



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 2023.*



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:
Dr. Ing. Leonardo Alfredo Forero Mendoza (PUC-RIO)
Dr. Ing. Javier Andrés Redolfi (UTN - FRSF)
Ing. Juan Ignacio Cavalieri (DeepAgro)

*Este trabajo fue realizado en la ciudad de Bogotá, Colombia,
entre octubre de 2021 y junio de 2023.*

Agradecimientos

Agradecimientos

A mis papás, la Neni y Juli por su paciencia.

A mis jurados por su invaluable ayuda.

A Hernán por su generosidad.

Índice general

| | |
|--|-----------|
| Resumen | I |
| 1. Introducción general | 1 |
| 1.1. Contexto de la problemática | 1 |
| 1.2. Motivación | 1 |
| 1.3. Estado del arte | 2 |
| 1.4. Objetivos y alcance | 4 |
| 2. Introducción específica | 7 |
| 2.1. Definición de intruso | 7 |
| 2.2. Modelo utilizado | 8 |
| 2.3. Protocolos de vídeo y transmisión | 10 |
| 2.4. Software y hardware utilizados | 11 |
| 3. Diseño e implementación | 13 |
| 3.1. Consideraciones generales | 13 |
| 3.2. Arquitectura del módulo | 14 |
| 3.3. Esquema detallado del módulo en el interior del sistema | 16 |
| 3.4. Esquema detallado de las modificaciones hechas al modelo | 17 |
| 3.5. Ajustes para cumplir con el rendimiento requerido | 18 |
| 3.6. Entrega de resultados obtenidos al resto del software y retransmisión del vídeo procesado | 19 |
| 4. Ensayos y resultados | 21 |
| 4.1. Descripción del proceso de pruebas | 21 |
| 4.2. Pruebas en Raspberry Pi | 22 |
| 4.3. Caso de Uso | 23 |
| 5. Conclusiones | 25 |
| 5.1. Resultados Obtenidos | 25 |
| 5.2. Tiempos de ejecución | 25 |
| 5.3. Trabajos futuros | 26 |
| A. Tablas extendidas | 27 |
| Bibliografía | 29 |

Índice general

| | |
|--|------------|
| Resumen | I |
| Agradecimientos | III |
| 1. Introducción general | 1 |
| 1.1. Contexto de la problemática | 1 |
| 1.2. Motivación | 1 |
| 1.3. Estado del arte | 2 |
| 1.4. Objetivos y alcance | 4 |
| 2. Introducción específica | 7 |
| 2.1. Definición de intruso | 7 |
| 2.2. Modelo utilizado | 8 |
| 2.3. Protocolos de vídeo y transmisión | 10 |
| 2.4. Software y hardware utilizados | 11 |
| 3. Diseño e implementación | 13 |
| 3.1. Consideraciones generales | 13 |
| 3.2. Arquitectura del módulo | 14 |
| 3.3. Esquema detallado del módulo en el interior del sistema | 16 |
| 3.4. Esquema detallado de las modificaciones hechas al modelo | 17 |
| 3.5. Ajustes para cumplir con el rendimiento requerido | 18 |
| 3.6. Entrega de resultados obtenidos al resto del software y retransmisión del vídeo procesado | 19 |
| 4. Ensayos y resultados | 21 |
| 4.1. Descripción del proceso de pruebas | 21 |
| 4.2. Pruebas en Raspberry Pi | 22 |
| 4.3. Caso de Uso | 23 |
| 5. Conclusiones | 25 |
| 5.1. Resultados Obtenidos | 25 |
| 5.2. Tiempos de ejecución | 25 |
| 5.3. Trabajos futuros | 26 |
| A. Tablas extendidas | 27 |
| Bibliografía | 29 |

- Tipo: opcional
 - Estado: descartado
2. Requerimientos de documentación:
- 2.1 El sistema deberá contar con un manual de usuario.
- Tipo: obligatorio
 - Estado: finalizado
- 2.2 Se deberá contar con un manual de instalación del módulo desarrollado.
- Tipo: obligatorio
 - Estado: finalizado
- 2.3 El código fuente deberá estar debidamente **documentado**.
- Tipo: obligatorio
 - Estado : finalizado
3. Requerimientos de pruebas:
- 3.1 El módulo deberá ser probado formalmente según indicaciones del cliente.
- Tipo: requisito modificado
 - Estado: descartado
- 3.2 El cliente debe poder acceder en cualquier momento a la versión más reciente del código a través de GitHub.
- Tipo: obligatorio
 - Estado: finalizado
4. Requerimientos de la interfaz:
- 4.1 El usuario debe poder iniciar fácilmente la ejecución del módulo.
- Tipo: obligatorio
 - Estado: finalizado
5. Requerimientos de interoperabilidad:
- 5.1 **Opción A:** el módulo debe estar en capacidad de recibir el *streaming* de vídeo de entre 4 y 8 drones.
- Tipo: requisito modificado
 - Estado: descartado
- 5.2 **Opción B:** el módulo debe ser capaz de recibir un *streaming* de vídeo y ejecutar la detección de intrusos en hardware dedicado a bordo.
- Tipo: requisito modificado
 - Estado: descartado

- Tipo: opcional
 - Estado: descartado
2. Requerimientos de documentación:
- 2.1 El sistema deberá contar con un manual de usuario.
- Tipo: obligatorio
 - Estado: finalizado
- 2.2 Se deberá contar con un manual de instalación del módulo desarrollado.
- Tipo: obligatorio
 - Estado: finalizado
- 2.3 El código fuente deberá estar debidamente **comentado**.
- Tipo: obligatorio
 - Estado : finalizado
3. Requerimientos de pruebas:
- 3.1 El módulo deberá ser probado formalmente según indicaciones del cliente.
- Tipo: requisito modificado
 - Estado: descartado
- 3.2 El cliente debe poder acceder en cualquier momento a la versión más reciente del código a través de GitHub.
- Tipo: obligatorio
 - Estado: finalizado
4. Requerimientos de la interfaz:
- 4.1 El usuario debe poder iniciar fácilmente la ejecución del módulo.
- Tipo: obligatorio
 - Estado: finalizado
5. Requerimientos de interoperabilidad:
- 5.1 **Opción A:** el módulo debe estar en capacidad de recibir el *streaming* de vídeo de entre 4 y 8 drones.
- Tipo: requisito modificado
 - Estado: descartado
- 5.2 **Opción B:** el módulo debe ser capaz de recibir un *streaming* de vídeo y ejecutar la detección de intrusos en hardware dedicado a bordo.
- Tipo: requisito modificado
 - Estado: descartado

las tarjetas gráficas apropiadas, dado que no se contó con el equipo. Sin embargo, se espera que este rendimiento mejore sustancialmente.

A fin de poder realizar estas pruebas, sin embargo, se requerirá contar con una de las siguientes opciones:

1. Computadora que cuente, idealmente, con una tarjeta gráfica NVidia Tesla T4 o NVidia Titan X (según referenciado en el capítulo 2).
2. Código que no requiera de la generación de una ventana para la retransmisión de los *frames*. Esto permitirá montar el módulo en un servicio de *Cloud Computing* (como Google Cloud), en donde es posible alquilar tarjetas gráficas con este tipo de especificaciones, o bien, ejecutar pruebas sobre el código en Google Colab, en donde se podrán medir más fielmente los tiempos de ejecución del módulo.

5.3. Trabajos futuros

Teniendo en cuenta que se trata de un mínimo producto viable, cabe aclarar que los siguientes trabajos son aún necesarios para que el cliente pueda contar con un módulo de inteligencia artificial que se ajuste de manera adecuada a sus necesidades. Dentro de estos trabajos futuros están:

- Añadir al módulo la capacidad de retransmitir los *frames* procesados. Esto eliminará la necesidad de utilizar software externo como OBS Studio.
- El módulo deberá ser optimizado muy fuertemente. Esto implica que se sigan los siguientes pasos:
 - Actualmente el módulo detecta la totalidad de las clases del *dataset* COCO y descarta las clases que no se requieren durante el paso de reporte. Esto implica que el sistema hace una gran cantidad de cálculos innecesarios, correspondientes a clases que no se requieren por el cliente.
 - Se deberá reentrenar los pesos a fin de evitar la ocurrencia de falsos positivos, así como reducir la ocurrencia de falsos negativos.
 - Para versiones más avanzadas del código, se hará necesaria la generación, en conjunto con el cliente, de definiciones más precisas de qué se considera aceptable en una matriz de confusión.
- Algunos de los requisitos del cliente (particularmente la detección en vuelos nocturnos) no pudieron ser cubiertos. Es imperativo entonces hacer las pruebas y ajustes correspondiente una vez exista el material de vuelos en horas de la noche.
- Empaquetar el módulo, idealmente en formato .deb, de forma que distribuir con facilidad.
- Durante el desarrollo de este trabajo se descartó el uso de Raspberry Pi para el desarrollo de módulos a bordo del dron dadas las limitaciones del modelo utilizado (Raspberry Pi 2 Modelo B). Se deberá utilizar placas Raspberry Pi más avanzadas, de preferencia Raspberry Pi 4.

las tarjetas gráficas apropiadas, dado que no se contó con el equipo. Sin embargo, se espera que este rendimiento mejore sustancialmente.

A fin de poder realizar estas pruebas, sin embargo, se requerirá contar con una de las siguientes opciones:

1. Computadora que cuente, idealmente, con una tarjeta gráfica NVidia Tesla T4 o NVidia Titan X (según referenciado en el capítulo 2).
2. Código que no requiera de la generación de una ventana para la retransmisión de los *frames*. Esto permitirá montar el módulo en un servicio de *Cloud Computing* (como Google Cloud), en donde es posible alquilar tarjetas gráficas con este tipo de especificaciones, o bien, ejecutar pruebas sobre el código en Google Colab, en donde se podrán medir más fielmente los tiempos de ejecución del módulo.

5.3. Trabajos futuros

Teniendo en cuenta que se trata de un mínimo producto viable, cabe aclarar que los siguientes trabajos son aún necesarios para que el cliente pueda contar con un módulo de inteligencia artificial que se ajuste de manera adecuada a sus necesidades. Dentro de estos trabajos futuros están:

- Añadir al módulo la capacidad de retransmitir los *frames* procesados. Esto eliminará la necesidad de utilizar software externo como OBS Studio.
- El módulo deberá ser optimizado muy fuertemente. Esto implica que se sigan los siguientes pasos:
 - Actualmente el módulo detecta la totalidad de las clases del *dataset* COCO y descarta las clases que no se requieren durante el paso de reporte. Esto implica que el sistema hace una gran cantidad de cálculos innecesarios, correspondientes a clases que no se requieren por el cliente.
 - Se deberá reentrenar los pesos a fin de evitar la ocurrencia de falsos positivos, así como reducir la ocurrencia de falsos negativos.
 - Para versiones más avanzadas del código, se hará necesaria la generación, en conjunto con el cliente, de definiciones más precisas de qué se considera aceptable en una matriz de confusión.
- Algunos de los requisitos del cliente (particularmente la detección en vuelos nocturnos) no pudieron ser cubiertos. Es imperativo entonces hacer las pruebas y ajustes correspondiente una vez exista el material de vuelos en horas de la noche.
- Empaquetar el módulo, idealmente en formato .deb, de forma que distribuir con facilidad.
- Durante el desarrollo de este trabajo se descartó el uso de Raspberry Pi para el desarrollo de módulos a bordo del dron dadas las limitaciones del modelo utilizado (Raspberry Pi 2 Modelo B). Se deberá utilizar placas Raspberry Pi más avanzadas, de preferencia Raspberry Pi 4.
- Se deberá implementar el reporte en formato JSON.