SISTEMA DE VISIÓN ARTIFICIAL INTEGRADO A PLATAFORMA AÉREA PARA LA DETECCIÓN DE PERSONAS EN TIEMPO REAL

Susana Andrea Parra Cabrera
u1802237@unimilitar.edu.co

William Ricardo Cuervo Lote
u1802206@unimilitar.edu.co

***Abstract*— The present grade work describes the development of a surveillance system based on artificial vision integrated at Unmanned Aerial Vehicle (UAV). The review realized about the methods dedicated to pedestrian detection, were found the following algorithms: Mean Shift, Histograms of Oriented Gradients (HOG), Local Binary Pattern (LBP), Viola-Jones and You Only Look Once (YOLO), selecting two algorithms to be evaluate HOG and YOLO, doing for each one the respective tests and its codification in the embedded system. The criteria used to determine the best algorithm was based on a qualitative statistical analysis realized with captures of images from a drone, which were used in tests off-line. The real-time validation obtains with an interface in ground that allows to the user to observe an image of the area, the number of detected persons and the position of the drone.**

***Resumen*— En el presente trabajo de grado, se presenta el desarrollo de un sistema de vigilancia basado en visión artificial integrado a una plataforma aérea no tripulada. En la revisión realizada con respecto a los algoritmos dedicados a la detección de personas, se encontraron Mean Shift, Histograms of Oriented Gradients (HOG), Local Binary Pattern (LBP), Viola-Jones y You Only Look Once (YOLO), dejando seleccionados dos algoritmos para evaluar HOG y YOLO, para los cuales se realizaron las respectivas pruebas y su codificación en el sistema embebido. Los criterios utilizados para determinar el mejor algoritmo se basan en un análisis estadístico cualitativo, realizado con tomas de imágenes desde un drone, las cuales fueron usadas en pruebas off-line. La validación en tiempo real se obtiene mediante una interfaz en tierra que permita observar al usuario el número de personas detectadas, una imagen de la zona y la posición del drone.**

***Palabras Clave*— Detección de personas, vigilancia aérea, HOG, YOLO.**

I. INTRODUCCIÓN

Países como China, Estados Unidos, Argentina, India, Francia, Bélgica, Suiza y España, que afrontan problemáticas de seguridad y de vigilancia, donde se ven obligados al uso de cámaras de vigilancia ubicadas en postes lo que les da un campo limitado de visión. Desde esta tecnología, constantemente se busca mejorar la calidad de las imágenes y la velocidad de procesado, al igual que la miniaturización de las cosas, de manera que baje peso y sean fácilmente embarcadas en estructuras aéreas como la que se usó en este trabajo.

Por otro lado, gran parte de la industria colombiana es orientada a los sectores hidroeléctricos y de hidrocarburos [1], además de tener problemáticas en torno a la vigilancia siendo complejo debido al conflicto armado del país [2] y la delincuencia organizada que realizan atentados terroristas contra los sistemas de transporte de petróleo [3], agua y energía [4]. Cuantiosos recursos se destinan para la vigilancia de esta infraestructura y nuevas tecnologías como vehículos aéreos no tripulados, [5].

Con el desarrollo de un sistema de vigilancia basado en visión artificial integrado a una plataforma aérea no tripulada, al ser una detección en tiempo real evita poner en riesgo el recurso humano, disminuyendo los costos logísticos y operativos, ya que se puede verificar el funcionamiento del sistema desde una estación en tierra donde se observa el número de personas detectadas, una imagen de la zona y la posición.

II. ESTADO DEL ARTE

En la universidad de Beihang en China, se realiza la identificación de personas desde un *drone* empleando el algoritmo *Histograms of Oriented Gradients* (HOG) [6], en la Universidad Politécnica de Cataluña se realiza el seguimiento de personas, basado en el algoritmo *Tracking Learning Decision (TLD)* [7]. En el artículo [8] se presenta un estudio para diferenciar los automóviles de los peatones teniendo en cuenta la velocidad del movimiento con la que se desplazan, con el método de *Mean Shift* [8], logran determinar las velocidades enmarcando en las más rápidas a los carros y las más lentas a los peatones.

Uno de los principales inconvenientes para identificar personas en una imagen es el fondo de esta, por ello, en el estudio [9] extraen el objeto de interés, en este caso la persona, de una imagen; para ello emplean diferentes filtros de luminosidad y contorno, el método empleado en este estudio es *Histograms of Oriented Gradients* (HOG), donde luego de separar el objeto de interés de la imagen se procede a identificar las partes principales del cuerpo, como brazos, torso y cabeza; al tener esto identificado se compara con una base de datos previamente entrenada, obteniendo una precisión del 89,6% [10].

Teniendo en cuenta que la vigilancia de las zonas de interés se realiza de día y de noche, en el artículo [6] efectúan la detección y seguimiento de personas desde una perspectiva aérea, empleando un *QuadRotor* y una cámara térmica; para la identificación de personas, primero se extrae con un método binario los posibles candidatos, y luego con un clasificador se define si es o no persona, cómo se muestra en la Figura 1.

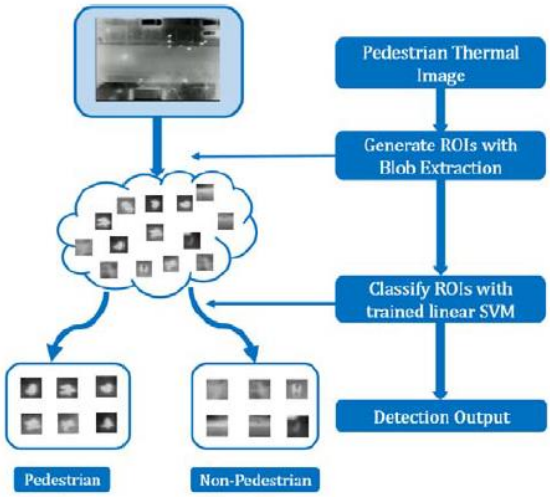

Figura 1. Diagrama de flujo de la identificación de personas. [6]

En la Figura 1 se observa el diagrama de flujo empleado en el estudio [6], donde primero se captura la imagen con una cámara térmica desde un *drone*, luego se clasifican los posibles objetos de interés, y finalmente se comparan los posibles candidatos con una base de datos, detectando así a las personas.

En los estudios [11] y [12] emplean el algoritmo HOG para la detección de peatones capturando un video desde un *drone*. En el primer estudio se analizaron tres situaciones, en la primera los peatones están estáticos, por ejemplo, esperando un bus, sin movimiento alguno, en la segunda situación, los peatones caminan (velocidad baja), observando cómo funciona el algoritmo ante movimiento, y en la tercera situación, se analiza cuando una persona está corriendo (velocidad alta); para la primera situación, se obtuvo un acierto en la detección de peatones del 93.3%, para la segunda situación de 92,7% y para la tercera de 94,1%. El estudio compara los resultados obtenidos con HOG respecto a dos algoritmos; donde el detector Viola-Jones [11] es quien obtiene la mayor cantidad de errores, HOG tiene una precisión alta sin importar la altura a la que se tome el video, sin embargo, LatSvm-V2 [11] es quien tiene mayor efectividad para los videos tomados a cuatro metros de altura. Para el segundo estudio, se realizó un clasificador en cascada, con una base de datos previamente entrenada para detectar a las personas, además se empleó un sistema térmico para ubicar donde se encuentran las personas en la imagen. A continuación, se muestra una base de datos empleada en el estudio [13] el cual emplea el algoritmo HOG y un clasificador de cascada.

En la Figura 2 se muestra la base de datos que fue entrenada para la detección de personas desde una perspectiva aérea descrita en el artículo [13]. En la parte superior de figura se muestra cómo se generan diferentes ángulos dependiendo de la ubicación de la cámara, por ello es necesario crear una base de datos donde se incluyan las diferentes posibilidades que puede tener el *drone* al momento de la detección; en la parte inferior de la Figura 2, se observa cómo se ve una persona desde diferentes perspectivas.

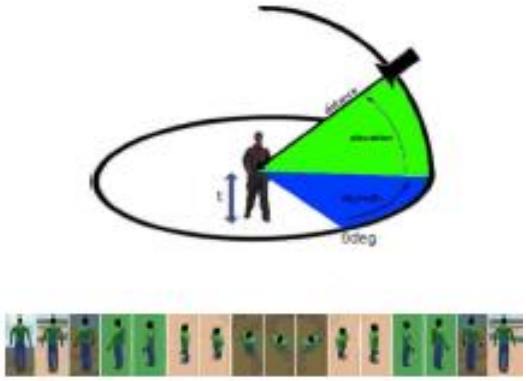

Figura 2. Base de datos para una persona desde perspectiva aérea.

Otro método empleado en la investigación con *drones* y sistemas de seguridad es explicado en [14], donde se capturan las imágenes con una resolución de 640x480 pixeles, se creó una base de datos y luego se procedió a detectar las personas desde un *drone*, en este se observa que dependiendo la cantidad de fotografías que se tengan en la base de datos, varia la efectividad del método; si se compara sólo con una imagen, se obtiene un 77,78% de efectividad, para tres imágenes un 91,11% y para cinco un 100%. Mostrando así la importancia de realizar una base de datos con fotografías de diferentes ángulos de la persona respecto a la cámara.

El seguimiento de personas se ha desarrollado con el algoritmo *Tracking Learning Detection (TLD)* [15], el cual, no utiliza GPS, su funcionamiento se basa en mantener el objetivo en una posición central en la imagen, comparando la diferencia entre dos posiciones, como se muestra en [7] quien empleó una tarjeta *Raspberry pi* con el sistema operativo tipo Linux en el cual se instaló el software ROS donde se implementaron sus respectivos algoritmos de rastreo para el *drone*.

En el sistema de vigilancia empleado para detectar personas que se acerquen a una subestación de transformadores [16] emplean un método basado en una red neuronal llamado *You Only Look Once (YOLO)*, obteniendo una exactitud del 92,7%; este método es más rápido que los métodos convencionales para la detección de personas. Otro sistema de vigilancia que ha empleado este método es [17] donde realizan el entrenamiento de la red neuronal con una base de datos desde una cámara de vigilancia estática.

En Colombia el uso de vehículos aéreos no tripulados para trabajos de vigilancia es poco común, generalmente se emplean en diferentes áreas como fotografía, cubrir eventos,

reconocimiento de grandes estructuras y terrenos. En cuanto al marco legal se han constituido una serie de normas y estamentos para el uso, monitoreo y control de estos dispositivos [18]. [19] propone integrar los *drones* a los centros de vigilancia donde son de apoyo para terrenos con grandes superficies, puesto que permiten tener una mayor cobertura y campo de acción, generando un gran impacto en la seguridad privada.

III. MARCO TEÓRICO

A. *Sistemas de detección*

Los sistemas de detección de objetos se clasifican en dos tipos [20]: los que emplean sensores, ya sean infrarrojos o térmicos y los que emplean cámaras, a estos se les conoce como sistema de visión artificial.

Para poder crear un sistema de visión artificial se requiere de dos componentes, una cámara, escáner, o cualquier periférico que permita capturar una imagen a tratar y de una tarjeta que permita el procesamiento de las imágenes y algoritmos a emplear. Los pasos por seguir para un sistema de visión artificial son:

- Adquirir imagen
- Segmentar dicha imagen
- Obtener las regiones de interés (ROI)
- Clasificar los objetos de interés

Partiendo de que la imagen que se va a obtener va a ser por medio de una cámara embarcada en el *drone* a emplear, se procede a explicar los métodos existentes para la detección de personas y el seguimiento de una persona en específica.

B. *Técnicas para la Detección De Personas*

Para la detección de personas se han desarrollado diferentes métodos, a continuación, se mencionarán los más relevantes.

1. *Pirámide de imágenes y ventana deslizante*

Dos términos que se emplean, independiente del algoritmo con el cual se desarrolle la detección de personas es la pirámide de imágenes [10] y ventana deslizante, como se muestra en la Figura 3.

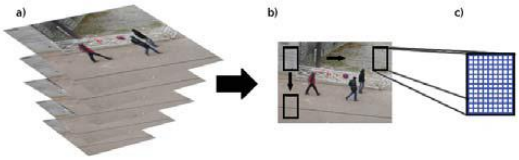

Figura 3. a) Pirámide de imágenes. b) Ventana deslizante. [21]

La pirámide de imágenes se emplea para encontrar objetos de diferentes tamaños, para ella se requieren de tres parámetros, una serie de niveles, un factor de escala y un rango para la escala. La ventana deslizante tiene en una cierta posición de la imagen, se desplaza tanto en el eje X como en el eje Y generando otra nueva ventana del mismo tamaño, pero desplazada según este pasó en el eje de las columnas y en el eje de las filas.

2. *Histograms of Oriented Gradients (HOG)*

Es un método por el cual se pueden reconocer peatones en una imagen, que tiene como ventajas la robustez ante diferentes condiciones de iluminación, cambios en el contorno de la imagen y diferencias de fondos [10]. Para implementar este algoritmo el primer paso es crear una base de datos con imágenes positivas (donde se van a evidenciar personas), e imágenes negativas (donde no hay personas), el algoritmo está definido para que la altura de las imágenes sea el doble que su ancho, a partir de esta base se pueden definir las características de detección; luego se divide la imagen en diferentes celdas, calculando los histogramas para cada una de ellas, siendo cada uno de estos histogramas el patrón a identificar con el algoritmo.

En [10] realizan un estudio con este método, donde 405 detecciones son positivas de 462 muestras, con una efectividad del 88%, esto se debe a que los peatones se encuentran a diferentes distancias de la cámara. En otra imagen, los peatones se encuentran a la misma distancia de la cámara obteniendo 426 detecciones positivas de 462 muestras.

3. *You Only Look Once (YOLO)*

Este algoritmo está basado en Darknet [22], que es una red neuronal escrita en C y CUDA de código abierto, emplea una red neuronal simple, procesa las imágenes en tiempo real entre 45-150 cuadros por segundo dependiendo de la tarjeta gráfica, en comparación con otros algoritmos este tiene mayor error en la ubicación del objeto de interés, pero tiene menos falsos positivos en el fondo. Este algoritmo es veloz debido a que solo necesita observar una vez la imagen para detectar donde se ubica el objeto de interés. El funcionamiento de este algoritmo se describe en tres pasos, primero escala la imagen, luego la procesa en la red neuronal y finalmente muestra la imagen original con las etiquetas de los objetos que identificó, como se muestra en la Figura 4.

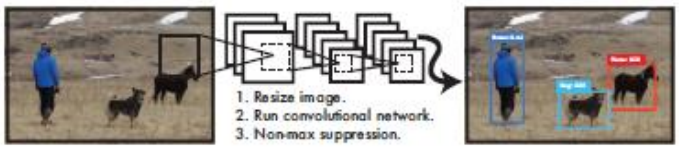

Figura 4. Sistema de detección YOLO [23]

De los anteriores métodos para la identificación de personas, las dos arquitecturas de software que se eligen para abordar el desarrollo del presente trabajo de grado son HOG y YOLO, debido a su robustez ante diferentes fondos, iluminación y la efectividad que presentan en los trabajos que se han realizado con estos dos métodos.

C. *Criterios de evaluación*

Para evaluar el rendimiento de los algoritmos seleccionados, se crea una matriz de confusión donde se escriben los resultados obtenidos, en este caso lo que se quiere

detectar son personas, por lo que la matriz queda como se muestra en la Figura 5.

Figura 5. Matriz de confusión. [24]

Donde la exactitud es la proximidad entre el resultado y la clasificación exacta, la precisión es la calidad de la respuesta del clasificador, la sensibilidad es la eficiencia en la clasificación de todos los elementos que son de la clase y la especificidad es la probabilidad de que una detección positiva tenga un resultado negativo. La exactitud, precisión, sensibilidad y especificidad vienen medidas por:

$$Exactitud = \frac{Reales\ positivos + Reales\ negativos}{Predicciones\ totales} \quad (1)$$

$$Precisión = \frac{Reales\ positivos}{Reales\ negativos + Falsos\ positivos} \quad (2)$$

$$Sensibilidad = \frac{Reales\ positivos}{Reales\ positivos + Falsos\ negativos} \quad (3)$$

D. Seguimiento De Personas

Enmarca la teoría existente y algoritmos desarrollados para el seguimiento de un objetivo dentro de una imagen Así se describen las técnicas *Tracking- Learning-Detection* (TLD), extracción mediante puntos y otros métodos desarrollados para la estimación del movimiento de un objetivo.

1) Tracking- Learning-Detection (TLD)

Este algoritmo, conocido por sus siglas en inglés TLD [7], se emplea para realizar el seguimiento de objetos en tiempo real, se selecciona el objeto de interés, y este algoritmo aprende su aspecto y lo detecta cada vez que aparece en el video en tiempo real, siguiendo así su trayectoria. Se descompone en tres características, *tracker*, como rastreador sigue al objeto a lo largo del video, *detector*, localiza todas las apariciones del objeto a lo largo del video y *learning*, como aprendiz, calcula el error y lo corrige para evitar cometer los mismos errores más adelante.

Otros algoritmos para la detección de movimiento emplean varias imágenes y el objetivo a seguir es extraído mediante puntos [8] [25], se comparan varias imágenes y se calcula la cantidad de puntos que se movieron de una imagen respecto a la otra, sin embargo, los métodos que comparan diferentes imágenes no son recomendados cuando la cámara se encuentra en movimiento debido a que el ángulo de la cámara puede variar y por ende un objeto puede aparentar estar en movimiento, cuando se encuentra quieto y es la cámara quien se ha movido.

El seguimiento de objetos empleando una cámara y un *drone*, consiste en aplicar un sistema de control al *drone* desde una base en tierra, la cual le envía una señal offset de cada motor utilizando un control PD. El procesamiento se realiza en tierra [26]. El seguimiento parte de la obtención de una imagen de la zona de interés luego se realiza una detección básica, se calculan la ubicación del objeto dentro de la imagen y el tamaño que tiene, basados en los datos obtenidos realizan una aproximación del movimiento con respecto al centro de la imagen.

E. Software

A continuación, se presenta el software empleado para la programación de los algoritmos, especializado en visión artificial y procesamiento gráfico para aumentar el rendimiento del sistema.

1) OpenCV

Es una librería abierta de visión artificial, desarrollada por *Inter Corporation*. Tiene alrededor de 2500 algoritmos [27] por lo que las aplicaciones desarrolladas han sido diversas, algunos ejemplos son: sistemas de seguridad, reconocimiento de objetos, calibración de cámara, visión robótica, estéreo, realidad aumentada, entre otros. Esta librería emplea diversos lenguajes de programación, como C, C++, Python y java, además viene desarrollada para todos los sistemas operativos. En esta librería se puede encontrar el algoritmo HOG, el cual tiene como base de datos INRIA. Otros algoritmos incluidos en esta librería para la detección de personas son LBP, SVM y HAAR.

2) CUDA

Compute Unified Device Architecture, mejor conocido como CUDA es una arquitectura de cálculo paralelo que emplea la unidad de procesamiento gráfico GPU para aumentar el rendimiento del sistema, CUDA trabaja de la mano OpenCV, por lo que se puede pasar de trabajar con la CPU a la GPU simplemente incluyendo las librerías de CUDA al código empleado, con esto se mejora la velocidad de procesamiento. A continuación, se muestra una gráfica comparativa entre una tarjeta Testa C2050 y un procesador Core i5-760 2.8Ghz [28].

F. Hardware

En la arquitectura de hardware de los sistemas para detección y seguimiento de personas u objetos existen diferentes configuraciones, donde diversos factores varían, como la distancia entre la cámara y el objeto, el sistema de procesamiento y la zona de interés que puede ser fija o en movimiento, en algunos casos se utilizan marcadores para determinar el objeto a seguir, lo que permite disminuir el costo computacional.

En [12] emplean un sistema de visión embarcado en una plataforma aérea no tripulada, realizando el procesamiento en tierra (offline). En otra distribución similar, emplean un *drone* que tiene incorporada una cámara, el procesamiento es en tierra y se realiza un control de posición [26]. Una arquitectura implementada online es una plataforma aérea no tripulada integrada con una cámara que están comunicados con una base en tierra donde se realiza el seguimiento y control del *drone* [15], sin embargo, la altura a la que realizan este seguimiento es de dos metros.

Los problemas de las arquitecturas existentes son: el limitado tiempo que puede estar en funcionamiento y el riesgo al que

está sometido el sistema al estar a una distancia corta de las personas, por otro lado, a mayor altura las imágenes tienen que ser de alta calidad lo que impide un procesamiento online. Con esto en mente, se plantea un sistema que realice un procesamiento en tiempo real, una plataforma aérea con las capacidades de carga para llevar acabo la detección de personas y seguimiento de ellas por un periodo de tiempo mayor y una cámara con una resolución que permita una altura segura para el sistema.

Teniendo en cuenta lo anterior, para la adquisición de imágenes se van a emplear dos cámaras, la primera de ellas es una cámara ZED y la segunda la cámara Zenmuse Z3, el procesamiento de estas imágenes se hará en una tarjeta NVIDIA Jetson TX1 y un Alienware Alpha (NVIDIA® GeForce® GTX960) todo esto va embarcado en el *dji Matrice 600 Pro*

IV. Desarrollo y simulación

A continuación, se presenta la integración de los códigos al sistema embarcado de procesado, para realizar las pruebas de los algoritmos seleccionados, HOG, YOLO y Tiny YOLO. Para esto, se describe la creación de la base de datos, punto de partida para correr los algoritmos que se van a emplear. Se debe tener en cuenta que el entrenamiento para HOG y para las redes neuronales es diferente, por lo cual se generan dos bases de datos. La última sección muestra el estudio estadístico de la detección de personas, para así definir que algoritmo se programa en el sistema embebido, para ello se presenta la arquitectura de hardware que permite integrar el sistema de procesado y los dispositivos.

A. Creación de la base de datos

La base de datos se generó a partir de las imágenes capturadas por dos cámaras diferentes, la ZED y la Zenmuse 3, para ello se debe tener en cuenta las propiedades de la imagen como tamaño y escala de color que se definen partiendo de la imagen original que se realiza con la cámara ZED, de esta se obtuvo las características de la imagen. La imagen obtenida es estéreo con dimensiones de 2560 x 720 píxeles en RGB. Para trabajar con esta imagen se realiza un preprocesado de la misma, debido a que no se requiere trabajar con estéreo se procede realizar un recorte sobre la imagen obteniendo la Figura 6.

Figura 6 Imagen reprocesada

Al tomar algunas pruebas iniciales con la cámara, simulando la altura y el ángulo de visión a la que va a trabajar se puede observar que debido al ángulo de enfoque de la cámara y la resolución que maneja no se puede tomar personas que se encuentren a una distancia superior a 40 metros, por lo que se determina volar el *drone* a una altura máxima de 15 metros, para evitar que la distancia entre la persona y la cámara sea mayor a 40 metros.

1) Base de datos método HOG

Teniendo definida la distancia desde la cual se van a tomar las fotos se procede a generar la base de datos para obtener así las muestras positivas y negativas con las cuales se va a desarrollar el algoritmo, el tamaño de las personas es aproximadamente de 28x60 pixeles, por lo que se decide mantener una ventana deslizante de 32x64 pixeles, ya que la altura de la ventana que se va a ingresar a la función del descriptor tiene que ser el doble de la base.

Luego de tener el tamaño de la ventana deslizante establecido, se procede a definir el solapamiento de las imágenes, se debe tener en cuenta la separación mínima entre las personas y cuál es la pérdida de información que resultaría al definir cada tipo de solapamiento, para aumentar la velocidad de procesamiento se va a trabajar con dos, ya que este factor permite disminuir el número de muestras sin perder información, en la Figura 7 se muestra cómo queda la imagen con este solapamiento.

Figura 7 .Ventana deslizante con solapamiento

En la generación de la ventana deslizante se emplea la librería de OpenCV la cual permite definir el tamaño de la ventana y el paso que va a tener, de este proceso se obtiene un conjunto de recortes de la imagen original que son guardados en un arreglo, luego se recorre dicho arreglo para extraer los descriptores de cada una de las ventanas y con este descriptor realizar el entrenamiento del algoritmo.

Esta base de datos se obtuvo luego de 45 minutos de grabación a 60 cuadros por segundo desde una altura de 13 metros, en diferentes horas del día y diversos escenarios del Campus de la Universidad, obteniendo 870 muestras positivas y 1274 muestras negativas.

2) Base de datos YOLO y Tiny YOLO

Para la generación de la base de datos para YOLO se tomaron cinco videos con la cámara ZED y cinco videos con la cámara Zemmuse Z3 cada uno con una duración aproximada de 3 minutos, de estos videos se extrajeron los fotogramas que mayor cantidad de información y variedad podían aportar al entrenamiento, quedando la base de datos de 517 imágenes, luego se toma cada una de estas imágenes y empleando el programa *BBox-Label-Tool-Master* [29] se define de forma manual la posición de las personas dentro de la imagen, como se muestra en la Figura 8.

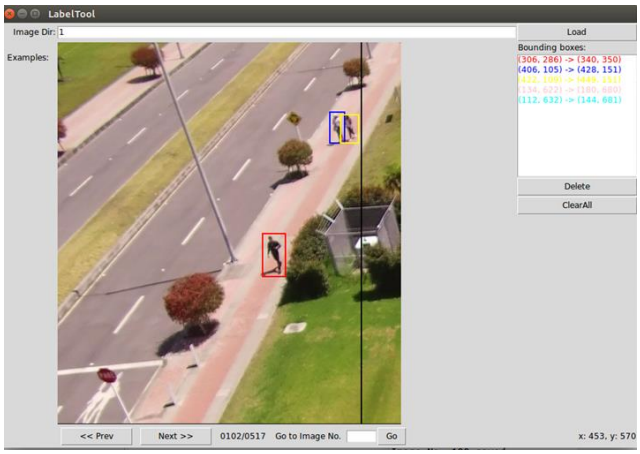

Figura 8. LabelTool selección de áreas de interés

El programa genera un archivo de tipo *.txt* para cada una de las imágenes, este contiene el número de personas y sus posiciones dentro de la imagen en términos de pixeles. El entrenamiento de YOLO necesita un formato distinto al que nos genera por defecto *BBox-Label-Tool*. Para esta conversión se emplea el código convert.py que viene por defecto en los scripts de YOLO.

Para realizar el entrenamiento en YOLO se generan dos archivos en los cuales se enlistan los nombres de las muestras que se van a ingresar. YOLO utiliza una red neuronal la cual tiene un entrenamiento de tipo supervisado por lo cual se necesita definir las imágenes de entrenamiento y prueba. Se decidió que la prueba fuera del 10% del total de muestras, escogiendo las imágenes muestras de manera aleatoria.

3) Entrenamiento YOLO y Tiny YOLO

El entrenamiento de YOLO necesita un archivo que configura los grupos de imágenes por cada iteración del entrenamiento, el número de subdivisiones del proceso debido a que el entrenamiento consume una gran cantidad de recurso de máquina y la cantidad de subdivisiones depende de cada computador, se define la velocidad de aprendizaje, el tamaño de la imagen de entrada, la cantidad de clases de detección, el lumbral de detección, los filtros y sus tamaños. Necesita un archivo que guarda el modelo de la red.

Al cumplir con todos los prerrequisitos del entrenamiento, se realizó una serie de modelos con una base de 3000 iteraciones, en los cuales se varió la velocidad de aprendizaje, el tamaño de la imagen de entrada, el tamaño de los filtros y la base de datos. Haciendo una comparación entre cada uno de los modelos generados se tomó el que mejores resultados produjo y se partió de ese modelo para un segundo entrenamiento que cuenta con 9200 iteraciones.

Tiny YOLO tiene el mismo procedimiento para su entrenamiento, la diferencia más significativa es la cantidad de filtros que se aplican pasando de 32 en YOLO a 12, dando como resultado un amento en la velocidad del algoritmo, pero una disminución en la precisión.

B. Pruebas de los algoritmos

Las pruebas de los algoritmos se realizaron con una recopilación de dos videos en diferentes escenarios del Campus de la universidad, cada video, el primer vídeo tiene una duración de 2 minutos y 30 segundos, fue tomado con la cámara ZED y embarcado en el *QuadRotor* AscTec Pelican; y el segundo video se realizó con la cámara Zenmuse Z3 y el *Matrice 600 Pro* con una duración de 2 minutos y 39 segundos. Estos algoritmos se desarrollaron con las librerías de OpenCV 3.2, CUDA 8.0, Python 2.7, scikit-learn 0.18.1. Se debe tener en cuenta que las pruebas offline se realizaron en computadores portátiles con tarjetas de video NVIDIA GeFoce 840M y NVIDIA GeForce 720M.

Para probar que método es el adecuado se definen los parámetros de calificación, esto lleva a determinar cuáles son las prioridades del proyecto y cuáles son las que tienen mayor peso a la hora de la puesta en marcha. Se define que la velocidad de procesamiento (FPS), la exactitud y la precisión son las cualidades que se van a evaluar.

Tabla I. Estudio estadístico cámara ZED

Algoritmo	Exactitud	Precisión	Sensibilidad	FPS
HOG	97,44%	37,25%	32,59%	6,6
YOLO	99,31%	97,41%	70,68%	5,2
Tiny YOLO	97,99%	52,70%	55,91 %	14,6

Tabla II. Estudio estadístico cámara Zenmuse Z3

Algoritmo	Exactitud	Precisión	Sensibilidad	FPS
HOG	99,40%	61,18%	41,15%	6,3
YOLO	99,80%	97,18%	74,78%	5,7
Tiny YOLO	99,62%	75,10%	75,40%	14,8

Teniendo en cuenta los datos de la Tabla y en la

Tabla , se define que el método YOLO es el que obtiene mejores resultados y por ende el método que se programará en el sistema embebido.

C. Diseño mecánico y arquitectura de hardware

Para integrar el sistema de procesado, el sistema de visión y el hardware del *drone* se diseña un soporte y la comunicación entre el *drone* y la base de control ubicada en tierra se utiliza el protocolo WiFi 802.11ac, lo que permite validar el funcionamiento del algoritmo desde tierra en tiempo real. La primera prueba embarcada se desarrolló en la tarjeta NVIDIA Jetson TX1, posterior a ello se empleó el Alienware Alpha que cuenta con una tarjeta GeForce GTX960, mejorando así la velocidad de procesamiento. En la Figura 9 se muestra la arquitectura de hardware empleada para las pruebas finales y en la Figura 10 se muestra el montaje físico.

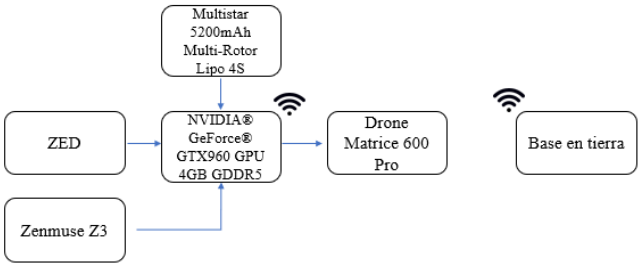

Figura 9. Arquitectura de hardware

Figura 10. Montaje físico del sistema.

D. Seguimiento de personas

Caracterizando el movimiento de una persona vista desde una altura de 20 metros, con un ángulo de la cámara 40° y a una distancia inicial del *drone* de 19 metros, se realiza la estimación del desplazamiento en pixeles como se muestra en la Tabla .

Tabla III. Estimación desplazamiento en X y en Y

Centímetros	Pixeles en X	Pixeles en Y
0	0	0
90	136	28
180	233	61
70	333	101
360	426	127
450	523	149
540	618	177

Realizando una línea de tendencia para cada desplazamiento se obtienen las funciones (5) *y* (6).

$$f(x) = 0.8\,x\,[cm] \quad (5)$$
$$f(y) = 0{,}0034 * y^2 + 2{,}386 * y\,[cm] \quad (6)$$

Las funciones 5 y 6 permiten tener la estimación de la distancia que se desplaza una persona que se encuentra en la zona de interés, estos valores se determinaron por la variación en términos de pixeles y la distancia recorrida por la persona en tierra. Se puede observar que la variación en el eje Y no se comporta de manera lineal ya que se obtiene una proyección 2D de un plano inclinado en 3D. Para hacer un cálculo que esté acorde a la realidad hay que tener en cuenta más variables de las que se pueden medir y controlar como una posición y ángulo fijo de la cámara, ya que por el movimiento del *drone* no se puede obtener un ángulo de visión fijo.

Con base a los diferentes métodos de tracking a objetos se dejó abierta la posibilidad de implementar distintos métodos para realizar el control del *drone*, se publica un *topic* que entrega un vector de ocho datos: posición en x, posición en y, Δ posicion x, Δ posición y, estimación x, estimación y, Error x, Error y; donde Error x y Error y son medidos con respecto al centro de la imagen. En la Figura 11 se muestran los datos publicados en el *topic* Datos_Tracking.

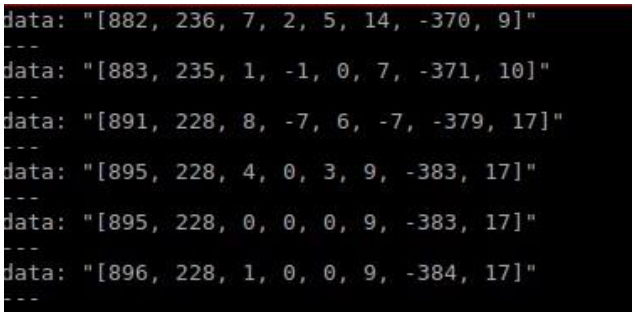

Figura 11. Datos obtenidos del *topic* Datos_tracking mediante un rostopic echo

V. EXPERIMENTACIÓN EN TIEMPO REAL

La zona de interés es sobrevolada por el *drone* con la cámara embarcada y el sistema embebido a bordo, la cámara obtiene una imagen de la zona, posteriormente esta es procesada por el algoritmo, el cual detecta y transmite si hay personas en la región. Al detectar una persona procede a realizar el *tracking* y transmitir los datos del GPS a la base en tierra donde el usuario observa la imagen de la detección y la posición global mediante una interfaz gráfica que se actualiza cada dos segundos, en la Figura 12 se muestra la arquitectura del sistema.

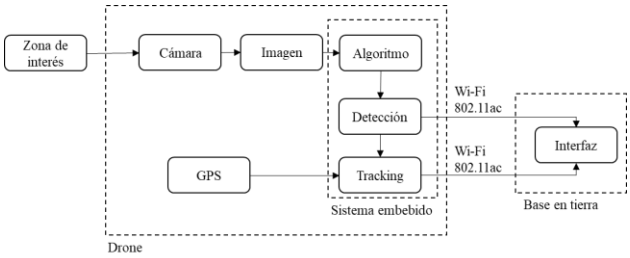

Figura 12. Arquitectura de programación

Para la detección de personas se implementó una arquitectura de software basada en nodos, la cual permite probar los algoritmos con las condiciones que tiene el sistema mientras está en marcha. El nodo *camera* publica el vídeo al nodo del algoritmo, de este nodo se obtienen tres mensajes: la dimensión y ubicación de la caja que enmarca a la persona, el número de personas detectadas y la imagen de salida; simultaneo a esto, el *drone* publica la posición global y en el caso de la cámara Zenmuse Z3 publica el ángulo de la cámara con respecto al plano horizontal del *drone*. Con la posición global del *drone*, la dimensión y ubicación de la caja se realiza el tracking a la persona detectada. La estación ubicada en tierra se suscribe a cuatro *topics*: La imagen de salida, el número de personas

detectadas, la ubicación del *drone* y el vector de datos entregado por el nodo Datos_Tracking.

A. *Puesta en marcha*

La interfaz se desarrolló en el programa Matlab bajo el sistema operativo Windows, el cual permite generar un ejecutable para que así cualquier persona pueda emplearlo sin tener conocimiento previo del programa. En la Figura 13 se observa la interfaz donde se ingresa la dirección IP del sistema embebido con él que se comunica; al estar conectado se muestra en la interfaz la imagen en tiempo real que se captura con la cámara, el número de personas detectadas y la ubicación global del *drone* que se obtiene mediante el GPS.

Figura 13. Interfaz que se presenta en la estación ubicada en tierra

B. *Validación en tiempo real*

Al tener el montaje físico del sistema, con el algoritmo programado en el sistema embebido y la comunicación con la estación en tierra, se procede a realizar pruebas finales obteniendo el estudio estadístico de la Tabla .

Tabla IV. Estudio estadístico en tiempo real

Algoritmo	Exactitud	Precisión	Sensibilidad	FPS
YOLO	99,23%	98,99%	99,78%	15,6

Solo fue necesaria una prueba online para ajustar los sistemas y los parámetros del algoritmo de reconocimiento, ya que en la puesta en marcha se usó la misma distribución lógica interna sustituyendo el nodo de la cámara por uno que entrega vídeo. La exactitud que se obtiene en promedio de las pruebas online y offline es del 99,45% y la precisión del 97,86%.

VI. Conclusiones

A partir de la literatura consultada y la experimentación, la detección de personas por medio de redes neuronales como YOLO y Tiny YOLO al requerir solo muestras positivas en su entrenamiento es más robusto respecto a los métodos convencionales, ya que el fondo de la imagen no afecta la precisión y sensibilidad de su detección, mientras que el algoritmo HOG requiere un entrenamiento previo del lugar donde se desea hacer la detección, ya que las muestra negativas varían en cada escenario.

Se debe tener en cuenta factores externos como la iluminación y la resolución de la cámara, ya que la calidad de la detección varía dependiendo de estos factores. En una condición ideal la velocidad de obturación de la cámara debe ser superior a 300Hz para que el movimiento del *drone* no afecte la precisión del algoritmo. Dependiendo de la capacidad de computo del sistema embebido la velocidad a la que el algoritmo puede procesar varía, en este caso, requiere que sea mayor a 3.0 para que la detección de personas sea en tiempo real.

Al detectar una persona se obtiene su ubicación dentro de la imagen, con este dato y el GPS se obtiene el vector de movimiento, el cual es impreciso debido a las condiciones ambientales y la perdida de datos provocada por la proyección de un plano 3D a 2D, esta condición es transmitida mediante el protocolo Wifi 802.11ac permitiendo comunicar el *drone* con la base de control ubicada en tierra.

El estudio estadístico realizado en tiempo real muestra que el método seleccionado para la detección de personas es el adecuado, puesto que su entrenamiento y los filtros propuestos han permitido tener una precisión del 98,99% y una exactitud del 99,23%, lo que permite inferir que los parámetros seleccionados en el experimento son los adecuados para la detección de personas.

A partir de la experiencia adquirida, se propone como trabajo futuro el seguimiento de una persona luego de ser detectada, esto se puede realizar mediante el vector de dirección del objeto; otro trabajo futuro sería etiquetar cada persona que aparezca en la escena con su respectivo vector de dirección y diferenciar las personas con marcadores, es decir, las personas que realicen la vigilancia empleen una prenda distintiva y así diferenciarlos. Para realizar una vigilancia en condiciones adversas de luz se propone integrar una cámara de visión nocturna o térmica y ampliar la base de datos para realizar un nuevo entrenamiento del algoritmo.

VII. Referencias

[1] ECOPETROL, Ed., *Ecopetrol anuncia plan de inversiones para 2016 por US$4.800 millones,* 2015.

[2] PORTAFOLIO, «Las válvulas ilícitas, otro drama para las petroleras,» *Portafolio,* 8 Junio 2017.

[3] M. Manosalve, «Derrame de petróleo por atentado en oleoducto de Caño Limón se extiende por 107 kilómetros,» *EL ESPECTADOR,* 27 Septiembre 2017.

[4] Extra, «Operación conjunta logró desactivar válvula ilegal de combustibles en Buenaventura,» *Extra,* 24 Junio 2017.

[5] EL TIEMPO, «La Policía de Bogotá planea utilizar drones para la vigilancia,» *EL TIEMPO,* 19 Octubre 2016.

[6] Y. Ma, X. Wu, G. Yu, Y. Xu y Y. Wang, «Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery,» *Sensors (Switzerland),* vol. 16, 2016.

[7] J. Navarro, «People exact-tracking using a Parrot AR.Drone 2.0,» 2015.

[8] P. Fang, J. Lu, Y. Tian y Z. Miao, «An Improved Object Tracking Method in UAV Videos,» *Procedia Engineering*, vol. 15, pp. 634-638, 2011.

[9] O. Oreifej, R. Mehran y M. Shah, «Human identity recognition in aerial images,» de *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010.

[10] N. M. Cruz, *Desarrollo de un sistema avanzado de asistencia a la conducción en tiempo real para la detección de peatones en entornos urbanos complejos*, 2013.

[11] Y. D. Guo, Z. G. Xie y H. B. Ma, «Pedestrian detection optimization algorithm based on low-altitude UAV,» 2014.

[12] D. Sugimura, T. Fujimura y T. Hamamoto, «Enhanced Cascading Classifier Using Multi-Scale HOG for Pedestrian Detection from Aerial Images.,» *International Journal of Pattern Recognition \& Artificial Intelligence*, vol. 30, pp. -1, 2016.

[13] P. Blondel, A. Potelle, C. Pegard y R. Lozano, «How to improve the HOG detector in the UAV context,» Compiegne, 2013.

[14] F. P. y. K. P. N. Davis, «Facial recognition using human visual system algorithms for robotic and UAV platforms,» 2013.

[15] S. Reyes, J. Rángel, J. Zavala, H. Romero, S. Salazar y R. Lozano, «Real-Time Tracking by a Quadrotor for Arbitrary Objects Defined Online*,» *IFAC Proceedings Volumes*, vol. 46, pp. 93-98, 2013.

[16] Q. Peng, W. Luo, G. Hong, M. Feng, Y. Xia, L. Yu, X. Hao, X. Wang y M. Li, «Pedestrian Detection for Transformer Substation Based on Gaussian Mixture Model and YOLO,» de *8th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Nanjing, China, 2016.

[17] V. Molchanov, B. Vishnyakov, Y. Vizilter, O. Vishnyakova y V. Knyaz, *Pedestrian detection in video surveillance using fully convolutional YOLO neural network*, Munich, 2017.

[18] J. Oviedo, *Uso de los drones en la seguridad privada*, Bogotá, 2016.

[19] W. Velásquez, *El Impacto De La Tecnología Dron En La Seguridad Privada*, Bogotá, 2017.

[20] A. Loaiza, D. Manzano y L. Múnera, *Sistema de visión artificial para conteo de objetos en movimiento*, 2012.

[21] P. Blondel, A. Potelle, C. Pégard y R. Lozano, «Human detection in uncluttered environments: From ground to UAV view,» de *Control Automation Robotics Vision (ICARCV), 2014 13th International Conference on*, 2014.

[22] R. Joseph, «Darknet: Open Source Neural Networks in C,» 2016. [En línea]. Available: https://pjreddie.com/darknet/.

[23] J. Redmon, S. Divval, R. Girshic y A. Farhad, «You Only Look Once: Unified, Real-Time Object Detection,» de *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] U. A. d. Barcelona, «Detección de objetos,» Barcelona, 2016.

[25] P. Negri y D. Garayalde, «Pedestrian tracking using probability fields and a movement feature space,» *DYNA*, 2016.

[26] J. Pestana, J. Sanchez-Lopez, P. Campoy y S. Saripalli, «Vision based GPS-denied Object Tracking and following for unmanned aerial vehicles,» de *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linkoping, 2013.

[27] OpenCV, «OpenCV,» [En línea]. Available: http://opencv.org/about.html.

[28] OpenCV, «CUDA,» [En línea]. Available: http://opencv.org/platforms/cuda.html.

[29] Qiushi, *Label object bboxes for ImageNet Detection data*, 2014.

[30] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. von Stryk, S. Roth y B. Schiele, «Vision based victim detection from unmanned aerial vehicles,» de *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.

[31] J. Li, X. Liang, S. Shen, T. Xu, J. Feng y S. Yan, «Scale-aware Fast R-CNN for Pedestrian Detection,» de *IEEE Transactions on Multimedia*, 2016.