SECR 2013
October 23 - 25, Moscow

Face Detection and Classification on Mobile Devices

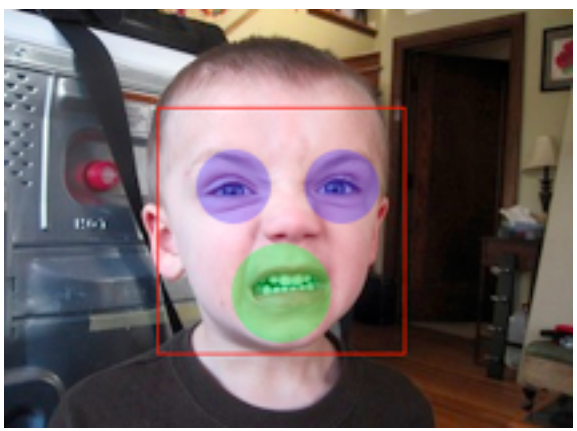
Igor Markov

Agenda

- Introduction
- Algorithms
- The project
- Free frameworks

What is face detection for?

- Camera focusing
- Tagging faces on photos
- Marketing studies
- Surveillance
- Special effects (Augmented Reality)
- Robotics

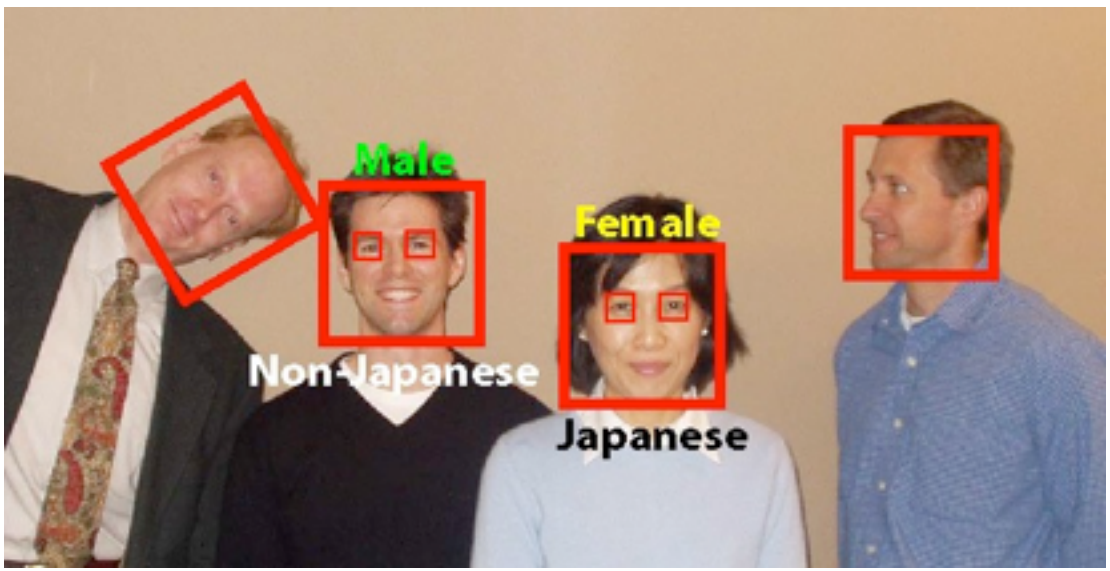


On mobile devices?

- The same thing.

Face classification

- Gender
- Age
- Emotion
- Ethnic group

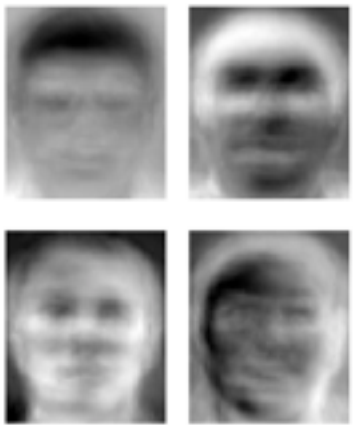


Face tracking

- Is this the same person in the next video frame?

Traditional algorithms

- Search for eyes, nose, mouth, etc
- Estimate relative positions of these points
- ... or, comparison with templates



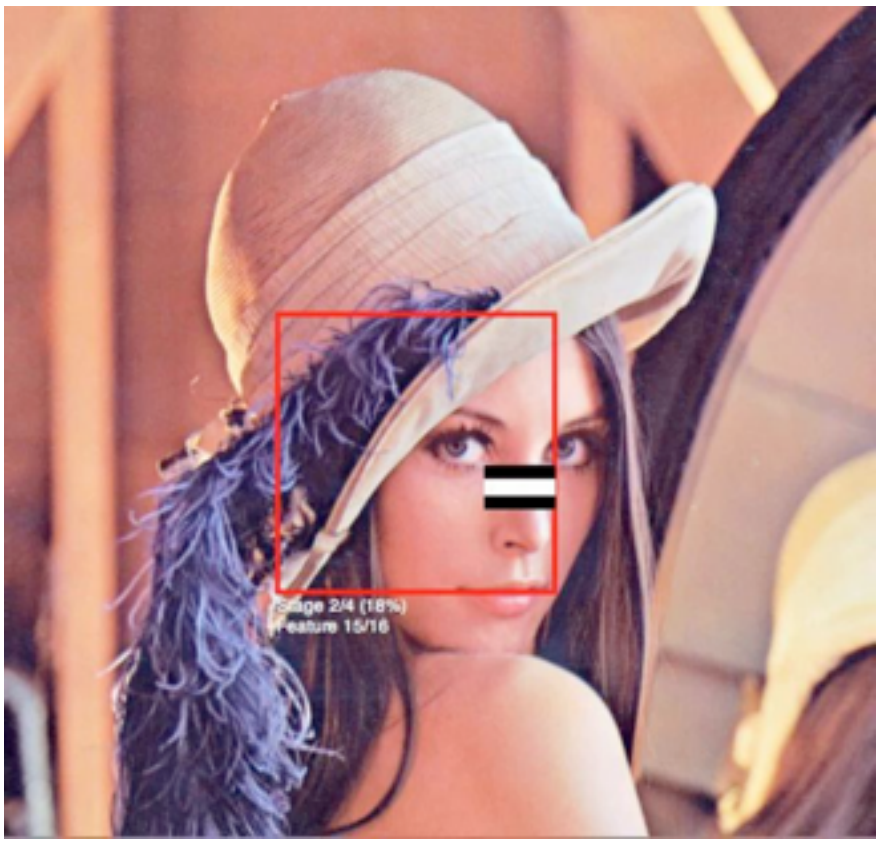
- Eigenfaces
- Linear Discriminate Analysis
- Elastic Bunch Graph Matching
- Multilinear Subspace Learning
- Dynamic link matching

Viola–Jones Object Detection Framework

- Proposed in 2001 by Paul Viola and Michael Jones
- Real-time enough
- A face can be rotated by angle up to 30°
- Good for embedded solutions
- Learning is rather slow

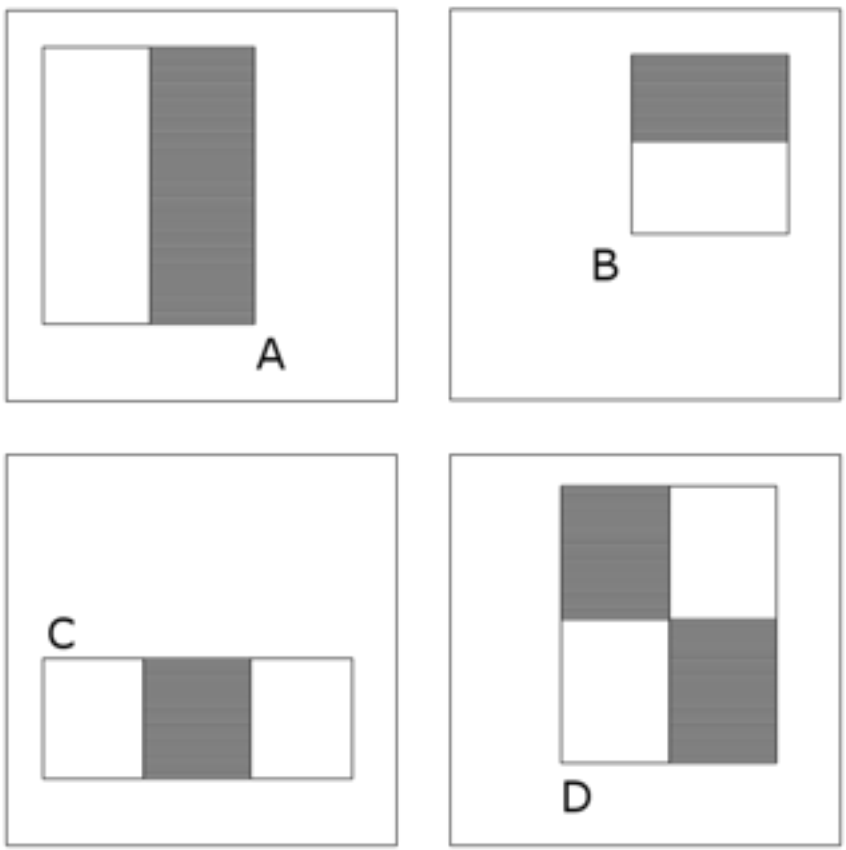
Sub-window

- Size is 24×24
- Moves through all possible positions

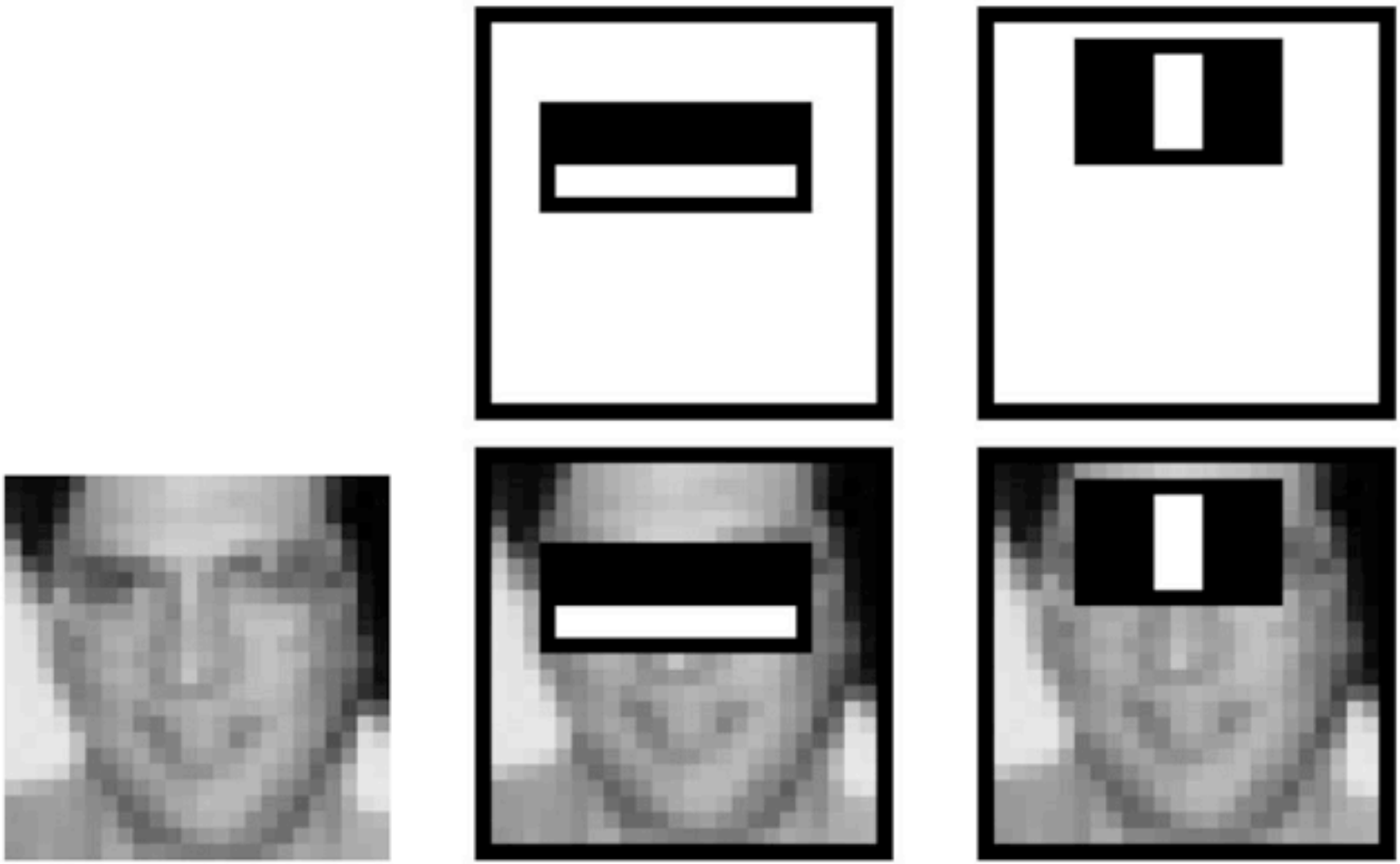


Haar-like features

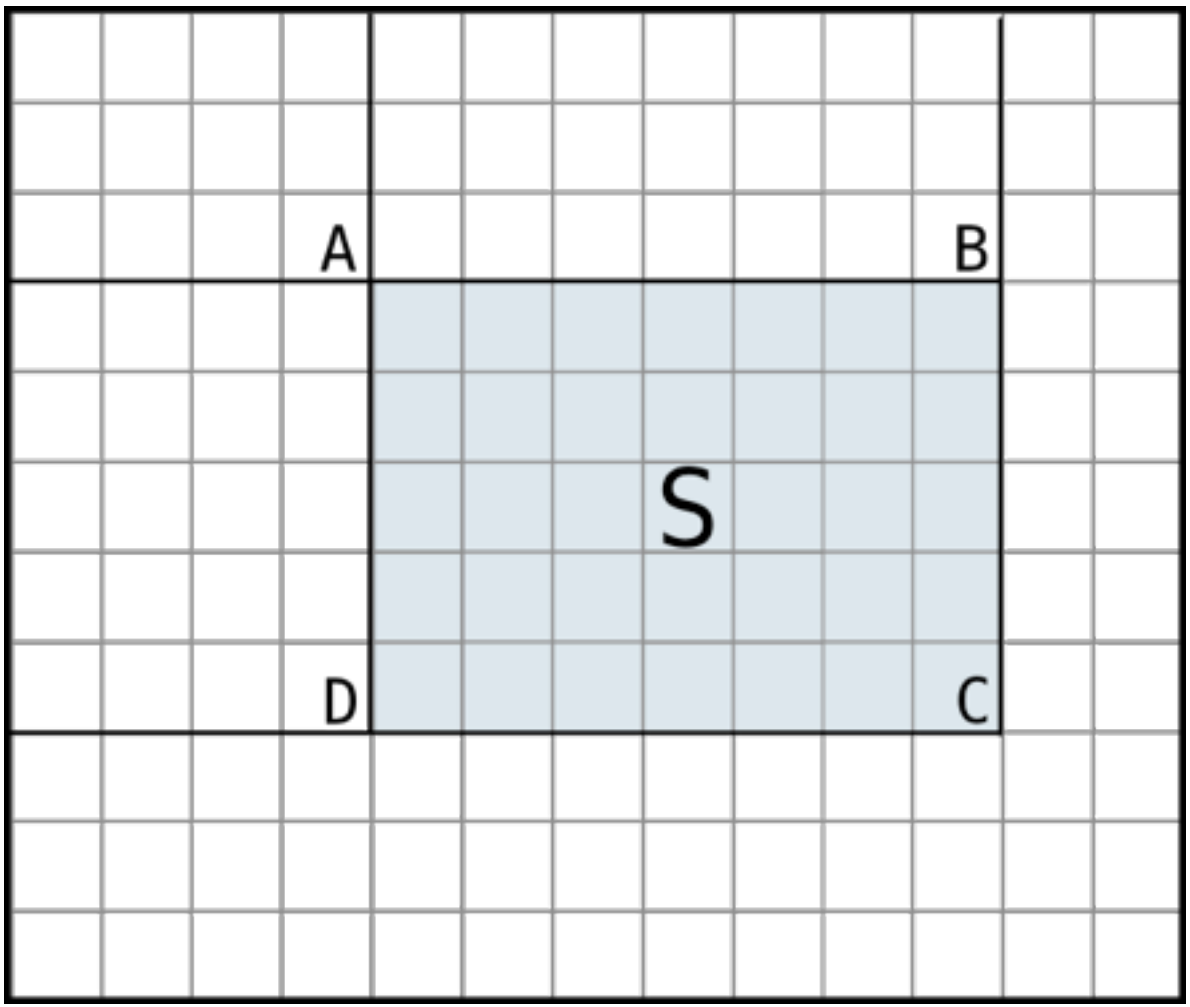
- The light part is added
- The dark part is subtracted



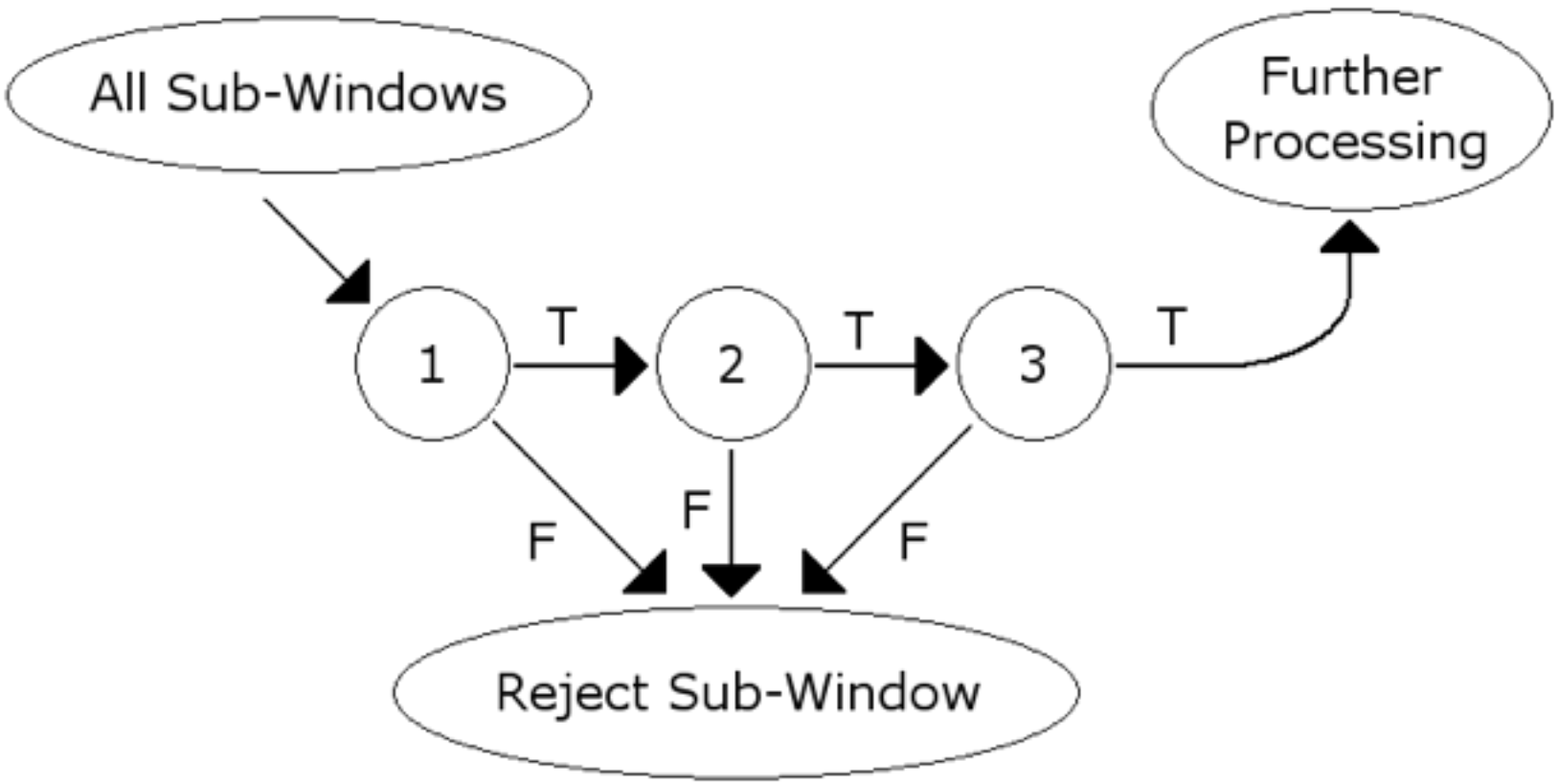
Haar Feature Example



Integral Image



Classifiers Cascade



Learning: Photo Set

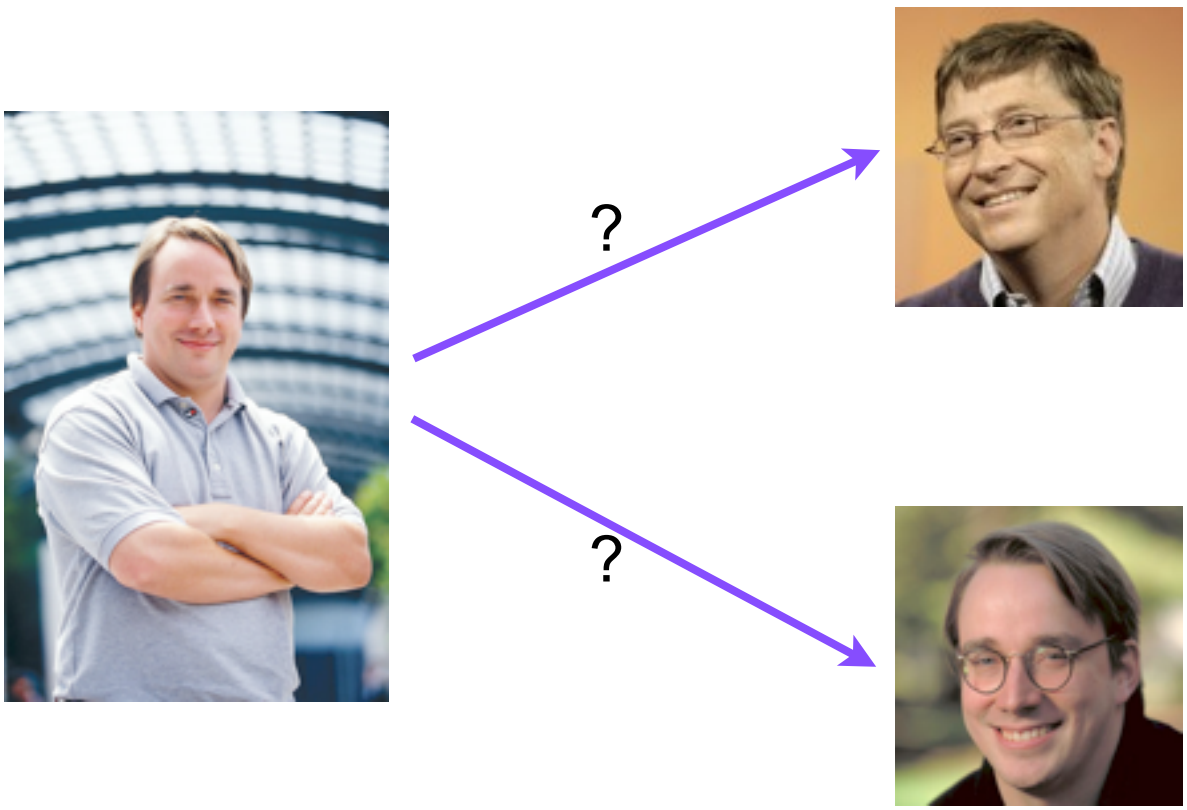
- At&T Facedatabase
- Yale Facedatabase A
- Extended Yale Facedatabase B
- FERET

Machine Learning

- Boosting
- AdaBoost

Classification

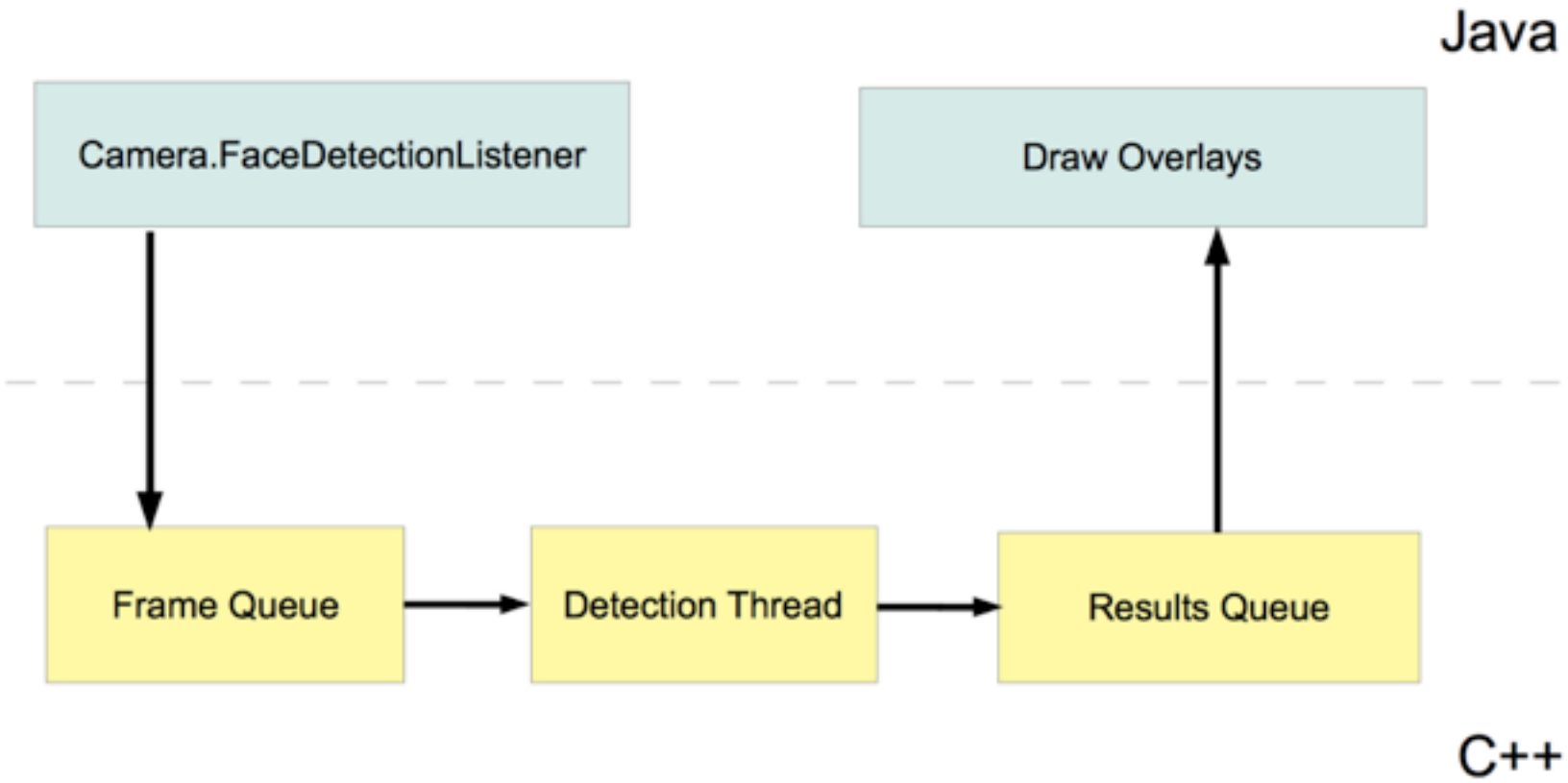
- Learning: AdaBoost
- Classifications: Local Binary Patterns, EigenFaces, etc.



Use Case

- Face detection and classification for marketing study
- Video stream from a camera, real time
- Using Android phone
- High performance

Generic scheme on Android



Optimizations

- Avoid large data copying
- double → int
- Early exit from loops
- Parallelization
- SIMD

Parallel Detection

- Thread pool (max threads = CPU cores number)
- For each possible sub-window size:
  - Put a task to the thread pool
- Wait for results

NEON code

```
loop:
    vldmia %0!, {%%d8, %%d9} //q4 <- data[i][j]
    vldmia %3!, {%%d28, %%d29} //q14 <- integral_fi[i-1][j]
    vldmia %5!, {%%d30, %%d31} //q15 <- sq_integral_fi[i-1][j]
    vmul.f32 %%q5, %%q4, %%q4 //q5 <- data^2
    vmov %%d1, %%d8 // q0[2-3] <- q4[0-1]
    vadd.f32 %%q4, %%q0
    vext.32 %%d3, %%d8, %%d9, #1 // q1[2-3] <- q4[1-2]
    vmov %%s5, %%s16 // q1[1] <- q4[0]
    vadd.f32 %%q4, %%q1 //data is summed in q4
    vmov %%d5, %%d10 // q2[2-3] <- q5[0-1]
    vadd.f32 %%q5, %%q2
    vext.32 %%d7, %%d10, %%d11, #1 // q3[2-3] <- q5[1-2]
    vmov %%s13, %%s20 // q3[1] <- q5[0]
```

Public Frameworks

- OpenCV (FaceRecognizer)
- Android SDK (Camera Face Listener)
- iOS SDK (Core Image)
- Lots of them (facedetection.com)

OpenCV

- Open source
- C++
- Many useful algorithms and primitives

```
FaceRecognizer model = createEigenFaceRecognizer();

....

int predictedLabel = model->predict(testSample);
```

Android SDK Face Detection

```
class MyFaceDetectionListener implements Camera.FaceDetectionListener {

    public void onFaceDetection(Face[] faces, Camera camera) {
        int i = 0;

        for (Face face : faces) {

            Log.i("FD", "face detected: " + (++i) + " of " + faces.length +
                    "X: " + faces.rect.centerX() +
                    "Y: " + faces.rect.centerY());

        }
    }
}
```

iOS Core Image

```
CIContext *context = [CIContext contextWithOptions:nil];

NSDictionary *opts = @{ CIDetectorAccuracy : CIDetectorAccuracyHigh };

CIDetector *detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:context
                                          options:opts];

opts = @{ CIDetectorImageOrientation :
          [[myImage properties] valueForKey:kCGImagePropertyOrientation] };

NSArray *features = [detector featuresInImage:myImage options:opts];
```

Contacts

Igor Markov

igor.markov@auriga.com