



## CARRERA DE ESPECIALIZACIÓN EN INTELIGENCIA ARTIFICIAL

MEMORIA DEL TRABAJO FINAL

# Reconocimiento de intrusos en video de vigilancia aérea

**Autor:**

**Ing. Juan Pablo Nieto Uribe**

Director:

Esp. Ing. Hernán Contigiani (FIUBA)

Jurados:

Nombre del jurado 1 (pertenencia)

Nombre del jurado 2 (pertenencia)

Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la ciudad de Bogotá, Colombia,  
entre octubre de 2021 y junio de 2020.*



## *Resumen*

La presente memoria detalla el proceso de creación de un módulo de inteligencia artificial diseñado para detectar personas y animales en vídeo de vigilancia capturado con cámaras montadas en drones. En particular, el objetivo de este trabajo es detectar intrusos en streaming de vídeo de seguridad. El módulo se desarrolló en colaboración con Manta Beach, empresa que hace parte del programa de vinculación de la universidad y que propuso el proyecto como parte de su plan de investigación y desarrollo.

El desarrollo de este trabajo hace un uso extensivo de los conocimientos de la rama de visión por computadora a través de redes neuronales convolucionales. De igual manera, se ha requerido una base sólida en Python, teniendo en cuenta que se partió de un algoritmo establecido con anterioridad.



## *Agradecimientos*



# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Motivación . . . . .	1
1.3. Estado del arte . . . . .	1
1.4. Objetivos y alcance . . . . .	1
<b>2. Introducción específica</b>	<b>3</b>
2.1. Definición de Intruso . . . . .	3
2.2. Modelo Utilizado . . . . .	3
2.3. Protocolos de vídeo . . . . .	3
2.4. Software utilizado . . . . .	3
<b>3. Diseño e implementación</b>	<b>5</b>
3.1. Consideraciones generales . . . . .	5
3.2. Arquitectura del proyecto . . . . .	6
3.3. Esquema del módulo . . . . .	6
3.4. Esquema de modificaciones . . . . .	6
3.5. Ajustes para cumplir con el rendimiento . . . . .	6
3.6. Entrega de resultados al resto del software . . . . .	6
<b>4. Ensayos y resultados</b>	<b>7</b>
4.1. Descripción del proceso de pruebas . . . . .	7
4.2. Pruebas en Raspberry Pi . . . . .	7
4.3. Caso de Uso . . . . .	7
<b>5. Conclusiones</b>	<b>9</b>
5.1. Resultados Obtenidos . . . . .	9
5.2. Tiempos de ejecución . . . . .	9
5.3. Monitoreo de resultados . . . . .	9





# Índice de figuras



# Índice de tablas



***A Iván. Por tu intuición.***



# Capítulo 1

## Introducción general

### 1.1. Introducción

$\text{\LaTeX}$  no es WYSIWYG (What You See is What You Get), a diferencia de los procesadores de texto como Microsoft Word o Pages de Apple o incluso LibreOffice en el mundo open-source. En lugar de ello, un documento escrito para  $\text{\LaTeX}$  es en realidad un archivo de texto simple o llano que *no contiene formato*. Nosotros le decimos a  $\text{\LaTeX}$  cómo deseamos que se aplique el formato en el documento final escribiendo comandos simples entre el texto, por ejemplo, si quiero usar texto en itálicas para dar énfasis, escribo `\it{texto}` y pongo el texto que quiero en itálicas entre medio de las llaves. Esto significa que  $\text{\LaTeX}$  es un lenguaje del tipo «mark-up», muy parecido a HTML.

### 1.2. Motivación

Si sos nuevo en  $\text{\LaTeX}$ , hay un muy buen libro electrónico - disponible gratuitamente en Internet como un archivo PDF - llamado, «A (not so short) Introduction to  $\text{\LaTeX}$ ». El título del libro es generalmente acortado a simplemente *lshort*. Puede descargar la versión más reciente en inglés (ya que se actualiza de vez en cuando) desde aquí: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

Se puede encontrar la versión en español en la lista en esta página: <http://www.ctan.org/tex-archive/info/lshort/>

### 1.3. Estado del arte

Acá tiene un ejemplo de una “subsubsección” que es el cuarto nivel de ordenamiento del texto, después de capítulo, sección y subsección. Como se puede ver, las subsubsecciones no van numeradas en el cuerpo del documento ni en el índice. El formato está definido por la plantilla y no debe ser modificado.

### 1.4. Objetivos y alcance

Si estás escribiendo un documento con mucho contenido matemático, entonces es posible que desees leer el documento de la AMS (American Mathematical Society) llamado, «A Short Math Guide for  $\text{\LaTeX}$ ». Se puede encontrar en línea en el siguiente link: <http://www.ams.org/tex/amslatex.html> en la sección «Additional Documentation» hacia la parte inferior de la página.





## Capítulo 2

# Introducción específica

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

### 2.1. Definición de Intruso

### 2.2. Modelo Utilizado

Si en el texto se hace alusión a diferentes partes del trabajo referirse a ellas como capítulo, sección o subsección según corresponda. Por ejemplo: “En el capítulo 1 se explica tal cosa”, o “En la sección ?? se presenta lo que sea”, o “En la subsección ?? se discute otra cosa”.

Cuando se quiere poner una lista tabulada, se hace así:

- Este es el primer elemento de la lista.
- Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

Si se desea poner una lista numerada el formato es este:

1. Este es el primer elemento de la lista.
2. Este es el segundo elemento de la lista.

Notar el uso de las mayúsculas y el punto al final de cada elemento.

### 2.3. Protocolos de vídeo

La recepción y análisis del vídeo depende únicamente de los formatos soportados por OpenCv. Dentro de los que, según la documentación encontramos están:

Sin embargo, es importante tener en cuenta que de todos esos se ha probado únicamente los siguientes:

### 2.4. Software utilizado

Para el desarrollo de este trabajo, y teniendo en cuenta el alcance propuesto desde el inicio, fue necesario el uso de tres tipos de software:

1. Paquetes de Python: Son paquetes importados en el código a través del comando `import`. Cabe mencionar que todos los paquetes son de código abierto. Dentro de ellos encontramos:

\* Numpy \* Matplotlib \* OpenCV \* Tensorflow

2. Código en Python: En especial para la implementación de la metodología YOLOV3. En internet está disponible el código fuente en formato Python (`.py`).

3. Software completo: En particular para la retransmisión del vídeo procesado, se utilizó OBS estudio. Esta decisión responde a que, dentro del alcance del proyecto, no se definió ninguna actividad que no se encuentre directamente relacionada con la implementación de la detección de intrusos. Sin embargo, dado que se trata de un proyecto en el que se debe devolver la imagen procesada, es importante asegurarse que la retransmisión de los datos funcione de la manera correcta.

## Capítulo 3

# Diseño e implementación

### 3.1. Consideraciones generales

La idea de esta sección es resaltar los problemas encontrados, los criterios utilizados y la justificación de las decisiones que se hayan tomado.

Se puede agregar código o pseudocódigo dentro de un entorno `lstlisting` con el siguiente código:

```
\begin{lstlisting}[caption= "un epígrafe descriptivo"]
las líneas de código irían aquí...
\end{lstlisting}
```

A modo de ejemplo:

```
1 #define MAX_SENSOR_NUMBER 3
2 #define MAX_ALARM_NUMBER 6
3 #define MAX_ACTUATOR_NUMBER 6
4
5 uint32_t sensorValue[MAX_SENSOR_NUMBER];
6 FunctionalState alarmControl[MAX_ALARM_NUMBER]; //ENABLE or DISABLE
7 state_t alarmState[MAX_ALARM_NUMBER]; //ON or OFF
8 state_t actuatorState[MAX_ACTUATOR_NUMBER]; //ON or OFF
9
10 void vControl() {
11
12     initGlobalVariables();
13
14     period = 500 ms;
15
16     while(1) {
17
18         ticks = xTaskGetTickCount();
19
20         updateSensors();
21
22         updateAlarms();
23
24         controlActuators();
25
26         vTaskDelayUntil(&ticks, period);
27     }
28 }
```

CÓDIGO 3.1. Pseudocódigo del lazo principal de control.

- 3.2. Arquitectura del proyecto**
- 3.3. Esquema del módulo**
- 3.4. Esquema de modificaciones**
- 3.5. Ajustes para cumplir con el rendimiento**
- 3.6. Entrega de resultados al resto del software**

## Capítulo 4

# Ensayos y resultados

### 4.1. Descripción del proceso de pruebas

La idea de esta sección es explicar cómo se hicieron los ensayos, qué resultados se obtuvieron y analizarlos.

### 4.2. Pruebas en Raspberry Pi

### 4.3. Caso de Uso



## Capítulo 5

# Conclusiones

### 5.1. Resultados Obtenidos

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

### 5.2. Tiempos de ejecución

Acá se indica cómo se podría continuar el trabajo más adelante.

### 5.3. Monitoreo de resultados