 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

DESARROLLO DE SOFTWARE PARA LA DETERMINACION DE PERSONAS EN ENTORNOS ROBOTIZADOS A PARTIR DE IMÁGENES RGBD

AUTORES

JUAN CARLOS ARANGO PULGARIN

PROGRAMA ACADEMICO

INGENIERIA DE SISTEMAS

DIRECTOR DEL TRABAJO DE GRADO

MAURICIO ARIAS CORREA

INSTITUTO TECNOLÓGICO METROPOLITANO

2018

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

Se propone implementar en entornos robotizados la manera de incrementar la seguridad de las personas que se encuentran en el entorno del robot, basados en la norma ANSI RIA TR R15.306 – 2014, la cual define distancias de acercamiento de las personas al volumen de trabajo de los robots en producción con el fin de reducir las muertes y / o accidentes causados por robots en entornos industriales.

Se trabajará con entorno de programación Processing 3.3.7, en conjunto con las librerías Java, OpenCv, Open Kinect, SDK browser V 2.0 y la aplicación Zadig, lo que permite la interacción del Kinect con el lenguaje de programación a través de imágenes RGBD.

Utilizando como base un ejemplo generado por el profesor Bryan Wai Ching CHUNG, quien implemento el código para la detección de personas con cámara RGB a raíz de librerías de OpenCv, usando Clasificador en cascada HAAR, Bryan, Ching. (2014). Realizamos la implementación para la conexión con el Kinect, el cual servirá para generar la información (Bases de datos RGBD), la cual permite demostrar la detección de personas en diferentes entornos, ya sea alimentando la aplicación desde una base de datos o desde la cámara del Kinect.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

A mi familia y amigos por el apoyo incondicional, al docente asesor quien estuvo siempre a disposición y generando retroalimentación durante todo el proceso y al grupo de investigación del laboratorio de Óptica, Fotonica y Visión Artificial.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

PROCESSING 3.3.7 Lenguaje de programación para procesamiento de imágenes.

KINECT V1.0(1414) Sensor RGBD desarrollado por Microsoft.

RGBD Red-Green-Blue-Deep, Rojo-Verde-Azul-Profundidad.

ZADIG Instalador Automático de drivers.

ANSI RIA TR R15.306 – 2014 Norma mundial de seguridad industrial en entornos de trabajo con robots.

JAVA Lenguaje de programación.

OPENCV biblioteca de código abierto para proyectos de visión artificial.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

1. INTRODUCCIÓN	6
2. MARCO TEÓRICO	7
3. METODOLOGÍA.....	8
4. RESULTADOS Y DISCUSIÓN.....	13
5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO	14
REFERENCIAS	16
APÉNDICE.....	18

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. INTRODUCCIÓN

JUSTIFICACIÓN

Buscar la reducción de accidentes o muertes de personas en entornos robotizados industriales, utilizando herramientas de bajo costo y lenguajes de programación de código abierto, lo genera valor agregado al momento de su desarrollo o implementación por el bajo costo.

OBJETIVOS

Desarrollar o implementar aplicación de software que permita la comunicación del sensor Kinect v1 (1414) o lectura de base de datos RGBD, para realizar la detección de personas en entornos robotizados industriales alrededor del volumen de trabajo del robot para utilizar imágenes RGBD, desde el sensor Kinect o desde base de datos de imágenes y lograr identificar las aproximaciones de las personas al robot y generar una alerta.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. MARCO TEÓRICO

La tendencia en los procesos industriales con robots está orientada en muchos aspectos a procesos de automatización debido a temas de productividad y costos, este proceso es directamente proporcional a los accidentes o muertes en estos entornos industriales robotizados. Según la norma ANSI RIA TR R15.306 – 2014 se busca proteger y salvaguardar la vida de las personas.

Actualmente existen sensores tales como el Kinect V1 (1414) que permite detectar personas y calcular su posición gracias a la capacidad de visualizar imágenes RGBD, podemos detectar proximidad con objetos y generar acciones en base a este tipo de información que ayudara a mejorar la seguridad de las personas en entornos industriales robotizados.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

3. METODOLOGÍA

El desarrollo del aplicativo se generó en base de un ejemplo de detección de personas con cámaras RGB del profesor Bryan Wai Ching CHUNG, quien a raíz de librerías de OpenCv, usando Clasificador en cascada HAAR, Bryan, Ching. (2014).

Se realiza la implementación de código para la conexión con el Kinect, el cual servirá para generar la información (Bases de datos RGBD) y que será utilizada para demostrar la detección de personas en diferentes entornos.

Usando Kinect V1 (1414), capturamos o visualizamos las imágenes en el entorno que se desea implementar el proceso.



Imagen 1. (Kinect V1 (1414) + Conexión USB) tomado de <https://co.pinterest.com/pin/515310382357256306/>

Se debe instalar JAVA (131) preferiblemente la última versión, processing 3.3.7, debe tener las librerías requeridas para este caso (OpenCv, Java, Open Kinect), las cuales pueden ser

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

descargadas desde la interfaz de programa processing 3.3.7 (menú, sketch, importar biblioteca).

Para utilizar el Kinect se deben reemplazar los controladores que se instalan automáticamente con la aplicación Zadig, los controladores de los dispositivos Xbox Nui Camera y Xbox Nui Motor, se reemplazan con el driver de versión libusbK (3.0.7.0), este driver genera mejor conexión entre el dispositivo Kinect y el processing 3.3.7

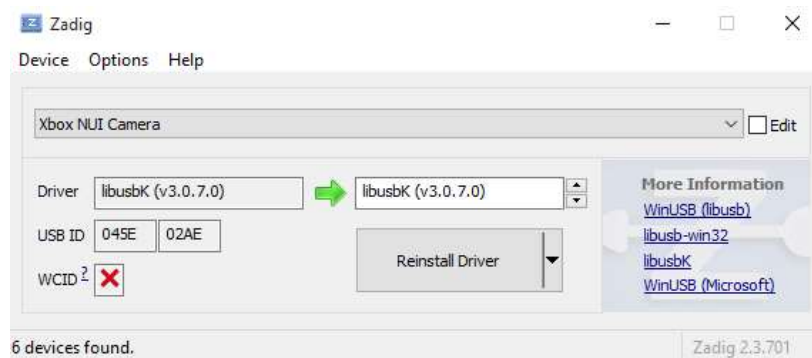


Imagen 2. Interfaz Zadig

Entorno de programación processing



Imagen 3. Interfaz Processing

Se realiza la codificación correspondiente, para que pueda leer la base de datos de videos, la cual reconoce los formatos .mov ò .mp4 y para la lectura del video se debe modificar el nombre y el

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

formato del video correspondiente, los videos se deben leer desde la ruta donde se ejecuta el software desde la carpeta Data.

Cuando requiera leer un video en especifico de la base de datos, debera modificar el nombre del video que desea visualizar, en la linea comentada de la siguiente manera : `movie = new Movie(this,"Personas2.mov");` //Reading of image or video:

```

33 h = int(height*ratio);
34 movie = new Movie(this,"Personas1.mov"); //Reading of image or video
35 movie.play();

```

Imagen 4. Linea a modificar según la lectura del video

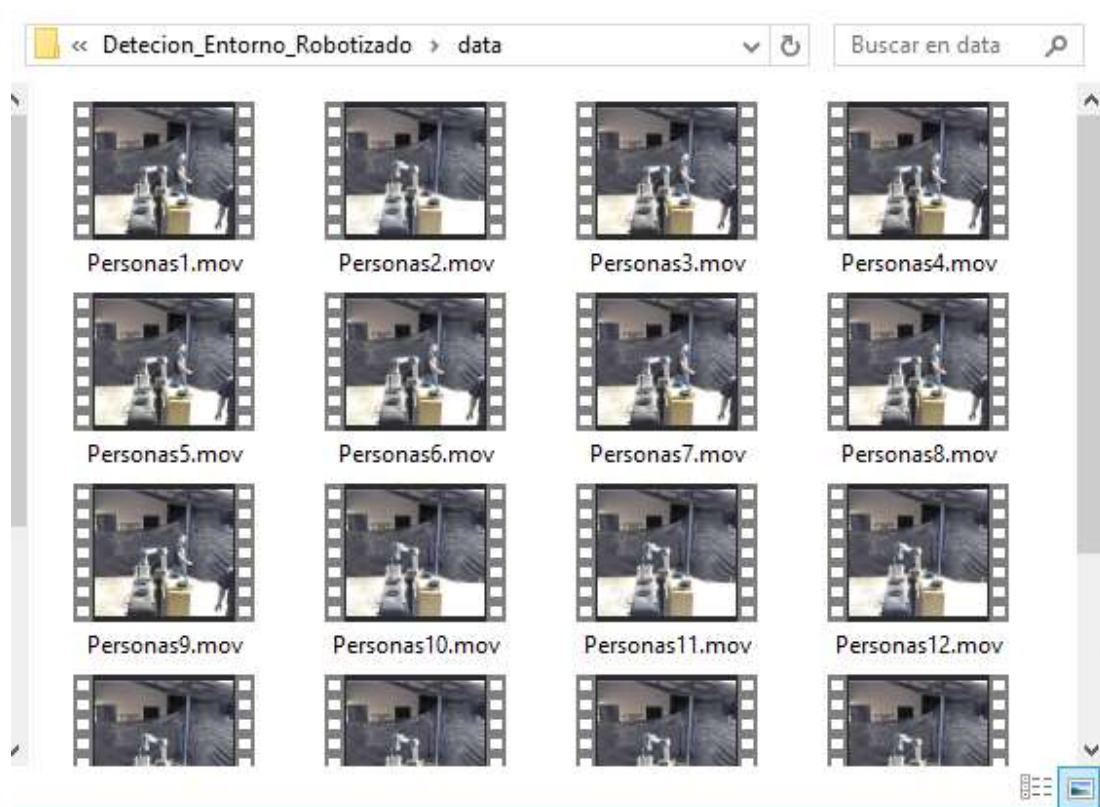


Imagen 5. Base de datos RGBD para reproduccion en la aplicacion

La base de datos de videos fue generada en un entorno real con un robot UR5.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Después deberá presionar el botón play para ejecutar la aplicación que se muestra de la siguiente manera para ambos casos reproducción de base de datos de imágenes e imagen de sensor kinect, como se muestra en las siguientes imágenes con la aplicación en ejecución:



Imagen 6. Interfaz ejecución del proyecto 1 persona

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

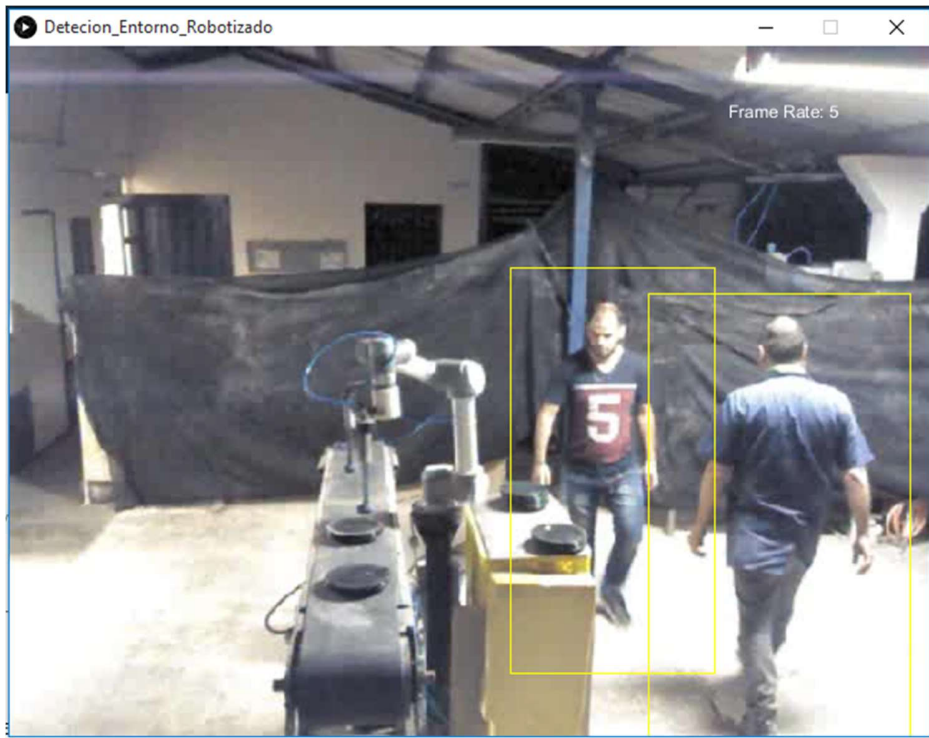


Imagen 7. Interfaz ejecución del proyecto 2 personas misma área

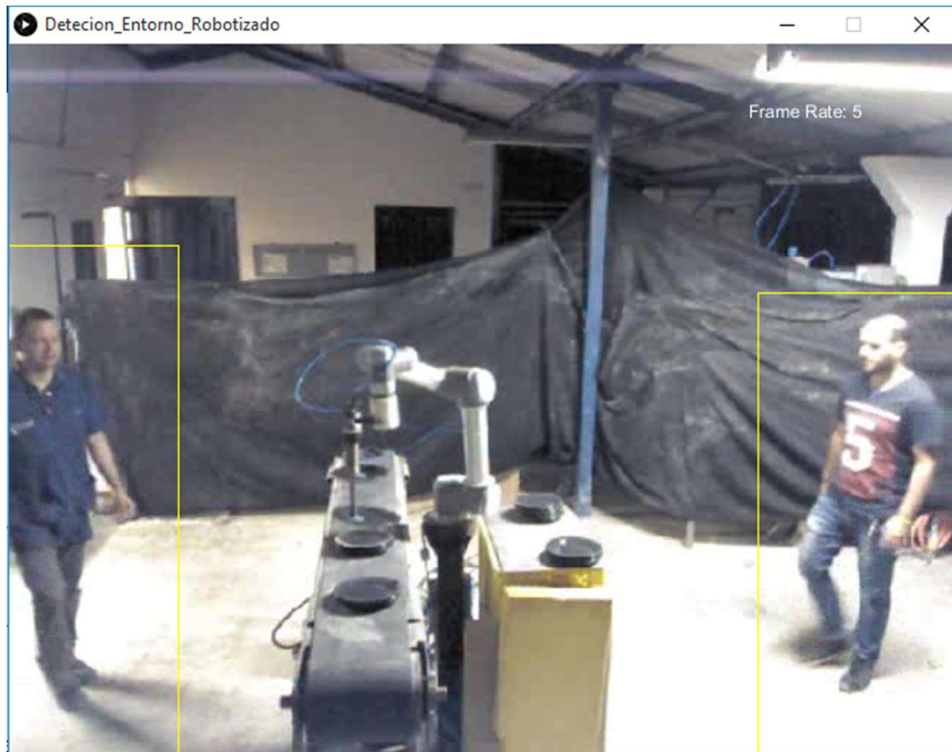


Imagen 8. Interfaz ejecución del proyecto 2 personas áreas diferentes

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

4. RESULTADOS Y DISCUSIÓN

DISCUSIÓN

El objetivo de este proyecto se fijó en el desarrollo de una aplicación de software, buscando utilizar de herramientas de software libre, lo que genera valor agregado en cuanto a inversión económica, por otro lado, el sensor utilizado Kinect v1 (1414) utilizado para generar las imágenes en tiempo real o generar las bases de datos RGBD cuenta con las características de resolución y sensores para la implementación en este tipo proyectos de detección de personas.

RESULTADOS

- Una de las herramientas principales de este proyecto fue el lenguaje de programación processing, por medio de este se logró unificar las diversas librerías que se requerían en este proyecto (OpenCv, Open Kinect, Java) para utilizar el clasificador en cascada (HOGDescriptor) que permite el reconocimiento de personas.
- Originalmente se evidencio el funcionamiento del proyecto con cámaras básicas, sin embargo, utilizamos la librería de Open Kinect en el código para generar la conectividad con el Kinect el cual tiene mejores características para este tipo de proyectos.
- Para el caso de la lectura de las imágenes RGBD o videos, processing utiliza una librería propia de video que permite reproducir los videos ante la configuración y algoritmo definido para el reconocimiento de personas.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

CONCLUSIONES

- Se evidencian las ventajas de realizar las tomas de imágenes desde la vista frontal para la detección de las personas, esto permite que el descriptor HOG utilizado para la detección de personas pueda realizar los reconocimientos de manera adecuada.
- Se logró identificar realizando las tomas imágenes desde la vista superior que se tiene como ventaja el cubrimiento del 100% del área del trabajo del robot y para este funcionamiento se debería realizar un entrenamiento del descriptor para esta posición de la imagen.
- Se logra evidenciar los beneficios a lo largo del proceso, una vez se puede tomar la imagen en tiempo real, directamente la imagen desde e Kinect en diferentes entornos con personas, además de poder alimentar la detección de personas a través de videos de una base de datos, videos tomados directamente desde el Kinect V1 (1414).

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECOMENDACIONES

- Para la detección y seguimiento de personas se recomienda tener alrededor de 5000 imágenes en RGBD y en video alrededor de 100 videos.
- Validar siempre al momento de utilizar la aplicación que se hallan importado las librerías mencionadas para evitar errores durante la ejecución.

TRABAJO FUTURO

- Realizar la conexión con el robot para convertir la señal emitida para que el robot real reciba las instrucciones, se recomienda a través de arduino.
- Realizar le predicción o seguimiento de las personas para el robot pueda predecir a donde se dirige la persona y estar preparado para disminuir la velocidad de trabajo o detenerse por completo.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

1. Automation, R. (2015). <https://www.mc-mc.com>. Obtenido de <https://www.mc-mc.com/ASSETS/DOCUMENTS/CMS/EN/PDH/RobotSystemSafety.pdf>
2. Batard, P. (26 de 07 de 2018). <https://zadig.akeo.ie/>. Obtenido de <https://zadig.akeo.ie/>
3. Borenstein, G. (2012). Making Things See. O'REILLY.
4. Bryan, Ching. (2014). <http://www.magicandlove.com/>. Obtenido de <http://www.magicandlove.com/blog/2014/03/06/people-detection-in-processing-with-opencv/>
5. Dalal, N. (2005). <https://docs.opencv.org/>. Obtenido de https://docs.opencv.org/2.4.13.6/modules/gpu/doc/object_detection.html
6. Fry, B. (26 de 07 de 2018). <https://processing.org>. Obtenido de <https://processing.org/reference/libraries/video/Movie.html>
7. <http://answers.opencv.org>. (19 de 04 de 2016). Obtenido de http://answers.opencv.org/question/93235/opencv-error-assertion-failed-s-0-in-cvsetsize-file-cbuildsmaster_packslave-win64-vc12-sharedopencvmodulescoresrcmatriccpp-line-306/
8. Miller, P. (30 de 03 de 2010). <https://www.engadget.com>. Obtenido de <https://www.engadget.com/2010/06/30/kinect-detailed-in-newly-precise-tech-specs/>
9. project, O. (7 de 3 de 2012). <https://openkinect.org/>. Obtenido de https://openkinect.org/wiki/Main_Page
10. Wikipedia. (25 de 02 de 2018). <https://es.wikipedia.org>. Obtenido de <https://es.wikipedia.org/wiki/OpenCV>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

11. Zananiri, E. (17 de 03 de 2008). www.silentlycrashing.net. Obtenido de <http://www.silentlycrashing.net/p5/libs/video/>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APÉNDICE

Apéndice A: Producto de laboratorio

Código Aplicación:

```

import org.openkinect.freenect.*;
import org.openkinect.freenect2.*;
import org.openkinect.processing.*;
import org.openkinect.tests.*;
import processing.video.*;
import java.util.*;
import java.nio.*;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.CvType;
import org.opencv.core.Scalar;
import org.opencv.objdetect.HOGDescriptor;
import org.opencv.core.MatOfRect;
import org.opencv.core.MatOfDouble;
import org.opencv.core.Rect;
import org.opencv.core.Size;
import org.opencv.imgproc.Imgproc;

Movie movie;
Capture cap;
PImage small;
HOGDescriptor hog;

byte [] bArray;
int [] iArray;
int pixCnt1, pixCnt2;
int w, h;
float ratio;

void setup() {
  size(640, 480);
  background(0);
  ratio = 0.5;
  w = int(width*ratio);
  h = int(height*ratio);
  movie = new Movie(this,"Personas1.mov"); //Reading of image or video
  movie.play();
}

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

//background(0);
// Define and initialise the default capture device.
cap = new Capture(this, width, height);
//cap.start();

// Load the OpenCV native library.
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
println(Core.VERSION);

// pixCnt1 is the number of bytes in the pixel buffer.
// pixCnt2 is the number of integers in the PImage pixels buffer.
pixCnt1 = w*h*4;
pixCnt2 = w*h;

// bArray is the temporary byte array buffer for OpenCV cv::Mat.
// iArray is the temporary integer array buffer for PImage pixels.
bArray = new byte[pixCnt1];
iArray = new int[pixCnt2];

small = createImage(w, h, ARGB);
hog = new HOGDescriptor();
hog.setSVMDetector(HOGDescriptor.getDefaultPeopleDetector());
noFill();
stroke(255, 255, 0);
}

void draw() {
  if (movie.available()) {
    movie.read();
    movie.play();
  }
  else {
    return;
  }
  image(movie, 0, 0);
  // Resize the video to a smaller PImage.
  small.copy(movie, 0, 0, width, height, 0, 0, w, h);
  // Copy the webcam image to the temporary integer array iArray.
  arrayCopy(small.pixels, iArray);

  // Define the temporary Java byte and integer buffers.
  // They share the same storage.
  ByteBuffer bBuf = ByteBuffer.allocate(pixCnt1);
  IntBuffer iBuf = bBuf.asIntBuffer();

  // Copy the webcam image to the byte buffer iBuf.

```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

iBuf.put(iArray);

// Copy the webcam image to the byte array bArray.
bBuf.get(bArray);

// Create the OpenCV cv::Mat.
Mat m1 = new Mat(h, w, CvType.CV_8UC4);

// Initialise the matrix m1 with content from bArray.
m1.put(0, 0, bArray);
// Prepare the grayscale matrix.
Mat m3 = new Mat(h, w, CvType.CV_8UC1);
Imgproc.cvtColor(m1, m3, Imgproc.COLOR_BGRA2GRAY);

MatOfRect found = new MatOfRect();
MatOfDouble weight = new MatOfDouble();

hog.detectMultiScale(m3, found, weight, 0, new Size(8, 8), new Size(32, 32), 1.05, 2, false);

Rect [] rects = found.toArray();
if (rects.length > 0) {
  for (int i=0; i<rects.length; i++) {
    rect(rects[i].x/ratio, rects[i].y/ratio, rects[i].width/ratio, rects[i].height/ratio);
  }
}
text("Frame Rate: " + round(frameRate), 500, 50);
}

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES	 <hr/> <hr/> <hr/>
FIRMA ASESOR	 <hr/>
FECHA ENTREGA: <u>31/07/2018</u>	

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____
RECHAZADO___ ACEPTADO___ ACEPTADO CON MODIFICACIONES___
ACTA NO. _____
FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____
ACTA NO. _____
FECHA ENTREGA: _____