

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 epigrafe 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     initGlobalVariables();
12
13     period = 500 ms;
14
15     while(1) {
16
17         ticks = xTaskGetTickCount();
18
19         updateSensors();
20
21         updateAlarms();
22
23         controlActuators();
24
25         vTaskDelayUntil(&ticks, period);
26     }
27 }
```

CÓDIGO 3.1: Pseudocódigo del lazo principal de control.

Capítulo 3

Diseño e implementación

En este capítulo se detallan los aspectos técnicos del desarrollo del trabajo. Incluye las consideraciones tomadas en cuenta durante el desarrollo, el detalle de las modificaciones realizadas al modelo de partida, así como el detalle del papel que juega el módulo en el sistema. De igual manera, se hace una descripción cronológica del desarrollo del módulo.

3.1. Consideraciones generales

El módulo de inteligencia artificial propuesto es una implementación del modelo YOLOv3, especialmente modificada para cumplir con las necesidades específicas del cliente, así como una serie de requisitos funcionales. Es por esto que fue necesario hacer modificaciones a los archivos originales de la biblioteca.

(Esta sección aún no se ha terminado)

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

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 4

Ensayos y resultados

A pesar de que esto se estipuló en la concepción inicial del proyecto, conforme se dio su avance no se hizo posible llevar a cabo un proceso formal de pruebas al módulo. Sin embargo, el cliente estuvo involucrado directamente en el proceso de verificación de la funcionalidad del módulo a través de reuniones quincenales.

4.1. Descripción del proceso de pruebas

A pesar de que esto se estipuló en la concepción inicial del proyecto, conforme se dio su avance no se hizo posible llevar a cabo un proceso formal de pruebas al módulo. Sin embargo, el cliente estuvo involucrado directamente en el proceso de verificación de la funcionalidad del módulo a través de reuniones quincenales.

4.2. Pruebas en Raspberry Pi

Desde la concepción inicial del proyecto, se estableció que se tendría dos caminos para poder desarrollarlo: Por un lado, se planteó que el módulo debía ser capaz de reconocer intrusos en varios streamings de video simultáneos (camino que finalmente se vio favorecido), o bien, hacer que el módulo estuviese en capacidad de correr en equipos a bordo de los drones. Para esto, se hizo una serie de pruebas en equipos Raspberry Pi. Más concretamente, las especificaciones de los equipos sobre los que se hicieron las pruebas son:

- Modelo: Raspberry Pi
- Sistema Operativo: Raspberry Pi OS (Debian 11 – Bullseye)
- CPU quad-core ARM Cortex-A7 900MHz
- Memoria ram: 1 Gb.
- Overclocking: Desactivado

Concretamente, el modelo con el que se tuvo los problemas más importantes, especialmente a la hora de la instalación fue la versión completa de Tensorflow, requerida para la ejecución del módulo. Cabe mencionar que las placas Raspberry Pi son compatibles con la versión Lite de Tensorflow. Dicho esto, se intentaron 5 métodos para la instalación de esta biblioteca. Ninguno de ellos fue exitoso. El proceso llevado a cabo en cada una de las cinco instalaciones fue el siguiente:

- Máquina 1: Aquí irá una descripción de la instalación de la máquina
- Máquina 2

- Máquina 3
- Máquina 4
- Máquina 5

Es importante destacar que, dado que el módulo se desarrolló tomando como base el modelo YOLOv3 original (no el modelo Lite), no se cumplía con las especificaciones requeridas por este tipo de equipos, y una segunda implementación del modelo utilizando la versión YOLOv3 Lite hubiese implicado en sí, el trabajo equivalente a un proyecto completo adicional.

De igual manera atrae particular atención mencionar que la placa con la que se contaba era un modelo antiguo de Raspberry Pi (de febrero de 2015), que no cuenta con una compatibilidad tan completa como la que se da con una Raspberry Pi 4. Es, entonces, de vital importancia que, para iteraciones posteriores del desarrollo del trabajo, y en caso de que se decida continuar con el desarrollo de módulos a bordo de los drones, considerar placas más avanzadas, como Raspberry Pi 4.

4.3. Caso de Uso