Package myproject;

Import java.awt.image.BufferedImage; Import java.awt.image.DataBufferByte; Import java.awt.image.WritableRaster;

Import java.io.FileNotFoundException; Import java.io.IOException;

Import javafx.application.Application; Import javafx.embed.swing.SwingFXUtils; Import javafx.scene.Group;

Import javafx.scene.Scene;

Import javafx.scene.image.ImageView;

Import javafx.scene.image.WritableImage; Import javafx.stage.Stage;

Import org.opencv.core.Core; Import org.opencv.core.Mat; Import org.opencv.core.MatOfRect; Import org.opencv.core.Point; Import org.opencv.core.Rect; Import org.opencv.core.Scalar;

Import org.opencv.imgcodecs.Imgcodecs; Import org.opencv.imgproc.Imgproc;

Import org.opencv.objdetect.CascadeClassifier;

Import org.opencv.videoio.VideoCapture;

Public class FaceRecognitionVideo extends Application{

Mat matrix = null;

@Override

Public void start(Stage stage) throws FileNotFoundException, IOException {

// Capturing the snapshot from the camera FaceRecognitionVideo obj = new FaceRecognitionVideo(); WritableImage writableImage = obj.capureFrame();

// Saving the image Obj.saveImage();

// Setting the image view

ImageView imageView = new ImageView(writableImage);

// setting the fit height and width of the image view imageView.setFitHeight(400); imageView.setFitWidth(600);

// Setting the preserve ratio of the image view imageView.setPreserveRatio(true);

// Creating a Group object

Group root = new Group(imageView);

// Creating a scene object

Scene scene = new Scene(root, 600, 400);

// Setting title to the Stage Stage.setTitle(“Capturing an image”);

// Adding scene to the stage Stage.setScene(scene);

// Displaying the contents of the stage Stage.show();

}

Public WritableImage capureFrame() { WritableImage writableImage = null;

// Loading the OpenCV core library System.loadLibrary( Core.NATIVE\_LIBRARY\_NAME );

// Instantiating the VideoCapture class (camera:: 0) VideoCapture capture = new VideoCapture(0);

// Reading the next video frame from the camera Mat matrix = new Mat();

Capture.read(matrix);

// If camera is opened If(!capture.isOpened()) {

System.out.println(“camera not detected”);

} else

System.out.println(“Camera detected “);

// If there is next video frame If (capture.read(matrix)) {

/////// Detecting the face in the snap /////

String file = “F:\\\\Manoj Folder\\\\VIT MTECH\\\\3RD SEM\\\\se java\\\\Project\\\\casecader.xml”;

CascadeClassifier classifier = new CascadeClassifier(file);

MatOfRect faceDetections = new MatOfRect(); Classifier.detectMultiScale(matrix, faceDetections); System.out.println(String.format(“Detected %s faces”,

faceDetections.toArray().length));

// Drawing boxes

For (Rect rect : faceDetections.toArray()) { Imgproc.rectangle(

Matrix,   
  
New Point(rect.x, rect.y), //bottom left

New Point(rect.x + rect.width, rect.y + rect.height), //top right New Scalar(0, 0, 255) //RGB colour

);

}

// Creating BuffredImage from the matrix

BufferedImage image = new BufferedImage(matrix.width(), matrix.height(), BufferedImage.TYPE\_3BYTE\_BGR);

WritableRaster raster = image.getRaster();

DataBufferByte dataBuffer = (DataBufferByte) raster.getDataBuffer();

Byte[] data = dataBuffer.getData();

Matrix.get(0, 0, data);

This.matrix = matrix;

// Creating the Writable Image

writableImage = SwingFXUtils.toFXImage(image, null);

}

Return writableImage;

}

Public void saveImage() {

// Saving the Image

String file = “F:\\\\Manoj Folder\\\\VIT MTECH\\\\3RD SEM\\\\se java\\\\Project\\\\images\\\\outputs\\\\facedetected.jpg”;

// Instantiating the imagecodecs class Imgcodecs imageCodecs = new Imgcodecs();

// Saving it again imageCodecs.imwrite(file, matrix);

}

Public static void main(String args[]) { Launch(args);

}

}