Import

```
import com.googlecode.javacv.CanvasFrame;
import com.googlecode.javacv.FrameGrabber;
import com.googlecode.javacv.OpenCVFrameGrabber;
import static com.googlecode.javacv.cpp.opencv_core.*;
import static com.googlecode.javacv.cpp.opencv_objdetect.*;
import static com.googlecode.javacv.cpp.opencv_imgproc.*;
import static com.googlecode.javacv.cpp.opencv_highgui.*;
```

Load an image and detect face

Image pre Process

1. convert to gray
2. Histogram equalization

Detection

require a classifer file named “ **haarcascade_frontalface_alt.xml**”.

keep this file in project directory.

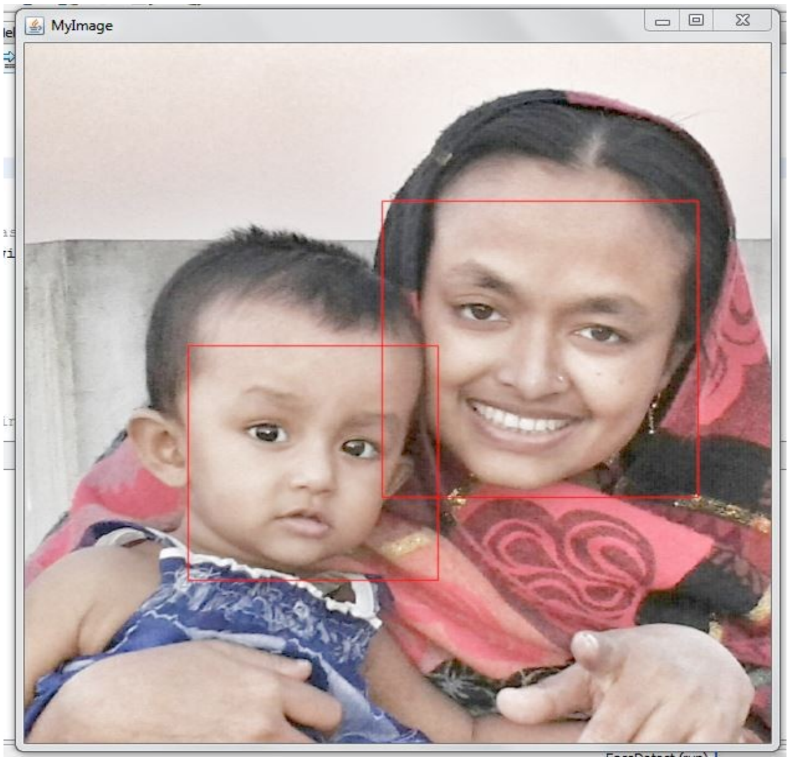
The code bellow is well commented for understanding face detection steps.

```
//first load an image
IplImage img = cvLoadImage("rr.jpg");


//make blank gray image, gray image has 1 channel
IplImage gray = IplImage.create(img.width(),
img.height(), IPL_DEPTH_8U, 1);
```

```
//now convert to gray iamge
cvCvtColor(img, gray, CV_BGR2GRAY);

//make the image sharp
cvEqualizeHist(gray, gray);

//now image is preprocessed for applying face
detection.

//prepare the face detection tools.
 String face_classifier =
"haarcascade_frontalface_alt.xml";

CvHaarClassifierCascade face_cascade = new
CvHaarClassifierCascade(cvLoad(face_classifier));


CvMemStorage storage = CvMemStorage.create();

//detect the face
CvSeq faces = cvHaarDetectObjects(gray,
face_cascade, storage, 1.1, 1, 0);


//now mark the faces on the original images.
for (int i = 0; i < faces.total(); i++) {
   CvRect r = new CvRect(cvGetSeqElem(faces, i));

   //make rectangle
    cvRectangle(img, cvPoint(r.x(), r.y()),
cvPoint((r.x() + r.width()), (r.y() + r.height())),
CvScalar.RED, 1, CV_AA, 0);
        }

//done, lets show the result.

CanvasFrame canvas = new CanvasFrame("MyImage");

canvas.setDefaultCloseOperation(CanvasFrame.EXIT_ON
_CLOSE);
```

```
canvas.showImage(img);
```

Output:



Capture continuously from camera and detect face

Same as capturing single frame. Here we just capture continuously.

```
CanvasFrame frame = new CanvasFrame("Face
Detector");
```

```
frame.setDefaultCloseOperation(CanvasFrame.EXIT_ON_
CLOSE);
frame.setLocation(200, 0);


//create camera object.
FrameGrabber grabber = new OpenCVFrameGrabber(0);
//start camera
try {
     grabber.start();
} catch (Exception ex) {
      System.out.println("No Camera");
      return;
}

//prepare the face detection tools.
String face_classifier =
"haarcascade_frontalface_alt.xml";
CvHaarClassifierCascade face_cascade = new
CvHaarClassifierCascade(cvLoad(face_classifier));


CvMemStorage storage = CvMemStorage.create();

IplImage image = null;
try {
    while ((image = grabber.grab()) != null)  {
//detect face and indicate with a rectangle

    //make a gray image.
    IplImage grayImage =
IplImage.create(image.width(), image.height(),
IPL_DEPTH_8U, 1);     //create grayscale image

cvCvtColor(image, grayImage,
CV_BGR2GRAY);        //convert to gray scale image

//sharp
cvEqualizeHist(grayImage,grayImage);
//detect face
CvSeq faces = cvHaarDetectObjects(grayImage,
face_cascade, storage, 1.1, 1, 0);  //face
//detection

/mark the faces
for (int i = 0; i < faces.total(); i++) {
     CvRect r = new CvRect(cvGetSeqElem(faces, i));
//make rectangle
     cvRectangle(image, cvPoint(r.x(), r.y()),
cvPoint((r.x() + r.width()), (r.y() + r.height())),
CvScalar.RED, 1, CV_AA, 0);
                }

    frame.showImage(image);
    cvClearMemStorage(storage);
}
} catch (Exception ex) {
}

//release the resources
try {
    grabber.stop();
    storage.release();
} catch (Exception ex) {
}
frame.dispose();
```